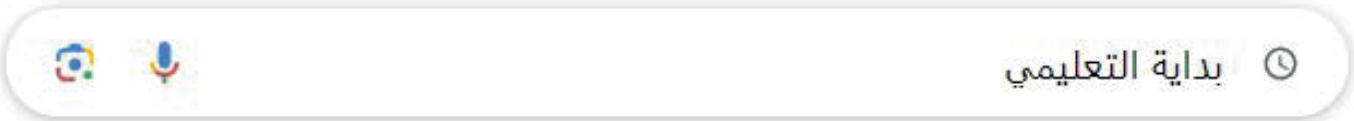
تم تحميل الملف
من موقع بداية

بداية
beadaya.com | موقع بداية التعليمي

للمزيد اكتب
في جوجل Google



موقع بداية التعليمي كل ما يحتاجه **الطالب والمعلم**
من ملفات تعليمية، حلول الكتب، توزيع المنهج،
بوربوينت، اختبارات، ملخصات، اختبارات إلكترونية،
أوراق عمل، والكثير...

حمل التطبيق

GET IN ON Google Play

Download on the App Store

المملكة العربية السعودية

وزارة التعليم
Ministry of Education

قررت وزارة التعليم تدريس
هذا الكتاب وطبعه على نفقتها

إنترنت الأشياء

التعليم الثانوي - نظام المسارات

السنة الثانية

طبعة 1445-2023

يوزع مجاناً ولا يباع
وزارة التعليم
Ministry of Education
2023 - 1445

ⓒ وزارة التعليم، ١٤٤٤ هـ

فهرسة مكتبة الملك فهد الوطنية أثناء النشر
التعليم ، وزارة
إنترنت الأشياء – التعليم الثانوي – نظام المسارات – السنة الثانية. /
وزارة التعليم.– الرياض، ١٤٤٤ هـ
٣٥٢ ص ؛ ٢١ x ٢٥,٥ سم
ردمك : ٠-٥٠٧-٥١١-٦٠٣-٩٧٨
١ ـ الحواسيب ـ تعليم ـ السعودية ٢ ـ التعليم الثانوي ـ السعودية ـ
كتب دراسية أ . العنوان
ديوي ٠٠٤,٠٧ ١١٦٧٠ / ١٤٤٤

رقم الإيداع : ١١٦٧٠ / ١٤٤٤
ردمك : ٠-٥٠٧-٥١١-٦٠٣-٩٧٨

مواد إثرائية وداعمة على "منصة عين الإثرائية"

ien.edu.sa

أعزاءنا المعلمين والمعلمات، والطلاب والطالبات، وأولياء الأمور، وكل مهتم بالتربية والتعليم:
يسعدنا تواصلكم؛ لتطوير الكتاب المدرسي، ومقترحاتكم محل اهتمامنا.

fb.ien.edu.sa

أخي المعلم/أختي المعلمة، أخي المشرف التربوي/أختي المشرفة التربوية:
نقدر لك مشاركتك التي ستسهم في تطوير الكتب المدرسية الجديدة، وسيكون لها الأثر الملموس في دعم العملية التعليمية، وتجويد ما يقدم لأبنائنا وبناتنا الطلبة.

fb.ien.edu.sa/BE

وزارة التعليم
Ministry of Education
2023 - 1445

الناشر: شركة تطوير للخدمات التعليمية

تم النشر بموجب اتفاقية خاصة بين شركة Binary Logic SA وشركة تطوير للخدمات التعليمية
(عقد رقم 2023/0003) للاستخدام في المملكة العربية السعودية

حقوق النشر © Binary Logic SA 2023

جميع الحقوق محفوظة. لا يجوز نسخ أي جزء من هذا المنشور أو تخزينه في أنظمة استرجاع البيانات أو نقله بأي شكل أو بأي وسيلة إلكترونية أو ميكانيكية أو بالنسخ الضوئي أو التسجيل أو غير ذلك دون إذن كتابي من الناشرين.

يُرجى ملاحظة ما يلي: يحتوي هذا الكتاب على روابط إلى مواقع إلكترونية لا تُدار من قبل شركة Binary Logic. ورغم أنَّ شركة Binary Logic تبذل قصارى جهدها لضمان دقة هذه الروابط وحداثتها وملاءمتها، إلا أنها لا تتحمل المسؤولية عن محتوى أي مواقع إلكترونية خارجية.

إشعار بالعلامات التجارية: أسماء المنتجات أو الشركات المذكورة هنا قد تكون علامات تجارية أو علامات تجارية مُسجَّلة وتُستخدم فقط بغرض التعريف والتوضيح وليس هناك أي نية لانتهاك الحقوق. تنفي شركة Binary Logic وجود أي ارتباط أو رعاية أو تأييد من جانب مالكي العلامات التجارية المعنيين. تُعد Excel علامة تجارية مُسجَّلة لشركة Microsoft Corporation. تُعد Tinkercad علامة تجارية مُسجَّلة لشركة Autodesk Inc. تُعد "Python" وشعارات Python علامات تجارية مسجلة لشركة Python Software Foundation. تُعد Jupyter علامة تجارية مُسجَّلة لشركة Project Jupyter. تُعد PyCharm علامة تجارية مُسجَّلة لشركة JetBrains s.r.o. تُعد Multisim Live علامة تجارية مُسجَّلة لشركة National Instruments Corporation. تُعد CupCarbon علامة تجارية مُسجَّلة لشركة CupCarbon. تُعد Arduino علامة تجارية مُسجَّلة لشركة Arduino SA. تُعد Micro:bit علامة تجارية مُسجَّلة لشركة Micro:bit Educational Foundation.
ولا ترعى الشركات أو المنظمات المذكورة أعلاه هذا الكتاب أو تصرح به أو تصادق عليه.

حاول الناشر جاهدا تتبع ملاك الحقوق الفكرية كافة، وإذا كان قد سقط اسم أيٍّ منهم سهوًا فسيكون من دواعي سرور الناشر اتخاذ التدابير اللازمة في أقرب فرصة.

وزارة التعليم
Ministry of Education
2023 - 1445

مقدمة

إن تقدم الدول وتطورها يقاس بمدى قدرتها على الاستثمار في التعليم، ومدى استجابة نظامها التعليمي لمتطلبات العصر ومتغيراته. وحرصًا من وزارة التعليم على ديمومة تطوير أنظمتها التعليمية، واستجابة لرؤية المملكة العربية السعودية 2030 فقد بادرت الوزارة إلى اعتماد نظام «مسارات التعليم الثانوي» بهدف إحداث تغيير فاعل وشامل في المرحلة الثانوية.

إن نظام مسارات التعليم الثانوي يقدم أنموذجًا تعليميًا متميزًا وحديثًا للتعليم الثانوي بالمملكة العربية السعودية يسهم بكفاءة في:

- تعزيز قيم الانتماء لوطننا المملكة العربية السعودية، والولاء لقيادته الرشيدة حفظهم الله، انطلاقًا من عقيدة صافية مستندة على التعاليم الإسلامية السمحة.
- تعزيز قيم المواطنة من خلال التركيز عليها في المواد الدراسية والأنشطة، اتساقًا مع مطالب التنمية المستدامة، والخطط التنموية في المملكة العربية السعودية التي تؤكد على ترسيخ ثنائية القيم والهوية، والقائمة على تعاليم الإسلام والوسطية.
- تأهيل الطلبة بما يتوافق مع التخصصات المستقبلية في الجامعات والكليات أو المهن المطلوبة؛ لضمان اتساق مخرجات التعليم مع متطلبات سوق العمل.
- تمكين الطلبة من متابعة التعليم في المسار المفضل لديهم في مراحل مبكرة، وفق ميولهم وقدراتهم.
- تمكين الطلبة من الالتحاق بالتخصصات العلمية والإدارية النوعية المرتبطة بسوق العمل، ووظائف المستقبل.
- دمج الطلبة في بيئة تعليمية ممتعة ومحفزة داخل المدرسة قائمة على فلسفة بنائية، وممارسات تطبيقية ضمن مناخ تعليمي نشط.
- نقل الطلبة عبر رحلة تعليمية متكاملة بدءًا من المرحلة الابتدائية حتى نهاية المرحلة الثانوية، وتُسهِّل عملية انتقالهم إلى مرحلة ما بعد التعليم العام.
- تزويد الطلبة بالمهارات التقنية والشخصية التي تساعدهم على التعامل مع الحياة، والتجاوب مع متطلبات المرحلة.
- توسيع الفرص أمام الطلبة الخريجين عبر خيارات متنوعة إضافة إلى الجامعات مثل: الحصول على شهادات مهنية، والالتحاق بالكليات التطبيقية، والحصول على دبلومات وظيفية.

ويتكون نظام المسارات من تسعة فصول دراسية تُدرّس في ثلاث سنوات، تتضمن سنة أولى مشتركة يتلقى فيها الطلبة الدروس في مجالات علمية وإنسانية متنوعة، تليها سنتان تخصصيتان، يُسكّن الطلبة بها في مسار عام وأربعة مسارات تخصصية تتسق مع ميولهم وقدراتهم، وهي: المسار الشرعي، مسار إدارة الأعمال، مسار علوم الحاسب والهندسة، مسار الصحة والحياة، وهو ما يجعل هذا النظام هو الأفضل للطلبة من حيث:

- وجود مواد دراسية جديدة تتوافق مع متطلبات الثورة الصناعية الرابعة والخطط التنموية، ورؤية المملكة 2030، تهدف لتنمية مهارات التفكير العليا وحل المشكلات، والمهارات البحثية.
- برامج المجال الاختياري التي تتسق مع احتياجات سوق العمل وميول الطلاب، حيث يُمكّن الطلبة من الالتحاق بمجال اختياري محدد وفق مصفوفة مهارات وظيفية محددة.
- مقياس ميول يضمن تحقيق كفاءة الطلبة وفاعليتهم، ويساعدهم في تحديد اتجاهاتهم وميولهم، وكشف مكامن القوة لديهم، مما يعزز من فرص نجاحهم في المستقبل.
- العمل التطوعي المصمم للطلبة خصيصًا بما يتسق مع فلسفة النشاط في المدارس، ويعد أحد متطلبات التخرج؛ مما يساعد على تعزيز القيم الإنسانية، وبناء المجتمع وتنميته وتماسكه.
- التجسير الذي يمكن الطلبة من الانتقال من مسار إلى آخر وفق آليات محددة.
- حصص الإتقان التي يتم من خلالها تطوير المهارات وتحسين المستوى التحصيلي، من خلال تقديم حصص إتقان إثرائية وعلاجية.

- خيارات التعليم المدمج، والتعلم عن بعد، والذي بُني في نظام المسارات على أسس من المرونة، والملاءمة والتفاعل والفعالية.
- مشروع التخرج الذي يساعد الطلبة على دمج الخبرات النظرية مع الممارسات التطبيقية.
- شهادات مهنية ومهارية تمنح للطلبة بعد إنجازهم مهامَّ محددة، واختبارات معينة بالشراكة مع جهات تخصصية.

وبالتالي فإن مسار علوم الحاسب والهندسة كأحد المسارات المستحدثة في المرحلة الثانوية يسهم في تحقيق أفضل الممارسات عبر الاستثمار في رأس المال البشري، وتحويل الطالب إلى فرد مشارك ومنتج للعلوم والمعارف، مع إكسابه المهارات والخبرات اللازمة لاستكمال دراسته في تخصصات تتناسب مع ميوله وقدراته أو الالتحاق بسوق العمل.

وتعد مادة إنترنت الأشياء أحد المواد الرئيسة في مسار علوم الحاسب والهندسة، حيث تسهم في توضيح ماهية إنترنت الأشياء والتقنيات المرتبطة بها بما يساعد على توظيف هذه التقنيات في عدة مجالات حياتية مثل المدن الذكية والتعليم والزراعة والطب وغيرها من المجالات الاقتصادية المتنوعة. وتهدف المادة إلى تعريف الطالب بأهمية إنترنت الأشياء ودورها في الجيل الرابع من الصناعة مع التعريف بالسياسات والتشريعات المتعلقة بالاستخدام الآمن والأخلاقي لتقنيات إنترنت الأشياء. وكذلك تركز على تعزيز مهارات الربط بين أجهزة إنترنت الأشياء، وكيفية إرسال واستقبال البيانات فيما بينها، ودورها في منظومة البيئات الذكية. كما تشتمل هذه المادة على مشاريع وتمارين تطبيقية لما يتعلمه الطالب؛ لحل مشاكل واقعية تحاكي مستوياته المعرفية، بتوجيه وإشراف من المعلم.

ويتميز كتاب إنترنت الأشياء بأساليب حديثة، تتوافر فيه عناصر الجذب والتشويق، والتي تجعل الطلبة يقبلون على تعلمه والتفاعل معه، من خلال ما يقدمه من تدريبات وأنشطة متنوعة، كما يؤكد هذا الكتاب على جوانب مهمة في تعليم إنترنت الأشياء وتعلمه، تتمثل في:

- الترابط الوثيق بين المحتويات والمواقف والمشكلات الحياتية.
- تنوع طرائق عرض المحتوى بصورة جذابة ومشوقة.
- إبراز دور المتعلم في عمليات التعليم والتعلم.
- الاهتمام بترابط محتوياته مما يجعل منه كلًّا متكاملًا.
- الاهتمام بتوظيف التقنيات المناسبة في المواقف المختلفة.
- الاهتمام بتوظيف أساليب متنوعة في تقويم الطلبة بما يتناسب مع الفروق الفردية بينهم.

ولمواكبة التطورات العالمية في هذا المجال، فإن كتاب مادة إنترنت الأشياء سوف يوفر للمعلم مجموعة متكاملة من المواد التعليمية المتنوعة التي تراعي الفروق الفردية بين الطلبة، بالإضافة إلى البرمجيات والمواقع التعليمية، التي توفر للطلبة فرصة توظيف التقنيات الحديثة والتواصل المبني على الممارسة؛ مما يؤكد دوره في عملية التعليم والتعلم.

ونحن إذ نقدم هذا الكتاب لأعزائنا الطلبة، نأمل أن يستحوذ على اهتمامهم، ويُلبي متطلباتهم، ويجعل تعلّمهم لهذه المادة أكثر متعة وفائدة.

والله ولي التوفيق

بسم الله الرحمن الرحيم

وزارة التعليم
Ministry of Education
2023 - 1445

الفهرس

الجزء الأول

1. أُسس إنترنت الأشياء 10

الدرس الأول

مفاهيم إنترنت الأشياء 11

تمرينات 18

الدرس الثاني

أجهزة إنترنت الأشياء 21

تمرينات 30

المشروع 34

2. إنترنت الأشياء في حياتنا 36

الدرس الأول

منصة إنترنت الأشياء 37

تمرينات 47

الدرس الثاني

تطبيقات وتحديات إنترنت الأشياء 50

تمرينات 62

المشروع 66

3. إنشاء تطبيقات إنترنت الأشياء باستخدام الأردوينو 68

الدرس الأول

إنشاء نظام منزل ذكي 69

تمرينات 85

الدرس الثاني

إنشاء نظام لري النباتات 88

تمرينات 98

الدرس الثالث

إنشاء نظام تَسرُّب الغاز 100

تمرينات 111

المشروع 114

4. إنشاء تطبيق سحابي لإنترنت الأشياء 116

الدرس الأول

إعداد بيئة تطوير الأردوينو 117

تمرينات 133

الدرس الثاني

برمجة الأردوينو في البايثون 136

تمرينات 148

الدرس الثالث

التفاعل مع خدمات الويب السحابية 150

تمرينات 161

المشروع 164

الجزء الثاني

5. تطبيقات إنترنت الأشياء المتقدمة..... 168

الدرس الأول

مجالات تطبيق إنترنت الأشياء 169

تمرينات..... 176

الدرس الثاني

تقنيات شبكات إنترنت الأشياء..... 180

تمرينات..... 190

الدرس الثالث

أمان وخصوصية أنظمة إنترنت الأشياء.... 194

تمرينات..... 200

المشروع 204

6. برمجة إنترنت الأشياء باستخدام C++ 206

الدرس الأول

تطبيقات الحماية الذكية ولغة C++ 207

تمرينات..... 223

الدرس الثاني

الانتقال من اللبنات البرمجية في تينكركاد إلى C++ 228

تمرينات..... 238

الدرس الثالث

برمجة المتحكم الدقيق باستخدام C++ 242

تمرينات..... 258

المشروع 260

7. الرسائل في إنترنت الأشياء .. 262

الدرس الأول

المدن الذكية وبروتوكول MQTT..... 263

تمرينات..... 269

الدرس الثاني

تصميم وبرمجة جهاز ذكي لإنترنت الأشياء273

تمرينات..... 289

الدرس الثالث

إنشاء حل ذكي لإدارة النفايات 292

تمرينات..... 304

المشروع 306

8. محاكاة شبكة مُستشعرات إنترنت الأشياء اللاسلكية 308

الدرس الأول

مقدمة إلى كاب كاربون 309

تمرينات..... 320

الدرس الثاني

الاتصال في شبكة إنترنت الأشياء 323

تمرينات..... 335

الدرس الثالث

إنترنت الأشياء والأجهزة المحمولة المؤتمتة..... 336

تمرينات..... 349

المشروع 350

الجزء الأول

الوحدة الأولى
أُسس إنترنت الأشياء

الوحدة الثانية
إنترنت الأشياء في حياتنا

الوحدة الثالثة
إنشاء تطبيقات إنترنت الأشياء باستخدام الأردوينو

الوحدة الرابعة
إنشاء تطبيق سحابي لإنترنت الأشياء

وزارة التعليم
Ministry of Education
2023 - 1445

1. أُسس إنترنت الأشياء

سيتعرف الطالب في هذه الوحدة على المفاهيم الأساسية لإنترنت الأشياء (IoT) وتطورها على مدار السنوات الأخيرة، وكيف أصبحت جزءًا لا يتجزأ من التقنيات الناشئة. وسيتعرف أيضًا على المكونات الرئيسة للكائنات الذكية، والتي تُعدُّ الأساس لنظام إنترنت الأشياء.

أهداف التعلُّم

بنهاية هذه الوحدة سيكون الطالب قادرًا على أن:

- يعرّف مصطلح إنترنت الأشياء.
- يناقش تطور إنترنت الأشياء.
- يتعرّف على الغرض من إنترنت الأشياء.
- يناقش تأثير إنترنت الأشياء على التقنيات الناشئة.
- يحدّد ماهية الكائن الذكي.
- يصنّف الكائنات الذكية واستخداماتها.
- يميّز بين أنواع مُستشعرات الأشياء الذكية المختلفة.
- يميّز بين أنواع مُشغّلات الأشياء الذكية المختلفة.

رابط الدرس الرقمي

www.ien.edu.sa

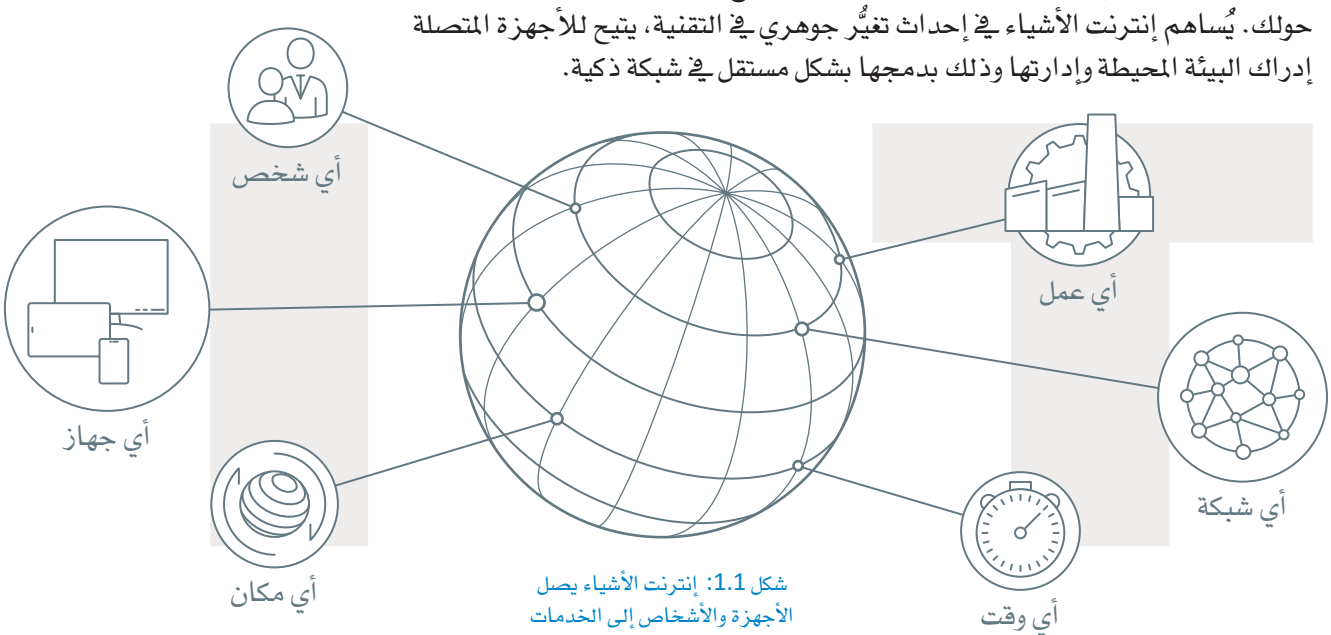
الدرس الأول

مفاهيم إنترنت الأشياء

ما المقصود بإنترنت الأشياء؟
What is the Internet of Things

في العصر الحالي، أصبح بإمكان كلّ الأجهزة والأشياء من حولك أن تتّصل بشبكة الإنترنت، وتتواصل بسهولة مع أجهزة أخرى، أو أشخاص، ومن ثَم الوصول إلى الخدمات التي تُحسِّن حياتك. يشهد العالم تحولًا تقنيًا كبيرًا، فقد صار من الممكن توصيل طلبات الطعام باستخدام الطائرات دون طيار، وطُوّرت المُستشعرات التي يمكن ارتداؤها لمتابعة حالتك الصحية. يُعرف هذا التقدم المتنامي في جميع المجالات باسم **إنترنت الأشياء** (IoT). إن الهدف الرئيس لإنترنت الأشياء هو توصيل الأجهزة المختلفة بشبكات الحاسوب الخاصة أو العامة (مثل شبكة الإنترنت) لتشارك بياناتها، وتتفاعل مع الأشخاص والأشياء الأخرى من حولك. يُساهم إنترنت الأشياء في إحداث تغيُّر جوهري في التقنية، يتيح للأجهزة المتصلة إدراك البيئة المحيطة وإدارتها وذلك بدمجها بشكل مستقل في شبكة ذكية.

إنترنت الأشياء (IoT):
هي شبكة من الأجهزة يستطيع كلٌ منها استشعار البيئة المحيطة أو مراقبتها أو التفاعل معها، بالإضافة إلى جمع البيانات وتبادلها.



شكل 1.1: إنترنت الأشياء يصل الأجهزة والأشخاص إلى الخدمات

عندما تتمكن الأجهزة والمعدات من اكتشاف بيئتها والتحكم فيها عن بُعد عبر الشبكة، فإنه يُمكن دمج العالم الواقعي والحواسيب وربطهما معًا. ويُسهم هذا الدمج في زيادة الإنتاجية والأتمتة، ويُخفض التكاليف في جميع الجوانب الحياتية تقريبًا. إن هدف إنترنت الأشياء هو توصيل الأجهزة بأنواعها المختلفة وتحويلها إلى أجهزة "إنترنت أشياء"، وبالطبع إنشاء تطبيقات جديدة لاستخدام تلك الأجهزة في الشبكة الذكية وإدارتها.

جهاز إنترنت الأشياء (IoT Device):
هو كائن مادي يتصل بشبكة، ويصبح معروفًا في تلك الشبكة. يمكن لذلك الجهاز جمع البيانات ونقلها، وكذلك التواصل مع أجهزة إنترنت الأشياء الأخرى ومنصاتها.

الكائنات الذكية Smart Objects

الكائنات المتصلة أو الذكية هي كائنات تتبادل البيانات عبر الشبكة. تحتوي بعض هذه الكائنات على واجهة مستخدم بسيطة، كمفتاح التحكّم بالحرارة، في حين تتسم الكثير من تلك الكائنات بالواجهات الأكثر تعقيدًا، كتلك الموجودة في السيارات الحديثة أو تطبيقات الهواتف الذكية. وقد تخلو بعض الكائنات الذكية من واجهة المستخدم، حيث تحتوي على مُستشعرات ومُشغّلات مستقلة تتفاعل مع بيئتها دون أي تدخل بشري. يتعرف المُستشعر على بيئته ويقيس القيم الموجودة بها، ثم يقوم ذلك المُشغِّل بتغيير العالم المادي. وتنقسم الكائنات الذكية إلى فئتين: كائنات رقمية، وكائنات مادية (ملموسة). تشمل الكائنات الرقمية أجهزة مثل الهواتف الذكية والساعات الذكية وأنظمة الإنذار المنزلية، والتي يتم تصميمها لتتفاعل مع البيئة المحيطة، في حين أن الكائنات المادية هي كائنات واقعية تتطلب إضافة مُستشعرات أو مُشغّلات لتصبح كائنات ذكية. فالأجهزة المنزلية كالثلاجات والمصابيح لا تستخدم البيانات أو تتبادلها إلا إذا تم تحسينها بإضافة مُستشعرات ووحدات تحكم دقيقة وهوائيات مخصصة لتوصيلها بالعالم الرقمي لإنترنت الأشياء.

شكل 1.2: كائن ذكي مادي (ملموس) وكائن ذكي رقمي

تاريخ إنترنت الأشياء The History of the Internet of Things

إن فكرة إضافة المُستشعرات إلى الأشياء المادية وإتاحة تفاعلها معًا عبر شبكات المعلومات ليست بالجديدة؛ فقد قام بعض طلبة الجامعات في ثمانينيات القرن الماضي بتطوير آلية للتعرف عن بُعد على محتويات آلة بيع المشروبات الغازية. وقد كان استخدام التقنية محدودًا للغاية في تلك الآونة، كما لم تكن شبكة الإنترنت متاحة، ثمّ ساهم تطور الشبكات لتشمل أي جهاز حاسب حول العالم، كما ساهم إصدار الشركات للأجهزة برقائق مصغرة ووحدات معالجة مركزية ومُستشعرات في تطوير المزيد من التطبيقات التقنية. كما تطوّرت شبكة الإنترنت والشبكة العنكبوتية العالمية (WWW) بواسطة شبكة وكالة مشاريع الأبحاث المتقدمة (ARPAnet) التي تأسست عام 1969، لتصبح أكبر حجمًا وأكثر تعقيدًا، وتعتمد أساسًا على بروتوكول الإنترنت (IP)، وبروتوكول التحكم في النقل (TCP)، ثمّ أوقفت الحكومة الأمريكية الدعم عن تلك الشبكة في أبريل 1995، وأُنشئ إطار عمل مفتوح للاتصال بالشبكة من جميع أنحاء العالم، مما أدى إلى ظهور شبكة الإنترنت كما هي معروفة اليوم. ويعدّ عنوان الإنترنت (IP address) أساسًا لهذا العالم المتصل، ويُمثّل عنوانًا فريدًا خاصًا بكل جهاز على الشبكة، ويمكّنه باستخدامه من الاتصال بأجهزة أخرى. ومن الأمثلة على هذه الأجهزة: الهواتف الذكية والأجهزة اللوحية وأجهزة الألعاب والسيارات والغسالات وأنظمة الإضاءة وأقفال الأبواب الأمامية ومحطات بطاقات الائتمان. تُعيّن عناوين IP لجميع الأجهزة المتصلة بإنترنت الأشياء، وقد يستخدم عنوان IP عامًا للوصول للجهاز عبر الإنترنت، أو للتعرف على ذلك الجهاز على شبكة محلية. ويحدد الجهاز الموجه (Router) هذه الأجهزة بناءً على الطلبات الواردة ويوجه الطلبات والبيانات وفقًا لذلك.

شكل 1.3: توجيه حزم البيانات استنادًا إلى عنوان IP عبر شبكة الإنترنت

أي مستخدم للحواسيب المكتبية يُدرك أن الهواتف الذكية أصبحت ذات قدرات حاسوبية كبيرة في السنوات الأخيرة، وذلك على الرغم من صغر حجمها؛ بل إن بعضًا منها أصبح يفوق الحواسيب من حيث السرعة وحجم الذاكرة، كما تمتلك تلك الهواتف مصدرها الخاص للطاقة وهي بطارياتها، ويمكن لهذه الهواتف الاتصال بسهولة بالشبكة اللاسلكية. يُدمج في هذه الهواتف العديد من المُستشعرات مثل الكاميرا، والميكروفون، و**نظام تحديد المواقع العالمي** (GPS)، ومقياس المغناطيسية، ومقياس التسارع، والجايروسكوب، ومُستشعر القرب، ومُستشعر الإضاءة المحيطة. وقد أصبح من الممكن تحويل جهاز حاسب صغير ذي طاقة معالجة وتخزين محدودة، وبضعة مُستشعرات، ليصبح جهازًا صغيرًا يمكن دمجه في أشياء مادية أخرى.

في الواقع فإن عصر إنترنت الأشياء بدأ فعليًا حوالي العام 2008. في ذلك الوقت تقريبًا، أصبح هناك المزيد من الأجهزة المتصلة بالإنترنت، وأصبح إنترنت الأشياء حقيقة واقعة. يعود الفضل في ذلك إلى عالم الحاسوب كيفين أشتون الذي استخدم مصطلح "إنترنت الأشياء" لأول مرة عام 1999 أثناء عمله في شركة كبيرة متعددة الجنسيات، حيث اُستخدم هذا المصطلح لوصف مفهوم جديد يتضمن وسوم التتبع وأجهزة الحواسيب المدعّمة بمُستشعرات متصلة بالإنترنت، والتي يمكنها جمع البيانات لتحسين عمليات سلسلة التوريد الخاصة بالشركة.

مراحل التطور The Phases of the Evolution

مرت عملية تطور الإنترنت بأربعة مراحل حددت أيضًا التطور الذي حدث في إنترنت الأشياء.

مرحلة الاتصال Connectivity Phase

في السنوات الأولى لظهور الإنترنت، اقتصر الاتصال بالإنترنت على المؤسسات والجامعات ولم يكن متاحًا لعامة الناس على نطاقٍ واسع. مكّنت هذه المرحلة بعض الأفراد من الحصول على المعلومات بسهولة عند الوصول إلى الويب.

الاقتصاد الشبكي Networked Economy

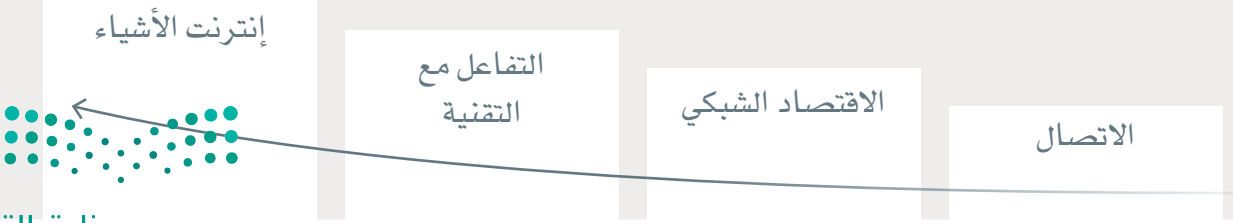
مع التقدم السريع للتقنية، استمرت سرعات الاتصال بالشبكات بالازدياد، ولم تَعُدْ عملية الاتصال هي العقبة الأساسية. ركزت هذه المرحلة على زيادة الكفاءة والربح من خلال الشبكات.

التفاعل مع التقنية Immersive Experiences

تميزت هذه الحقبة الزمنية بظهور وسائل التواصل الاجتماعي والتعاون وانتشار الأجهزة على نطاق واسع. تم فيها رقمنة التفاعلات البشرية، وتحول التطبيقات تدريجيًا إلى البنية التحتية السحابية.

إنترنت الأشياء Internet of Things

اهتمت هذه المرحلة الأخيرة بتوفير الاتصال وعمليات البيانات بين جميع الأجهزة المتصلة بالإنترنت تقريبًا، وذلك لتقديم حلول تقنية متقدمة لمختلف القطاعات والصناعات.

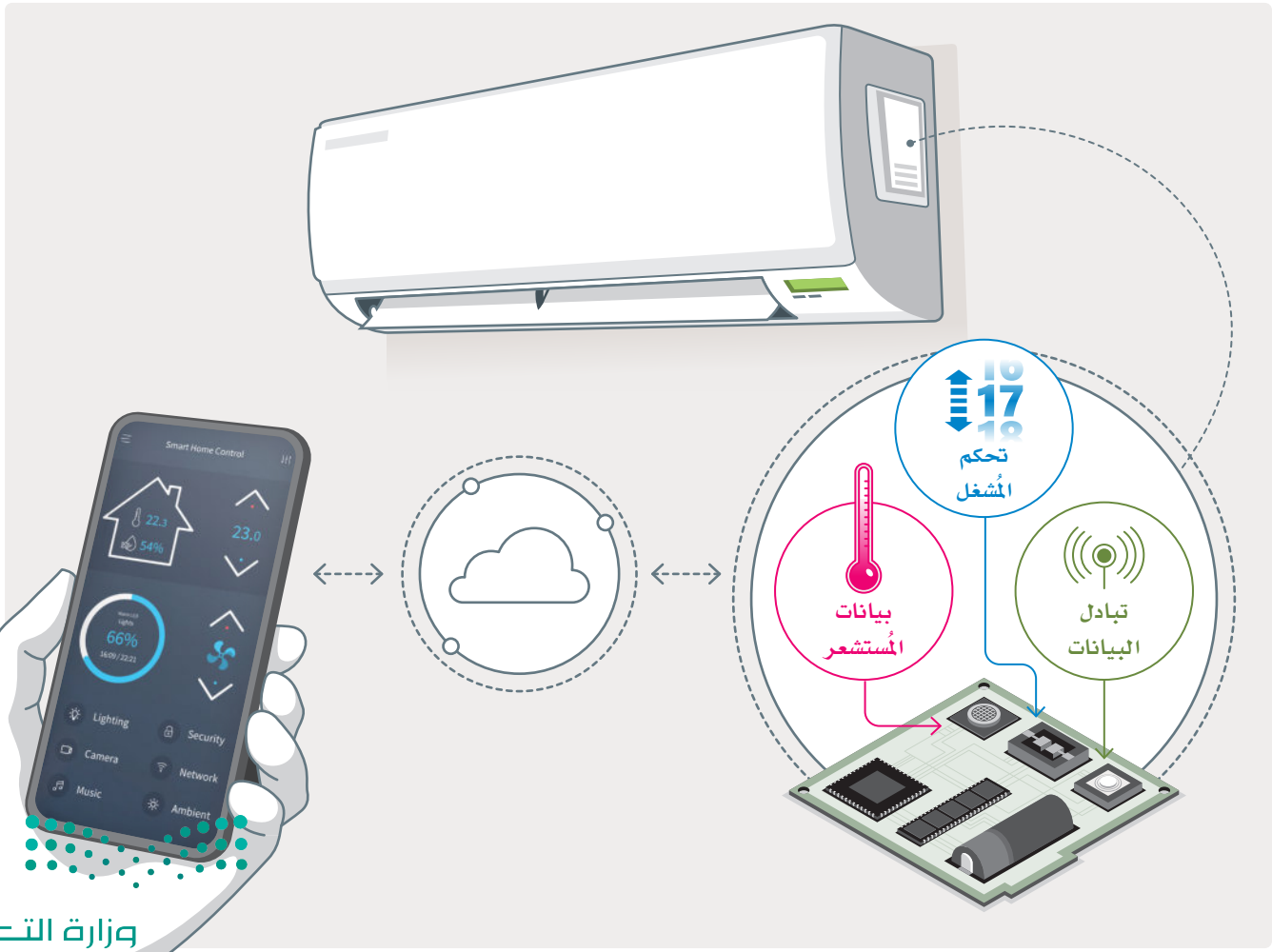


شكل 1.4: ثورة الإنترنت

من الضروريّ إدراك أنّ إنترنت الأشياء هو عبارة عن مجموعة من التقنيات والأُطُر المترابطة. فكما أن شبكة الإنترنت تربط بين أجهزة الحاسب والمحتوى، فإن إنترنت الأشياء يربط الأجهزة والبيانات والأشخاص معًا. كذلك فإن ازدياد الترابط بين التقنيات والأنظمة، وتسارع عملية جمع البيانات، يُكسب هذا العالم المتصل قوة ويزيده قيمة. تقود التقنيات الناشئة مثل: إنترنت الأشياء والذكاء الاصطناعي (AI) والروبوتات، التحول الرقمي والذي يُعرف بالثورة الصناعية الرابعة.

ما الذي يفعله إنترنت الأشياء؟ What the Internet of Things Does

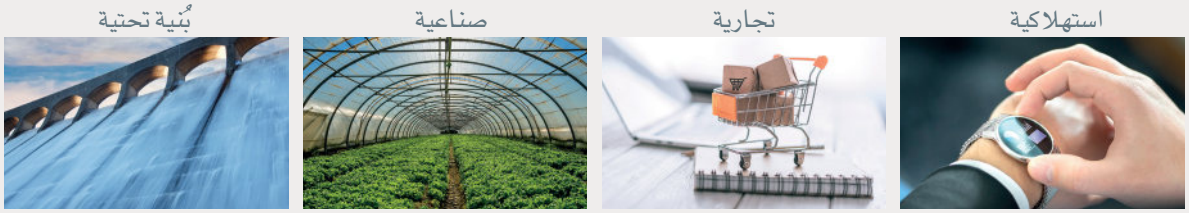
ساهم إنترنت الأشياء في تحقيق إنجازات تقنية وإنتاج أنواع جديدة من المنتجات والخدمات. وتكمن أهمية إنترنت الأشياء في إتاحة نقل البيانات المُجمّعة (التي تم التوصل إليها) من حيز محدد إلى مركز بيانات، ثم إلى جميع أنحاء العالم. يمكن معالجة البيانات بجانب المعلومات الأخرى في موقع مركزي، ويمكن القيام بهذا الإجراء في الموقع نفسه من خلال مُشغل. فعلى سبيل المثال، يستشعر منظم الحرارة في غرفتك كلًا من درجة الحرارة ودرجة الرطوبة، وتجمع خوارزمية عملية معالجة هاتين القيمتين مع بيانات الطقس في مدينتك، وذلك لتُشغل نظام تكييف الهواء بناءً على خوارزمية الذكاء الاصطناعي، والتي من شأنها التوفير في استهلاك الطاقة. يحدث كل هذا في الوقت الفعلي ودون أي تدخل بشري.



شكل 1.5: استشعار منظم الحرارة في غرفتك لدرجة الحرارة

تُصنّف تطبيقات إنترنت الأشياء إلى أربعة مجالات رئيسة: **استهلاكية وتجارية وصناعية وبُنية تحتية**.

تُصنَّفُ الأجهزة القابلة للارتداء والمنازل الذكية في مجال إنترنت الأشياء الاستهلاكية، أما إنترنت الأشياء التجارية فيوجد في المدارس والمكاتب ومحلات البيع بالتجزئة. وتستخدمُ تطبيقات إنترنت الأشياء الصناعية على نطاق واسعٍ في المصانع والمزارع وشبكات النقل، أما إدارة الطاقة والمياه فتُصنّف في مجال إنترنت الأشياء في البنية التحتية.



شكل 1.6: المجالات الأساسية لتطبيقات إنترنت الأشياء

لقد أصبح إنترنت الأشياء جزءًا لا يتجزأ من الحلول التقنية الحديثة، وبات يُدمج بصورة متزايدة مع التقنيات الأخرى مثل الذكاء الاصطناعي وعلم الروبوت، والتي تُوظّف لتحسين تطبيقات إنترنت الأشياء، أو دعمها بالكائنات الإضافية . أتاح هذا الأمر تطبيق مجموعة من التقنيات الحالية والناشئة لحل المشكلات الحالية والجديدة بأكثر الطرائق فعالية. يوضح الجدول 1.1 تطبيقات التقنيات الناشئة المُحسنة بواسطة تقنيات إنترنت الأشياء.

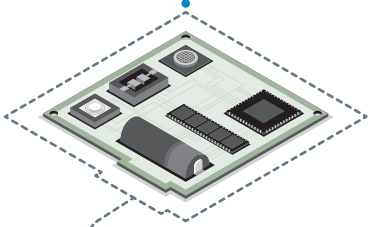
الجدول 1.1: التطبيقات المُحسّنة من خلال إنترنت الأشياء

التطبيق	الوصف
الأتمتة (Automation)	يمكن العثور على الآليات والتقنيات التي ساعدت في الأتمتة على مدى العصور. غالبًا ما تؤدي أتمتة الأنشطة المختلفة إلى زيادة السرعة والكفاءة والسلامة وتقليل التكلفة. تشمل عمليات الأتمتة في الوقت الحاضر المنازل الذكية، والمباني الذكية، والمصانع الذكية، والتي تشتمل على المنتجات الذكية المختلفة: أدوات التحكم في الإضاءة ومكبرات الصوت الذكية، وأنظمة الأمان، والروبوتات.
الرؤية الحاسوبية (Computer Vision)	تُمكّن المُستشعرات المقترنة بخوارزميات الذكاء الاصطناعي أجهزة الحاسب من فهم الصور ومقاطع الفيديو بالطريقة نفسها التي يقوم بها الإنسان، ولكن بقدراتٍ فائقة. إن تقنيات التعرف على الوجوه ومعالجة الصور تساعد الطائرات دون طيار والمركبات ذاتية القيادة على الملاحة وتجنب الاصطدامات. تقوم هذه التقنيات أيضًا بتحسين نماذج التعلم الآلي وذلك لتقييم دقة العلاج الكيميائي والعلاجات الأخرى من خلال تحليل الصور والمسح الضوئي، وتُعدّ جميعها تطبيقات تقنية حديثة. يمكن لهذا الجانب من التطبيقات الصناعية زيادة معدل اكتشاف الأخطاء بنسبة 90 بالمائة أو أكثر في العمليات المختلفة.
معالجة اللغات الطبيعية (NLP)	يستخدم هذا المجال اللغويات والحوسبة والذكاء الاصطناعي لفهم ومحاكاة اللغة البشرية. تُعدُّ أليكسا (Alexa) وسيري (Siri) ومساعد قوقل (Google Assistant) واجهات مستخدم لمعالجة اللغات الطبيعية القياسية، وقد انتشرت هذه الواجهات الصوتية بوتيرة سريعة في الأجهزة والمعدات المختلفة. تُطبّق هذه التقنية أيضًا على روبوتات المحادثة وخدمات الويب الآلية التي تتطلب الكتابة أو التحدث، كما يقوم الباحثون أيضًا بتطوير أنظمة للتعرف على العواطف والمشاعر.

التطبيق	الوصف
تعلُّم الآلة (Machine Learning)	يُعدُّ تعلم الآلة فرعًا من فروع الذكاء الاصطناعي، وتتنبأ التقنية المرتبطة به بالنتائج المستقبلية لسيناريوهات مختلفة، وتفسّرها باستخدام نماذج رياضية تُدرَّب باستخدام ما يسمى "بيانات التدريب". يساعد التعلم الآلي الأنظمة الموزعة عالميًا داخل إنترنت الأشياء على إكمال المهام دون برمجة محددة مما يفيد على وجه الخصوص في عمليات المراقبة والتنبؤ وتطبيقات القياس عن بعد.
إيدج للذكاء الاصطناعي (Edge AI)	ازداد عدد الأجهزة الرقمية القادرة على معالجة البيانات بشكل مستقل. حيث تُنقل البيانات من المُستشعرات المدمجة في الأجهزة مثل الروبوت أو السيارات ذاتية القيادة أثناء قيام نظام **إيدج للذكاء الاصطناعي** (Edge AI) بالعمليات الرياضية، ويقوم الجهاز بتخزين النتائج. في بعض الحالات قد تُنقل هذه البيانات سحابيًا. وتتيح هذه البنية للأجهزة العمل بشكل أسرع وبصورة أكثر ذكاءً وبطاقة أقل. لقد ساهم هذا المجال في تغيير آليات تشغيل الأجهزة المستقلة، وأتاح إطالة عمر بطارية المُستشعرات لسنوات.
التحليلات المتقدمة (Advanced Analytics)	نظرًا للطبيعة المشتتة للبيانات، تختلف الإجراءات التحليلية لتلك البيانات في إنترنت الأشياء، حيث يقوم البرنامج مثلًا بتجميع وتفسير البيانات المناسبة. يُعدُّ التحليل المرتكز على إنترنت الأشياء مفيدًا للغاية في عمليات التصنيع والرعاية الصحية والنقل والخدمات المالية والطاقة والاتصالات وأتمتة المنازل.
علم الروبوت (Robotics)	شهدت الآلات المستقلة مثل الطائرات دون طيار والروبوتات المحمولة والمركبات ذاتية القيادة تطورًا كبيرًا بسبب الذكاء الاصطناعي وتقنيات الاستشعار القوية، فظهر مفهوم جديد وهو **إنترنت التقنيات الروبوتية** (IoRT)، والذي يشير إلى الأنظمة التي تراقب الأحداث من حولها، وتحسب البيانات الموجودة داخليًا أو سحابيًا، لكي تَستخدِم هذه المعلومات في التعامل مع العالم الحقيقي.
الواقع المعزز (AR)	تكمُن قوة الواقع المعزز في قدرته على تعديل العالمين الافتراضي والواقعي ودمجهما. تُستخدم تطبيقات الواقع المعزز في الهواتف الذكية في الأعمال التجارية لإدخال التحسينات على الصور، ولتجربة الملابس رقميًا، ولممارسة الألعاب المختلفة. تستخدم نظارات الواقع المعزز المختلفة في التدريب والهندسة والمجالات المختلفة. يُنشأ النص والرسومات في بيئة الواقع المعزز بواسطة **محرك تقديم** (Rendering Engine) يتلقى البيانات المناسبة من إنترنت الأشياء ويوصلها إلى الجهاز.
الواقع الافتراضي (VR)	أصبحت عمليات المحاكاة ثلاثية الأبعاد الإبداعية المُنشأة بواسطة الحاسب تتطلب وجود البنية التحتية لإنترنت الأشياء. فعلى سبيل المثال، تحولت أنظمة المؤتمرات عبر الفيديو إلى أماكن واقع افتراضي تُمكن الأفراد من جميع أنحاء العالم من الانضمام إلى اجتماع أو المشاركة في ندوة عبر الإنترنت، أو حضور مؤتمر افتراضي من خلال شاشة ثنائية الأبعاد مثل جهاز حاسب محمول أو هاتف ذكي أو نظارات مخصصة. يتيح الواقع الافتراضي دمج مجموعة من الأجزاء من مواقع مختلفة داخل عالم افتراضي واحد.
تقنية سلسلة الكُتل (Blockchain)	تلعب تقنية سلسلة الكُتل التي ارتبطت في بدايتها بالعملات الرقمية دورًا مهمًا في إنترنت الأشياء. فيمكن مراقبة البيانات والمصادقة عليها أثناء مرورها للأجهزة وقواعد البيانات والخدمات المصغرة. وبالتالي يمكن أن تساعد في الأتمتة واكتشاف المخالفات مثل التلاعب أو التزوير. يفيد هذا في سياق إنترنت الأشياء اللامركزي بشكلٍ خاص، حيث تمر البيانات باستمرار عبر المؤسسات والخوادم والأنظمة.

مكونات تطبيق إنترنت الأشياء
The Components of an IoT Application

الكائن: جهاز إنترنت الأشياء.



يتكون تطبيق إنترنت الأشياء من أجهزة وبرامج ومكونات بنية تحتية، يُعدّ بعضها ضروريًا، بينما يعتمد البعض الآخر على نوع التطبيق نفسه. المكون الرئيس هنا هو "الشيء أو الكائن"، أي جهاز إنترنت الأشياء الذي يتفاعل مع بيئته بطرائق مختلفة. قد يحتوي جهاز إنترنت الأشياء على مُستشعرات أو مُشغّلات، ولكن يجب تزويده بمتحكّم دقيق مُدعم بمصدر للطاقة وذاكرة ووحدة اتصال بالشبكة لتبادل البيانات عبر تلك الشبكة. قد تحتوي بعض أجهزة إنترنت الأشياء على شاشات أو أزرار للتفاعل مع المستخدم. تُعدُّ آلية الاتصال أمرًا بالغ الأهمية، حيث تُنَفّذ عادةً من خلال بوابات إنترنت الأشياء المتخصصة وبروتوكولات الشبكة المحسّنة، فهي تُمكّن جهاز إنترنت الأشياء من الاتصال بأجهزة الحوسبة المحلية أو المركزية أو السحابية.

قد تكون مجرد أجهزة حاسب بسيطة، أو خوادم سحابية في مراكز البيانات الضخمة في دولة أخرى. يتعامل الخادم مع البيانات المُخزنة ليعالجها وليحدد ما إذا كانت ستُرسل أوامر لإجراءات معينة مرة أخرى إلى جهاز إنترنت الأشياء. تُنشأ تحليلات البيانات لتوفير رؤى مفيدة حول استخدام تطبيق إنترنت الأشياء، خاصةً عند مشاركة الآلاف أو ملايين الأجهزة في بعض الحالات. سوف تستكشف جميع مكونات تطبيق إنترنت الأشياء بالتفصيل في الدروس القادمة.

الجدول 1.2: أمثلة على أجهزة إنترنت الأشياء

تلفاز ذكي.
سماعات أذن ذكية.
تحكم المنزل الذكي مثل أليكسا (Alexa).
نظام الشبكة اللاسلكية (WiFi).
منظم حرارة ذكي مع حساسات للغرفة.
مراقب الصحة أو اللياقة البدنية.
مكيف الهواء الذكي.
أضواء بمفاتيح ومصابيح ذكية .
مقابس الطاقة الذكية.
مراقب جودة الهواء.
مراقب استهلاك الكهرباء.
ثلاجة متصلة بالإنترنت.
نظام الحماية المنزلي المزود بكاميرا جرس الباب الذكية.
باب مرآب بقفل ذكي.

شكل 1.7: مساعد قوقل المنزلي الذكي

تمرينات

1

حدد الجملة الصحيحة والجملة الخاطئة فيما يلي:	صحيحة	خاطئة
1. يتكون إنترنت الأشياء من شبكة من الأجهزة المترابطة التي تتواصل ببعضها.	○	○
2. يتضمن إنترنت الأشياء الأجهزة غير المتصلة بالإنترنت.	○	○
3. تعمل الكائنات الذكية بصورة مستقلة دون تدخل بشري.	○	○
4. الكائنات الرقمية هي الأجهزة التي ترسل البيانات وتستقبلها فقط.	○	○
5. تطورت شبكة الإنترنت في البداية من مشروع ARPAnet.	○	○
6. ساعدت وسائل التواصل الاجتماعي خلال مرحلة التفاعل مع التقنية على تسريع استخدام البيانات الضخمة.	○	○
7. تقدم أبحاث إنترنت الأشياء والذكاء الاصطناعي في وقت واحد لتطوير تطبيقات مشتركة.	○	○
8. لا يمكن استخدام البيانات من مُستشعرات إنترنت الأشياء لتطبيقات الواقع الافتراضي.	○	○
9. تتواصل الكائنات الذكية حصريًا مع بعضها البعض.	○	○
10. لا يمكن لشيء بسيط مثل باب المرآب أن يكون جزءًا من نظام إنترنت الأشياء.	○	○

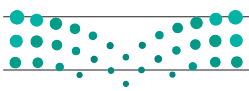
2

حدد السمات الرئيسة لإنترنت الأشياء، والتي تميزها عن التقنيات الناشئة الأخرى. اعرض أفكارك أدناه.

3 هل يمكنك التفكير في التطور التقني الأكثر أهمية في التاريخ الحديث، والذي جعل إنترنت الأشياء ممكنًا؟ اعرض أفكارك أدناه.

4 أيٌّ من مراحل الإنترنت الأربع تعتقد أنها كانت الأفضل تأثيرًا من الناحية التقنية والاقتصادية؟ اعرض أفكارك أدناه.

5 أيٌّ من التقنيات الناشئة التي تقود الثورة الصناعية الرابعة هي الأكثر تأثيرًا على الاقتصاد؟ اعرض أفكارك أدناه.



6 ابحث في الإنترنت عن مثال لتطبيق إنترنت الأشياء يوظف الرؤية الحاسوبية ومعالجة اللغة الطبيعية. دوّن ما عثرت عليه هنا وقم بوصفه.

7 ابحث في الإنترنت عن معلومات حول تطبيق للواقع الافتراضي ينشئ بيئات افتراضية مزودة ببيانات مُدخلة من مُستشعرات إنترنت الأشياء.

8 ابحث في الإنترنت عن معلومات حول كيفية مساعدة تقنيات سلسلة الكتل في إنشاء أنظمة إنترنت الأشياء اللامركزية وتطبيقاتها.

رابط الدرس الرقمي

www.ien.edu.sa

الدرس الثاني

أجهزة إنترنت الأشياء

ما المقصود "بالأشياء"؟ What is a "Thing"

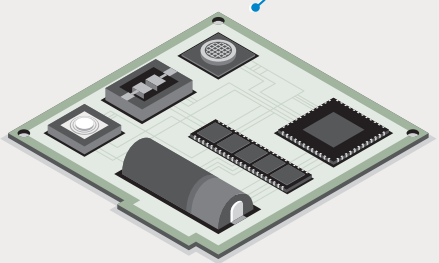
الكائنات الذكية The Smart Objects

إن "الأشياء" أو "الكائنات الذكية" هي اللبنات الأساسية لإنترنت الأشياء، فهي أجهزة محوسبة صغيرة منخفضة التكلفة تتفاعل مع بيئتها المادية المحيطة بها، وذلك بجمع البيانات من المُستشعرات، والتفاعل الفوري مع هذه البيانات عبر المُشغّلات. تقوم المُستشعرات والمُشغّلات بتحويل الأشياء اليومية إلى كائنات ذكية قادرة على الحصول على المعلومات من بيئتها وتتفاعل معها بطريقة مفيدة. وتكمن القوة الحقيقية للكائنات الذكية في حلول إنترنت الأشياء التي تربطها ببعضها، بدلًا من عملها بشكلٍ مستقلٍ كأجهزة قائمة بذاتها.

تُشغّل الكائنات الذكية بواسطة مصدر للطاقة مثل الشبكة الكهربائية أو البطارية، أو بمصدر للطاقة الذاتية من خلال الطاقة الشمسية أو طاقة الرياح. يُعدُّ استهلاك هذه الكائنات للطاقة منخفضًا جدًا لدرجة أنه في بعض الأحيان يمكن تشغيل الكائن الذكي لأشهر أو سنوات باستخدام البطاريات. يوجد جيل جديد من الكائنات (المُستشعرات) الذكية الخاصة بالصحة، والتي يمكن تشغيلها بالتيار الكهربائي المنبعث من جسم الإنسان.

يحتوي كل كائن ذكي على جهاز اتصال يرسل البيانات التي تُجمع من المُستشعرات ويتلقى التعليمات اللازمة للمُشغّلات. ويقوم جهاز الاتصال بتوصيل الكائن الذكي بالتخزين السحابي.

حاسب صغير منخفض التكلفة يُدمج في الكائنات لجعلها ذكية، ويمكن تنظيمها في شبكات.



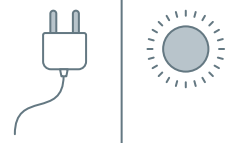
شكل 1.8: حاسب صغير منخفض التكلفة

المكون المشترك هو وحدة المعالجة على هيئة جهاز التحكم الدقيق.

يقوم جهاز التحكم الدقيق بالتنسيق بين المُستشعرات والمُشغّلات وجهاز الاتصال. إن أجهزة التحكم الدقيقة المستخدمة في التطوير أو الأغراض الأكاديمية مثل: **الأردوينو** (Arduino) أو **رازبيري باي** (Raspberry Pi)، هي عبارة عن حواسيب صغيرة.

تستخدم تطبيقات إنترنت الأشياء الفعلية وحدات تحكم دقيقة صغيرة الحجم يصل حجمها أحيانًا إلى 2×2 ملم. أحد الأمثلة على ذلك شبكة **الغبار الذكي** (Smart dust)، وهي شبكة لاسلكية لمنصات حوسبة واستشعار لا يتجاوز حجمها حبة الرمل الواحدة والتي يمكنها أن تعمل بمفردها. يمكن لشبكة الغبار الذكي أن تستشعر أشياء مثل الإضاءة ودرجة الحرارة والصوت ووجود السموم أو الاهتزازات ثم تسجّل هذه المعلومات وترسلها لاسلكيًا إلى أنظمة الحاسب المركزية.

تصنيفات الكائنات الذكية Classifications of Smart Objects

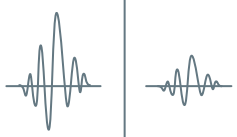


تعمل بالطاقة الذاتية أو تتصل بمزود للطاقة

قد يحتوي الكائن على مصدر الطاقة الذاتي الخاص به كالبطارية أو الألواح الشمسية، أو يمكن أن يعمل باستمرار بواسطة مصدر خارجي. يمكن للكائنات التي تعمل بالطاقة الذاتية أن تكون متحركة، أمّا الكائنات التي تُزود ببطاريات فتكون مقيدةً من حيث فترة استخدامها وقدرتها على جمع البيانات وإرسالها.

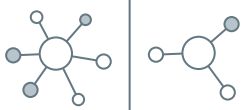
متحرك أو ثابت

يمكن للكائن الذكي أن يكون متحركًا، ويمكنه أن يبقى ثابتًا في مكانه. كما يمكن أن يكون الجسم متحركًا إذا تم توصيله بجسم متحرك أكبر.



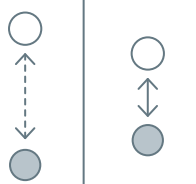
معدل إرسال بيانات منخفض أو مرتفع

يمكن أن تكون عمليات إرسال البيانات الخاصة بعمليات المراقبة للكائن الذكي منخفضة أو مرتفعة. فمثلًا قد يُرسل مُستشعر الصدأ الموجود على أحد الجسور القيم المسجلة شهريًا، وقد يُرسل مُستشعر الحركة في السيارة قيم التسارع مئات المرات في الثانية. تؤدي مُعدلات التقارير المرتفعة إلى ازدياد استهلاك الطاقة مما قد يفرض قيودًا على مصدر الطاقة.



بيانات بسيطة أو معقدة

يعتمد هذا التصنيف على كمية البيانات التي تم جمعها وتبادلها خلال كل دورة تقارير. يمكن لمُستشعر الرطوبة في حقل زراعي تسجيل قيمة يومية غير معقدة، في حين يُسجل مُستشعر المحرك مئات القيم، مثل درجة الحرارة والضغط وسرعة الغاز وسرعة الانضغاط.يُحدّد معدل نقل البيانات بناء على عاملين هما تصنيف البيانات (بسيطة أو معقدة)، و معدل إرسال البيانات (منخفض إلى مرتفع). والنتيجة هنا عبارة عن مقياس مدمج. قد ينقل كائن متوسط الإنتاج بيانات غير معقدة بمعدل مرتفع نسبيًا (في هذه الحالة يظهر مخطط التدفق بصورة متصلة)، أو قد ينقل بيانات كثيرة بمعدل منخفض نسبيًا (وفي هذه الحالة يبدو مخطط التدفق متقطعًا).



نطاق التقرير

تحدد المسافة بين الكائن الذكي وجامع البيانات نطاق التقرير. فبالنسبة للسوار الرياضي الذي يرسل البيانات إلى هاتفك على سبيل المثال، يصل نطاق التقرير إلى بضعة أمتار، وفي المقابل قد يحتاج مُستشعر الرطوبة المضمن في سطح الأسفلت في الطريق إلى الاتصال بهوائي يقع على بعد مئات الأمتار أو حتى كيلومترات.

كثافة الكائنات في كل خلية

يعتمد هذا التصنيف على كمية الكائنات الذكية ذات احتياجات الاتصال المتماثلة والمتصلة بنفس البوابة. قد يستخدم خط أنابيب النفط مُستشعرا واحدًا كل بضعة أميال. وفي المقابل تستخدم تلسكوبات علم الفلك المئات أو حتى الآلاف من المرايا على مساحة صغيرة يزود كل منها بمقاييس الجايروسكوب (أداة لتحديد زاوية الاتجاهات والدوران) ومُستشعرات للجاذبية والاهتزاز.

المكونات الرئيسة للكائن الذكي The Main Components of a Smart Object

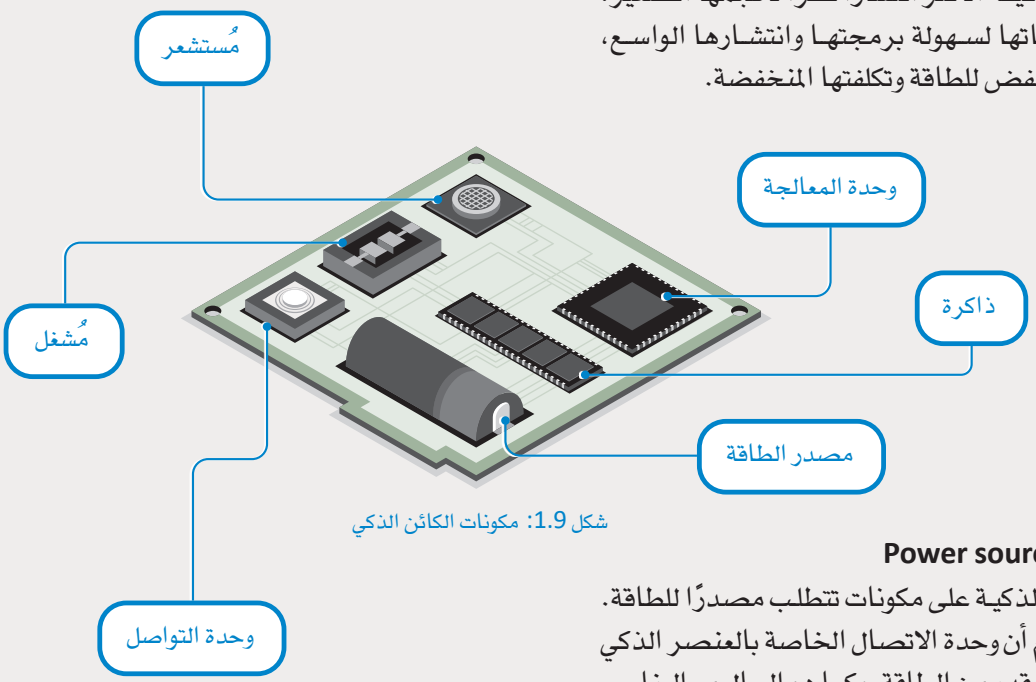
إن الكائن الذكي هو جهاز يمتلك المكونات الأربعة المدرجة أدناه كحدٍ أدنى. قد يحتوي الكائن الذكي على مُستشعر واحد فقط أو على مجموعة من المستشعرات، وكذلك على مُشغل واحد فقط أو على مجموعة من المُشغّلات، وكذلك على مجموعة من المُشغّلات والمستشعرات معًا، وذلك حسب الغرض من تطبيق إنترنت الأشياء.

وحدة المعالجة Processing unit

يحتوي الكائن الذكي على وحدة معالجة لجمع البيانات ومعالجتها وتحليل معلومات الاستشعار التي يتلقاها المُستشعر (أو المُستشعرات)، ولتنسيق إشارات التحكم إلى أي مُشغل، وتشغيل مجموعة متنوعة من العمليات بما فيها أنظمة الاتصالات والطاقة. يمكن أن يختلف نوع وحدة المعالجة المستخدمة بشكل كبير وذلك اعتمادًا على متطلبات المعالجة لتطبيق محدد. تُعدّ أجهزة التحكم الدقيقة الأكثر انتشارًا نظرًا لحجمها الصغير، وتتعدد استخداماتها لسهولة برمجتها وانتشارها الواسع، ولاستهلاكها المنخفض للطاقة وتكلفتها المنخفضة.

المُستشعرات والمُشغّلات Sensors and Actuators

يستطيع الكائن الذكي التفاعل مع العالم المادي عبر المُستشعرات والمُشغّلات الخاصة به. فليس ضروريًا أن يضم الكائن الذكي كلًا من المُستشعرات والمُشغّلات، فقد يحتوي الكائن الذكي على واحد أو أكثر من المُستشعرات والمُشغّلات اعتمادًا على نوع التّطبيق.



شكل 1.9: مكونات الكائن الذكي

مصدر الطاقة Power source

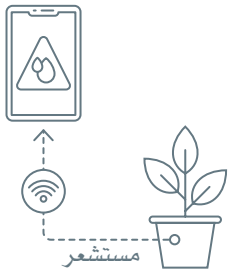
تحتوي الكائنات الذكية على مكونات تتطلب مصدرًا للطاقة. من المثير للاهتمام أن وحدة الاتصال الخاصة بالعنصر الذكي تستهلك عادةً أكبر قدر من الطاقة. وكما هو الحال مع العناصر الثلاثة الأخرى للأشياء الذكية، تختلف احتياجات الطاقة بشكل كبير حسب التطبيقات. فعادةً تعمل الكائنات الذكية بطاقة محدودة، ويمكنها الاستمرار لمدة زمنية طويلة، وبشكل خاص عندما يكون من الصعب الوصول إليها. ويتطلّب هذا المزيج كفاءة الطاقة، وإدارتها بفعالية، وتفعيل أوضاع السكون، ووجود أجهزة استهلاك طاقة منخفضة للغاية، وغيرها، خاصةً عند اعتماد الكائن الذكي على طاقة البطارية. عند وجود عناصر ذكية بتركيبات طويلة المدى ولجميع المقاصد والأغراض تُستخدم عادةً مصادر بديلة لتوفير الطاقة.

وحدة الاتصالات Communication unit

وحدة الاتصالات مسؤولة عن ربط العنصر الذكي بأشياء ذكية أخرى وبالعالم الخارجي (بواسطة الشبكة). يمكن أن تكون أجهزة اتصالات الكائنات الذكية سلكية أو لاسلكية. وتُربط العناصر الذكية في شبكات إنترنت الأشياء لاسلكيًا لعدة أسباب، أهمها التكلفة ومحدودية توافر البنية التحتية وسهولة التنفيذ. توجد العديد من بروتوكولات الاتصال للعناصر الذكية.

المُستشعرات The Sensors

يُنفِّذ المُستشعر ما يشير إليه اسمه، فهو يستشعر. فعلى وجه التحديد، يقيس المُستشعر كمية فيزيائية ويحولها إلى بيانات يمررها لتُستخدم بواسطة الأجهزة الذكية أو الإنسان. لا تقتصر وظيفة المُستشعرات على جمع البيانات الحسية المشابهة لحواس الإنسان، فهي توفر مجموعة واسعة من بيانات القياس وبدقة أكبر من الحواس البشرية. يمكن تضمين المُستشعرات في أي كائن مادي وتوصيلها بالإنترنت عن طريق الشبكات السلكية أو اللاسلكية.

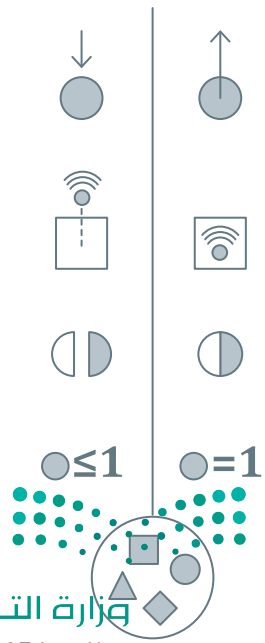


شكل 1.10: مستشعر يجمع البيانات ويرسلها من أصيص النبات إلى نظام إنترنت الأشياء

تحتوي السيارة الحديثة على مجموعة متنوعة من المُستشعرات التي توفّر كمًا هائلًا من البيانات يمكن استخدامها في أنظمة ذكية، أو مشاركتها مع المركبات الأخرى على الطريق. يمكن للسائق فحص كل شيء بالسيارة والتحكم بها باستخدام مجموعة متنوعة من المُستشعرات مثل قياس درجة حرارة الماء والزيت، والموقع وضغط الإطارات والسرعة، والتي توفر البيانات ذات العلاقة لتحسين السلامة وصيانة السيارة.

شكل 1.11: مُستشعرات سيارة حديثة

تصنيف المُستشعرات Classification of Sensors



نشطة أو سلبية

يمكن تصنيف المُستشعرات بناءً على مصدر الطاقة، فإذا كانت تتطلب مصدر طاقة خارجي للعمل وتنقل الطاقة وتكتشفها في نفس الوقت فهي مستشعرات (نشطة)، وإذا كانت لا تتطلب توفر أي مصدر طاقة خارجي ولا تنقل الطاقة، ولكن تكتشفها فقط فهي مستشعرات (سلبية).

توغلية أو غير توغلية

يمكن أن تكون المُستشعرات جزءًا من البيئة التي تقيسها (توغلية)، أو قد تكون مكونًا خارجيًا (غير توغلية).

تلامسية أو غير تلامسية

قد تتطلب المُستشعرات ملامسة مادية للجسم الذي يتم قياسه (تلامسية) وقد لا تتطلّب ذلك (غير تلامسية).

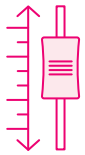
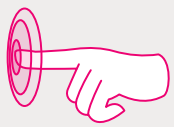
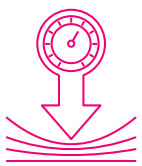
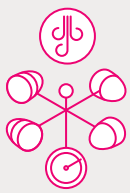
مُطلقة أو نسبية

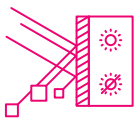
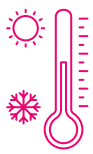
يمكن للمُستشعرات جمع البيانات وفق مقياس مطلق، أو نسبةً إلى قيمةٍ مرجعية.

مجال التطبيق

يمكن تصنيف المُستشعرات وفقًا للتطبيق المحدد الذي تستخدم به.

الجدول 1.3: أنواع المُستشعرات وأمثلتها

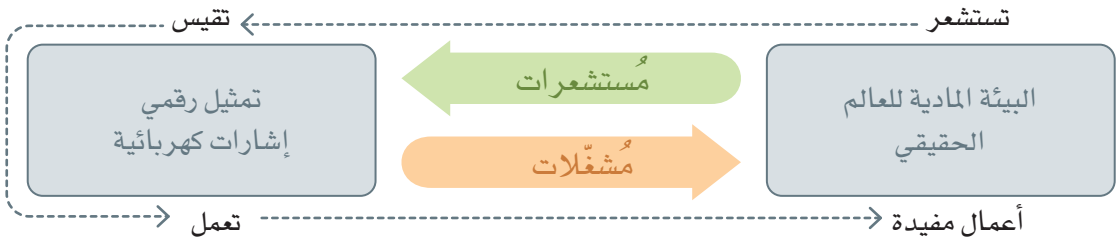
النوع	الوصف	أمثلة
 الموضع	يقيس هذا المُستشعر موضع الجسم؛ فيمكن أن يكون القياس بشكل مطلق أو نسبي. هناك ثلاثة أنواع من مُستشعرات الموضع: خطية وزاوِيَّة ومتعددة المحاور.	مقياس الجهد، ومقياس الميل، ومُستشعر القُرب.
الإشغال والحركة	تكتشف مُستشعرات الإشغال وجود الأشخاص والحيوانات في المنطقة التي تُراقب، بينما تكتشف مُستشعرات الحركة حركة الأشخاص والأشياء. وتبعث مُستشعرات الإشغال إشارة حتى عندما يكون الشخص خاملًا، على عكس مستشعراتِ الحركة.	عين كهربائية، رادار.
السرعة والتسارع	قد تكون مُستشعرات السرعة خطية أو زاويّة، مما يشير إلى مدى سرعة تحرك الجسم في خط مستقيم أو مدى سرعة دورانه. وتقيس مُستشعرات التسارع تغيرات السرعة.	مقياس التسارع، والجايروسكوب.
 القوة	تحدد مُستشعرات القوة الحالة التي تُطبّق بها قوة فيزيائية معينة.	مقياس القوة، مقياس اللزوجة، مُستشعر اللمس.
 الضغط	على غرار مُستشعرات القوة، تقيس مُستشعرات الضغط القوة الناجمة عن ضغط السوائل أو الغازات.	بارومتر، بيزومتر.
 التدفق	تكتشف مُستشعرات التدفق معدل تدفق السوائل.	مقياس شدة الريح، مُستشعر تدفق الكتلة الحرارية، عداد المياه.

النوع	الوصف	أمثلة
صوتي	تقيس المُستشعرات الصوتية مستويات الصوت الموجودة في البيئة.	ميكروفون، مسجل أصوات تحت الأرض، مسجل أصوات تحت الماء.
رطوبة	تقيس مُستشعرات الرطوبة كمية الرطوبة في الهواء أو في أي حيِّز.	مقياس الرطوبة، مُستشعر الرطوبة، مُستشعر رطوبة التربة.
 ضوء	تكتشف مُستشعرات الضوء وجود الضوء بأنواعه ودرجاته المختلفة.	مُستشعر الأشعة تحت الحمراء، كاشف الضوء، كاشف اللهب.
إشعاعي	تكتشف مُستشعرات الإشعاع أي إشعاع في البيئة المحيطة.	عداد جيجر مولر، كاشف النيوترون.
 حراري	تحدد مُستشعرات درجة الحرارة كمية الحرارة أو البرودة داخل النظام. يجب أن تكون مُستشعرات درجة الحرارة التلامسية على اتصال بالجسم المستهدف. تعمل مُستشعرات درجة الحرارة غير التلامسية على قياس درجة الحرارة من مسافة بعيدة.	ميزان الحرارة، مقياس السعرات الحرارية، مقياس درجة الحرارة.
كيميائي	تحدد المُستشعرات الكيميائية التركيز الكيميائي داخل النظام.	جهاز قياس الكحول، كاشف الدخان.
مؤشرات حيوية	يمكن للمُستشعرات الحيوية اكتشاف الخصائص البيولوجية في الكائنات الحية.	مُستشعر الجلوكوز في الدم، مقياس أكسجين الدم، جهاز تخطيط القلب.

المُشغّلات The Actuators

تُعدُّ المُشغّلات مُكمّلة للمُستشعرات.

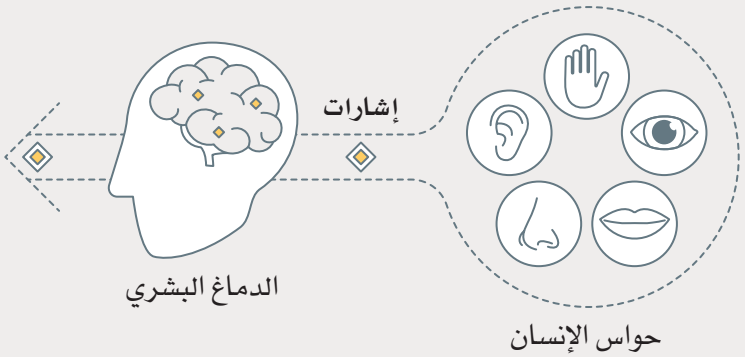
تستقبل المُشغّلات إشارة تحكم، وهي غالبًا إشارة كهربائية أو أمر رقمي يؤدي إلى تأثير فيزيائي على النظام.

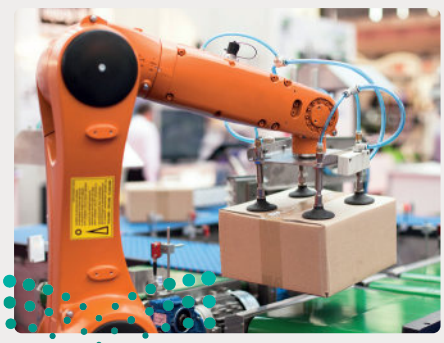


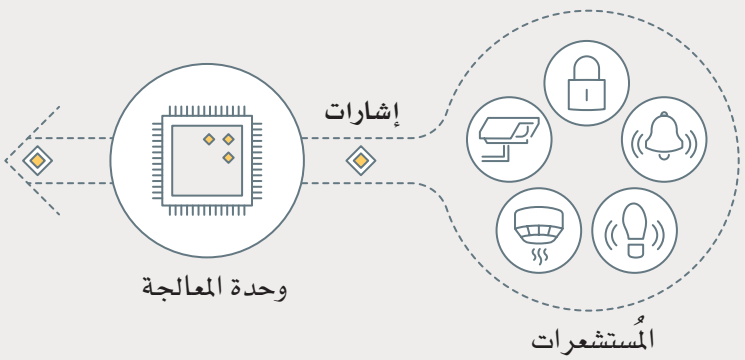
شكل 1.12: العلاقة بين المُستشعرات والمُشغّلات

التشابه مع الإنسان Human Analogy

يستخدم البشر حواسهم الخمس لاستشعار بيئتهم المحيطة وقياسها. فتقوم أعضاء الحواس بتحويل هذه المعلومات إلى نبضات كهربائية يرسلها الجهاز العصبي إلى الدماغ لمعالجتها. وكذلك فإن مُستشعرات إنترنت الأشياء هي أجهزة تستشعر وتقيس العالم الفيزيائي، وترسل قياساتها كإشارات كهربائية إلى معالج دقيق أو وحدة تحكم دقيقة من أجل المزيد من المعالجة. يتحكم الدماغ البشري بالوظيفة والحركة، ويحمل الجهاز العصبي هذه المعلومات إلى الجزء المناسب من الجهاز العضلي. وفي المقابل، يمكن للمعالج إرسال إشارة كهربائية إلى مُشغل يحول الإشارة إلى فعل ذي تأثير قابل للقياس في بيئته. يُعدّ هذا التفاعل بين المُستشعرات والمُشغّلات والمعالجات والوظائف المماثلة في الأنظمة البيولوجية الأساس لمجالات علم الروبوت والمؤشرات الحيوية.







شكل 1.13: التشابه مع البشر

تصنيف المُشغّلات Classification of Actuators

نوع الحركة
يمكن تصنيف المحركات وفقًا لنوع الحركة الصادرة منها.
أمثلة: خطي، ودوراني، وذو محور واحد، وذو محورين، وذو ثلاثة محاور.

القوه الناتجة
يمكن تصنيف المحركات وفقًا للقوة الناتجة.
أمثلة: طاقة عالية، وطاقة منخفضة، وطاقة ضئيلة.

نوع الإنتاج
يمكن تصنيف المحركات وفقًا لطبيعة المخرجات المستقرة.
أمثلة: ثنائية، ومستمرة.

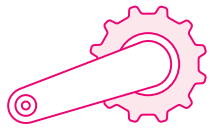
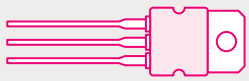
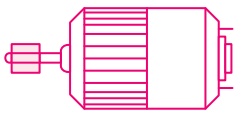
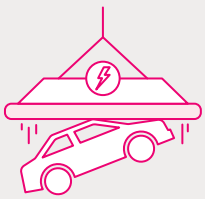
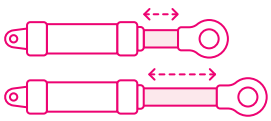
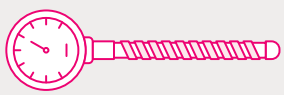
مجال التطبيق
يمكن تصنيف المحركات وفقًا لنوع الصناعة التي تُستخدم فيها المحركات.
أمثلة: التصنيع والمركبات والطب.

نوع الطاقة
يمكن تصنيف المحركات بناءً على نوع الطاقة التي تستخدمها.
أمثلة: كهربائية، وكيميائية، وحركية.

شكل 1.14: مُشغل صغير (مفتاح) بحركة ميكانيكية

شكل 1.15: نظام مُشغل هيدروليكي (مكبس) للآلات الثقيلة

الجدول 1.4: أنواع المُشغّلات مع أمثلة

نوع المشغل	أمثلة
 مُشغِّل ميكانيكي	رافعة، جاك لولبي، الساعد اليدوي.
 مُشغِّل كهربائي	ثايرستور، ترانزستور ثنائي القطب، الصمام الثنائي.
 مُشغِّل كهروميكانيكي	محرك تيار متردد، محرك تيار مستمر، محرك خطوي.
 مُشغِّل كهرومغناطيسي	مغناطيس كهربائي، ملف لولبي خطي.
 مُشغِّل هيدروليكي وهوائي	إسطوانة هيدروليكية، إسطوانة هوائية، مكبس، صمام التحكم في الضغط.
 مُشغِّلات حرارية ومغناطيسية	المواد الممغنطة، الشريط ثنائي المعدن، ثنائي الشكل الكهروإجهادي (piezoelectric bimorph).
مُشغِّلات دقيقة و مُشغِّلات نانوية	محرك إلكتروستاتيكي، صمام مايكروي، محرك مُشطي.

تمرينات

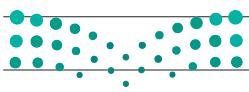
1

حدد الجملة الصحيحة والجملة الخاطئة فيما يلي:	صحيحة	خاطئة
1. الكائنات الذكية هي أجهزة إلكترونية معقدة للغاية تتطلب كميات كبيرة من الطاقة للمعالجة.	○	○
2. يتم تشغيل الكائنات الذكية حصريًا من خلال مصادر الطاقة المتجددة.	○	○
3. يمكن للكائنات الذكية إرسال البيانات من خلال مجموعة متنوعة من ترددات الاتصالات.	○	○
4. ترسل وحدة المعالجة بيانات المُستشعر التي جُمعت إلى خدمات خارجية على الإنترنت.	○	○
5. يمكن لمُستشعرات السرعة الكشف عن وجود أي كائن في البيئة المحيطة.	○	○
6. يمكن للمُستشعرات الإشعاعية الكشف عن أي قراءات حرارية في البيئة المحيطة.	○	○
7. تتماثل العلاقة بين وحدة المعالجة والمُشغل مع العلاقة بين حواس الإنسان والدماغ.	○	○
8. يمكن أن تأخذ المُشغّلات مدخلات بيانات متقطعة فقط.	○	○
9. يمكن أن تأخذ المُشغّلات مدخلات من المُستشعرات مباشرة دون الحاجة إلى خدمات البيانات الخارجية.	○	○

2 صِف المكونات الرئيسة للكائن الذكي.

3 حلل أنواع التطبيقات التي تتطلب مُستشعرات تعمل بالطاقة الذاتية، وأيها يتطلب مُستشعرات يجب تزويدها بالطاقة. اعرض أفكارك أدناه.

4 تتطلب تطبيقات إنترنت الأشياء المختلفة أنواعًا مختلفة من الكائنات الذكية. ضع قائمة بالسمات الرئيسة التي تُصنّف الكائنات الذكية بناءً عليها.

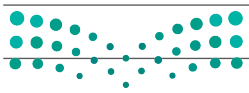


5 حدد ثلاثة أنواع من المُستشعرات المهمة المستخدمة لقياس البيئة المحيطة وناقش استخدامها.

6 كيف يُمكن لدراسة التواصل بين أنظمة جسم الإنسان المختلفة أن تساعد المهندسين على إنشاء حلول أكثر ترابطًا لأنظمة إنترنت الأشياء؟

7 قارن بين أنواع المُشغّلات الأكثر شيوعًا في التطبيقات الروبوتية.

8 عدِّد المُشغّلات التي تتطلب بيانات أكثر تعقيدًا لإنجاز المهام المطلوبة. وضح سبب ذلك.



المشروع

تتسع تقنية إنترنت الأشياء لتشمل معظم الجوانب الحياتية اليومية والعملية، فعند دمج إنترنت الأشياء في أحد التطبيقات الحياتية، تصبح الأجهزة شائعة الاستخدام كائنات ذكية مُنتجة ومستهلكة لبيانات إنترنت الأشياء.

1

اختر جهازًا إلكترونيًا شائعًا تستخدمه يوميًا وقدِّم مُقترحًا لتطبيق إنترنت الأشياء باستخدام هذا الجهاز. سيرسل هذا الجهاز البيانات ويستقبلها من نظام إنترنت الأشياء لإنشاء التوقُّعات ولتحسين كفاءته.

2

أنشئ عرضًا تقديميًا باستخدام برنامج **باوربوينت** (PowerPoint) يوضح مُقترحك، ويصف كيفية توسيعه ليشتمل على المزيد من الأجهزة من نفس النوع أو أنواع أخرى من الكائنات الذكية.

ماذا تعلّمت

- تعريف إنترنت الأشياء وتاريخه.
- التمييز بين الاستخدامات المختلفة لتطبيقات إنترنت الأشياء.
- فهم مدى مساهمة إنترنت الأشياء في تعزيز التقنيات الناشئة.
- وصف الكائنات الذكية واستخداماتها.
- تصنيف المُستشعرات والمُشغّلات الموجودة في الكائنات الذكية.

المصطلحات الرئيسة

الكائنات المادية/ الملموسة	Physical-First
مُستشعر	Sensor
كائن ذكي	Smart Object
شيء	Thing

مُشغِّل	Actuator
الكائنات الرقمية	Digital-First
إنترنت الأشياء	Internet of Things
أجهزة إنترنت الأشياء	IoT Device

2. إنترنت الأشياء في حياتنا

ستتعرف في هذه الوحدة على كيفية تمكين تقنيات الشبكات والاتصالات لأنظمة ومنصات إنترنت الأشياء. ستتعرف أيضًا على تأثير هذه التقنيات على المجتمع، وعلى تطوراتها المتوقعة في المستقبل القريب، وفي الختام سيتم تناول الجوانب الرئيسة لأنظمة إنترنت الأشياء وتحدياتها وكيفية تنظيمها.

أهداف التعلُّم

بنهاية هذه الوحدة سيكون الطالب قادرًا على أن:

- يتعرّف على مفاهيم طبقات الحوسبة السحابية والضبابية والطرفية لتطبيقات إنترنت الأشياء.
- يصف المُمكِّنات الرئيسة لأنظمة إنترنت الأشياء.
- يتعرّف على بروتوكولات الشبكات والتقنيات التي تُشكِّل بُنية الاتصالات لإنترنت الأشياء.
- يصنّف التطبيقات الرئيسة لحلول إنترنت الأشياء.
- يتعرّف على أهمية إنترنت الأشياء في المستقبل القريب.
- يحدّد المخاطر الأمنية لأنظمة إنترنت الأشياء المعقَّدة.
- يتعرّف على التحديات التقنية لأنظمة إنترنت الأشياء.
- يتعرّف على مفاهيم الرقابة والتنظيم لتطبيقات إنترنت الأشياء.

رابط الدرس الرقمي

www.ien.edu.sa

الدرس الأول

منصة إنترنت الأشياء

الإنترنت في إنترنت الأشياء The Internet in the Internet of Things

يحتوي مصطلح إنترنت الأشياء على كلمتين رئيستين: الإنترنت والأشياء. تم مُسبقًا شرح ماهية الأشياء (الكائنات الذكية)، وستستكشف الآن الجزء الخاص بالإنترنت في حلول إنترنت الأشياء. تتيح عملية الاتصال والخدمات السحابية للكائنات الذكية جمع القياسات من المُستشعرات، وإرسال أوامر للتحكم إلى المُشغّلات. عادةً ما تتصل أجهزة إنترنت الأشياء بخدمة إنترنت الأشياء السحابية باستخدام بروتوكول اتصال، ومن خلال هذه الخدمة يتخذ تطبيق إنترنت الأشياء الرئيس القرارات بناءً على البيانات المُجمَّعة. ستتعرف في هذا الدرس على بُنية الحوسبة السحابية والضبابية والطرفية، وشبكاتها وبروتوكولاتها المستخدمة، ونوع البيانات التي يجري تبادلها لدعم حل إنترنت أشياء فعّال.

بُنية الحوسبة السحابية والضبابية والطرفية
Cloud, Fog and Edge

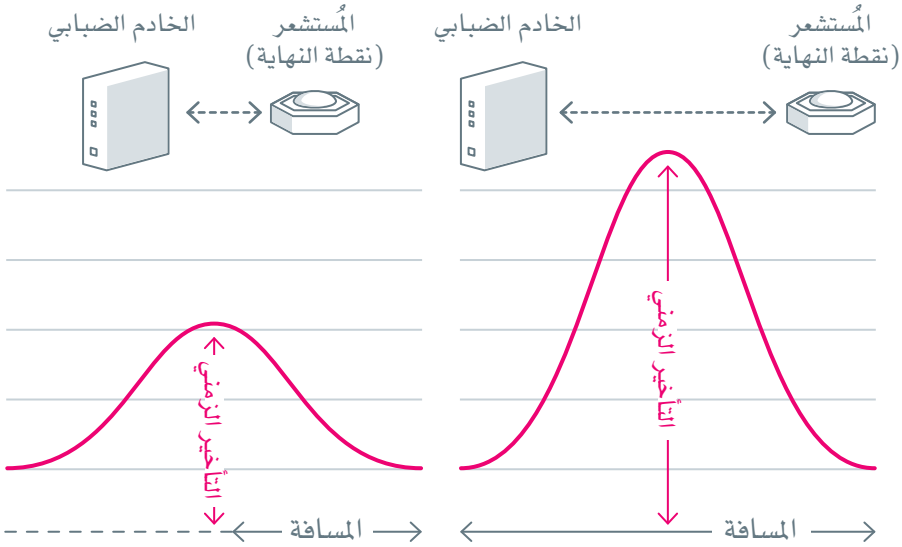
يُطلق على البُنية التحتية المحوسبة الأكثر شيوعًا اسم بنية الحوسبة السحابية والضبابية والطرفية. يصف هذا النموذج باختصار ثلاثة مستويات من التخزين والاتصال والتطبيقات، حيث تُعَدُّ الحوسبة السحابية بمثابة البُنية التحتية لمركز البيانات، بينما تُستخدم الحوسبة الطرفية لمعالجة البيانات عند أطراف الشبكة بالقرب من الكائن المادي الذي ينشئ البيانات، وأخيرًا فإن الحوسبة الضبابية هي الوسيط ما بين الحوسبة السحابية والطرفية، وذلك للأغراض المتعددة. لقد تعرفت سابقًا على كيفية تمكين الحوسبة السحابية لتخزين ومعالجة البيانات لمجموعة من التطبيقات. ستتعرف الآن على جزأين آخرين من البُنية التحتية لحوسبة إنترنت الأشياء.

شكل 2.1: بُنية الحوسبة الطرفية الضبابية السحابية

أساسيات الحوسبة الضبابية

Fog Computing Fundamentals

يتمثل الهدف التقني الثابت لأنظمة إنترنت الأشياء في توزيع إدارة البيانات إلى أقرب مدى من عُقَد المُستشعر أو المُشغّل. تُعدُّ الحوسبة الضبابية أشهر مثال على الخدمات الطرفية في إنترنت الأشياء، وهي أقرب إلى الأشياء المُنتجة لبيانات إنترنت الأشياء. يمكن لأي جهاز محوسب ذي قدرات تخزين واتصال بالشبكة أن يُشكّل عُقدةً ضبابية. ومن الأمثلة على ذلك وحدات التحكم الصناعية، والمحولات، والموجهات، والخوادم المضمنة، وبوابات إنترنت الأشياء، ويؤدي تحليل بيانات إنترنت الأشياء بالقرب من مصدرها إلى تقليل التأخير الزمني، وتقليص عمليات التحميل الكبيرة للبيانات من الشبكة الأساسية، والحفاظ على البيانات الحساسة داخل الشبكة المحلية.



شكل 2.2: مثال على ازدياد مدة التأخير الزمني حسب المسافة

غالبًا ما تُنفّذ الخدمات الضبابية بالقرب من جهاز إنترنت الأشياء، وكذلك بالقرب من نقاط النهاية الطرفية. تتمثل إحدى ميزات هذا الأمر في إيجاد معرفة ضمنية للعُقدة الضبابية بالمُستشعرات التي تديرها بسبب قربها المادي بهذه المُستشعرات. ونظرًا لأن العُقدة الضبابية يمكنها تحليل بيانات جميع المُستشعرات في هذا الجزء، فيمكن توفير تحليل ضمني للرسائل المُستقبَلة بحيث يمكن اختيار إرسال البيانات ذات العلاقة فقط إلى الخدمة السحابية. يقلل هذا من حجم البيانات المُرسلة في المراحل الأولى بشكل كبير، مما يجعلها أكثر فائدة في التطبيقات السحابية وخوادم التحليلات. كذلك فإن توافر المعرفة السابقة يسمح للعقد الضبابية بالاستجابة للأحداث في شبكة إنترنت الأشياء بشكل أسرع بكثير من النموذج السحابي التقليدي الذي عادةً ما يعاني من التأخير الكبير والاستجابة البطيئة في نقل البيانات ومعالجتها. وبذلك توفّر الطبقة الضبابية قدرة شبكية موزعة تسمح بمراقبة الأجهزة والتحكم بها وتحليلها في الوقت الفعلي دون انتظار الاتصال من التطبيق المركزي السحابي وخوادم تحليلاته.

التأخير الزمني (Latency):

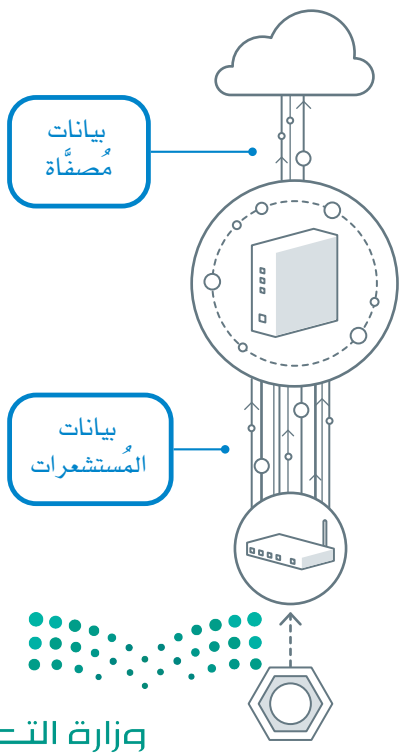
هو التأخر في معالجة البيانات عبر الشبكة، أو التأخر الزمني ما بين إجراءات المستخدم ووقت الاستجابة.

نقطة النهاية (Endpoint):

هي خدمة توجيه البيانات، والتي تختص بإرسال البيانات واستقبالها من الخدمات الأخرى وإليها. قد تكون هذه النقطة مجرد برنامجٍ أو جهازٍ حاسوبي متخصص.

البوابة (Gateway):

تتيح البوابة القدرة على الاتصال للأجهزة التي لا يمكنها الاتصال مباشرة بالإنترنت. وتعمل نقطة الاتصال اللاسلكي كبوابة أيضًا.



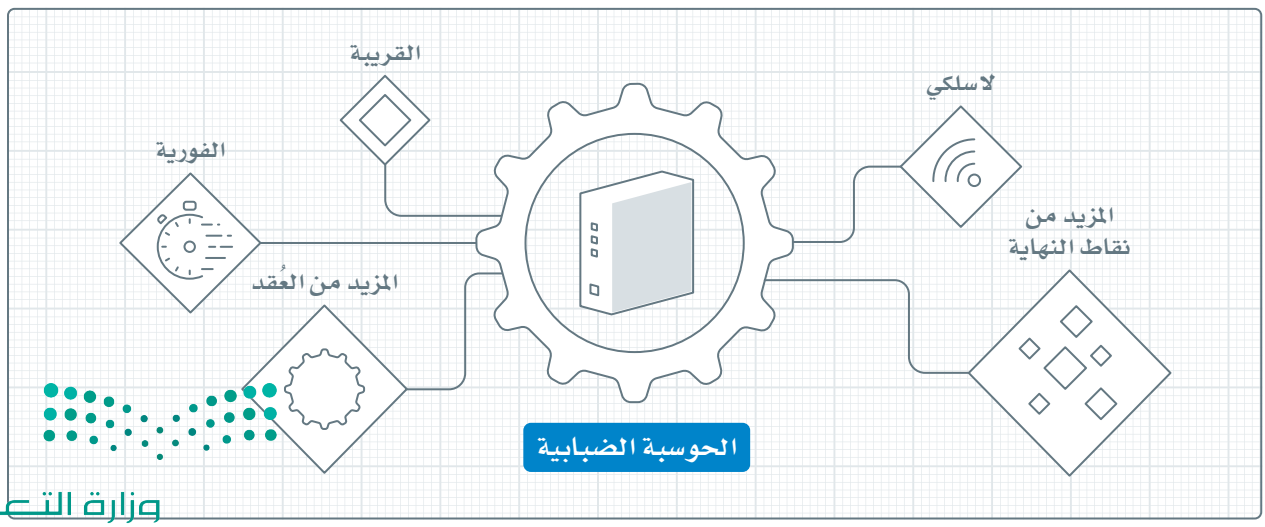
شكل 2.3: تحليل البيانات ضبابيًا

مزايا الحوسبة الضبابية Fog Computing Advantages

تتنوع التطبيقات الضبابية كما تتنوع إنترنت الأشياء نفسها. فتشمل مهامها القياسية اختزال البيانات، والمراقبة، وتحليل البيانات في الوقت الفعلي من قِبل الأجهزة المتصلة بالشبكة.

الجدول 2.1: مزايا الحوسبة الضبابية

الميزة	الوصف
المعرفة الضمنية بالموقع، وانخفاض التأخير الزّمني	تقع العُقدة الضبابية (fog node) في أقرب موقع ممكن من نقطة نهاية إنترنت الأشياء لدعم الحوسبة الموزّعة.
التوزيع الجغرافي	على عكس الحوسبة السحابية الأكثر مركزية، تتطلب خدمات وتطبيقات العُقد الضبابية التثبيت على نطاق أوسع وأكثر انتشارًا.
النشر في نقاط نهاية إنترنت الأشياء	يجري في العادة نشر العُقد الضبابية مع العديد من نقاط نهاية إنترنت الأشياء. تتكون عمليات النشر القياسية النموذجية في العادة من 3000 إلى 4000 عُقدة لكل بوابة، وتعمل كعُقدة حوسبة ضبابية أيضًا.
الاتصال اللاسلكي بين الحوسبة الضبابية وجهاز إنترنت الأشياء	على الرغم من إمكانية توصيل العُقد سلكيًا، إلا أن الحوسبة الضبابية ذات فائدة أكبر خاصةً عندما يتعلق الأمر بعدد كبير من نقاط النهاية، فيُعدُّ الوصول اللاسلكي أبسط طريقة لتحقيق قابلية التوسع.
استخدام التفاعلات الفورية	تتضمن التطبيقات الضبابية المهمة تفاعلات فورية بدلًا من المُعالجة المُجمّعة. وتتيح المعالجة المسبقة للبيانات في العُقد الضبابية لتطبيقات الطبقة العليا معالجة مجموعة فرعية من حِزم البيانات الأكبر.



شكل 2.4: ميزات الطبقة الضبابية

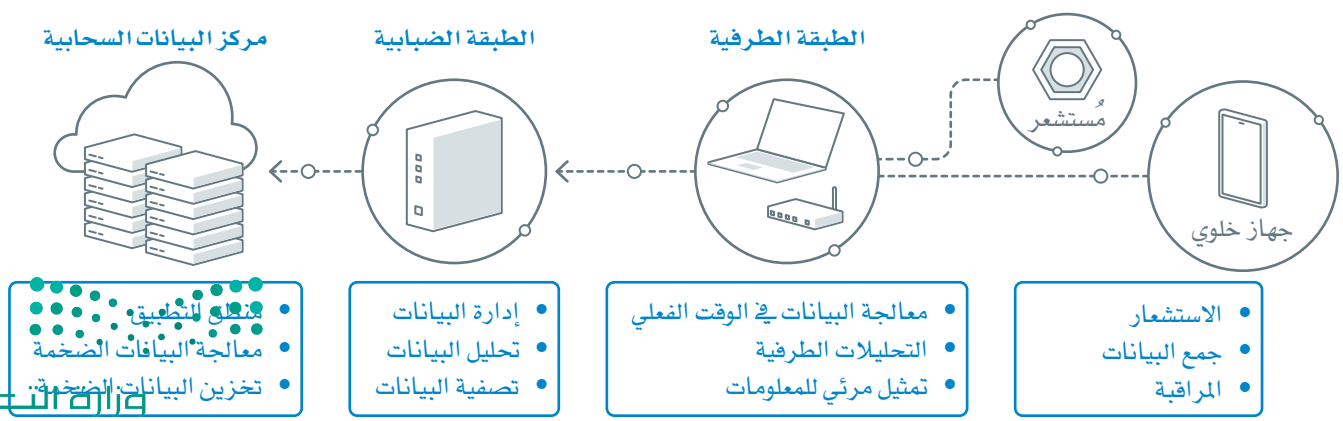
نقاط نهاية الحوسبة الطرفية
Edge Computing Endpoints

تتمتع الأنواع الأحدث من نقاط نهاية إنترنت الأشياء بقدرة محوسبة كافية لإجراء التحليلات والتصفية على مستوىً منخفض. يُطلق على هذه الأجهزة تسمية نقاط نهاية الحوسبة الطرفية أو الأجهزة الطرفية. توفر هذه الطبقة في البُنية الطرفية الضبابية السحابية المزيد من الكفاءة في حلول إنترنت الأشياء. فلا تُستبدل الحوسبة السحابية بالحوسبة الطرفية أو الضبابية، بل تُكمل جميع هذه الطبقات بعضها. تساعد طبقات الحوسبة الطرفية والضبابية في تصفية البيانات وتحليلها وإدارتها. تَحُدُّ هذه النقاط من عمليات الاستعلام السحابي المستمرة عن كل حدث من جميع أجهزة إنترنت الأشياء. يشير هذا النموذج إلى أن تنظيم النطاق الترددي للشبكة، والحسابات، وموارد تخزين البيانات يتم بشكل هرمي ويتمّ جمع البيانات وتحليلها وإرسالها في كل مرحلة بناءً على إمكانيات الموارد المتوفرة في كل طبقة. ينخفض وقت الوصول مع إرسال المزيد من البيانات إلى نقاط النهاية الطرفية الأقرب إلى أجهزة إنترنت الأشياء. وتتمثّل فائدة هذا التسلسل الهرمي في سرعة الاستجابة للأحداث من الموارد القريبة من جهاز إنترنت الأشياء وبنتيجة فورية. كما تتوفر أيضًا في الوقت نفسه موارد تخزين البيانات الضخمة ومعالجتها في مراكز البيانات السحابية عند الضرورة.

الجهاز الطرفي (Edge device):
الأجهزة الطرفية هي بوابات ذكية قادرة على معالجة البيانات محليًا. يُمكن لأجهزة إنترنت الأشياء الاتصال بالأجهزة المتطورة عبر الشبكات المحلية مثل الشبكة اللاسلكية المحلية (Wi-Fi) أو عبر تقنية البلوتوث (Bluetooth).

الحوسبة الطرفية والضبابية معًا Edge and Fog Working Together

تتطلب الحوسبة الطرفية والضبابية استخدام طبقة اختزال لتمكين التطبيقات من التواصل مع بعضها. تعرض طبقة الاختزال **واجهات برمجة التطبيقات الموحدة** (Application Programming Interfaces – APIs) لمراقبة الموارد المادية وتوفيرها والتحكم بها. ولدعم المرونة والاتساق عبر نظام إنترنت الأشياء، تتطلب طبقة الاختزال أيضًا آلية لدعم المحاكاة الافتراضية، مع القدرة على تشغيل أنظمة تشغيل متعددة أو حاويات خدمة على الأجهزة المادية. من ناحية البنية (أو هندسة الشبكة)، فتقوم العقد الضبابية الأقرب الى طرف الشبكة باستقبال البيانات من أجهزة إنترنت الأشياء. ويقوم تطبيق إنترنت الأشياء الضبابي بعد ذلك بتوجيه أنواع البيانات المختلفة إلى أفضل موقع ليتم تحليلها، حيث تُحلّل البيانات الأكثر حساسية للوقت بشكل أقرب إلى الكائنات الذكية التي تولّد البيانات طرفيًا أو إلى العُقدة الضبابية، ثم توجّه البيانات التي يمكن التعامل معها في ثوانٍ أو دقائق إلى عُقدة التجميع لتحليلها واتخاذ الإجراءات بشأنها. وتُرسل البيانات ذات الأهمية الزمنية الأقل إلى السحابة للقيام بعمليات التحليل الزمني وتحليلات البيانات الضخمة والتخزين طويل الأجل. على سبيل المثال، يمكن لآلاف العقد الضبابية إرسال ملخصات البيانات إلى السحابة للتحليل الزمني والتخزين، وتساعد مراعاة هذه العوامل في تحديد ما إذا كانت الحوسبة الطرفية أو الضبابية ستعمل على تحسين كفاءة النظام.



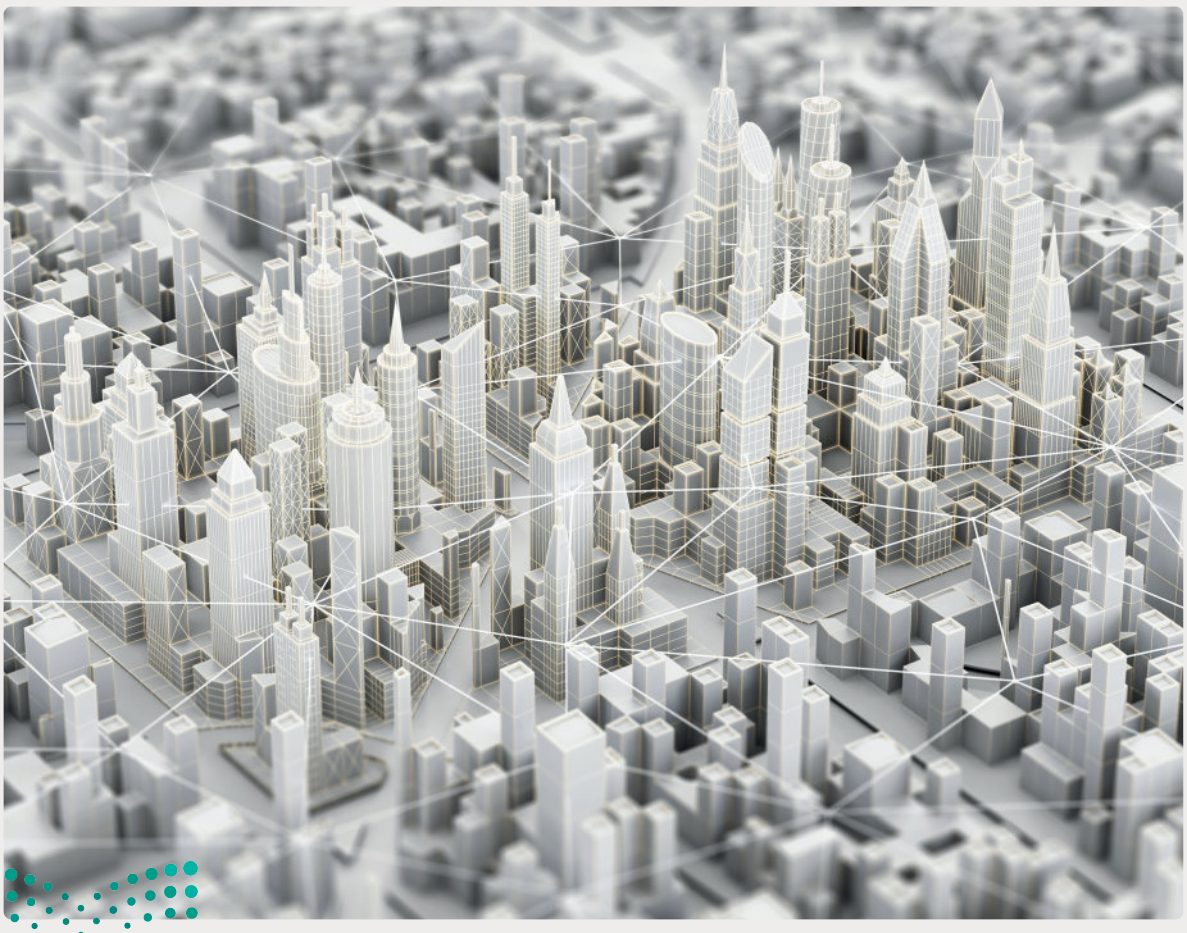
شكل 2.5: الطبقات الطرفية والضبابية في نظام إنترنت الأشياء

مُمكِنات إنترنت الأشياء IoT Enablers

بيانات إنترنت الأشياء IoT Data

تُعدُّ البيانات المُنشأة بواسطة مليارات من أجهزة إنترنت الأشياء ذات قيمة كبيرة، حيث إنها تسمح للمؤسسات بتحليل تلك البيانات وذلك لتقديم خدمات جديدة تعمل على تحسين تجربة المستخدم أو خفض التكاليف أو خلق مصادر جديدة للإيرادات. على الرغم من أن البيانات غير المنظّمة تُمثل غالبية البيانات الناتجة من إنترنت الأشياء، فإن الرؤى التي تقدمها من خلال التحليلات يمكن أن تساعد في ضبط وترشيد العمليات، وكذلك تساهم في تطوير نماذج أعمال جديدة. وقد تكون عملية إدارة هذه البيانات غير المنظّمة وتقييمها أمرًا صعبًا. ولحل هذا المشكلة، تُصمّم عمليات نشر إنترنت الأشياء بحيث تقوم بتقنين إنتاج البيانات وتصفية البيانات الأقل أهمية، وكذلك تقليص البيانات الأولية غير الضرورية، والاستجابة للأجهزة بأسرع وقت ممكن.

تخيّل وجود مدينة ذكية تُضاء من خلال شبكة تحتوي على مئات الآلاف من المصابيح الذكية المتصلة عبر نظام إنترنت الأشياء. ستكون معظم المعلومات المنقولةً بين وحدات شبكة الإضاءة ومركز التحكم قليلة الأهمية، ورغم ذلك فإن أنماط هذه البيانات يُمكنها أن تُقدم معلومات تساعد في التنبؤ بالوقت الذي يتوجب فيه إصلاح الإنارة أو التنبؤ بتوقيت تشغيل الإنارة وإطفائها، وذلك لتوفير النفقات التشغيلية.



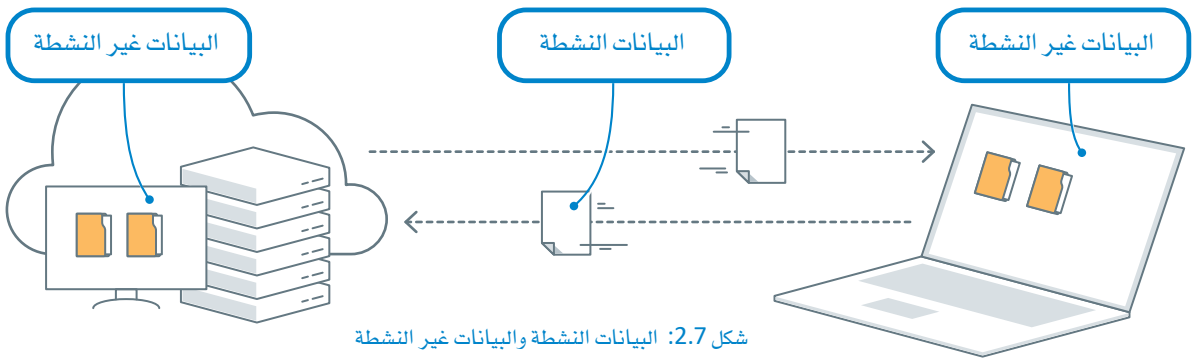
شكل 2.6: مدينة ذكية بأجهزة ذكية متصلة

تصنيف البيانات Data Classification

عامةً لا يجب التعامل مع جميع البيانات بالطريقة نفسها؛ بل يمكن تصنيفها وتقييمها بعدة طرائق. يمكن استخدام العديد من أدوات التحليل وتقنيات معالجة البيانات. تشمل التصنيفات المتعلقة بإنترنت الأشياء جميع البيانات، سواءً كانت البيانات منظمة أو غير منظمة وسواءً كانت البيانات نشطة أو غير نشطة.

البيانات النشطة وغير النشطة Data in Motion Vs. at Rest

عندما تكون البيانات قيد النقل فإنه يطلق عليها "البيانات النشطة"، أما حينما تُخزّن في مكان ما فتُسمى "البيانات غير النشطة". يُعدُّ نقل البيانات داخل الشبكة، وتبادل البيانات بين العميل والخادم كعمليات تصفح الويب ونقل الملفات، ورسائل البريد الإلكتروني أمثلةً على البيانات النشطة. تُحفظ البيانات غير النشطة عادةً في محركات أقراص USB، أو محرك أقراص ثابت في حاسب محمول، أو في مركز بيانات سحابي. تُعدُّ البيانات الناشئة من الكائنات الذكية في عالم إنترنت الأشياء بيانات نشطة، وذلك نظرًا لانتقالها خلال الشبكة إلى وجهتها النهائية. يتم التعامل مع هذه البيانات بشكل متكرر في الأجهزة الطرفية أو العُقد الضبابية. عندما تُعالج تلك البيانات على جهاز طرفي، يمكن تصفيتها أو التخلص منها، أو يمكن نقلها إلى عُقدة ضبابية أو مركز بيانات للمعالجة الإضافية والتخزين المحتمل.



شكل 2.7: البيانات النشطة والبيانات غير النشطة

التحليلات الطرفية

لقد ساهم التحول إلى الخدمات السحابية في التطور الكبير لإنترنت الأشياء وعمليات تحليلات البيانات في السنوات الأخيرة. تُجمع في عالم إنترنت الأشياء كميات هائلة من البيانات على الأجهزة، ويتعين تحليل تلك البيانات بشكل متكرر لاتخاذ الإجراءات المناسبة بناءً عليها وذلك في الوقت الفعلي. تحتاج البيانات الضخمة التي تُنشأ طرفيًا إلى المزيد من النطاق الترددي للشبكة لنقل البيانات سحابيًا، وقد تكون تلك البيانات ذات طبيعة حساسة بحيث تتطلب اهتمامًا فوريًا، وتستدعي تحليلًا عميقًا يستحيل القيام به عبر الخدمات السحابية. وتعمل التقنية الحديثة للتحليلات الطرفية على حل هذه المشكلة من خلال توفير وظائف تحليل البيانات داخل جهاز إنترنت الأشياء ذاته، حيث يجري تحليل البيانات على الجهاز في مدة قياسية مقارنة بتلك التي تتطلبها عملية إرسال البيانات للتحليل في الخدمات السحابية.

شكل 2.8: معالجة التحليلات الطرفية

بروتوكولات الشبكات Networking Protocols

بروتوكولات الشبكات الأساسية Basic Networking Protocols

توفر بروتوكولات شبكات الإنترنت الأساسية (Internet Protocol – IP)، و (Transmission Control Protocol – TCP) و (User Datagram Protocol – UDP) الاتصال لشبكات إنترنت الأشياء. تجري عملية معالجة نقل البيانات بين الكائنات الذكية وأي نظام آخر في تطبيق إنترنت الأشياء عبر بروتوكولات عالية المستوى. تم تطوير هذه البروتوكولات لتلبية متطلبات نقل بيانات إنترنت الأشياء. تعتمد بعض شبكات إنترنت الأشياء على نموذج شبكة معتمد على دفع البيانات، فعلى سبيل المثال، يقوم المُستشعر بعملية إرسال البيانات على فترات منتظمة، أو من خلال الاستجابة لأحداث مُعينة. يعتمد البعض الآخر على نموذج سحب البيانات، مثل التطبيق الذي يستعلم من المُستشعر عن البيانات في شبكة إنترنت الأشياء.

الجدول 2.2: آلية عمل بروتوكولات TCP وUDP

بروتوكول التحكم في الإرسال (TCP)	بروتوكول حِزم بيانات المستخدم (UDP)
يحتاج هذا البروتوكول المُخصص للاتصال إلى إعداد ربط بين المَصدر والوِجهة قبل إرسال البيانات. يمكن مقارنة هذا البروتوكول بعملية إجراء محادثة هاتفية عادية، حيث يجب توصيل الهاتفين معًا وإنشاء قناة اتصال قبل تمكّن الطرفين من التواصل.	باستخدام هذا البروتوكول يُمكن إرسال البيانات بسرعة من المصدر إلى الوجهة، ولكن دون ضمان وصولها إلى هناك. يشبه هذا إرسال البريد، حيث تُرسل رسالة بالبريد إلى الشخص المناسب، ولكن دون إمكانية للتأكد من استلامها حتى يُشعر المُرسل باستلام الرسالة.

بروتوكولات الوصول اللاسلكي Wireless Access Protocols

الاتصال قريب المدى NFC

هو مجموعة من البروتوكولات بمدى لا يتجاوز 4 سنتيمترات. إن المدى القصير والتكلفة المنخفضة لهذه التقنية يجعلانها مثالية للاستخدام في الأغراض الشخصية اليومية، حيث تراها بشكل كبير في التطبيقات أنظمة الدفع دون اتصال باستخدام بطاقة الائتمان أو الدفع بالهاتف الذكي الذي يدعم تقنية NFC.

البلوتوث Bluetooth

هي تقنية لاسلكية تستخدم الترددات اللاسلكية لتبادل البيانات عبر مسافات قصيرة، مما يُغني عن التوصيلات السلكية. يصل مدى اتصال البلوتوث بين الأجهزة حتى 10 أمتار. توجد تقنية البلوتوث في العديد من الأجهزة والأدوات، وحتى السيارات، وتتيح تبادل البيانات بسرعات عالية بين الأجهزة المقترنة مع بعضها. ومن الأجهزة الأخرى التي تستخدم هذه التقنية سماعات الرأس ومكبرات الصوت ولوحات المفاتيح اللاسلكية والأقفال الذكية وساعات المعصم.

IEEE 802.15.4

هي تقنية وصول لاسلكية للأجهزة تتميز بانخفاض تكلفتها وبمعدل نقل بيانات منخفض للأجهزة التي تعمل بالطاقة الكهربائية أو البطاريات. تُعدُّ هذه التقنية غير مُكلفة، ويمكنها أن تدعم عمرًا أطول للبطارية، كما أنها سهلة الإعداد لاستخدامها مجموعة بروتوكولات قليلة وسهلة ومرنة.

بروتوكولات شبكات إنترنت الأشياء IoT Networking Protocols

يحتوي جدول 2.3 على بعض أحدث بروتوكولات الشبكات المُستخدمة في أجهزة إنترنت الأشياء للتواصل مع بعضها والإنترنت. تعتمد هذه البروتوكولات على البروتوكولات الأساسية لشبكة الإنترنت.

الجدول 2.3: بروتوكولات شبكات إنترنت الأشياء

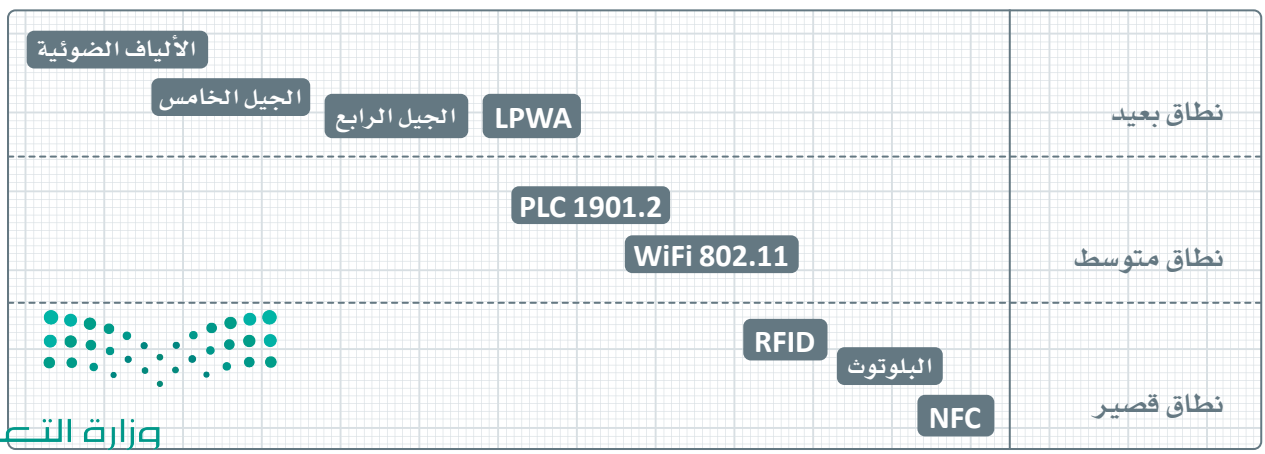
اسم البروتوكول	الميزات
6LoWPAN	هو اختصار لبروتوكول IPv6 عبر شبكات المنطقة الشخصية اللاسلكية منخفضة الطاقة. يوفر هذا البروتوكول اتصالات إنترنت أشياء منخفضة التكلفة وآمنة.
ZigBee	هو تطوير لمعيار 6LoWPAN يوفر طريقة اتصال أسهل وأقل تكلفة من البلوتوث (Bluetooth) والواي فاي (Wi-Fi). تشمل التطبيقات الشائعة أتمتة المباني والمنازل والرعاية الصحية.
ISA100.11a	بروتوكول ISA100.11a هو معيار للأتمتة الصناعية للأنظمة اللاسلكية، ويُستخدم للتحكم في العمليات.
WirelessHART	يعدّ حزمة بروتوكولات لإنشاء بُنية شبكية متزامنة زمنيًا وذاتية التنظيم والتصحيح.
Thread	يعدّ مجموعة بروتوكولات لإنشاء شبكة تشعبية آمنة وموثوقة لربط الأجهزة معًا والتحكم بها خاصةً في المنازل.

تقنيات اتصال إنترنت الأشياء IoT Communication Technologies

تُصنّف تقنيات الاتصال المختلفة لحلول إنترنت الأشياء حسب نطاق المعلومات والبيانات المنقولة خلالها. يجب الأخذ بعين الاعتبار أن الأجهزة التي تستخدم تقنيات الاتصالات بعيدة المدى تستهلك طاقة أكثر بكثير من نظيراتها قصيرة المدى.

الجدول 2.4: تصنيف تقنيات اتصالات إنترنت الأشياء حسب المسافة

المسافة	تقنيات اتصالات إنترنت الأشياء
نطاق قصير	يُعدّ الكابل التسلسلي مثالًا تقليديًا على النظام السلكي. عادةً ما تكون الحلول اللاسلكية قصيرة المدى، والتي يبلغ أقصى مدى لها عشرات الأمتار بين جهازين، بديلًا للكابلات التسلسلية. تشتمل التقنيات اللاسلكية قصيرة المدى على تقنية البلوتوث **والاتصال قريب المدى** (Near-Field Communication – NFC) ومُعرف **تحديد الهوية بموجات الراديو** (Radio Frequency IDentification – RFID).
نطاق متوسط	يعدّ هذا النوع الأكثر شيوعًا في تقنيات الوصول إلى إنترنت الأشياء. فهناك تطبيقات مختلفة في نطاق يتراوح بين عشرات ومئات الأمتار. غالبًا ما تكون أقصى مسافة بين الجهازين أقل من كيلومتر واحد، ولكن تقنيات **التردد اللاسلكي** (Radio Frequency – RF) ليس لها مسافات قصوى محددة سابقًا، حيث أنها تعمل طالما تُبثّ الإشارة اللاسلكية وتُستقبل بشكل صحيح. تشمل التقنيات اللاسلكية متوسطة المدى تقنية الشبكة اللاسلكية IEEE 802.11 Wi-Fi . يمكن أيضًا تصنيف التقنيات السلكية مثل IEEE 802.3 Ethernet و اتصالات خطوط الطاقة ضيقة النطاق IEEE 1901.2 (Power Line Communications – PLC) على أنها متوسطة المدى.
نطاق بعيد	تُستخدم التقنيات بعيدة المدى عندما تزيد المسافات بين جهازين عن كيلومترين على الأقل، وتُعدُّ التقنيات الخلوية (الجيل الثاني، والجيل الثالث، والجيل الرابع، والجيل الخامس)، وكذلك **التقنيات منخفضة الطاقة واسعة النطاق** (LPWA) أمثلة على التقنيات اللاسلكية. ويُمكن لاتصالات التقنيات منخفضة الطاقة واسعة النطاق أن تغطي منطقة واسعة وذلك بمتطلبات قليلة من الطاقة، ونتيجة لذلك، تُعدُّ هذه التقنيات مناسبة لمُستشعرات إنترنت الأشياء المزودة ببطارية، ويتم تصنيف كل من IEEE 802.3 عبر الألياف البصرية **واتصالات خط الطاقة ذات النطاق العريض** (IEEE 1901 Broadband Power Line Communications) على أنها من النطاق البعيد، ولكنها لا تُعدّ تقنيات وصول إلى إنترنت الأشياء تحديدًا.



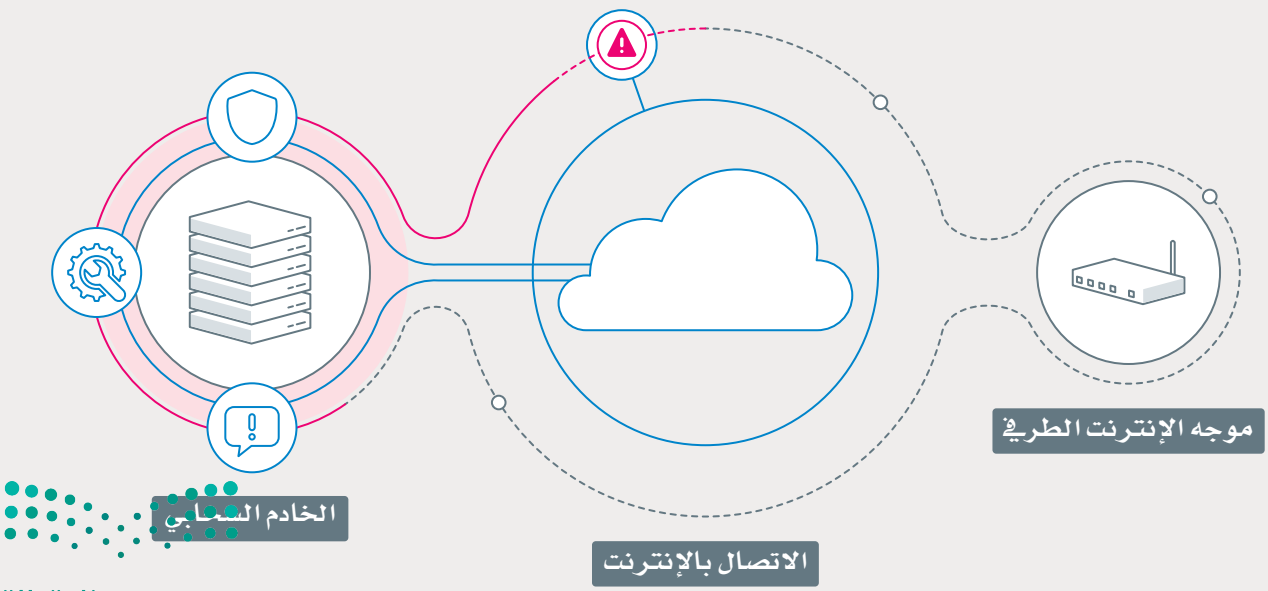
شكل 2.9: نطاق تقنيات الاتصال

بعض المسائل المتعلقة بالاتصالات Connectivity Issues

قد يكون الاتصال بالإنترنت غير ثابت، وقد يتعرض لفقدان الاتصال لفترة قصيرة أو طويلة وبصورة غير متوقعة. فما الذي يجب أن يفعله جهاز إنترنت الأشياء عند فقدان الاتصال بالشبكة؟ تتمثل الخيارات في هذه الحالة في الاستغناء عن البيانات أو تخزينها محليًا إلى حين استعادة الاتصال بالشبكة. ويعتمد الاختيار على عدة عوامل أهمها التطبيق نفسه، وأهمية تلك البيانات، وطريقة استخدام البيانات للتحكم في المُشغِّلات، كما يعتمد أيضًا على مدة فُقدان الاتصال بالشبكة.

يمكن تخزين البيانات مؤقتًا محليًا (داخل الجهاز) إذا تمت استعادة الاتصال بسرعة، ولكن لا يمكن لجهاز إنترنت الأشياء الاحتفاظ بكمية كبيرة من البيانات في وسائط التخزين، ويجب أن تكون برمجة جهاز إنترنت الأشياء ذكية بما يكفي للتعامل مع مشكلات الاتصال، ويمكن في بعض الأحيان لأجهزة إنترنت الأشياء اتخاذ قرارات محدودة للتحكم في المُشغّلات دون الاتصال بتطبيق إنترنت الأشياء الرئيس، كما يجب أن تكون الخدمات الطرفية أو السحابية قادرة على العمل ببيانات منقوصة أو مُتأخرة. على سبيل المثال يجب ألا تُرسل أوامر إلى جهاز إنترنت الأشياء بناءً على البيانات التي لم يعد لها أهمية.

تتضمن العديد من المشكلات السحابية فقدانًا مؤقتًا للاتصال بالشبكة، أو انتهاء الوقت عند عدم توفر الخدمة، ولكن يُمكن لأجهزة إنترنت الأشياء التغلب على هذه المشكلات تلقائيًا. يدعم مزودو إنترنت الأشياء السحابية مثل AWS IoT Core وMicrosoft Azure IoT Hub إمكانية الكشف عن مشكلات الاتصال المُتكررة ومعالجتها عند ظهورها على الأنظمة التي تُنفّذ باستخدام هذه التقنيات. يمكن لخدمات إنترنت الأشياء السحابية تشخيص المشكلة وتوفير الحلول المؤقتة والمساعدة في التوجيه لتصحيحها. يُنبّه القائمون على نظام إنترنت الأشياء عند تعرُّض الأجهزة المهمة والبُنية التحتية لمثل هذه المشكلات واتخاذ الإجراءات اللازمة.



شكل 2.10: توفير خدمات إنترنت الأشياء السحابية إجراءات الاسترداد التلقائي

تمرينات

1

حدِّد الجملة الصحيحة والجملة الخاطئة فيما يلي:	صحيحة	خاطئة
1. تُعدُّ طبقة الحوسبة الضبابية أقرب إلى الكائنات الذكية من الطبقة الطرفية.	○	○
2. يمكن للإنترنت الاتصال بطبقة الحوسبة الطرفية مُباشرة.	○	○
3. يمكن للطبقة الضبابية التواصل مباشرة مع الخدمات السحابية.	○	○
4. يمكن أن تحدث معالجة البيانات في كلٍ من الطبقات الضبابية والسحابية.	○	○
5. تُعدُّ البيانات المنقولة إلى القرص الصلب "بيانات ثابتة".	○	○
6. تحل التحليلات الطرفية محل معالجة البيانات سحابيًا.	○	○
7. ينتظر بروتوكول بيانات المستخدم (UDP) إشعارًا من المُستقبل يؤكد استلامه للحزمة.	○	○
8. يرسل بروتوكول (ZigBee) مزيدًا من المعلومات حول الكائن المُرسل، وبشكلٍ أكثر من البروتوكولات الأخرى.	○	○
9. تستخدم الشبكات الخلوية الاتصالات قصيرة المدى بين الكائنات الذكية.	○	○
10. تفقد كافة بروتوكولات الشبكة البيانات أثناء نقلها عند حدوث مشكلات في الاتصال.	○	○

2 وضِّح كيف أسهمت إضافة طبقة الحوسبة الضبابية إلى تطبيقات إنترنت الأشياء في تحسين فعاليتها. اكتب أفكارك أدناه.

3 ارسم مخططًا يُمثل العلاقة بين طبقات الحوسبة السحابية والضبابية والطرفية لبُنية إنترنت الأشياء.

4 ناقش مدى مساعدة التحليلات الطرفية في حلول إنترنت الأشياء المجدية.

5 صَنِّف أنواع التطبيقات التي تستخدم بروتوكولات اتصالات TCP وUDP على التوالي.

6 صِفْ الخصائص الرئيسة لبروتوكول الشبكة IEEE.802.15.4، والتي تُكسبه أهمية كبيرة في تطبيقات إنترنت الأشياء. اكتب أفكارك أدناه.

7 ابحث في الإنترنت عن معلومات حول الاختلافات الرئيسة في طرائق الاتصال بين الشبكات الخلوية وتقنيات البلوتوث.

رابط الدرس الرقمي

www.ien.edu.sa

الدرس الثاني

تطبيقات وتحديات إنترنت الأشياء

تطبيقات إنترنت الأشياء IoT Applications

تُعدُّ إنترنت الأشياء واحدة من أسرع التقنيات نموًا وتطورًا، وبينما تنتقل البشرية من عصر المنتجات إلى عصر الخدمات؛ فإن إنترنت الأشياء تقوم بدور هام في إحداث هذه الثورة التقنية وربما تشهد حياتك تغيرات في المستقبل القريب، ومنها أنك قد تعود إلى منزلك في سيارة ذاتية القيادة، حيث سيكتشف باب المرآب وجودك ويفتح تلقائيًا. فيما يلي بعض الأمثلة على المجالات التي غيرت فيها إنترنت الأشياء طريقة حياتنا وأعمالنا:

الأجهزة القابلة للارتداء Wearables

تُعدّ الأجهزة القابلة للارتداء من أكثر العناصر رواجًا بشكل تجاري، ورغم أن عامة الناس يعتقدون بأنها مجرد أجهزة غير معقدة ، لكنها في واقع الأمر من أجهزة إنترنت الأشياء المتطورة والتي توفر مجموعة متنوعة من الوظائف المختلفة، وذلك بدءًا من المراقبة الطبية إلى تتبُّع الصحة واللياقة البدنية. يمكن لهذه الأجهزة التواصل مع الخدمات السحابية لتوفير مراقبة صحية فورية لمستخدميها، وكذلك عرض التنبيهات بشأن المشكلات الصحية المحتملة.

شكل 2.11: كائن ذكي قابل للارتداء

التطبيب عن بعد Telemedicine

لم يصل مجال التطبيب أو الرعاية الصحية عن بعد إلى كامل إمكاناته بعد. تُقدّم خدمة التطبيب عن بعد المبنية على إنترنت الأشياء من خلال الأجهزة الطبية المتصلة دائمًا، والتي يمكن لمتخصصي الرعاية الصحية مراقبتها. يحدث التشخيص الطبي عن بعد بشكل استباقي، مما يوفر وقتًا ثمينًا لتوفير العلاج المناسب للمرضى. على سبيل المثال، يمكن أن تستشعر أنظمة الكشف عن النوبات القلبية نبضات قلب الشخص بصورة فورية لترسل رسائل للطبيب لإنقاذ حياة المريض.

شكل 2.12: تطبيق التطبيب عن بعد

المنازل الذكية Smart Homes

تُعدُّ المنازل الذكية واحدة من أفضل تطبيقات إنترنت الأشياء وأكثرها استخدامًا، وتوفر الكثير من الراحة للأشخاص والحماية المنزلية. توجد مجموعة واسعة من تطبيقات إنترنت الأشياء الخاصة بالمنازل الذكية، ولكن أكثرها كفاءة هو ما يدمج بين أنظمة المرافق الذكية وأغراض الترفيه. وتُعزّز الحماية المنزلية من خلال أنظمة الأقفال المتطورة وأنظمة المراقبة الشبكية. مع تطور إنترنت الأشياء، يمكنك أن تكون على ثقة بأن منزلك سيصبح أكثر ذكاءً. فعلى سبيل المثال، سيتعرف نظام الإضاءة الأوتوماتيكي بصورة تلقائية على عدم وجود أحد في المنزل؛ ليوفر الطاقة.

شكل 2.13: مراقبة المنزل الذكي

التعليم Education

يمكن للمدرسة أو المؤسسة الدراسية التي تدعم إنترنت الأشياء مساعدة المعلمين والإدارة في تسجيل الحضور اليومي. يمكن للنظام أيضًا إخطار أولياء الأمور بتغيب الطلبة تلقائيًا. تُعدّ أجهزة السبورة الذكية، وأقفال الأبواب، وأنظمة الحرائق والحماية من أبرز أجهزة إنترنت الأشياء الأخرى المستخدمة في قطاع التعليم.

شكل 2.14: تطبق جامعة سعودية نظام أمان قائم على إنترنت الأشياء

الشبكات الذكية Smart Grids

تعتمد إدارة شبكات الكهرباء على أنظمة معقدة وحيوية، وذلك لأداء مهمتها في توفير الطاقة الكهربائية للمباني والمصانع ووسائل النقل وكل شيء في الحياة اليومية تقريبًا. تستخدم شبكة الكهرباء الذكية تقنيات إنترنت الأشياء لتقليل الهدر في الطاقة الكهربائية وتعزيز كفاءة نقلها وتحسين وقت الصيانة وتقليل تكاليف التشغيل.

شكل 2.15: اتصالات الشبكة الذكية

السيارات ذاتية القيادة Self-Driving Cars

تعمل شركات التقنية الكبرى على تطوير إصدارات من السيارات والمركبات الأخرى ذاتية القيادة. على سبيل المثال قامت إحدى شركات الاتصالات في المملكة بتنفيذ خدمة الحافلات الذاتية القيادة لمجمع الرياض. تستخدم جامعة الملك عبد الله للعلوم والتقنية حافلة ذاتية القيادة مماثلة في حرمها الجامعي. تزود المُستشعرات المتعددة والأجهزة المضمنة المتصلة سحابيًا بيانات ضخمة لاتخاذ القرارات بناءً على خوارزميات التعلم الآلي، ويمكن القول بأنه حتى وإن استغرق تطوير التقنية بأكملها سنوات أخرى، وإن احتاج الأمر إلى قيام الدول بتعديل قوانينها ولوائحها لتنظيم هذه التقنيات، فإن تطبيقات إنترنت الأشياء ستستمرّ في النمو بشكل سريع للغاية. ويجب على مطوري ومقدمي هذه التقنيات والخدمات التأكد من جاهزيتها وقدرتها على ضمان سلامة البشر ووسائل النقل كافة في الطرق.

شكل 2.16: سيارة ذاتية القيادة على طريق سريع

أسواق التجزئة Retail Shops

شهد العالم مؤخرًا ظهور نوع جديد من متاجر البيع بالتجزئة التي تعتمد على إنترنت الأشياء، وتقدم نموذجًا مختلفًا عن المتاجر الفعلية والمتاجر الإلكترونية. يضيف المتجر الذي يدعم إنترنت الأشياء المنتجات إلى عربة التسوق الخاصة بك فور اختيارها، ثم يتيح لك الدفع مقابل مشترياتك بخصم الأموال من محفظتك الرقمية على هاتفك الذكي، كما تتيح التقنية إضافة المنتجات وإزالتها واستبدالها في سلة التسوق، ولا تتطلب عملية الشراء هذه رسومًا أو كلفةً إضافية، وبالطبع فإنك لست بحاجة إلى الانتظار في الطابور للدفع.

شكل 2.17: تسوق رقمي فعال

يُعدّ المعرض السعودي لإنترنت الأشياء (Saudi IoT) حدثًا إقليميًا محوره الأساسي هو تقنيات إنترنت الأشياء وتطويرها. يركز المعرض على الفرص الحالية والمستقبلية الناتجة عن تطبيق تقنيات إنترنت الأشياء، تطبيقاتها وخدماتها. يتماشى إنترنت الأشياء السعودي مع رؤية المملكة العربية السعودية 2030 والتي تنظر إلى التقنيات الناشئة على أنها مطلب ومفتاح لاكتشاف المزيد من الأساليب المبتكرة لإدارة الأعمال وتعزيز الاقتصاد.

إدارة سلسلة التوريد الذكية
Smart Supply-Chain Management

يجب أن تتسم شبكات التوريد بالفعالية وأن تعمل بالشكل الأمثل، وإلا فإنها ستزيد من تكلفة البضائع. توفر إنترنت الأشياء حلولا لتتبع العناصر أثناء وجودها في المستودعات أو أثناء النقل، وذلك باستخدام رقاقات إلكترونية توفر معلومات فورية، مما يسهم في الحد من حدوث الأخطاء، والتقليل من التأخير في عملية توريد المنتجات.

شكل 2.18: شبكات سلسلة توريد ذكية

إنترنت الأشياء الصناعي Industrial IoT

يتكون إنترنت الأشياء الصناعي من مُستشعرات وأدوات وأجهزة إنترنت الأشياء الأخرى التي ترتبط بتطبيقات إدارة الإنتاج والطاقة. يتوقع خبراء الصناعة بأن يتمتع إنترنت الأشياء بإمكانيات أكبر في جميع تطبيقات إنترنت الأشياء، وذلك بإمكانية تحسين جودة المنتجات وكفاءة الإنتاج. على سبيل المثال، قامت إحدى الشركات المُصنعة للطائرات التجارية بدمج مُستشعرات في الأدوات والآلات ومنحت العاملين نظارات ذكية لتقليل الأخطاء وتعزيز السلامة في موقع العمل.

شكل 2.19: تطبيق إنترنت الأشياء في الصناعة

الزراعة الذكية

تُنبّئ التطورات الحالية في إنترنت الأشياء في الصناعة الزراعية بمستقبلٍ واعد، حيث تجري باستمرار عمليات تطوير أدوات الري بالتنقيط، والتعرف على أنماط المحاصيل، وتوزيع المياه، واستخدام الطائرات دون طيار لمراقبة المزارع. تُمكّن هذه الابتكارات المزارعين من زيادة الإنتاجية والحد من المخاطر المحيطة بالزراعة بشكلٍ أكثر فعالية.

شكل 2.20: تطبيق إنترنت الأشياء للحديقة الذكية

النقل الذكي

يُتيح نظام النقل العام الجديد في الرياض الذي يتكون من مترو الرياض ونظام حافلات الرياض السريع للأفراد القدرة على التنقل في المدينة بسهولة. يستخدم مترو الرياض قطارات دون سائق مزودة بإنترنت الأشياء، ويوفر تركيب حلولًا متكاملة تُعزز تجربتهم في التنقل، ويتضمن المشروع مركزًا متطورًا لأنظمة للمراقبة والتحكم للمحطات والخطوط والبُنية التحتية الأخرى.

شكل 2.21: شبكات النقل الذكية

إدارة الحركة المرورية Traffic Management

يمكن تحسين إدارة الحركة المرورية بمساعدة تقنيات إنترنت الأشياء في المدن الكبيرة. يتم ذلك باستخدام الهواتف المحمولة ككائنات ذكية مزودة بمُستشعرات وتطبيقات تحديد المواقع الجغرافية مثل خرائط قوقل، إضافةً إلى المعلومات التي يُتحصّل عليها من المركبات من خلال الأنظمة السحابية كنظام الإنذار بالمخاطر الموجود في بعض وسائل النقل. يمكن لتقنيات إنترنت الأشياء تحسين الحركة المرورية والسلامة على الطرق، ويعدُّ التحليل طويل المدى لأنماط الحركة المرورية تطبيقًا آخر لإنترنت الأشياء، مما يُمكِّن المسافرين من تجنب الازدحام المروري والحصول على معلومات وافية عن الطرق البديلة خلال ساعات الذروة بصورة أفضل.

شكل 2.22: تطبيق إنترنت الأشياء للطريق السريع

تتابع وزارة النقل والخدمات اللوجستية تنفيذ أنظمة النقل الذكية على الطرق السريعة لتحسين السلامة على الطرق وإدارة الحركة المرورية بصورة فعّالة. ستُجهّز شبكات الطرق والطرق السريعة بأنظمة إنترنت الأشياء للتحكم في حركة السير والحركة المرورية وتحسين كفاءتها. تحدد خطة الوزارة الإطار الأساسي للتنفيذ المستقبلي لأنظمة النقل الذكية.

شكل 2.23: مراقبة الحركة المرورية باستخدام إنترنت الأشياء

إدارة المياه / النفايات Water / Waste Management

تقوم العديد من البلديات بتنفيذ إعادة تدوير المياه من خلال استخدام وحدات معالجة المياه. باستخدام تطبيق إنترنت الأشياء يُمكن تحديد كمية المياه المُستهلكة في موقع معين، وكذلك كمية المياه التي تُجمع، ومدى التغير في كَمِّ النفايات المُنتجة بمرور الوقت. يمكن للبلديات من خلال تقنيات إنترنت الأشياء التنبؤ بكَمِّ النفايات الناتجة في منطقة معينة، وتحديد كيفية معالجتها وآليات التخلص منها، وكذلك التخطيط المستقبلي لمثل تلك العمليات. يمكن أيضًا تحليل حجم النفايات المُنتجة في كل حي ويمكن استخدام جميع هذه المعلومات للتخطيط لمبادرات تحسين المدينة، وتُقدم أنظمة تحليل البيانات التخطيط الأمثل لجمع النفايات.

شكل 2.24: الإدارة الذكية للنفايات للحد من النفايات الصلبة

مثال

يشهد شهر رمضان وموسم الحج قدوم أكثر من مليوني مسلم سنويًا لزيارة مكة والمشاعر المقدسة. يتعين على السلطات المحلية التعامل مع الزيادة المتنامية في إنتاج النفايات في تلك الفترة. تهدف استراتيجية رؤية المملكة 2030 إلى تقليل جميع أنواع النفايات بتنفيذ أنظمة إدارة النفايات. يسعى نظام إدارة النفايات الصلبة القائم على تقنيات إنترنت الأشياء إلى تحسين عملية جمع النفايات من مصادرها. قدمت الجامعات السعودية مقترحًا باستخدام الحاويات الذكية التي تسمح بأتمتة عملية تصنيف النفايات مما يسمح بإعادة تدويرها بسهولة.

أهمية إنترنت الأشياء في الوقت الحاضر وفي المستقبل

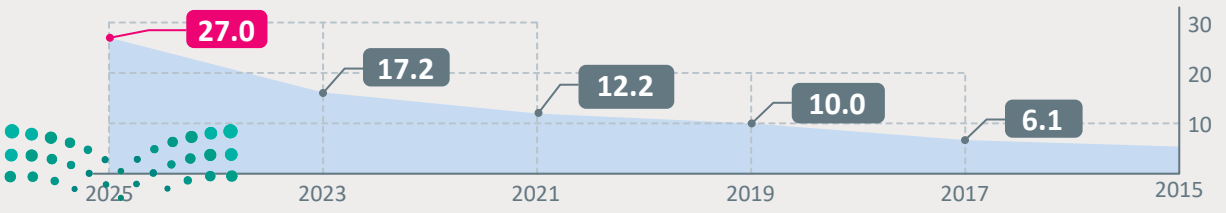
The Importance of the Internet of Things Now and in the Future

مارس البشر لعقود من الزمن عمليات إدخال البيانات في أنظمة الحاسب، حيث تُعالج هذه البيانات لعمليات أتمتة المهام أو للتوصل لمعلومات قيّمة تُسهم في اتخاذ القرارات. تعمل تقنيات إنترنت الأشياء على تغيير نموذج الحوسبة هذا، حيث أصبحت أجهزة الحاسب تدرك البيئات المختلفة بشكل مستقل، وتجمع البيانات من خلال مُستشعراتها الخاصة. تمثل تقنيات إنترنت الأشياء تحولًا تقنيًا كبيرًا، ولكن توجد عدة اعتبارات تتعلق بتأثيراتها على التطورات التقنية المستقبلية.

يُغيِّر إنترنت الأشياء من طريقة تفاعل الأفراد والمؤسسات والشركات مع محيطهم، حيث يوفر استخدام الاتصال الفوري لإدارة الأجهزة الذكية ومراقبتها مستوىً جديدًا من اتخاذ القرارات المُستند إلى البيانات، ويؤدي هذا الأمر إلى تحسين الأنظمة والعمليات وتقديم خدمات جديدة توفر الوقت والجهد للأفراد والشركات، وتعزز الجودة الحياتية الشاملة. سيزداد عدد الأشياء الذكية كأجهزة إنترنت أشياء مُستقلة أو مُدمجة في الأشياء المادية في الحياة اليومية بشكل كبير في السنوات القادمة. ومنذ إنشاء أول جهاز لإنترنت الأشياء عام 1990 وهو محمصَة خبز مُتصلة بالإنترنت، أصبح بالإمكان تحويل أي كائن مادي تقريبًا إلى كائن ذكي. تُقدِّر مؤسسة **تحليلات إنترنت الأشياء** (IoT Analytics) الخاصة بأبحاث سوق إنترنت الأشياء بأن هناك حوالي 14 مليار جهاز لإنترنت الأشياء في جميع أنحاء العالم، وتتوقع أن يصل هذا الرقم إلى 27 مليار جهاز بحلول العام 2025. لا تشمل هذه الأرقام أجهزة الحاسب والهواتف الذكية أو المُستشعرات البسيطة جدًا أو أجهزة الاتصالات أحادية الاتجاه. وستبنى الحلول الصناعية في مجالات الطاقة والمياه وإدارة النفايات وتجارة التجزئة والجُملة والنقل استخدام إنترنت الأشياء على نطاق عالمي واسع.

الاتجاهات التقنية في الكائنات الذكية Technological Trends in Smart Objects

- **الحجم في تناقص:** تستمر عملية تصغير حجم وحدات التحكم الدقيقة والمُستشعرات، وقد يصل الحال ببعضها لأن تكون صغيرة جدًا لا يمكن رؤيتها بالعين البشرية. يجعل هذا الحجم الصغير دمج الذكاء والاتصال بالشبكة في الكائنات المادية الشائعة أكثر سهولة.
- **خفض استهلاك الطاقة:** أصبحت المكونات المادية لأجهزة إنترنت الأشياء تتطلب طاقةً أقل بمرور الوقت. يُعدُّ هذا ضروريًا للمُستشعرات، حيث إن هناك الكثير من المستشعرات السلبية. تتمتع بعض المُستشعرات التي تعمل بالبطارية بعمر افتراضي يصل إلى 10 سنوات أو أكثر.
- **رفع قدرة المعالجة:** تستمر قدرات وحدات المعالجة المركزية في الارتفاع مع تصغير حجمها، ويُعدُّ هذا تطورًا مهمًا للأشياء الذكية التي تزداد قدراتها المحلية تعقيدًا وكذلك إمكانياتها في التحليلات الطرفية كما تعرفت سابقًا.
- **قدرة الاتصال في تحسن:** بالإضافة إلى تحسين سرعة نقل البيانات، تتحسن الاتصالات اللاسلكية أيضًا في مداها مع الحفاظ على انخفاض استهلاك الطاقة.
- **زيادة توحيد الاتصالات:** تعزز إنترنت الأشياء تطوير بروتوكولات الاتصال المتخصصة بشكل متزايد، والتي تدعم حالات الاستخدام المختلفة. تبذل الصناعة جهدًا كبيرًا لإنشاء معايير مفتوحة لبروتوكولات اتصالات إنترنت الأشياء.

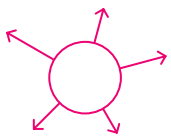
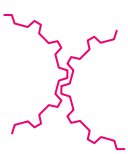


شكل 2.25: اتصالات إنترنت الأشياء النشطة تعد عالميًا بالمليارات باستثناء أجهزة الحاسب، أو الأجهزة المحمولة أو الهواتف الذكية أو الأجهزة اللوحية

تحديات أنظمة إنترنت الأشياء Challenges of Internet of Things Systems

بينما تهدف مجموعة واسعة من مكونات إنترنت الأشياء إلى تحقيق فوائد كبيرة في الإنتاجية والأتمتة، بيد أن هناك مشكلات جديدة تظهر. فقد أصبح العديد من جوانب إنترنت الأشياء حقيقة واقعة، ولكن يجب معالجة بعض التحديات لكي تصبح إنترنت الأشياء سائدة في مختلف الصناعات وفي حياتنا اليومية. يوضح جدول 2.5 بعض المشكلات والتحديات الأكثر شيوعًا التي يواجهها كل تقدُّم تقني بما فيها أنظمة إنترنت الأشياء.

الجدول 2.5: تحديات إنترنت الأشياء الشائعة

التحدي	الوصف
 قابلية التوسع	قد تكون شبكات تقنية المعلومات التقليدية كبيرة، ولكن شبكات إنترنت الأشياء يمكنها أن تكون أكبر بعدة مرات. ومع ازدياد عدد الأجهزة في النظام، يزداد تعقد الاتصالات ويصبح حجم الشبكة مشكلة. فتنشأ المشاكل في تأخر الاستجابة ووقت المعالجة بالإضافة إلى تضخم حجم الشبكات مما يجعل من الصعب إدارة أنظمة الوقت الفورية.
الأمن والحماية	نظرًا لارتباط المزيد من الكائنات الذكية ببعضها البعض وبالمستخدمين، يصبح أمان إنترنت الأشياء مهمة صعبة بشكل متزايد. لقد ازدادت المخاطر الأمنية بشكل كبير مع دمج الأجهزة في الشبكات. فأصبح اختراق اتصال أحد أجهزة إنترنت الأشياء يشكل مشكلة كبيرة بذاته، كما يُمكن أن يُستخدم هذا الجهاز لمهاجمة أجهزة وأنظمة أخرى.
الخصوصية	نظرًا لانتشار المُستشعرات في الحياة اليومية، فإن الكثير من البيانات الخاصة بالأفراد وسلوكياتهم يتم جمعها، وقد تتضمن هذه البيانات معلومات خاصة بصحة الأفراد وأنماط التسوق. يمكن للشركات الاستفادة ماديًا من هذه البيانات، وعليه نجد جدلًا واسعًا حول ملكية هذه البيانات وكيفية تمكين الأفراد من إدارة الوصول لبياناتهم الشخصية.
تحليلات البيانات والبيانات الضخمة	ينتج عن إنترنت الأشياء ومُستشعراتها المختلفة كمية هائلة من البيانات التي يجب تحليلها، وإذا عُولجت هذه البيانات بكفاءة، فإنه يمكن الحصول منها على معلومات ورؤى مهمة للغاية. تكمن المشكلة الأساسية في كيفية دمج وتقييم هذه الكميات الضخمة من البيانات المتعددة الأنواع والمصادر، وذلك قبل أن تصبح عديمة القيمة.
 التوافقية	كما هو الحال مع أي تقنية ناشئة أو قيد التطوير، تسعى بروتوكولات وبُنيات إنترنت الأشياء جاهدة للحصول على توحيد المعايير وزيادة حصتها السوقية. تعتمد بعض البروتوكولات والتطبيقات لإنترنت الأشياء على معايير تجارية، بينما يعتمد بعضها الآخر على معايير مفتوحة.

معوّقات إنترنت الأشياء الأخرى Other IoT Barriers

يوضح الجدول أدناه المعوّقات الحالية التي تحد من نشر أنظمة إنترنت الأشياء وتطويرها. إن التغلب على هذه المعوّقات يتطلب الاستمرار في تطوير التقنية، ودمج أفضل الممارسات والتجارب، والاستفادة من الدروس السابقة للأنظمة التي واجهت بعض الإخفاقات، وبالطبع السعي الدؤوب لتوفير اللوائح الحكومية الخاصة بإرشادات الأمان والخصوصية.

الجدول 2.6: معوّقات إنترنت الأشياء الشائعة

معوّقات إنترنت الأشياء	أمثلة
نشر بروتوكول IPv6	يربط إنترنت الأشياء المليارات من الأجهزة الصغيرة، لذلك يجب أن يكون لكل منها عنوان IP فريد، ويمكن فقط لعنوان الإنترنت من الجيل السادس IPv6 دعم العدد الحالي لأجهزة إنترنت الأشياء، وتؤخر عملية الانتقال إلى معيار IP الجديد التطور السريع للنظام البيئي لإنترنت الأشياء، وسيؤدي إلى ازدياد نقاط ضعف أمان الشبكة.
استهلاك طاقة المُستشعر	يجب أن تعمل مُستشعرات إنترنت الأشياء بشكل شبه مستقل، فتغيير البطاريات مثلًا على مليارات الأجهزة المنتشرة يتطلب الكثير من الوقت والجهد، كما يجب أن تكون المُستشعرات ذات كفاءة أيضًا في استهلاك الطاقة لتجنب ارتفاع التكلفة.
المسائل القانونية والتنظيمية	يتسبب استخدام أجهزة إنترنت الأشياء في العديد من التعقيدات القانونية، وتبرز مشكلات الخصوصية المتعلقة بالإنترنت بشكل خاص، وتُعدّ تحديات نقل البيانات عبر الحدود الدولية من أهم المشكلات التي تواجهها هذه التقنيات الناشئة.
المرونة وتطور التطبيقات	تواصل المُستشعرات والأجهزة كل يوم توسيع قدراتها، ويؤدي هذا إلى تطوير خدمات جديدة ومحسّنة. أصبحت التطبيقات الحديثة ذات تجربة المستخدم المخصصة التي تدعم هذه الخدمات أكثر تعقيدًا وأصبح هناك حاجة ماسة إلى وجود مطورين ومصممي تجربة مستخدم محترفين.
تكامل البيانات من مصادر متعددة	تتدفق العديد من البيانات من مصادرها بما فيها المُستشعرات، والأجهزة المحمولة، وموجزات الشبكات الاجتماعية، وموارد الويب الأخرى، في تطبيقات إنترنت الأشياء التي تُنتج البيانات بتنسيقات متنوعة، وتزداد صعوبة تصفية هذه البيانات ومعالجتها بكفاءة.

شكل 2.26: معوّقات إنترنت الأشياء الشائعة

مشكلات أمان إنترنت الأشياء IoT Security Issues

أحد أكبر المشكلات الناشئة عن الاستخدام العالمي للإنترنت، والازدياد المتسارع في الأجهزة في إنترنت الأشياء واستخدام السحابة هو أمان هذه البيئة الرقمية العالمية بأكملها. لقد وُجدت شبكات البيانات منذ عقود عديدة، ولكن معظمها كان يتعذر الوصول إليه علنًا، حيث اتسمت تلك الشبكات بوجود معدات وبروتوكولات الأمان الخاصة. إن اتصال مليارات الأجهزة على شبكات البيانات التي أصبحت جزءًا من شبكة الإنترنت هو السبب في ازدياد الاختراقات الأمنية. قد تعمل أجهزة إنترنت الأشياء ببساطة على تشغيل وإطفاء إنارة الشوارع للحفاظ على الطاقة، ولكن بعضها قد يتفاعل أيضًا مع البيانات الحساسة كالبيانات الطبية للأشخاص، ومن الضروري التعامل مع تلك المخاوف الأمنية وذلك بدءًا من مرحلة تصميم النظام. وتتعرض شبكات إنترنت الأشياء لمجموعة متنوعة من الهجمات أكثر من الشبكات الأخرى، وتزداد قوة وتعقيد هذه الهجمات بشكلٍ سريعٍ ومتطور.

يجب أن تضمن أنظمة إنترنت الأشياء إجراء تفاعلات المستخدم في بيئة آمنة. يجب على خبراء أمن إنترنت الأشياء تطبيق ومراعاة الأخذ بالاعتبارات الآتية لتجنب الوصول غير المرغوب فيه إلى البيانات الخاصة:

- نماذج مُصادقة لا مركزية موثوقة.
- تقنيات التشفير وحماية البيانات الفعّالة.
- أمن الحوسبة السحابية وموثوقيتها.
- التحكم في البيانات.
- المخاوف القانونية والمسؤولية.
- نقاط ضعف الاتصالات والشبكات.
- الوصول وحقوق المستخدم وقوانين المشاركة لتقاسم القيمة.
- أجهزة آمنة وغير مكلفة.
- إدارة سياسات الخصوصية.

شكل 2.27: معايير أمان إنترنت الأشياء

أمثلة: مشكلات الأمان الشائعة بتقنيات تحديد الهوية بموجات الراديو

Examples: Security Issues with RFID Technologies

يُعدّ بروتوكول (RFID) أحد بروتوكولات الاتصالات الأكثر استخدامًا لمعالجة بيانات تحديد الهوية في الكائنات الذكية. يستخدم (RFID) موجات الترددات اللاسلكية للتفاعل، ولتبادل البيانات دون الحاجة إلى الاتصال الجسدي. يتكون نظام (RFID) من مُكونين، وهما أجهزة إرسال (RFID) وأجهزة استقبال (RFID)، أما رمز المنتج الإلكتروني (EPC) فهو المُعرِّف الفريد للكائن الذكي. تتميز رقاقات (RFID) بأنها إما نَشطة أو سَلبية. تسمح البطارية المدمجة بالرقاقة النشطة بالتفاعل عن بُعد ما بين مُعرف (EPC) الفريد الخاص بها مع المُعرفات (EPCs) المحيطة على مسافة محدودة. تعمل الرقاقات السلبية دون بطارية، ولا تُقرأ البيانات إلا عندما يُنشِّط جهاز الإرسال والاستقبال الرقاقة في المدى القصير، مما يعرض البيانات الموجودة في رقاقة (RFID) لخطر التلاعب والتخريب والإزالة على الرغم من استخدام تقنيات التشفير لضمان الخصوصية وإثبات صحة البيانات.

الجدول 2.7: أمثلة على نقاط الضعف الأمنية لأنظمة إنترنت الأشياء من خلال استغلال RFID

الثغرة الأمنية	مثال على الهجوم
هجوم على الموثوقية تعطيل رقاقة غير مُصرَح به	تتسبب هذه الهجمات في تعطيل رقاقات (RFID) مما يجعل قارئ (RFID) يتصرف بطريقة خاطئة عند قيامه بمسحها. فمُعرف رمز المُنتج الإلكتروني (EPC) سيشير إلى معلومات خاطئة لا تتطابق مع خصائص وجهة رقاقة (RFID). في العادة يُنفّذ مثل هذا الهجوم عن بُعد، مما يسمح للمهاجم بتغيير تصرف الرقاقة من مسافة بعيدة.
الهجوم على سلامة البيانات استنساخ الرقاقات غير المُصرَح به	تندرج سرقة معلومات هوية مُعرفات (EPC)، وتعديل الرقاقات من قِبل أجهزة قراءة (RFID) غير مُصرح بها ضمن هذه الفئة، ويُمكن نسخ الرقاقة بسهولة بمجرد الحصول على معلومات الهوية الخاصة بها، وبالتالي استخدامها للتحايل على آليات الأمان والحماية أو تزويرها وإنشاء ثغرات أمنية جديدة أثناء عمليات التحقق التلقائي.
الهجوم على السرية	قد يؤدي تتبع رقاقة (RFID) من خلال أجهزة القراءة غير المصرح لها إلى الكشف عن معلومات حساسة تحتويها هذه الرقاقة.
هجوم على الإتاحة (التوفر) هجوم إيقاف الخدمة (DoS)	يتم ذلك بتشويش النظام من خلال التداخلات اللاسلكية أو بحجب الإشارات اللاسلكية أو بتعطيل رقاقة (RFID).

مشكلات الأمان مع تقنيات شبكات الاستشعار اللاسلكية

Security Issues with Wireless Sensor Networks Technologies

تُعدّ شبكات الاستشعار اللاسلكية (WSN) مسؤولة عن نقل البيانات والمعلومات بين الكائنات الذكية في أنظمة إنترنت الأشياء، وتتألف هذه الشبكات من عُقَد مستقلة تتواصل بتردد وقدرة محدودة، كما تتكون عُقدة الاتصال من بطارية ومُستشعرٍ وذاكرة وجهاز إرسال واستقبال لاسلكي ومعالج دقيق. ونظرًا لمدى الاتصال المحدود، يكون لكل عُقدة مُستشعر. تُرحّل المعلومات بين المصدر والمحطة الأساسية في مراحل متعددة، وتقوم المُستشعرات اللاسلكية بجمع ونقل البيانات المطلوبة بالتنسيق مع العقد الأخرى للتوجيه إلى النظام المركزي، وتتسم المُستشعرات اللاسلكية بقدرات حاسوبية محدودة وطاقة محدودة كذلك، مما يجعل العديد من طرائق الحماية التقليدية صعبًا أو مستحيل التنفيذ.

شبكة الاستشعار اللاسلكية (Wireless sensor network):

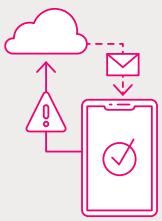
تتكون شبكة المُستشعرات اللاسلكية (WSN) من مُستشعرات مستقلة مشتتة تراقب الظروف المادية أو البيئية التي تنقل البيانات بشكل جماعي إلى موقع مركزي

وزارة التعليم
Ministry of Education
2023 - 1445

مخاوف تتعلق بالأمان والخصوصية Security and Privacy Concerns

يمكن أن يُشكل أي جهاز مُتصل بالشبكة جزءًا محتملًا من البُنية التحتية لإنترنت الأشياء وبياناتها الحساسة. تُعدّ المخاوف بشأن أمن البيانات وخصوصيتها مُهمة جدًا حيث يرتبط مستوى تعقيد الأنظمة بوجود المزيد من نقاط الضعف المتعلقة بتوفير خدمات إنترنت الأشياء.

الجدول 2.8: المخاطر الأمنية بناءً على مستويات نظام إنترنت الأشياء

مستويات نظام إنترنت الأشياء	المخاطر الأمنية
مستوى الجهاز	يجب أن تُثبت أجهزة إنترنت الأشياء هويتها للحفاظ على الموثوقية، وعليها التقليل من البيانات المُخزنة محليًا لحماية الخصوصية. نظرًا لوجود أجهزة إنترنت الأشياء في كل مكان في البيئة المحيطة، فإن الأمان المادي مُهم أيضًا، ويؤدي هذا إلى الحاجة إلى تصميم حماية لاختراقات الأجهزة بحيث يصعب استخراج العناصر الحساسة مثل البيانات الشخصية أو مفاتيح التشفير أو بيانات الاعتماد، كما يجب تحديث البرامج بشكل مستمر لضمان استمراريّة الخدمة لمدّة طويلة.
 مستوى الشبكة	يمثل هذا المستوى من نظام إنترنت الأشياء الاتصال والمراسلة بين أجهزة إنترنت الأشياء والخدمات السحابية، وعادةً ما تدمج اتصالات الإنترنت بين الشبكات الخاصة والعامة، لذا فإن تأمين حركة نقل البيانات أمرٌ بالغ الأهمية. تتواصل العديد من أجهزة إنترنت الأشياء أيضًا من خلال بروتوكولات أخرى غير Wi-Fi. تُعدُّ بوابة إنترنت الأشياء المسؤولة عن الحفاظ على السرية والسلامة والتوافر عند الترجمة بين البروتوكولات اللاسلكية المختلفة.
مستوى الخدمة	يمثل هذا المستوى نظام إدارة إنترنت الأشياء، وهو المسؤول عن إدارة الأجهزة والمستخدمين وتنفيذ السياسات والقواعد وتنسيق الأتمتة عبر الأجهزة. يُعدُّ التحكم في الوصول القائم على الوظيفة لإدارة هوية المستخدمين والأجهزة والإجراءات المصرح لهم باتخاذها أمرًا بالغ الأهمية في هذا المستوى، ويجب تمكين تتبع الإجراءات لضمان إمكانية تحديد الأجهزة التي يُحتمل تعرضها للخطر عند اكتشاف سلوك غير طبيعي.
مستوى البيانات	غالبًا ما يوصف تحليل البيانات الكبيرة المُجمعة الناتجة عن إنترنت الأشياء بأنه الجانب الأكثر قيمة في إنترنت الأشياء لمقدمي الخدمات، ويعدُّ الحفاظ على خصوصية البيانات أولوية قصوى للهيئات الحكومية **كهيئة الاتصالات وتقنية المعلومات** (Communications and Information Technology Commission – CITC) في المملكة العربية السعودية، **ولجنة التجارة الفيدرالية** (Federal Trade Commission – FTC) في الولايات المتحدة الأمريكية، **ووكالة أمن الشبكات والمعلومات** (European Network and Information Security Agency – ENISA) في الاتحاد الأوروبي، حيث تُصدر هذه الهيئات إرشادات متطلبات الأمان المتعلقة بالخصوصية.

أساليب التغلب على التحديات الأمنية
Approaches to Solving Security Challenges

يجب مراعاة الأمان أثناء مرحلة تصميم نظام إنترنت الأشياء. تبدأ مرحلة حماية النظام في مرحلة التصميم على مستوى الأجهزة والبُنية التحتية للاتصالات ومستوى نظام التشغيل، متبوعة بمستوى التصميم لتتوسع حتى نشر التطبيق. يجب على الشركات والمؤسسات الحكومية اتباع سياسات حماية البيانات والامتثال للتشريعات المحلية. ويتعين على مهندسي الشبكات والحماية ذوي الخبرة والمتخصصين في تصميم أنظمة إنترنت الأشياء اختبار وتأمين أجهزة إنترنت الأشياء وشبكاتها وذلك بتطبيق أفضل الممارسات في مجال الأمن السيبراني. يجب على هؤلاء المهندسين الجمع بين المعرفة التقنية والخبرة الميدانية من مجالات الحوسبة المختلفة.

مخاوف الخصوصية Privacy Concerns

يختلف مفهوم الخصوصية باختلاف الثقافات، كما يتطوّر ويتغير بمرور الوقت. ففي الماضي كان أمرُ تركيب كاميرات المراقبة يُعدّ انتهاكًا للخصوصية، لكنه الآن أصبح أكثر انتشارًا وقَبولًا على نطاق واسع، ويعد إنترنت الأشياء مزيجًا من التطبيقات العامة والتجارية التي تنتج عنها الكثير من البيانات، ومن الضروري توجيه الاهتمام بمن لهم حق الوصول والتحكم في تلك البيانات، كما يجب فرض الخصوصية على معلومات التعريف الشخصية في أنظمة إنترنت الأشياء، ويجب فرض القيود على التخزين والكشف عن البيانات، ويجب كذلك وضع إطار ملائم للخصوصية والحماية، ويجب ضمان أن تكون البيانات خاصة وآمنة.

التنظيم الحكومي Governmental Regulation

أثار النمو السريع لتطبيقات إنترنت الأشياء العديد من المخاوف حول قضايا خصوصية البيانات والهجمات الضارة التي يمكن أن تعطل العمليات الهامة الحكومية أو الصناعية. بدأت الحكومات في جميع أنحاء العالم بالتركيز على حل هذه المشكلة من خلال المبادرات التنظيمية والتشريعية التي تشمل النظم البيئية لإنترنت الأشياء.

هيئة الاتصالات وتقنية المعلومات
Communications & Information Technology Commission

الإطار التنظيمي لإنترنت الأشياء IoT Regulatory Framework

تهدف المملكة العربية السعودية إلى أن تصبح دولة رائدة في التطوير والتطبيق لتقنيات إنترنت الأشياء وخدماتها. وقد طوّرت **هيئة الاتصالات وتقنية المعلومات** الإطار التنظيمي لإنترنت الأشياء لتنظيم متطلبات توفير خدمة إنترنت الأشياء لدعم هذا المسعى. يحدد إطار العمل اللوائح الخاصة بمعدات إنترنت الأشياء، ومُعرفات إنترنت الأشياء مثل عناوين IP التي تميز الكائنات بصورة فريدة لتسهيل الاتصالات وتقنيات إنترنت الأشياء الأخرى. وبالإضافة إلى ذلك، يتضمن الإطار التنظيمي لإنترنت الأشياء أساسيات أخرى ومعايير لمُقدمي خدمات إنترنت الأشياء، مثل التواصل مع المستفيدين فيما يتعلق بأهمية الشبكة وأمن البيانات وإرشادات حمايتها.

شكل 2.28: وثائق هيئة الاتصالات وتقنية معلومات الخاصة بإنترنت الأشياء القابلة للتنزيل

البيئة التنظيمية التجريبية للتقنية الناشئة

Emerging Technology Regulatory Sandbox

أنشأت **هيئة الاتصالات وتقنية المعلومات** (CITC) أيضًا البيئة التجريبية التنظيمية للتقنية الناشئة لتطوير تطبيقات مبتكرة وتقديمها في المملكة العربية السعودية. تأتي هذه المبادرة ضمن مسؤولية الهيئة للإشراف والرقابة على قطاع تقنية المعلومات والاتصالات، بما فيها ترخيص وتنظيم تطبيقات الاتصالات التي تدمج تقنيات إنترنت الأشياء، ويهدف صندوق الحماية التنظيمي هذا إلى دعم وتسهيل واستدامة التوسع في النظام البيئي لتطوير تطبيقات إنترنت الأشياء في المملكة العربية السعودية ونفع جميع أصحاب المصلحة في هذا القطاع بمن فيهم الشركات والعملاء. وتحاول البيئة التجريبية التنظيمية للتقنية الناشئة تقليل الوقت المطلوب لتقديم التطبيقات إلى السوق، وتقليل تكلفة تقديم الخدمة، بينما يُمكن لمطوري التطبيقات اختبار منتجاتهم وخدماتهم الجديدة في بيئة خاضعة للرقابة، وتساهم هيئة الاتصالات وتقنية المعلومات أيضًا في تطوير نظام بيئي ملائم للابتكار وذلك من خلال تحسين الوصول إلى التمويل، وتتماشى هذه الجهود مع رؤية المملكة العربية السعودية 2030 للمساعدة في نمو القطاع الخاص وتوفير الوظائف المتعلقة بالتقنية.

تمرينات

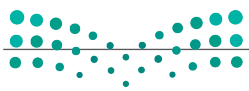
1

حدد الجملة الصحيحة والجملة الخاطئة فيما يلي:	صحيحة	خاطئة
1. يُعدُّ التطبيب عن بُعد أحد تطبيقات إنترنت الأشياء التي تشهد تراجعًا.	○	○
2. تُستخدم تطبيقات الشبكة الذكية لإنترنت الأشياء لتوفير الطاقة فقط.	○	○
3. حدثت أهم الابتكارات في تقنيات إنترنت الأشياء في السنوات العشرين الماضية.	○	○
4. ستستهلك الكائنات الذكية المزيد من الطاقة في المستقبل.	○	○
5. يُعدُّ تطبيق معايير الأمان نفسِها لجميع أنظمة إنترنت الأشياء المشكلة الأقل تعقيدًا في أنظمة إنترنت الأشياء.	○	○
6. لن يتمكن بروتوكول IPv6 من دعم العدد المتوقع للكائنات الذكية في المستقبل.	○	○
7. ستزيد كمية مصادر بيانات الإدخال للكائنات الذكية.	○	○
8. إن تقنيات RFID وWSN غير معرضة لهجمات قطع الخدمة.	○	○
9. تُعدّ تطبيقات إنترنت الأشياء وأنظمتها غير مُنظمة من قِبل السلطات الحكومية.	○	○
10. تُعدُّ خصوصية بيانات إنترنت الأشياء مصدر قلق كبير للحكومات والمنظمات.	○	○

2 **قارن بين السيارات ذاتية القيادة المزودة بتقنية إنترنت الأشياء ووسائل النقل العام الذكية. هل تعدّ تطبيقات إنترنت الأشياء هذه مُستقلة أم مُكمِّلة لبعضها؟ اعرض أفكارك أدناه.**

3 تُوفِّر مصادر الطاقة الشمسية أو طاقة الرياح المتجددة تنوعًا في توليد الطاقة. هل تعتقد بأن تقنيات الشبكة الذكية (smart grid) يمكنها جعل توزيع وإدارة الطاقة المتجددة أكثر كفاءة؟ اكتب أفكارك أدناه.

4 حدد الاتجاه التقني في الكائنات الذكية الذي سيكون الأكثر أهمية في تطوير أنظمة إنترنت الأشياء. اكتب إجابتك أدناه.

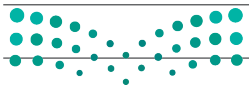


5 حسب اعتقادك، ماهي تحديات إنترنت الأشياء الأكثر شيوعًا والتي تُعدّ الأكثر صعوبة والأعلى تكلفة للتغلّب عليها؟ اكتب إجابتك أدناه.

6 ابحث في الإنترنت عن حدث أدَّت فيه ثغرةٌ أمنية إلى هجوم إلكتروني على نظام إنترنت الأشياء. ما الأضرار التي سببتها وكيف يمكن منعها؟ اكتب إجابتك أدناه.

7 باعتقادك ما النقطة الأكثر ضعفًا في شبكات الاستشعار اللاسلكية وما مدى تأثيرها على أنظمة إنترنت الأشياء. قدِّم أفكارك أدناه.

8 قدِّم وصفًا للإجراءات التي اتخذها مجلس إدارة هيئة الاتصالات وتقنية المعلومات لتنظيم اعتماد تطبيقات إنترنت الأشياء في المملكة العربية السعودية.



المشروع

تُعدُّ تطبيقات إنترنت الأشياء أنظمة معُقدة على العديد من المستويات التقنية والتشغيلية، وذلك لكي تعمل بصورة صحيحة وبفعالية.

1 اختر صناعة تُستخدم فيها إنترنت الأشياء بشكل شائع، ولكنها عُرضة للهجمات الإلكترونية واستغلال البيانات، ثم صِف كيف يمكن استخدام ثغرة أمنية لمهاجمة هذا النظام، وما التداعيات المُحتملة على المستخدمين النهائيين.

2 أنشئ عرض باوربوينت تقديمي يصفُ الصناعة التي اخترتها، ويوضِّح مشكلة الثغرة الأمنية، ويحتوي على اقتراح لحل هذه المشكلة.

ماذا تعلّمت

- التمييز بين الطبقات السحابية والضبابية والطرفية لتطبيقات إنترنت الأشياء.
- تصنيف عوامل التمكين التقنية الرئيسة لأنظمة إنترنت الأشياء.
- تصنيف بروتوكولات الشبكات والأنظمة التي تربط تطبيقات إنترنت الأشياء.
- تحديد استخدامات تطبيقات إنترنت الأشياء في المجال الاقتصادي والقطاعات الحكومية.
- وصف مدى تطور تطبيقات إنترنت الأشياء في المستقبل القريب.
- تحديد التحديات التقنية الرئيسة التي يواجهها مهندسو إنترنت الأشياء.
- تمييز التحديات الأمنية لأنظمة إنترنت الأشياء.
- وصف كيفية تنظيم تطبيقات إنترنت الأشياء حاليًا.

المصطلحات الرئيسة

IoT Services	خدمات إنترنت الأشياء
Latency	تأخير زمني
Near-Field Communication	الاتصال قريب المدى
Protocols	بروتوكولات
Radio Frequency IDentification	تحديد الهوية بموجات الراديو
Regulations	قوانين
Regulatory Framework	الإطار التنظيمي
Smart Grid	شبكة ذكية
Transmission Control Protocol	بروتوكول التحكم في الإرسال
User Datagram Protocol	بروتوكول حِزم بيانات المستخدم
Wireless Sensor Networks	مُستشعر لاسلكي

Authentication	مصادقة
Authorization	ترخيص
Data at Rest	البيانات غير النشطة
Data in Motion	البيانات النشطة
Denial of Service	إيقاف الخدمة
Edge Analytics	التحليلات الطرفية
Edge Computing	حوسبة طرفية
Edge Device	جهاز طرفي
Electronic Product Code	رمز المنتج الإلكتروني
Endpoint	نقطة النهاية
Fog Computing	حوسبة ضبابية
Gateway	بوابة
Internet Protocol	بروتوكل الإنترنت
Internet Protocol Version 6	بروتوكول الإنترنت الجيل السادس
IoT Enabler	مُمَكّنات إنترنت الأشياء

3. إنشاء تطبيقات إنترنت الأشياء باستخدام الأردوينو

ستتعرف في هذه الوحدة على الخصائص الرئيسة لتطبيق إنترنت الأشياء وطريقة تصميم وبناء تطبيقات عملية باستخدام جهاز تحكم الأردوينو الدقيق وبيئة محاكاة دوائر تينكركاد (Tinkercad Circuits).

أهداف التعلُّم

بنهاية هذه الوحدة سيكون الطالب قادرًا على أن:

- يتعرّف على مكونات جهاز التحكم الدقيق وعلى طريقة برمجته.
- يقيس البيانات المُجمَّعَة من مُستشعرات الإدخال المختلفة.
- يفهم كيفية التكامل بين بيانات المُستشعرات وخوارزميات البرمجة وطريقة عملهما معًا.
- يستخدم بيانات المُستشعرات لتشغيل التنبيهات والاستجابات الآلية.
- يصمم دوائر إنترنت الأشياء باستخدام جهاز تحكم الأردوينو الدقيق في بيئة دوائر تينكركاد (Tinkercad Circuits).
- يبرمج جهاز تحكم الأردوينو الدقيق باستخدام لغة برمجة قائمة على اللبنات البرمجية في بيئة دوائر تينكركاد (Tinkercad Circuits).
- يستخدم مُستشعر الغاز لإنشاء إنذار تَسرُّب الغاز في البيئات ذات الظروف الخطرة.

الأدوات

- بيئة محاكاة دوائر أوتوديسك تينكركاد (Autodesk Tinkercad Circuits)

وزارة التعليم
Ministry of Education
2023 - 1445

رابط الدرس الرقمي

www.ien.edu.sa

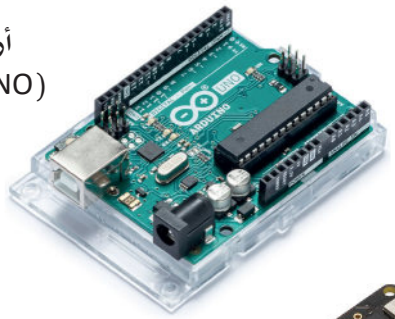
الدرس الأول

إنشاء نظام منزل ذكي

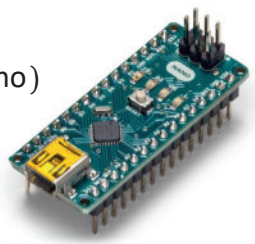
أجهزة تحكم الأردوينو الدقيقة Arduino Microcontrollers

تُصمّم أجهزة التحكم الدقيقة لاستخدامها في الحواسيب المُصغرة أحادية اللوحة، وذلك على نطاق أوسع بكثير من استخدامها في الحواسيب المكتبية أو الشخصية. فعلى سبيل المثال، تنتج أردوينو مجموعة أجهزة تحكم دقيقة مُستقلة في عملها تمامًا، مُدعَّمة بمجموعة من المعالجات الدقيقة المُدمجة، وتتميز أجهزة تحكم الأردوينو الدقيقة بتعدد وظائفها، ورغم هذا تبقى هذه الوظائف محدودة مقارنةً بإمكانات الحواسيب الشخصية، وذلك لأن أجهزة تحكم الأردوينو الدقيقة قد صُممت لأداء مهام بسيطة فقط. من لوحات الأردوينو الأكثر شيوعًا:

أردوينو أونو
(Arduino UNO)

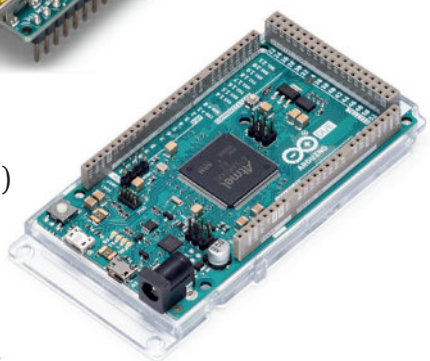


أردوينو نانو
(Arduino Nano)



أردوينو بورتينتا
(Arduino Portenta)

أردوينو ديو
(Arduin Due)



شكل 3.1: لوحات أجهزة تحكم الأردوينو الدقيق

جدول 3.1: خصائص نماذج لوحات الأردوينو

النموذج	نوع منفذ USB	المداخل / المخارج	سرعة المعالج	ذاكرة فلاش	ذاكرة SRAM
أردوينو نانو 33	Mini-B	26 منفذ	48 MHz	256 KB	32 KB
أردوينو أونو R3	Type B	20 منفذ	16 MHz	32 KB	2 KB
أردوينو ديو	Micro-B	68 منفذ	84 MHz	512 KB	96 KB
أردوينو بورتينتا H7	Type-C	80 منفذ	480 MHz	تصل إلى 128 MB	تصل إلى 64 MB

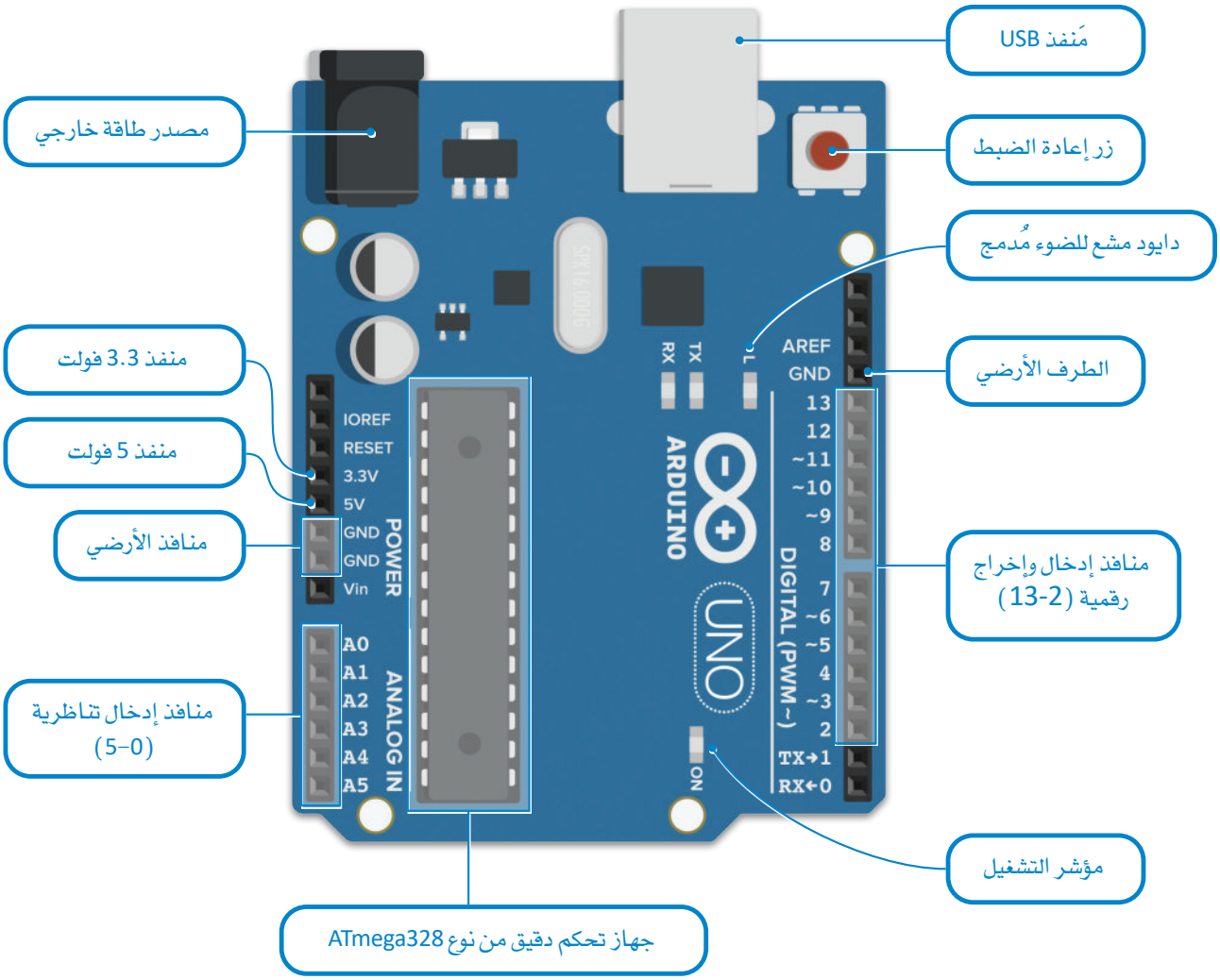
ملاحظة

يُمكن تصنيع وحدات التحكم الدقيقة بسرعة وسهولة بالمقارنة مع صناعة الحواسيب الشخصية أو المحمولة، مما يُقلل تكلفة الإنتاج بشكل كبير.

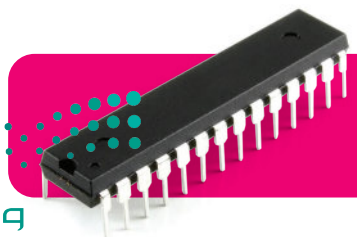
أردوينو أونو R3 Arduino UNO R3

تعتمد لوحة أردوينو **R3** (Arduino UNO R3) على جهاز تحكم دقيق من نوع ATmega.

تحتوي هذه اللوحة على 14 منفذ إدخال وإخراج رقمي، حيث يُمكن استخدام 6 منها كمُخرجات يُطلق عليها تسمية **تضمين عرض النبضة** (Pulse Width Modulation– PWM)، ويُستخدم 2 منها لإرسال البيانات التسلسلية (Tx→1)، ولاستقبالها (Rx←0) وتستخدم 6 منها كمداخل تناظرية، ومنفذ لتوصيل USB، ومقبس للطاقة، وزر لإعادة الضبط.



شكل 3.2: لوحة الأردوينو أونو R3 (Arduino Uno R3)

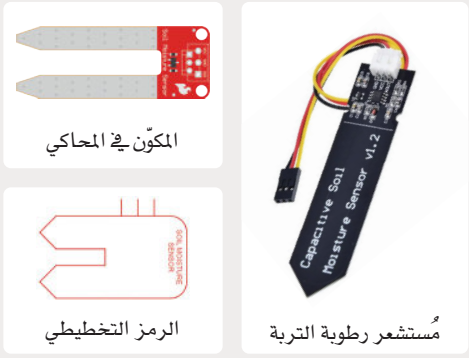


يعدّ ATMega328P جهاز تحكم دقيق أحادي الرقاقة يُستخدم بصورة شائعة في منتجات الأردوينو. ويتميز بأدائه العالي واستهلاكه المنخفض للطاقة.

أمثلة على مُستشعرات خارجية مُلحقة بأجهزة التحكم الدقيقة
Some Examples of External Sensors for Microcontrollers

مُستشعرات رطوبة التربة
Soil Moisture Sensors

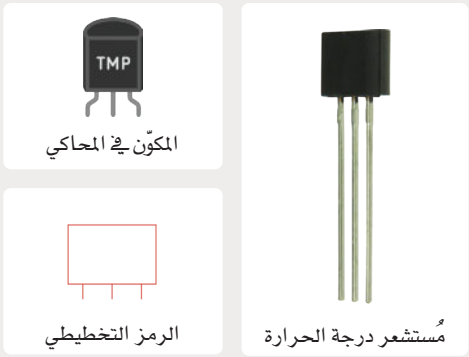
تقيس مُستشعرات رطوبة التربة حجم الماء الموجود داخل التربة، ونظرًا لأن هذا القياس ينطوي على كمٍ كبير من عمليات معالجة التربة، فإنّ رطوبة التربة تقاس بشكل غير مباشر، وذلك باستخدام خصائص أخرى للتربة مثل المقاومة الكهربائية (كلما زادت الرطوبة قلت المقاومة)، وتُعدّ مُستشعرات رطوبة التربة ضرورية في مجال الزراعة، كما تُستخدم في تطبيقات المراقبة مثل التحكم في الري لأغراض الصناعة، والأغراض المنزلية، وري النباتات المكتبية والحدائق الطبيعية.



شكل 3.3: مُستشعر رطوبة التربة

مُستشعرات درجة الحرارة
Temperature Sensors

يُستخدم مُستشعر درجة الحرارة TMP36 في قياس درجة الحرارة، ويُنتج جهد إخراج تناظري يتناسب مع درجة الحرارة التي يستشعرها، ويُحوّل هذا الجهد إلى قراءة لدرجة الحرارة بالدرجات المئوية. يُمكن لهذا المُستشعر قياس درجات الحرارة في نطاق يتراوح بين 40- وحتى 125 درجة مئوية، ويُستخدم المستشعر TMP36 بشكل أساسي في التطبيقات التي تتضمن تنظيم درجة الحرارة وقياسها، ويتميز بعدم حاجته إلى المعايرة، وبالتالي يمكن استخدامه دون أي ملحقات إضافية.



شكل 3.4: مُستشعر درجة الحرارة TMP36

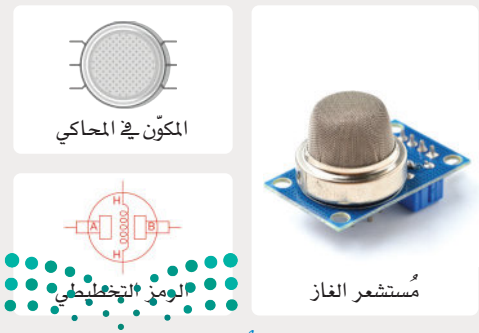
مُستشعرات الحركة
PIR Sensors

تستكشف مُستشعرات الحركة (Passive Infrared Sensors – PIR Sensors) الإلكترونية وجود الأشياء ضمن مجال مُعين، وتعمل هذه المُستشعرات عن طريق قياس إشارات الموجات تحت الحمراء الموجودة في مجال رؤيتها.

شكل 3.5: مُستشعر الحركة (PIR sensor)

مُستشعرات الغاز
Gas Sensors

هي مقاومات كيميائية تكتشف وجود مستويات مُرتفعة من الدخان والغازات الأخرى مثل البروبان والهيدروجين وأول أكسيد الكربون، حيث تتغير قيمة المقاومة الكيميائية عند ملامسة الغاز لها ويُمكن لهذه المُستشعرات اكتشاف تركيز غاز بين 200 و 10,000 جزء في المليون، كما تُستخدم مثل هذه المُستشعرات لمراقبة المناطق التي قد تتعرض لخطر الحرائق أو انبعاث غازات سامة.



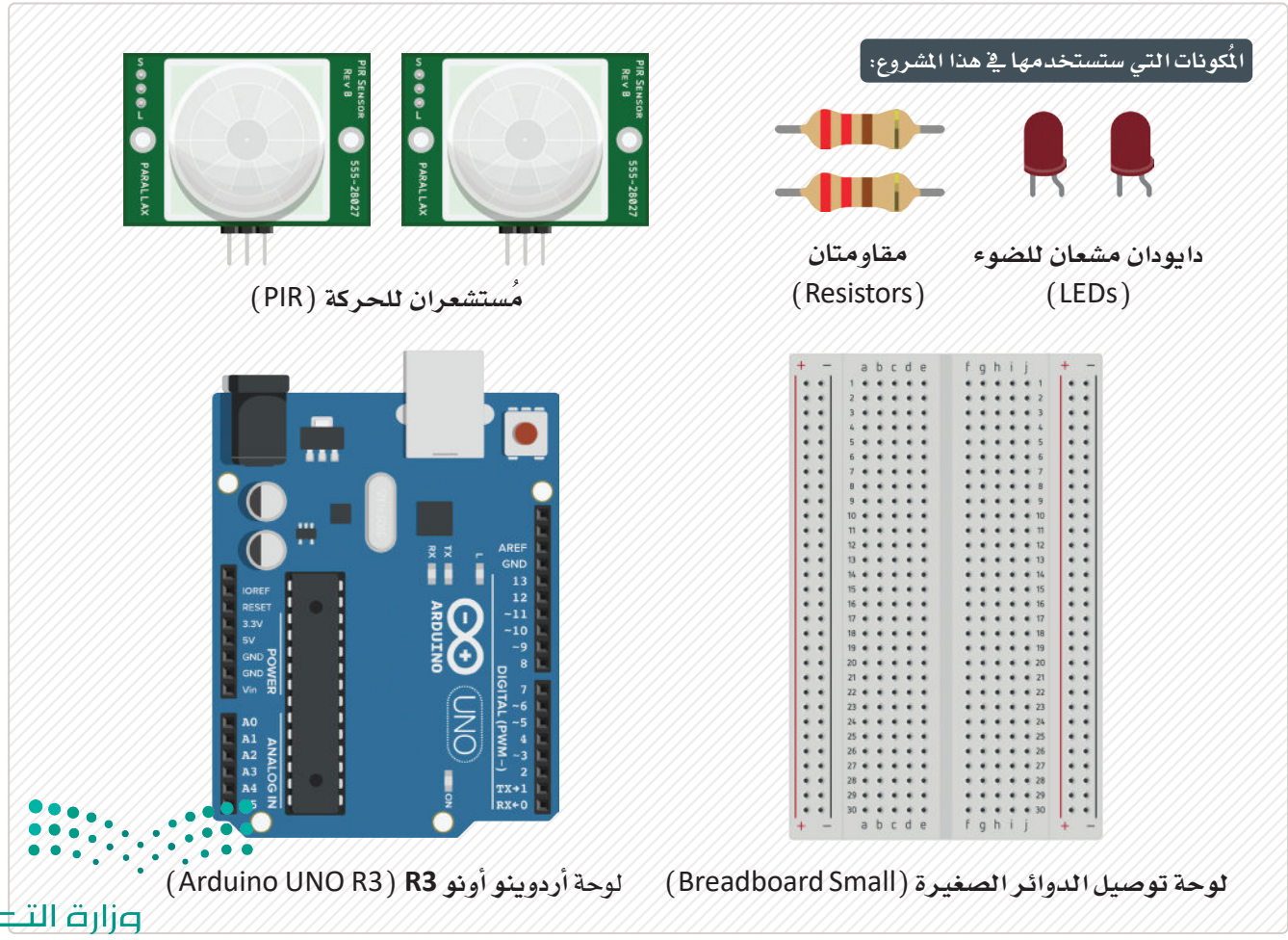
شكل 3.6: مُستشعر الغاز

إنشاء نظام المنزل الذكي Build a Smart Home System

يزداد كل يوم تجهيز المنازل بمُستشعرات ذكية، حيث تُستخدم هذه المستشعرات لتحسين جودة الحياة وتسهيل القيام بالأعمال المنزلية، وتتمثل إحدى التقنيات "الذكية" في تقنية المصابيح المنزلية الذكية التي يكون تشغيلها وإيقافها تلقائيًا عن طريق استشعار حركة الأشخاص في غُرف المنزل. ستستخدم في هذا المثال لوحة جهاز التحكم الدقيق **أردوينو أونو R3** (Arduino UNO R3) لمحاكاة نظام تلقائي لإضاءة غرفة في منزل ذكي، وذلك في محاكي **تينكركاد** (Tinkercad)، كما ستستخدم **مُستشعري حركة** (PIR sensors) ليكتشفا وجود أي كائن ضمن **مجال رؤيتهما** (Field of View) في أي من الغرفتين، وعند وجود شخص في مجال رؤية المُستشعر، سيضيء الدايود المشع للضوء المُلحق به، وعند مغادرته ستنطفئ الإنارة. سيُمثّل المُستشعر الثاني غرفةً أخرى تنتظر دخول شخصٍ ما.

ستستخدم المكونات الآتية لهذا المشروع:

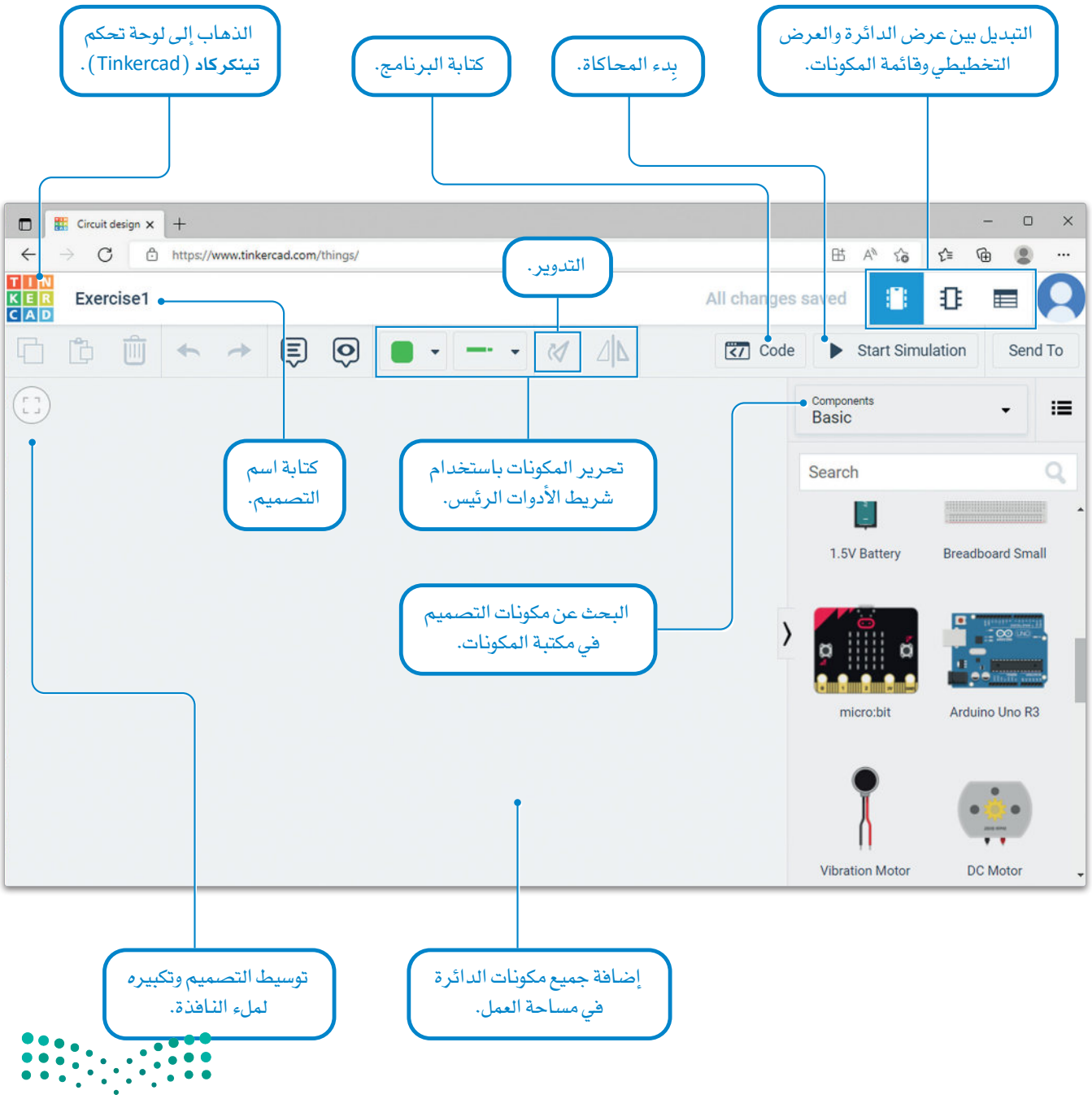
- لوحة أردوينو أونو **R3** (Arduino UNO R3).
- **مُستشعران للحركة** (PIR) يعملان بالموجات تحت الحمراء.
- **دايودان مشعان للضوء** (LEDs).
- **مقاومتان** (Resistors).
- **لوحة توصيل الدوائر الصغيرة** (Breadboard Small).



شكل 3.7: مكونات مشروع المنزل الذكي

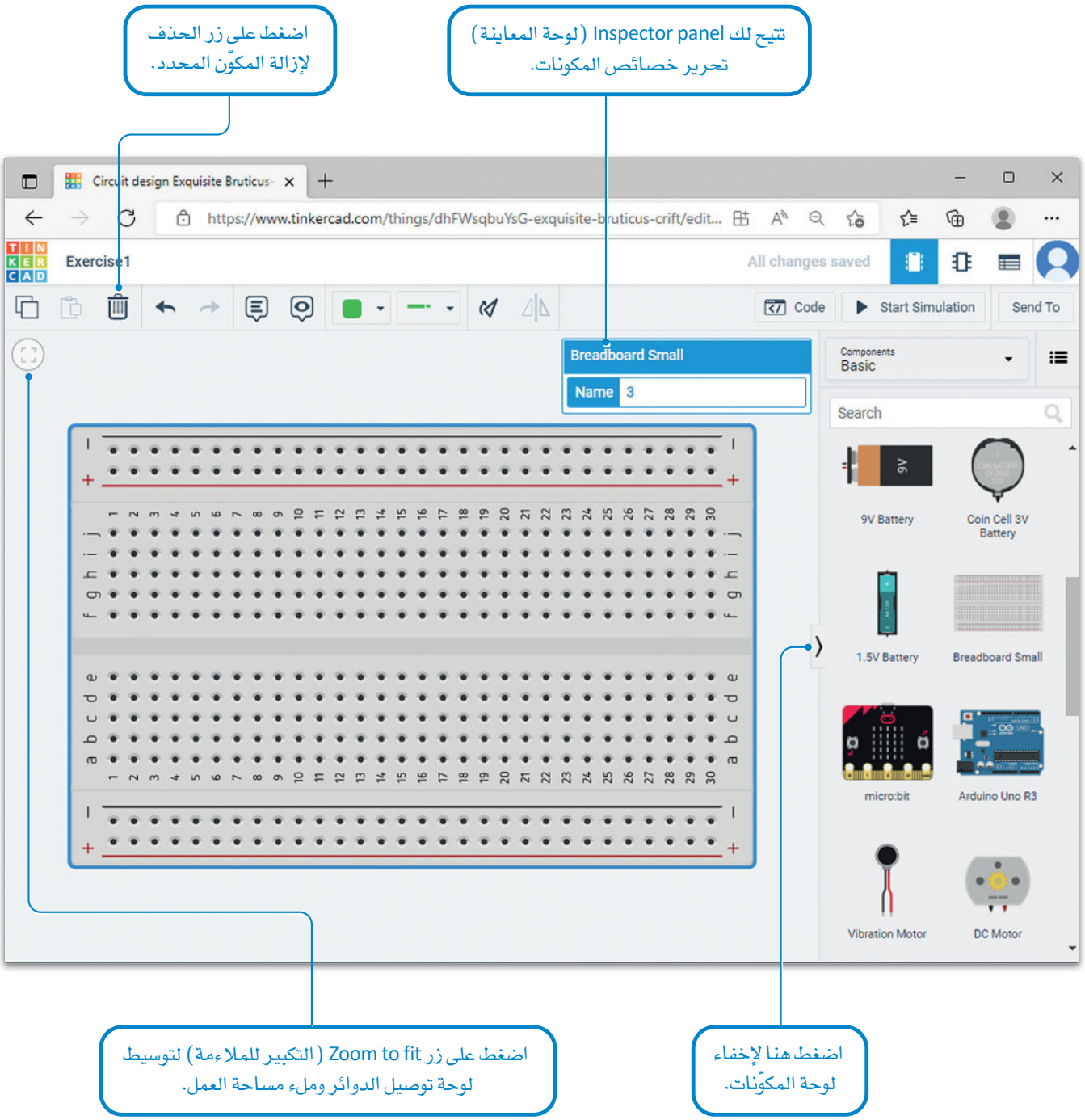
ابدأ الآن بتصميم دائرة جديدة في محاكي **تينكركاد** (Tinkercad).

إن **دوائر تينكركاد** (Tinkercad Circuits) هي محاكي قائم على الويب يُستخدم في إنشاء نماذج أولية للدوائر الإلكترونية، وبرمجتها باستخدام أجهزة تحكم دقيقة مثل الأردوينو. افتح دوائر تينكركاد من موقع الويب https://www.tinkercad.com واختر Circuits (الدوائر) لفتح النافذة الرئيسة للمحاكي.



شكل 3.8: النافذة الرئيسة لمحاكي دوائر تينكركاد (Tinkercad Circuits)

أضف مكونًا إلى التصميم بالضغط على المُكون من مكتبة **المكونات** (Components)، ثم اضغط على أي مكان في مساحة العمل.



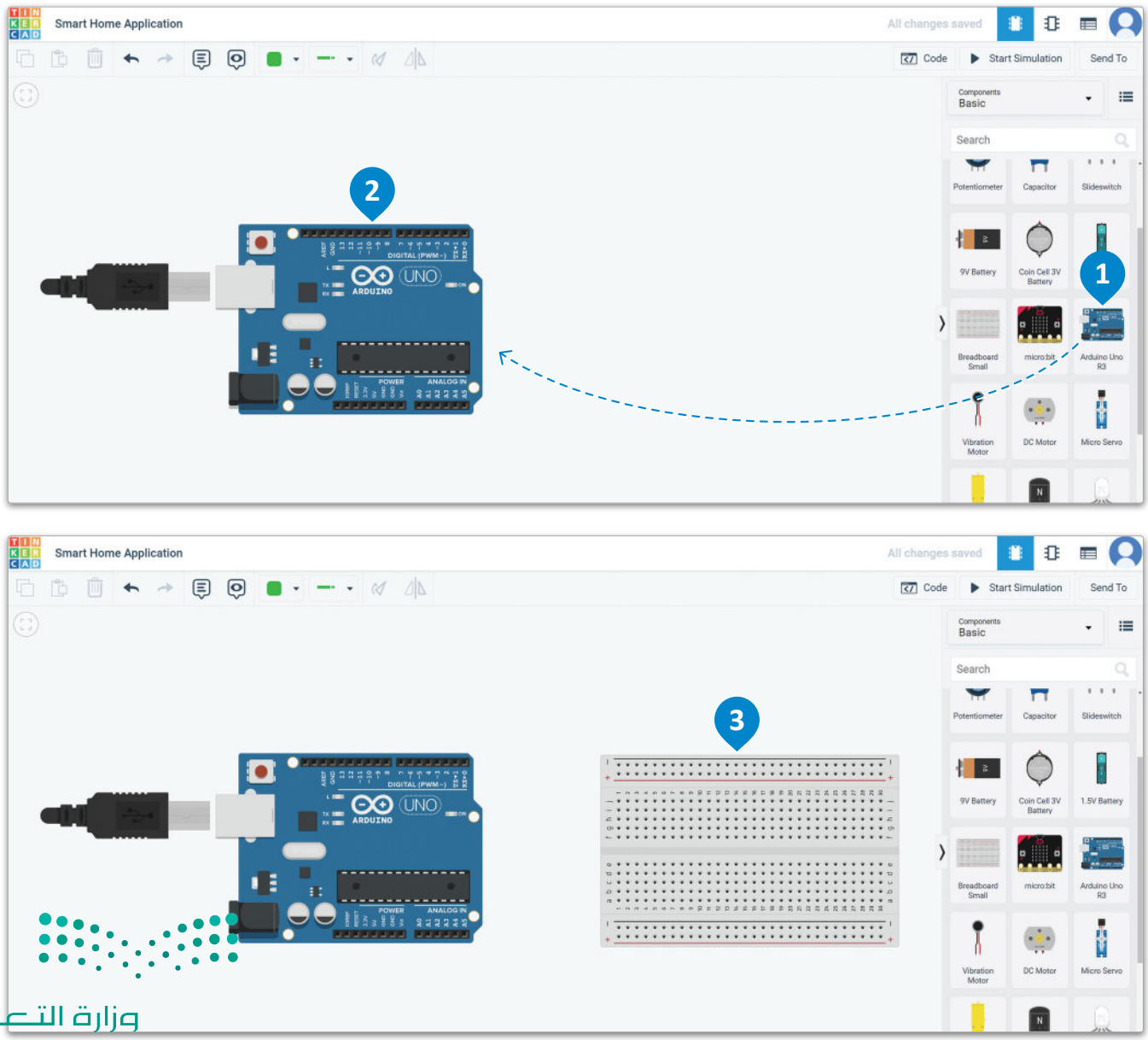
شكل 3.9: تحرير المكونات

ابحث عن المكونات التي ستحتاج إليها لهذا المشروع وانقلها إلى مساحة العمل، ثم ابحث عن لوحة **أردوينو أونو R3** (Arduino UNO R3) وقم بإضافتها، وكرر الشيء نفسه لإضافة **مُستشعِرَي الحركة** (PIR)، واثنين من **الدايودات المشعة للضوء** (LED) ومقاومتين (Resistor)، و**لوحة توصيل الدوائر الصغيرة** (Breadboard small) في مساحة العمل.

لإضافة المكونات إلى مساحة العمل:

> ابحث عن Arduino UNO R3 (لوحة أردوينو أونو R3) في مكتبة Components (المكونات)، 1 واسحبها وأفلتها في مساحة العمل. 2

> ابحث عن Breadboard Small (لوحة توصيل الدوائر الصغيرة) في مكتبة Components (المكونات)، واسحبها وأفلتها في مساحة العمل. 3

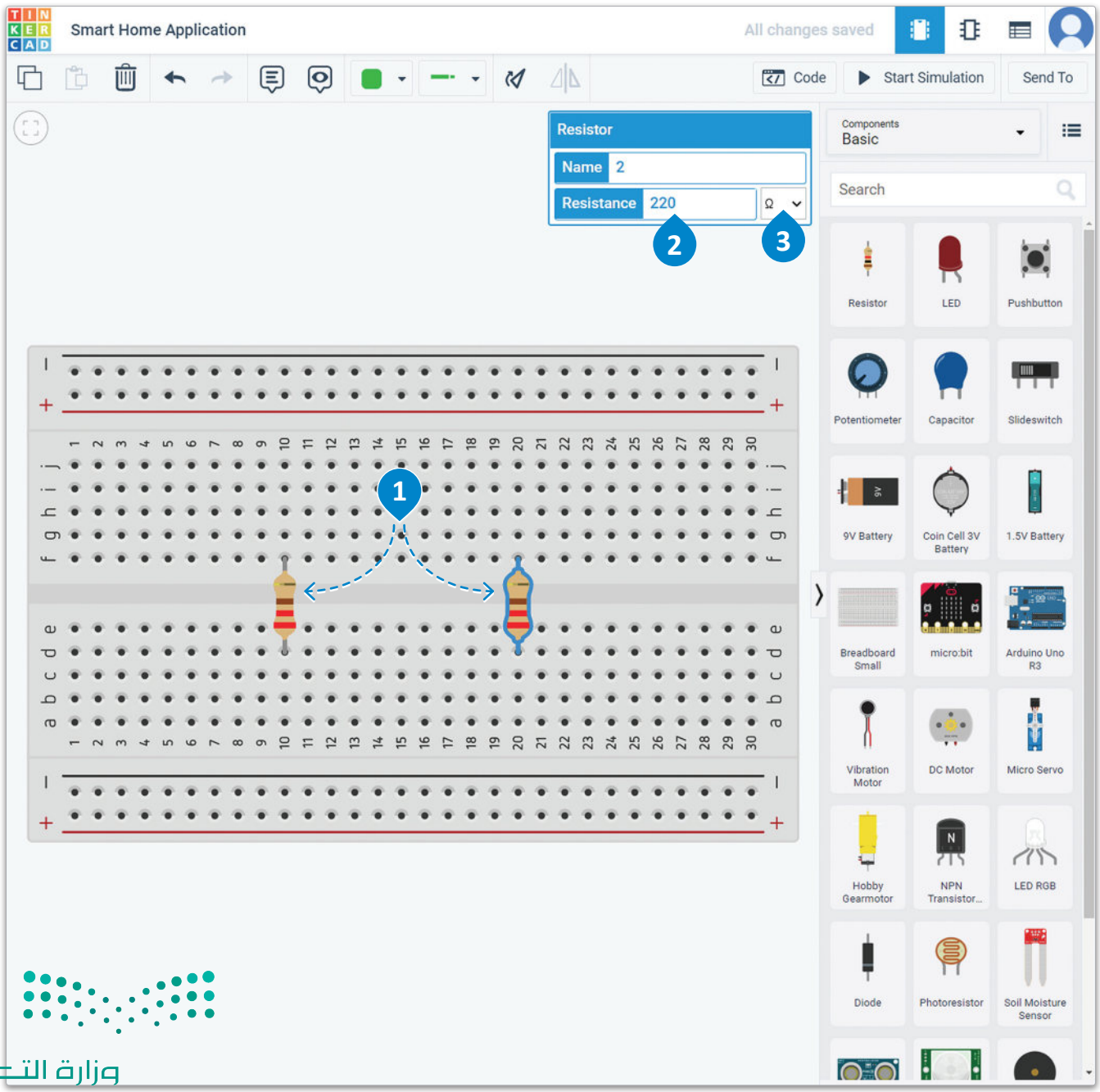


شكل 3.10: إضافة المكونات إلى مساحة العمل

ستضيف الآن مقاومتين إلى **لوحة توصيل الدوائر الصغيرة** (Breadboard Small).

لإضافة المقاومات إلى لوحة توصيل الدوائر:

- اسحب وأفلت Resistors (**المقاومات**) من مكتبة Components (**المكونات**) على Breadboard Small (**لوحة توصيل الدوائر الصغيرة**). 1
- من Inspector panel (لوحة المعاينة)، اضبط قيمة كل **مقاومة** (Resistor) إلى 220، 2 واضبط الوحدة Ω. 3



شكل 3.11: إضافة المقاومات إلى لوحة توصيل الدوائر

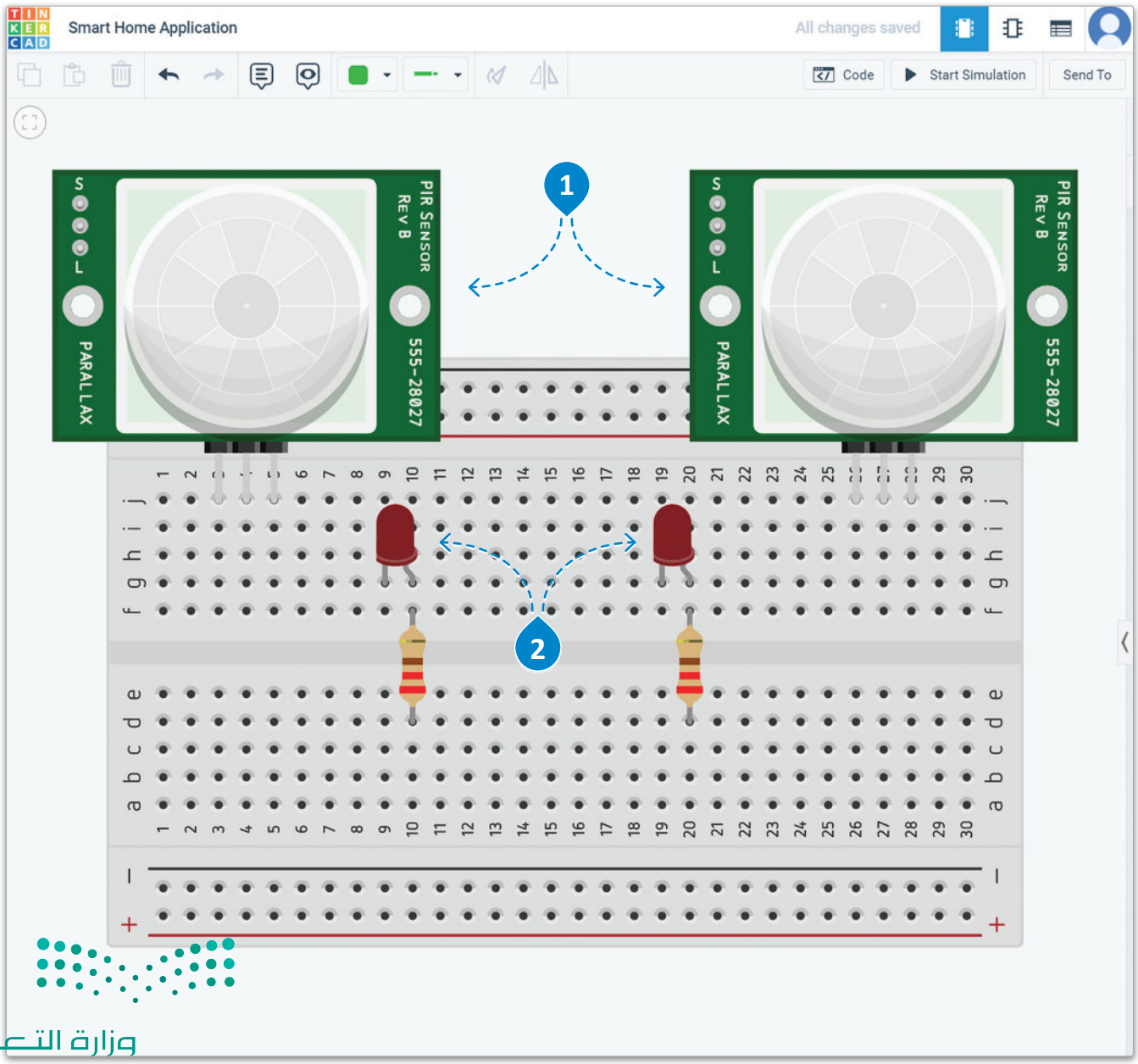
استمر بإضافة دايودين مشعين للضوء إلى لوحة توصيل الدوائر، وآخرين من مُستشعرات الحركة.

ستحتاج إلى توصيل مِصعد كل دايود مشع للضوء على التوالي مع مقاومته المُقابلة.

لإضافة المكونات إلى لوحة توصيل الدوائر:

> اسحب وأفلت PIR sensors (مُستشعرات الحركة) من مكتبة Components (المكونات) وضعها في Breadboard Small (لوحة توصيل الدوائر الصغيرة). 1

> اسحب وأفلت LEDs (الدايودات المشعة للضوء) من مكتبة Components (المكونات)، ثم صِل مِصعد كل دايود مشع للضوء بالتوازي مع مقاومته المقابلة في Breadboard Small (لوحة توصيل الدوائر الصغيرة). 2



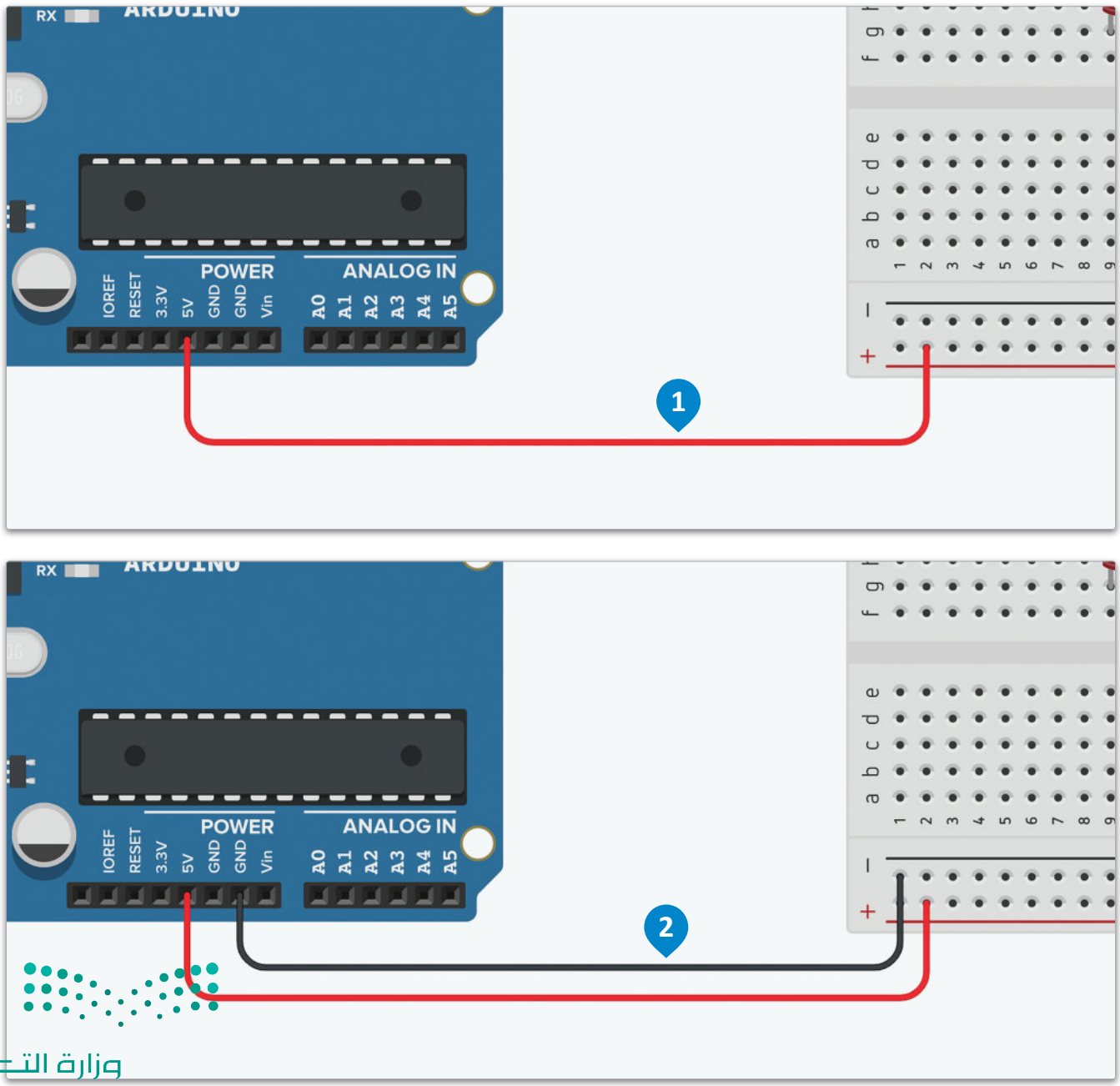
شكل 3.12: إضافة الدايودات المشعة للضوء

قُم بتوصيل لوحة الأردوينو **بلوحة توصيل الدوائر الصغيرة** (Breadboard Small) عن طريق توصيل طرف **جهد 5 فولت** (5V) بالعمود الموجب، و**الطرف الأرضي** (GND) بالعمود السالب.

لتوصيل لوحة الأردوينو أونو R3 (Arduino UNO R3):

> قُم بتوصيل 5V (جهد 5 فولت) في لوحة الأردوينو بالعمود الموجب من Breadboard Small (لوحة توصيل الدوائر الصغيرة) وغيّر لون السلك إلى Red (الأحمر). 1

> قُم بتوصيل GND (الطرف الأرضي) من لوحة الأردوينو أونو R3 بالعمود السالب من Breadboard Small (لوحة توصيل الدوائر الصغيرة) وغيّر لون السلك إلى black (الأسود). 2



شكل 3.13: توصيل الأسلاك بلوحة الأردوينو

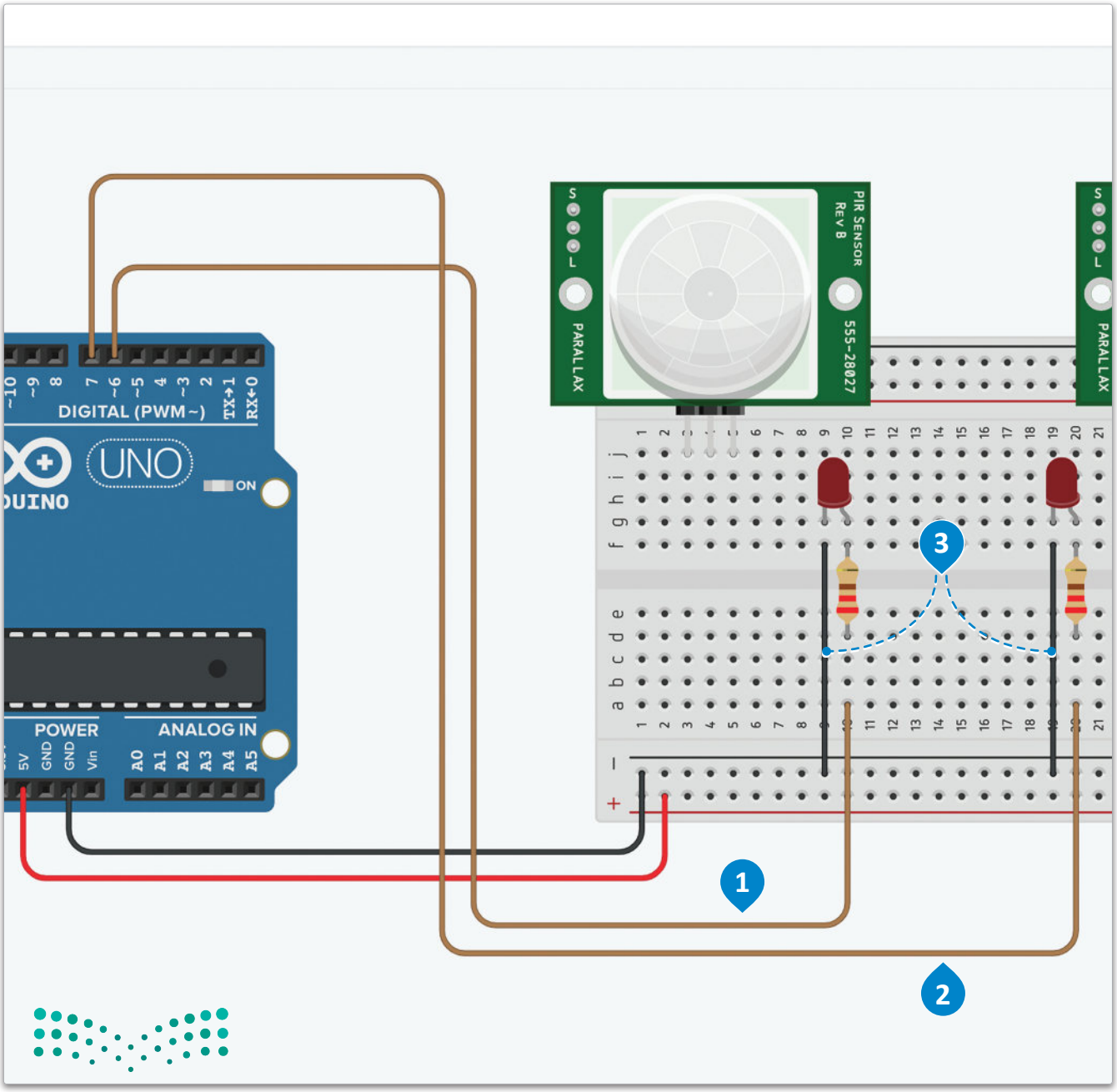
بعد ذلك قُم بتوصيل مقاومات الدايودات المشعة للضوء بمنافذ الأردوينو الرقمية 6 و7.

لتوصيل مقاومات الدايودات المشعة للضوء:

> قُم بتوصيل أحد المقاومات بـ Digital pin 6 (الطرف الرقمي 6) من لوحة الأردوينو وغيّر لون السلك إلى البني. 1

> قُم بتوصيل المقاومة الأخرى بـ Digital pin 7 (الطرف الرقمي 7) من لوحة الأردوينو وغيّر لون السلك إلى البني. 2

> قُم بتوصيل مِهبطي الدايودات المشعة للضوء بالعمود السالب في Breadboard Small (لوحة توصيل الدوائر الصغيرة). 3



شكل 3.14: توصيل أزواج أسلاك مقاومات الدايودات المشعة للضوء

ختامًا ولإتمام التوصيلات، ستحتاج إلى توصيل **مُستشعر حركة الغرفة الأولى** (PIR Room1) و **مستشعر حركة الغرفة الثانية** (PIR Room2) بلوحة الأردوينو وبالأعمدة السالبة والموجبة في لوحة توصيل الدوائر الصغيرة.

لتوصيل مُستشعرات الحركة:

> قُم بتوصيل طرف إشارة PIR Sensor (مُستشعر الحركة) بـ Digital pin 8 (الطرف الرقمي 8)، وغيّر لون السلك إلى اللون green (الأخضر). 1

> قُم بتوصيل طرف طاقة PIR Sensor (مُستشعر الحركة) بالعمود الموجب من Breadboard Small (لوحة توصيل الدوائر الصغيرة) وغيّر لون السلك إلى اللون red (الأحمر). 2

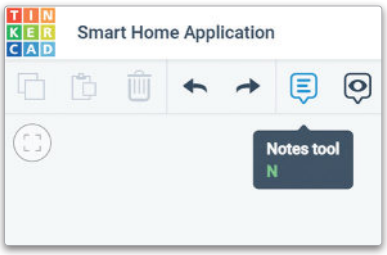
> قُم بتوصيل Ground (الطرف الأرضي) لمُستشعر الحركة بالعمود السالب من Breadboard Small (لوحة توصيل الدوائر الصغيرة) وغيّر لون السلك إلى black (الأسود). 3

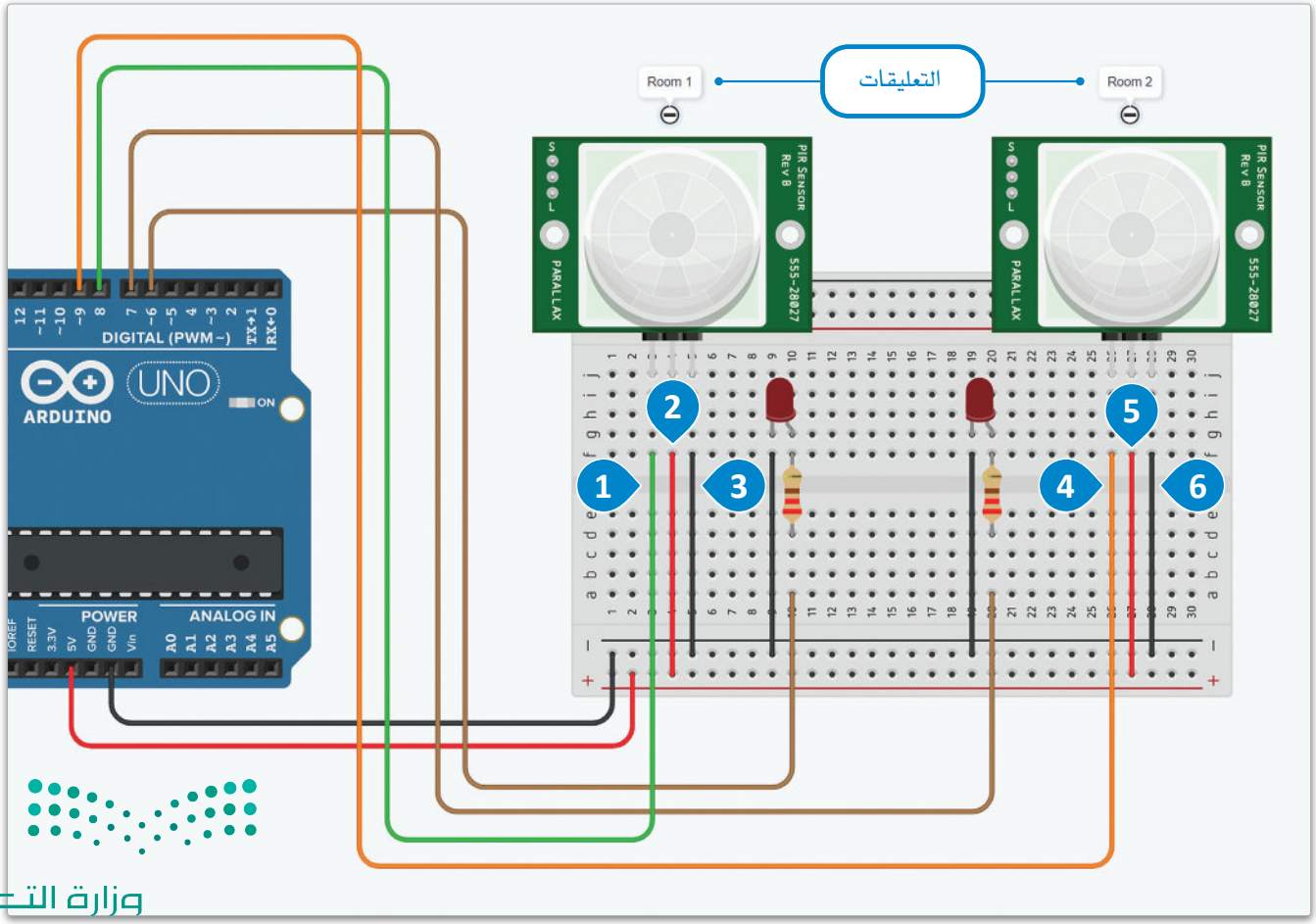
> قُم بتوصيل طرف إشارة PIR Sensor (مُستشعر الحركة) بـ Digital pin 9 (الطرف الرقمي 9) وغيّر لون السلك إلى اللون orange (البرتقالي). 4

> قُم بتوصيل طرف طاقة PIR Sensor (مُستشعر الحركة) بالعمود الموجب من Breadboard Small (لوحة توصيل الدوائر الصغيرة) وغيّر لون السلك إلى اللون red (الأحمر). 5

> قُم بتوصيل Ground (الطرف الأرضي) لمُستشعر الحركة بالعمود السالب من Breadboard Small (لوحة توصيل الدوائر الصغيرة) وغيّر لون السلك إلى black (الأسود). 6

من المهم جدًا إضافة التعليقات والملاحظات أثناء عملك كما هو الحال عندما تقوم بالبرمجة. يمكنك في Tinkercad إضافة التعليقات في مساحة العمل. على سبيل المثال، يمكنك إضافة تعليقات للإشارة إلى الغرفتين باستخدام أداة الملاحظات (Note tool).





شكل 3.15: توصيل مُستشعرات الحركة

لبنات التعليمات البرمجية Code Blocks

الآن وبعد أن انتهيت من إعداد المكونات، ستستكشف بيئة البرمجة التي ستستخدمها في هذه الوحدة. يوفر تينكركاد تقنية البرمجة القائمة على اللبنات البرمجية لتبسيط عملية برمجة وحدة التحكم الدقيقة.

شكل 3.16: اللبنات البرمجية

تصنيفات اللبنات البرمجية

الإخراج Output

تسمح لك فئة لبِنات **الإخراج** (Output) بتحديد المنافذ **الرقمية** (Digital) و**التناظرية** (Analog) وإرسال الأوامر إلى مكونات وحدة التحكم الدقيقة.

التحكم Control

تسمح لك فئة لبنات **التحكم** (Control) بإضافة أحداث وتحديد التكرارات البرمجية لتكرار الإجراءات واستخدام العبارات الشرطية لاتخاذ القرارات.

الإدخال Input

تسمح لك فئة لبِنات **الإدخال** (Input) بقراءة البيانات من وحدة التحكم الدقيقة.

العمليات الرياضية Math

تسمح لك فئة اللبِنات **الرياضية** (Math) باستخدام الرموز والعمليات الرياضية.

التعليقات Notation

تسمح لك فئة لبِنات **التعليقات** (Notation) باستخدام التعليقات على التعليمات البرمجية الخاصة بك.

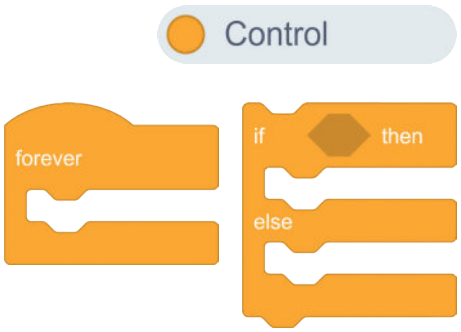
المتغيرات Variables

تسمح لك فئة لبِنات **المتغيرات** (Variables) بإنشاء متغيرات.

ملاحظة

تُعدُّ لبِنات التعليمات البرمجية الرسومية في تينكركاد مفيدة في إنشاء برامج الأردوينو، كما تساعد في تجنب الأخطاء الشائعة مثل أخطاء تراكيب الجُمل وأخطاء كتابة أسماء الدوال، ونسيان الفاصلة المنقوطة (؛) وغيرها من الأخطاء.

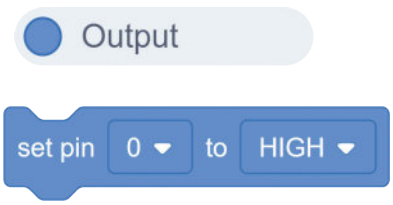
بعض اللبِنات المهمة المُستخدمة في هذا المشروع:



يمكنك العثور على لبِنات forever و if ( ) then else في فئة لبِنات **التحكم** (Control).

يُعدّ عمل لبِنة forever ضروريًا في كل برنامج بصفتها اللبِنة التي تضاف إليها بقية الأوامر. ستُشغّل جميع اللبِنات الأخرى الموجودة بداخلها إلى الأبد وعلى التوالي، وذلك حتى يتم إيقاف تشغيل جهاز إنترنت الأشياء.

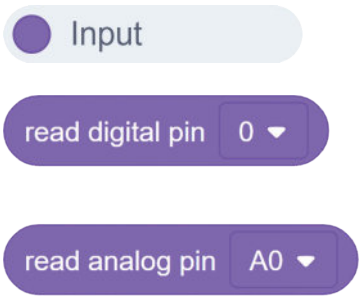
تتحقق لبِنة if ( ) then else مما إذا كانت معايير الشرط مستوفاة. إذا كان الأمر كذلك، فإن جهاز التحكم الدقيق يُنفّذ اللبِنات بين if ( ) then و else، وبخلاف ذلك تُنفّذ اللبِنات أدنى else.



يمكن العثور على لبِنة set pin ( ) to ( ) في تصنيف لبِنات **الإخراج** (Output).

يمكن للبِنة set pin ( ) to ( ) تعيين الحالة **للطرف** (pin) الرقمي أو التناظري إلى قيمة **مُرتفعة** (HIGH) أو **مُنخفضة** (LOW).

يمكن أن تتراوح قيم المنافذ التناظرية بمجموعة من قيم الجهد من 0 فولت إلى 3.3 فولت أو 5 فولت. يعني هذا أنه عند ترجمتها بواسطة البرنامج فإنّ قيم الجهد هذه تتوافق مع مجموعة كبيرة ومتنوعة من القيم.



يمكن العثور على لبِنة read digital pin ( ) ولبِنة read analog pin ( ) في تصنيف لبنات **الإدخال** (Input).

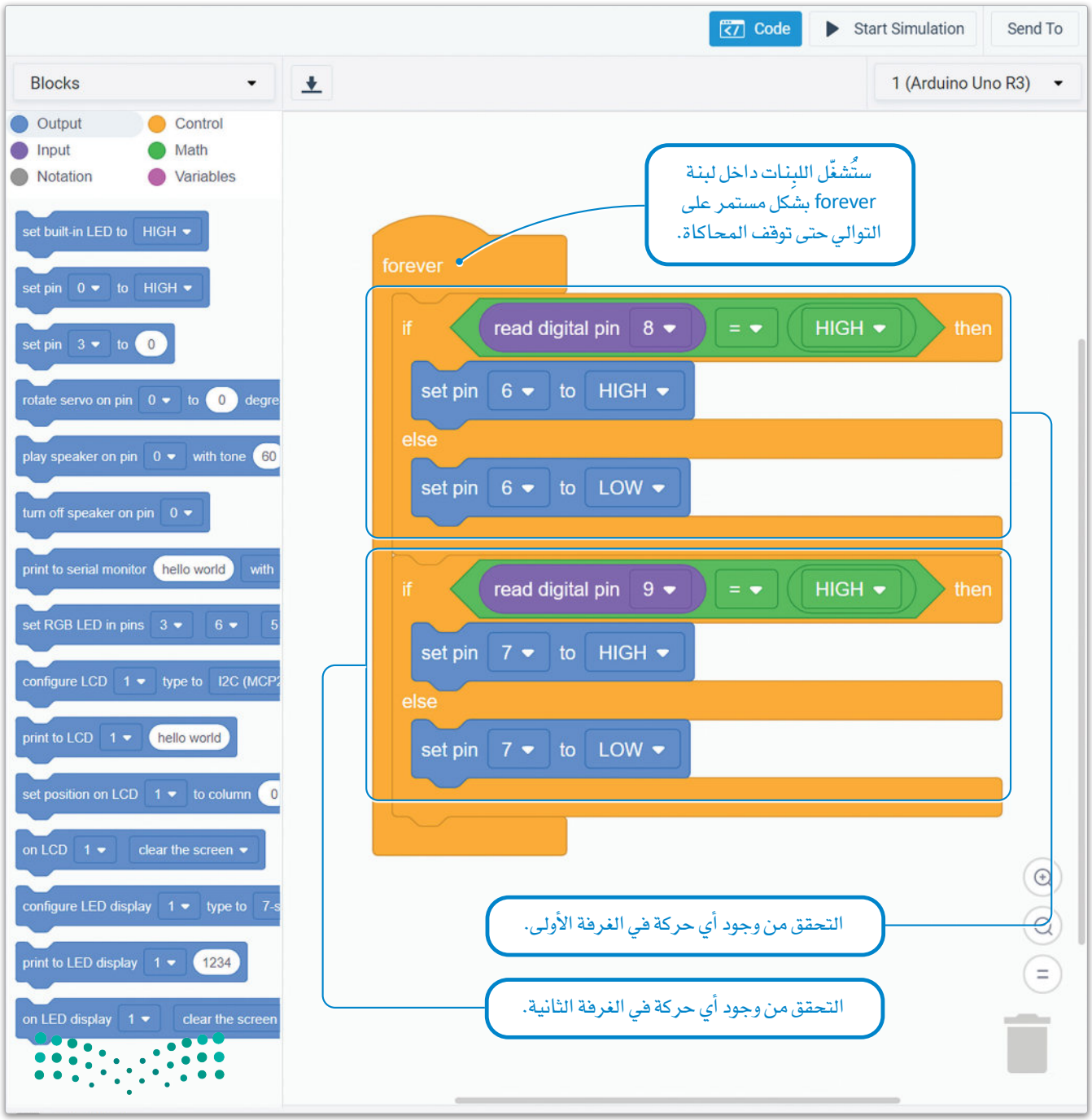
لبِنة read digital pin ( ) هي لبِنة منطقية تقرأ حالة الطرف الرقمي للجهاز (**مُرتفع** (HIGH) أو **مُنخفض** (LOW)).

يُمكن للبِنة read digital pin ( ) قراءة مستوى جُهد يتراوح بين جُهد اللوحة 3.3 فولت أو 5 فولت وصولًا إلى 0 فولت أي **الأرضي** (GND).

للأطراف الرقمية حالتان: مُرتفع (HIGH) عند مرور جُهدٍ خلالها مقداره (3.3 فولت أو 5 فولت)، ومنخفض (LOW) عندما لا يمر أي جهد (0 فولت). تُفسّر هاتان الحالتان بواسطة البرنامج على أنهما 1 (صواب) True و 0 (خطأ) False على التوالي.

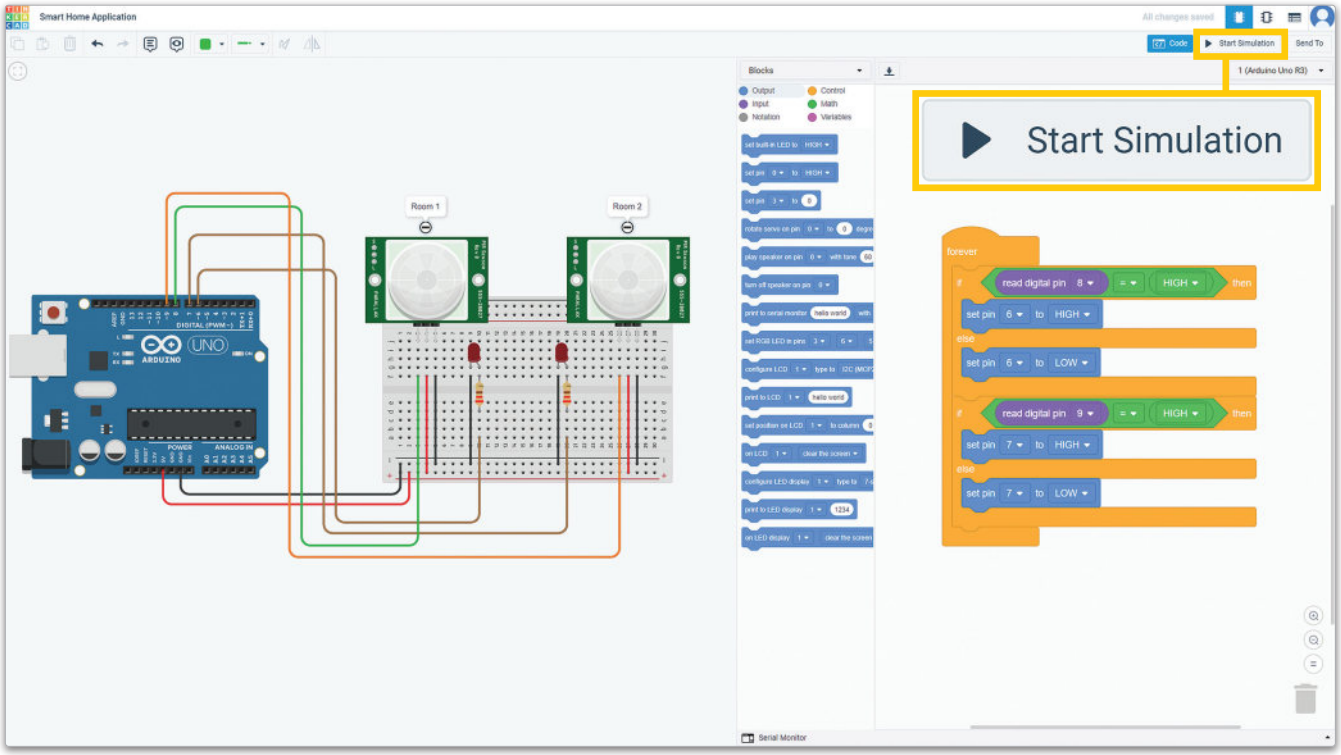
التعليمات البرمجية Let's Take a Look at the Code

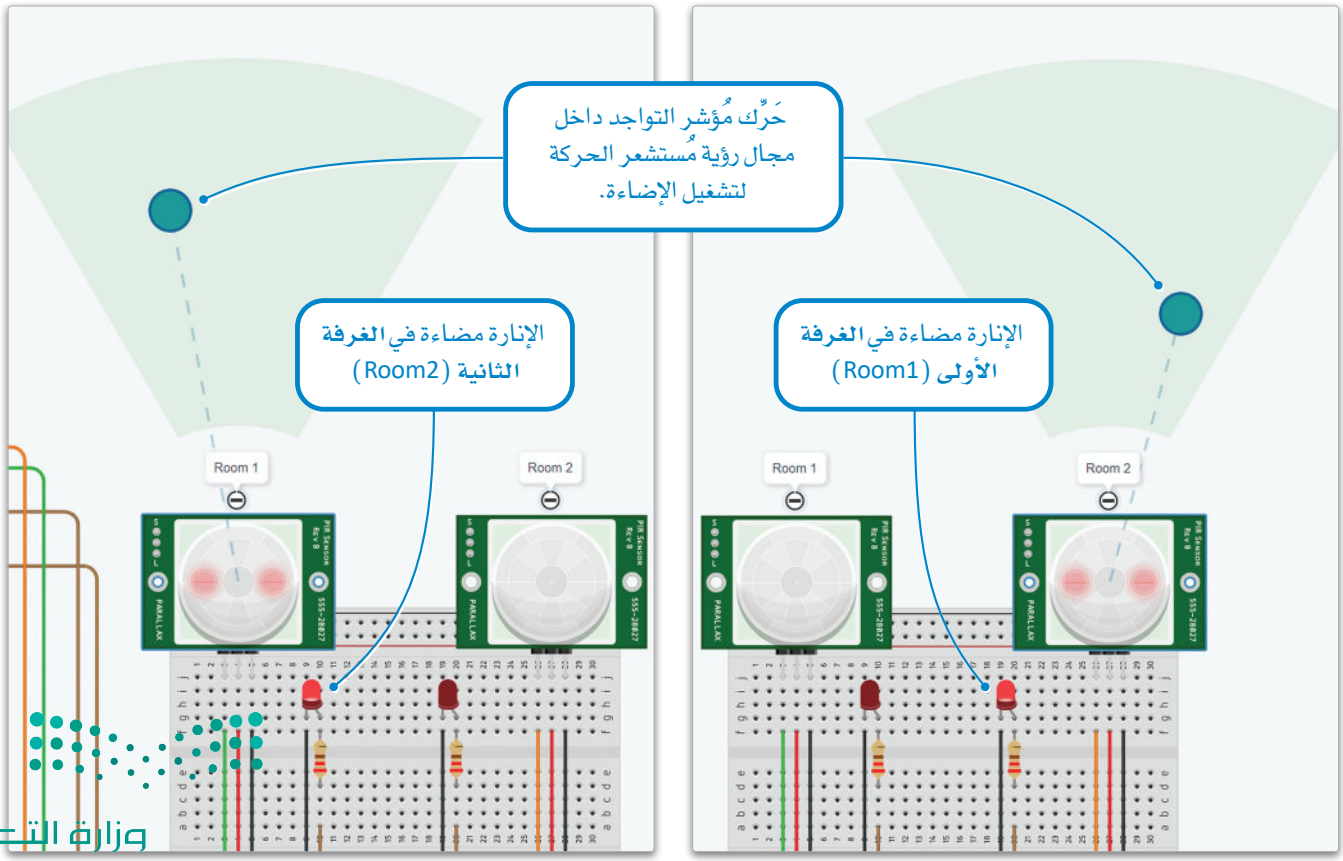
سيُنفّذ برنامجك بلا توقف forever، وسيتم تنفيذ لبنتي if /else، إحداها **للغرفة الأولى** (Room1) والأخرى **للغرفة الثانية** (Room2). سوف تتحقق مُستشعرات الحركة من وجود أي حركة داخل الغُرفتين. إذا اكتشف مُستشعر الحركة أي تحرك، فسيشتغل الدايود المشع للضوء الخاص به، وعند توقفه عن استشعار الحركة، سيتوقف الدايود المشع للضوء عن الإضاءة. تُحاكي هذه الدائرة تطبيق **المنزل الذكي** (Smart Home) الذي يقوم بإنارة الغرفة تلقائيًا بمجرد دخول أي شخص إلى الغرفة.



شكل 3.17: تنفيذ البرنامج

ابدأ المحاكاة بالضغط على زر بِدء **المحاكاة** (Start Simulation) لاختبار برنامجك.

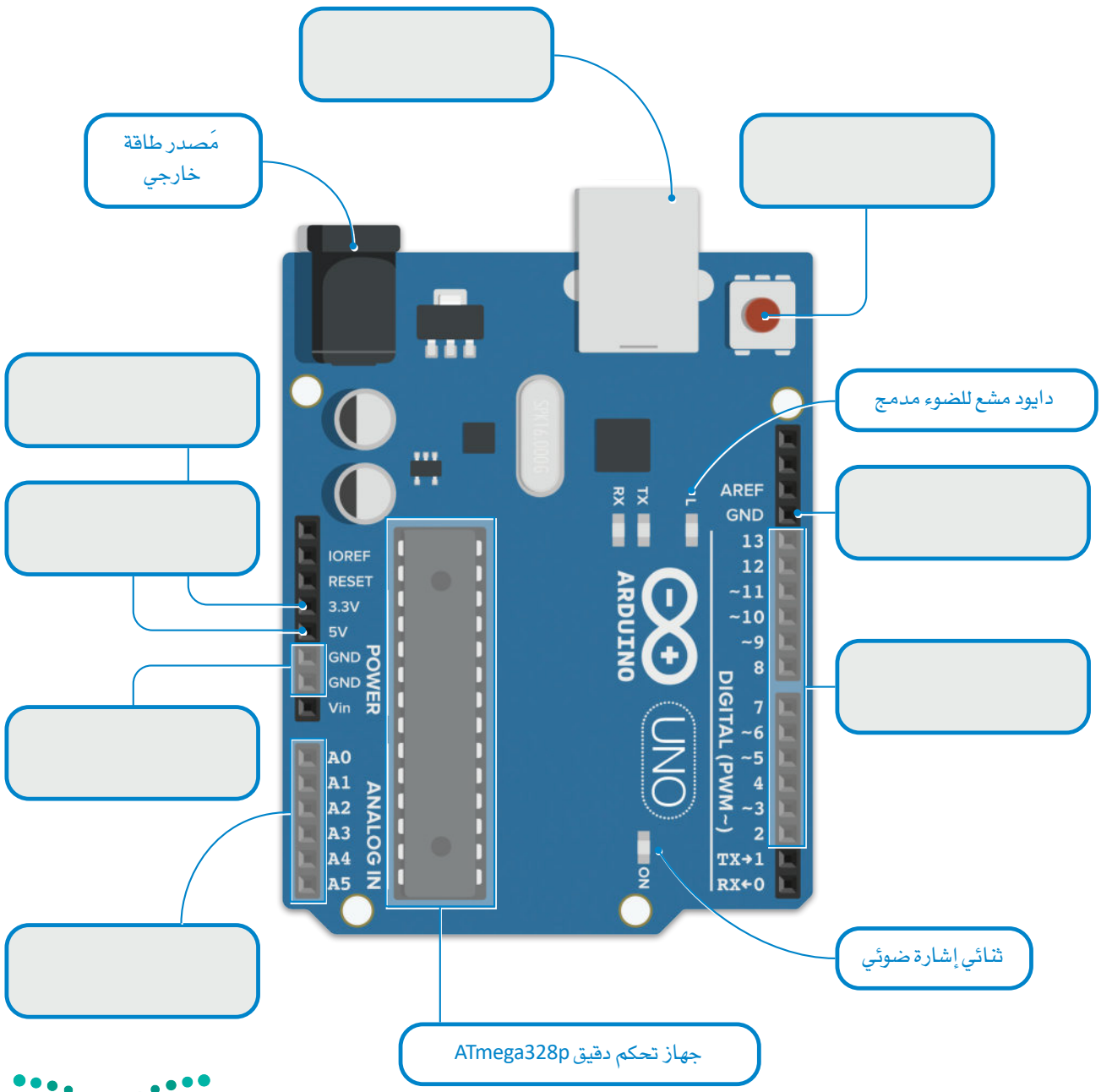




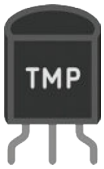
شكل 3.18: محاكاة البرنامج

تمرينات

1 املأ المربعات الفارغة بأسماء المكونات الناقصة.



2 صِل العناصر في السطر الأول بمسمياتها في السطر الثاني.

المُكون في المحاكي			
الاسم	مُستشعر درجة الحرارة	مُستشعر الغاز	مُستشعر رطوبة التربة

اكتب الحالات التي يمكنك فيها استخدام مُستشعر:

1. درجة الحرارة

2. الغاز

3. رطوبة التربة

3 ما الفرق بين المداخل الرقمية والتناظرية؟

4 ابحث في الإنترنت عن الاختلافات الرئيسة بين مجموعة الأردوينو أونو (Arduino Uno) ومجموعة الأردوينو نانو (Arduino Nano). ما أنواع التطبيقات التي تُستخدم في كل مجموعة؟ اكتب إجابتك أدناه.

5 ما ميزة برمجة جهاز تحكم الأردوينو الدقيق باللبنات البرمجية بدلًا من كتابة التعليمات البرمجية نصيًا في تينكركاد؟ اذكر ما تتوقعه من سلبيات استخدام هذه الطريقة؟

6 قم بتوسعة نظام المنزل الذكي لمراقبة ثلاث غُرفٍ بدلًا من اثنتين.

7 غيّر نظام المنزل الذكي ليحتوي على دايودين مشعين للضوء (أحمر وأخضر) ومُستشعر الحركة. عند اكتشاف المستشعر لحركة يشتغل الدايود المشع للضوء الأخضر فقط، وعند عدم اكتشاف حركة، يشتغل الدايود المشع للضوء الأحمر فقط.

رابط الدرس الرقمي

www.ien.edu.sa

الدرس الثاني

إنشاء نظام لري النباتات

إنشاء نظام لري النباتات Build a Plant Watering System

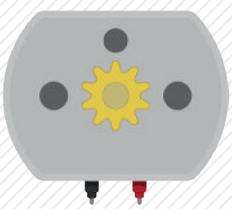
أدَّت الزيادة الكبيرة في عدد السكان خلال القرن الماضي إلى ظهور الحاجة الماسة لتوفير المزيد من المحاصيل لتأمين الغذاء والمنتجات الأساسية الأخرى. وتطلّب هذا الأمر توسّعًا كبيرًا في زراعة المحاصيل والنباتات لتلبية حاجات البشر. إلا أنّ عمليّة التوسّع هذه واجهتْ صعوبة الاعتناء بهذه الأراضي الزراعية بشكلٍ يدوي.

طُوّرت الأنظمة الآلية للري لزيادة فاعليّته وتحسين كفاءة الإنتاج الزراعي بصورة كبيرة. ستستخدم في هذا الدرس لوحة **الأردوينو أونو R3** (Arduino UNO R3) لمحاكاة نظام آلي لري النبات. وسيستخدم النظام مُحركًا لتشغيل نظام الري عند اكتشاف المُستشعرات انخفاض رطوبة التربة وارتفاع درجة الحرارة.

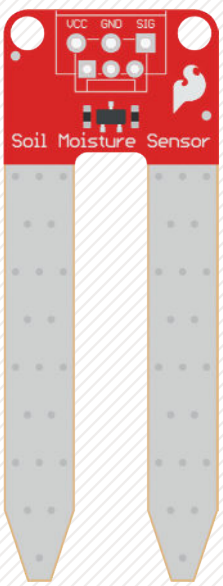
ستحتاج في هذا المشروع إلى المكونات الآتية:

- مُحرك تيار مستمر (DC motor).
- مُستشعر درجة الحرارة (TMP36).
- مُستشعر رطوبة التربة.
- لوحة أردوينو اونو **R3** (Arduino UNO R3).

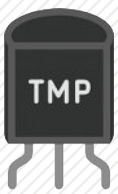
المُكونات التي ستستخدمها في هذا المشروع:



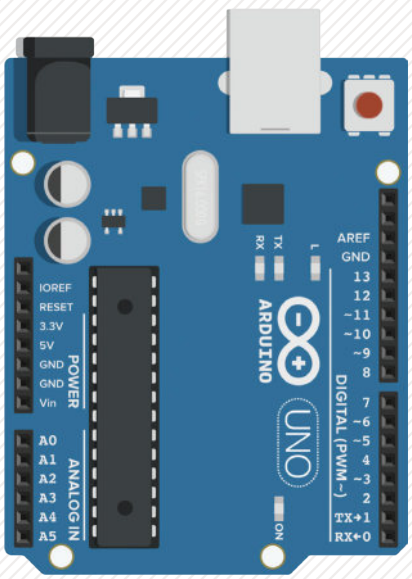
مُحرك تيار مستمر (DC motor).



مُستشعر رطوبة التربة.



مُستشعر درجة الحرارة (TMP36)



لوحة أردوينو اونو R3 (Arduino UNO R3).

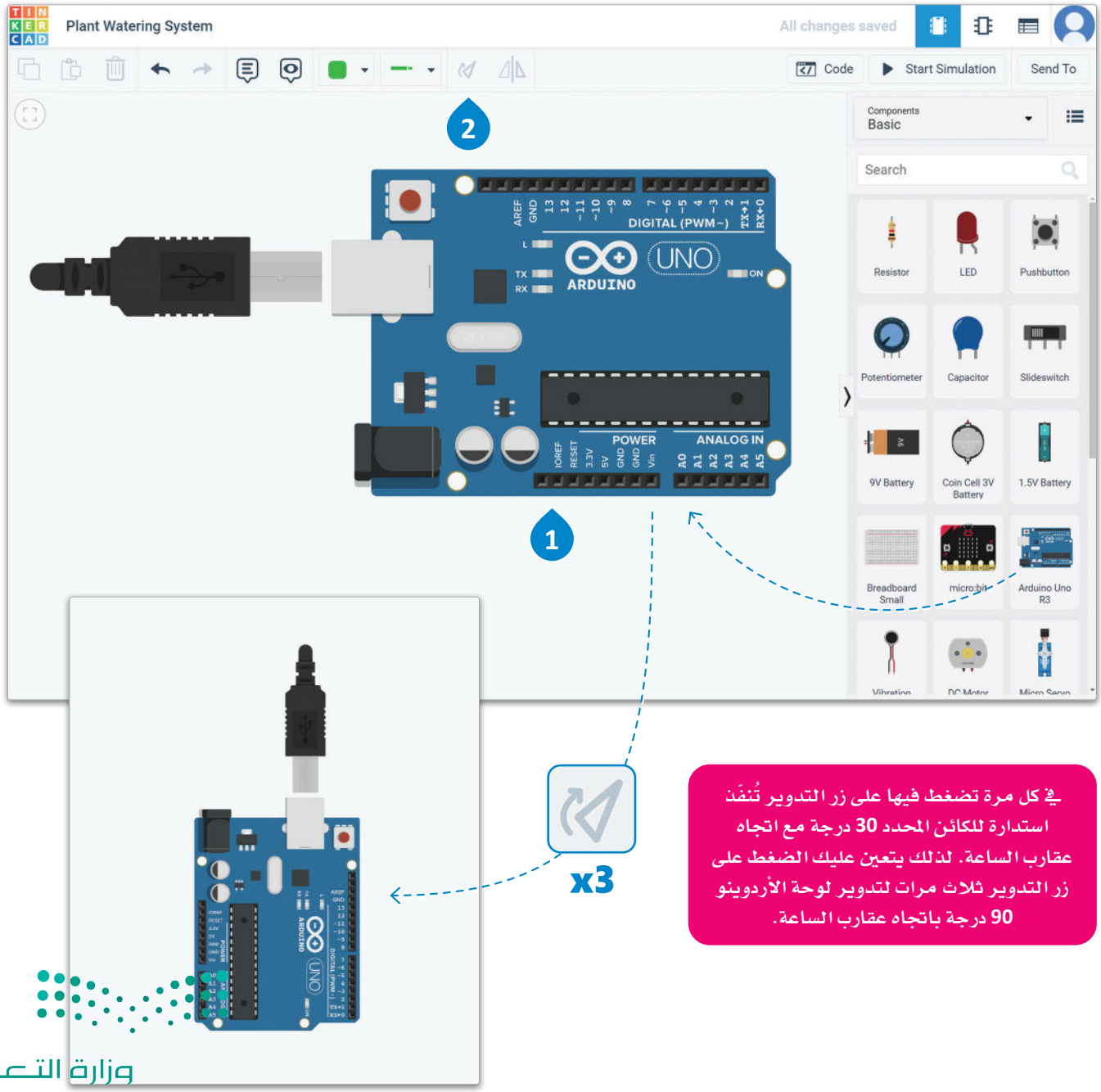
شكل 3.19: مكونات المشروع

ابـدأ بإضافـة المكونـات التـي سـتحتاج إليهـا لهـذا المشـروع في مسـاحة العمـل. أولًا، ابحـث عـن لوحـة **الأردوينـو أونـو R3** (Arduino UNO R3) وأضفها إلى مساحة العمل، وقم بتدويرها 90 درجة.

لإضافة جهاز تحكم الاردوينو الدقيق:

> اسحب وأفلت Arduino UNO R3 (لوحة أردوينو أونو R3) من مكتبة Components (المكونات) في مساحة العمل. 1

> اضغط على زر التدوير ثلاث مرات. 2



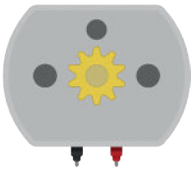
في كل مرة تضغط فيها على زر التدوير تُنفّذ استدارة للكائن المحدد 30 درجة مع اتجاه عقارب الساعة. لذلك يتعين عليك الضغط على زر التدوير ثلاث مرات لتدوير لوحة الأردوينو 90 درجة باتجاه عقارب الساعة.

شكل 3.20: إضافة جهاز تحكم الاردوينو الدقيق

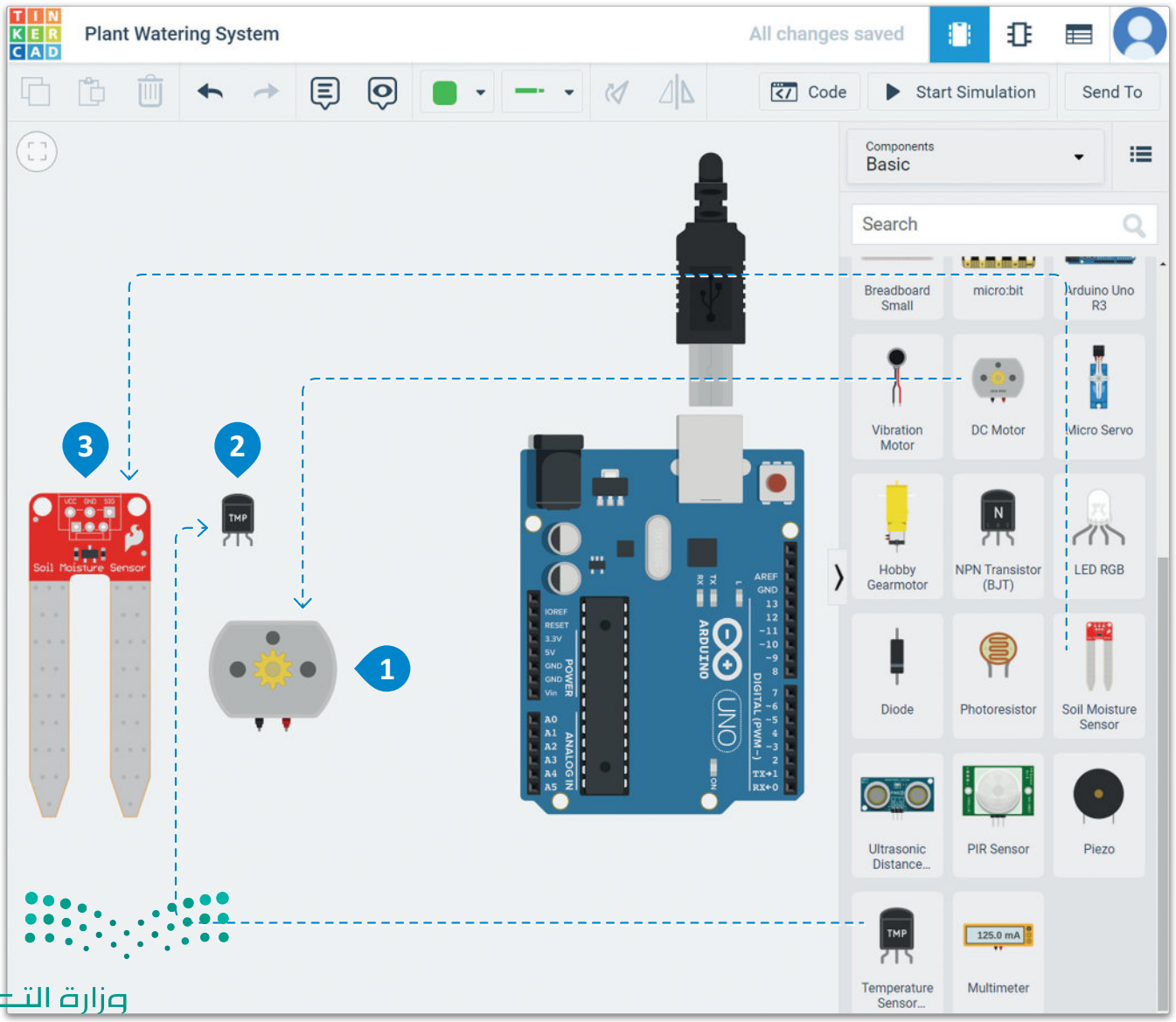
ابحث بعد ذلك عن بقية المكونات، وهي مُحرك تيار مستمر (DC motor)، ومُستشعر درجة الحرارة (TMP36)، ومُستشعر رطوبة التربة (Soil Moisture Sensor)، ولوحة أردوينو أونو R3 (Arduino UNO R3). أضف هذه المكونات إلى مساحة العمل كما يلي:

لإضافة المكونات إلى مساحة العمل:

- ابحث عن DC motor (مُحرك تيار مستمر) من مكتبة Components (المكونات) واسحبه وأفلته في مساحة العمل. 1
- ابحث عن TMP36 Temperature Sensor (مُستشعر درجة الحرارة TMP36) من مكتبة Components (المكونات) واسحبه وأفلته في مساحة العمل. 2
- ابحث عن Soil Moisture Sensor (مُستشعر رطوبة التربة) من مكتبة Components (المكونات) واسحبه وأفلته في مساحة العمل. 3



يُستخدم في هذا المشروع محرك تيار مستمر (DC motor) كوسيلة لتشغيل مضخة المياه التي تزود النباتات بالمياه.



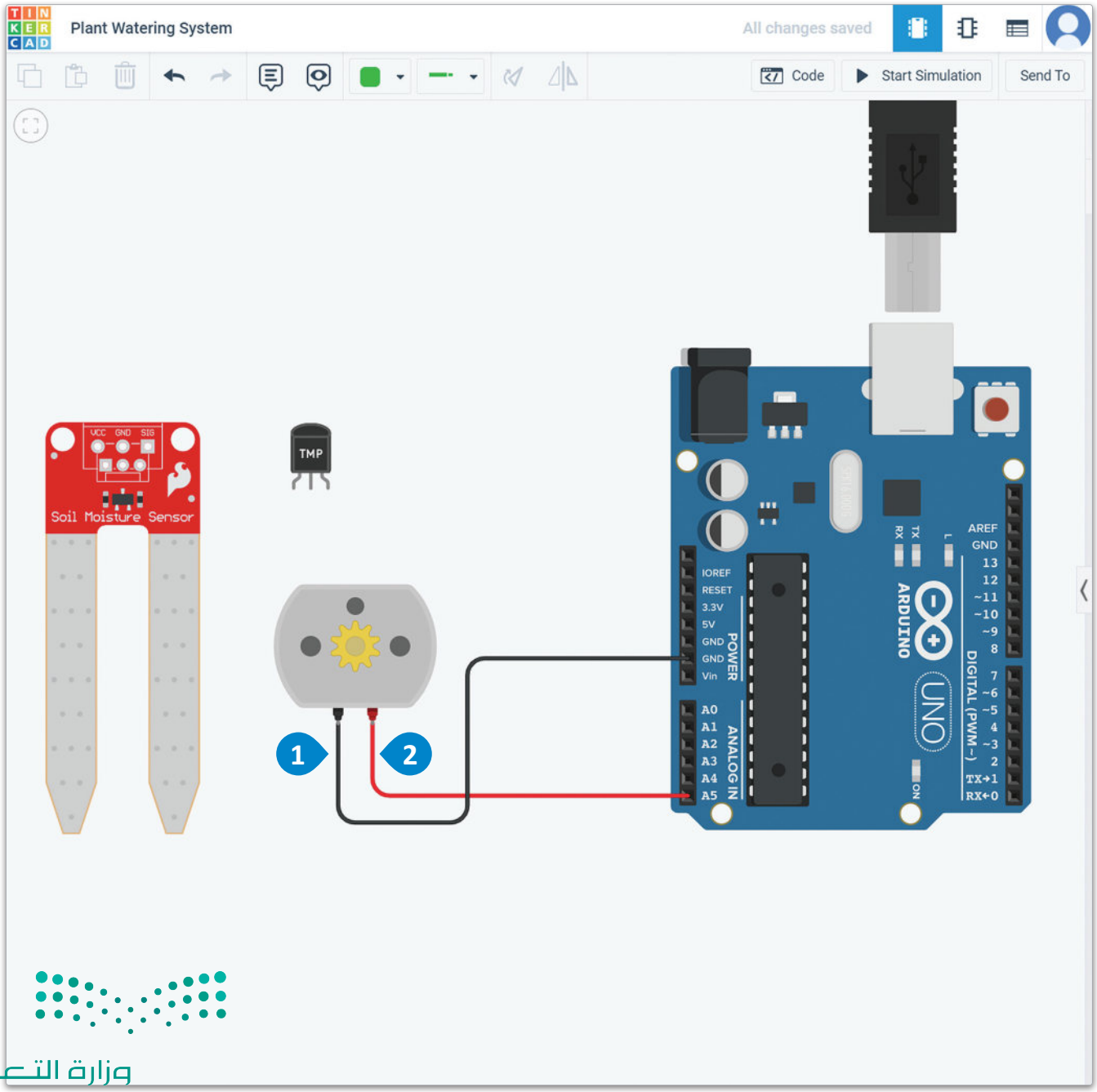
شكل 3.21: إضافة المكونات إلى مساحة العمل

الآن، قُم بتوصيل **محرك تيار مستمر** (DC motor) إلى لوحة **أردوينو أونو** R3 (Arduino UNO R3).

لتوصيل محرك التيار المستمر (DC motor):

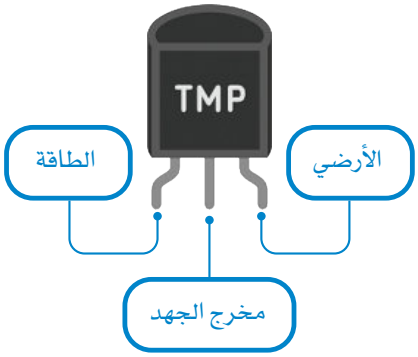
> قُم بتوصيل الطرف 1 من محرك التيار المستمر بـ GND (الطرف الأرضي) للوحة الأردوينو أونو R3 وغيّر لون السلك إلى اللون black (الأسود). 1

> قُم بتوصيل الطرف 2 من محرك التيار المستمر إلى الطرف التناظري A5 للوحة الأردوينو أونو R3 وغيّر لون السلك إلى اللون red (الأحمر). 2



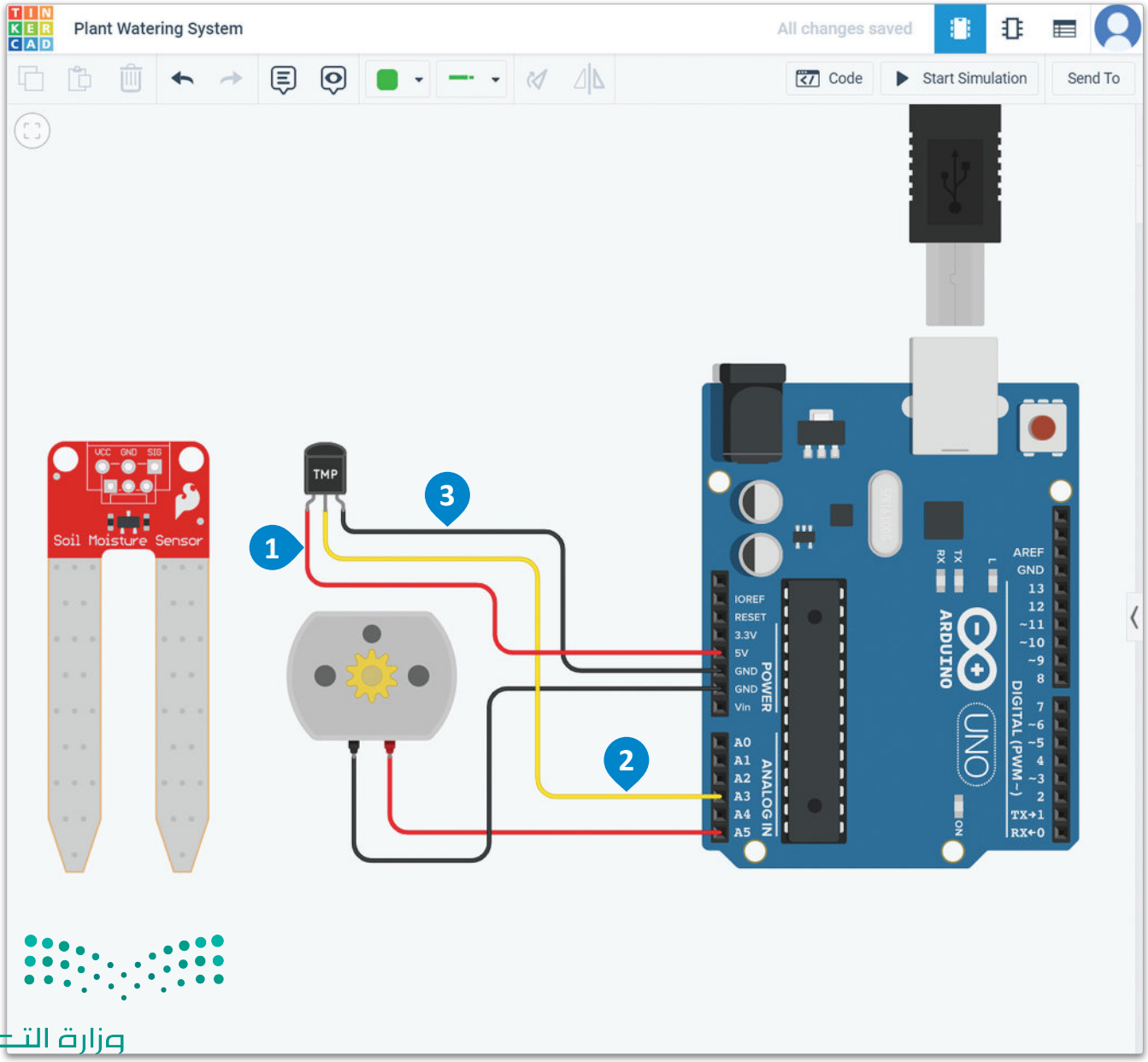
شكل 3.22: توصيل محرك التيار المستمر

بعد ذلك قُم بتوصيل مُستشعر درجة الحرارة بلوحة **الأردوينو أونو** R3 (Arduino UNO R3).



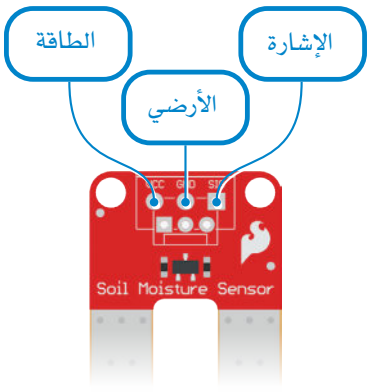
لتوصيل مُستشعر درجة الحرارة:

- قُم بتوصيل طرف Power (الطاقة) الخاص بمُستشعر درجة الحرارة بالطرف 5V (جهد 5 فولت) من لوحة الأردوينو، وغيِّر لون السلك إلى red (الأحمر). 1
- قُم بتوصيل طرف مخرج Vout (الجهد) الخاص بمُستشعر درجة الحرارة بالطرف التناظري A3 للوحة الأردوينو وغيّر لون السلك إلى اللون yellow (الأصفر). 2
- قُم بتوصيل GND (الطرف الأرضي) الخاص بمُستشعر درجة الحرارة بـ GND (الطرف الأرضي) للوحة الأردوينو أونو وغيّر لون السلك إلى black (الأسود). 3



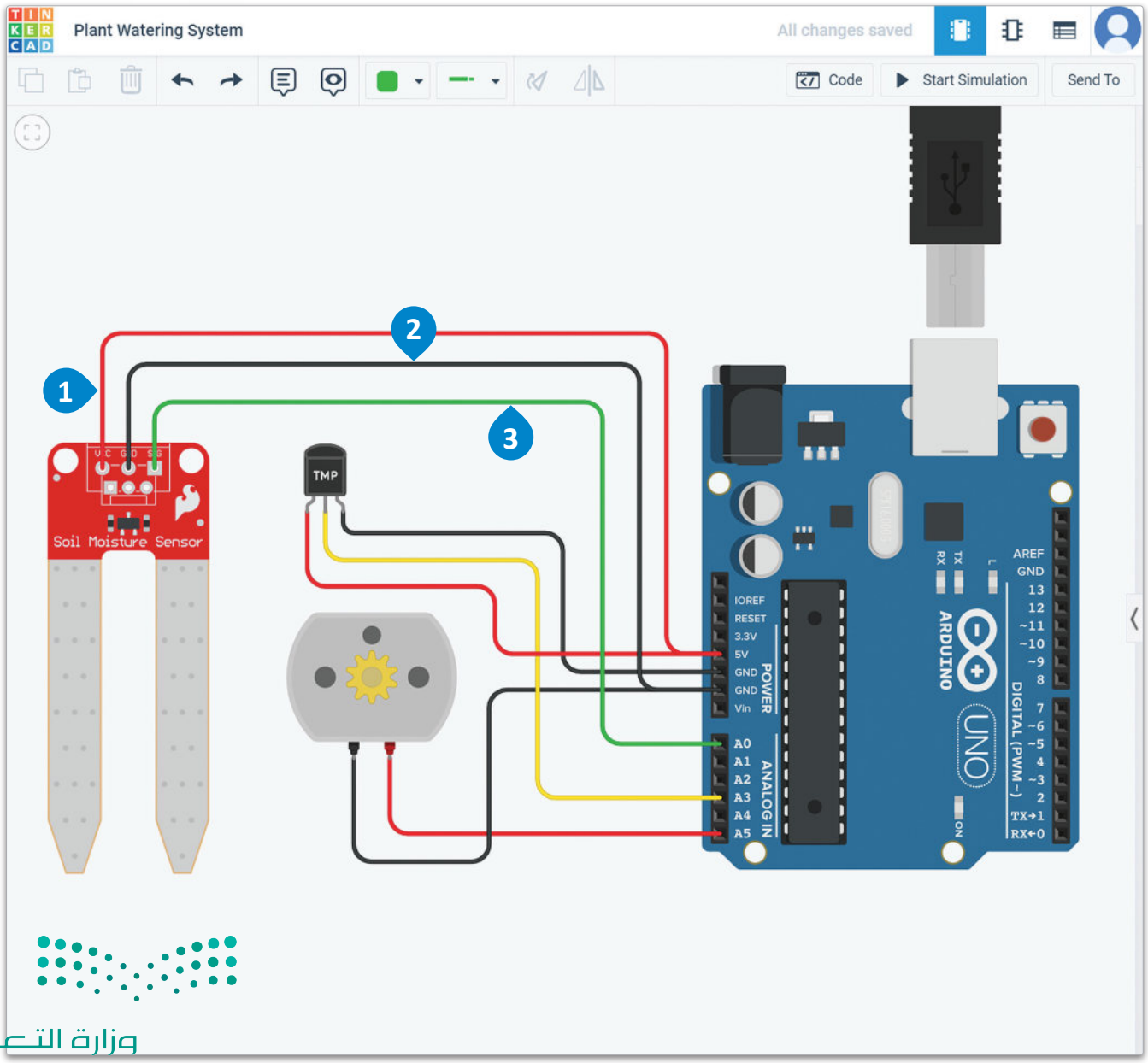
شكل 3.23: توصيل مُستشعر درجة الحرارة

ثم أكمل العملية بتوصيل مُستشعر رطوبة التربة بلوحة **الأردوينو أونو** R3 (Arduino UNO R3).



لتوصيل مُستشعر رطوبة التربة:

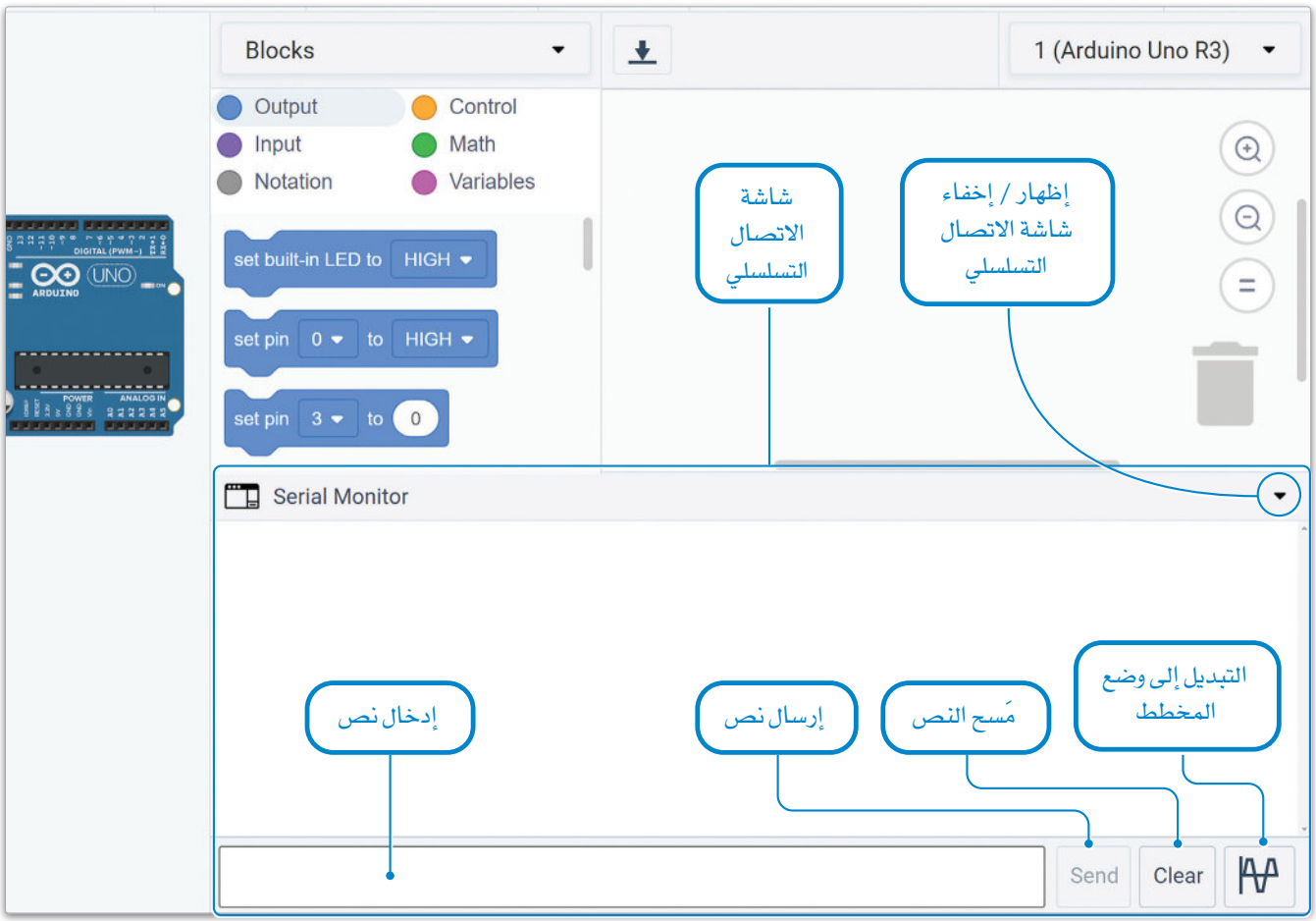
- قُم بتوصيل طرف Power (الطاقة) الخاص بمُستشعر رطوبة التربة بالطرف 5V (جهد 5 فولت) من لوحة الأردوينو، وغيّر لون السلك إلى red (الأحمر). 1
- قُم بتوصيل GND (الطرف الأرضي) الخاص بمُستشعر رطوبة التربة بـ GND (الطرف الأرضي) للوحة الأردوينو أونو R3 وغيّر لون السلك إلى black (الأسود). 2
- قُم بتوصيل طرف Signal (الإشارة) الخاص بمُستشعر رطوبة التربة بالطرف A0 للوحة الأردوينو وغيّر لون السلك إلى اللون green (الأخضر). 3



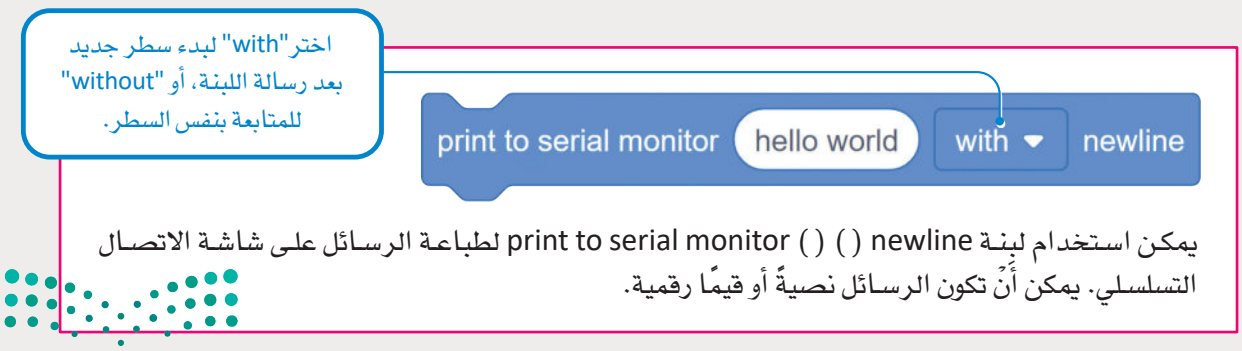
شكل 3.24: توصيل مُستشعر رطوبة التربة

شاشة الاتصال التسلسلي Serial Monitor

تُستخدم شاشة الاتصال التسلسلي كأداة مراقبة للقيم التي يتم تلقيها من المُستشعرات. تُسهّل هذه الأداة عملية تحديد مشكلات الدوائر، وكذلك حل المشاكل البرمجية من خلال سماحها للمستخدم بمعاينة قيم الإخراج الناتجة من المُستشعرات والبرنامج. يمكن استخدامها أيضًا لطباعة رسالة خاصة للمستخدم، أو لعرض معلومات وإرشادات مفيدة. يمكن الوصول إلى **شاشة الاتصال التسلسلي** (Serial Monitor) أسفل لوحة البرمجة.



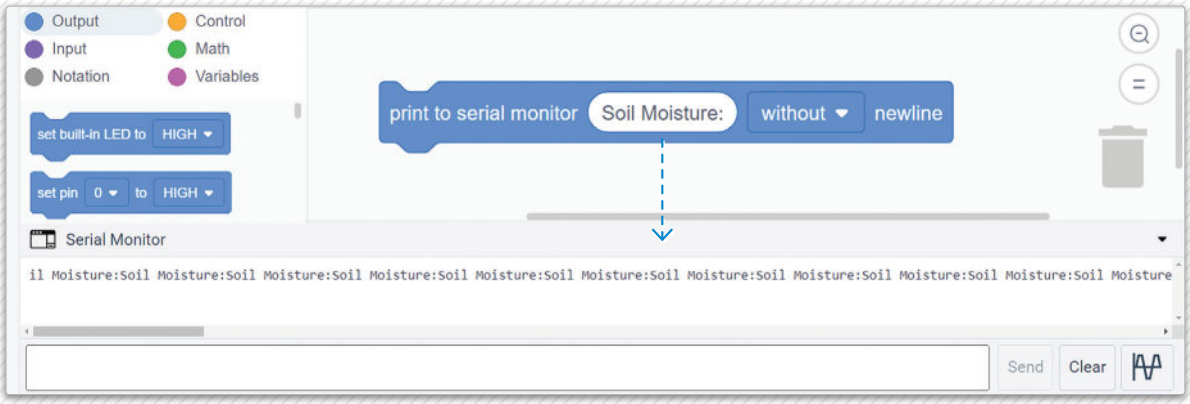
شكل 3.25: شاشة الاتصال التسلسلي



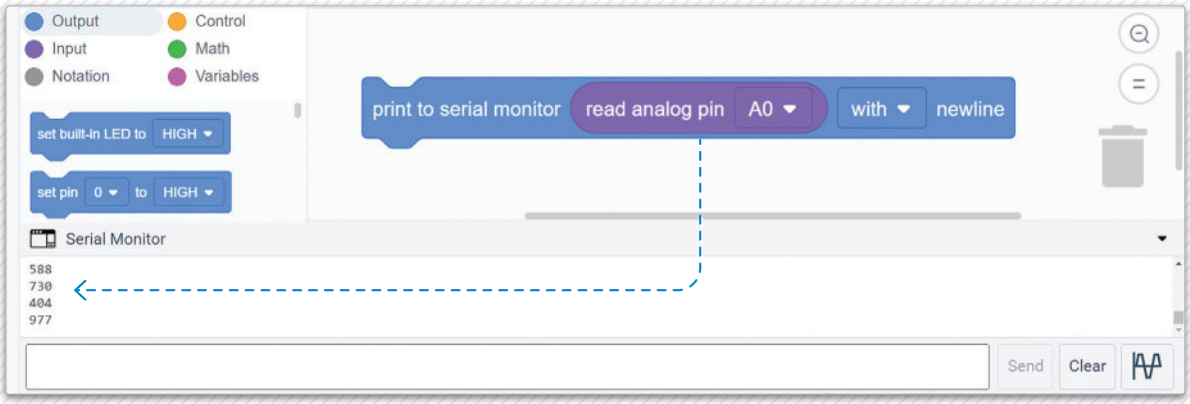
يمكن استخدام لبنة print to serial monitor ( ) ( ) newline لطباعة الرسائل على شاشة الاتصال التسلسلي. يمكن أن تكون الرسائل نصيةً أو قيمًا رقمية.

مثال

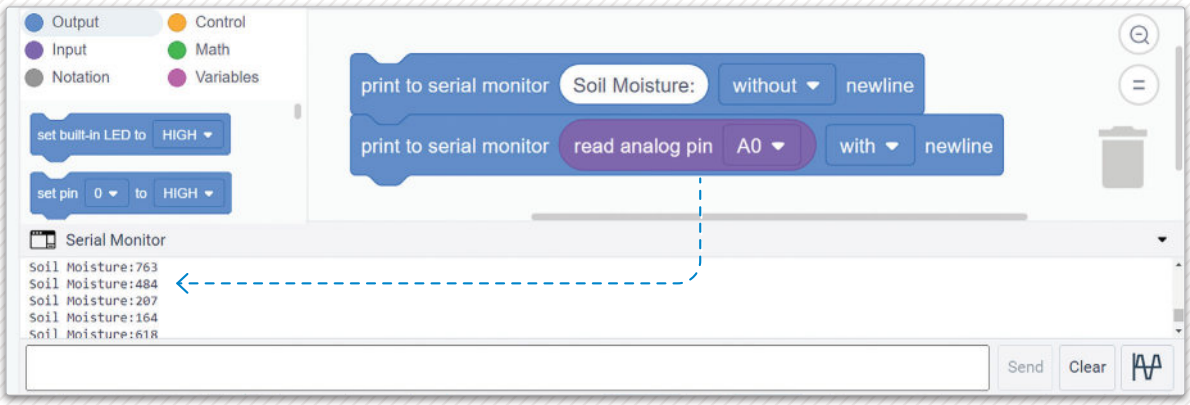
على سبيل المثال، إذا أردت طباعة النص "Soil Moisture" دون سطر جديد، استخدم اللبنة أدناه:



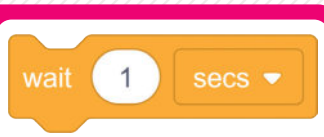
وإذا كنت تريد طباعة قيمة الطرف التناظري A0 في سطر جديد، استخدم اللبنة أدناه:



إذا دمجت هاتين اللبنتين، ستكون مخرجات شاشة الاتصال التسلسلي أكثر تنظيمًا.

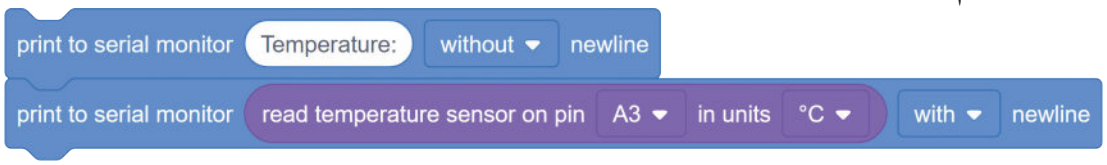


تعمل لبِنات () () **wait** على إضافة تأخير زمني على تنفيذ البرنامج ومُخرجات طباعته، مما يوفر إخراجًا أوضح للمستخدم. ويمكن لهذه اللبنات إضافة تأخير للبرنامج لعددٍ مُحدد من الثواني أو أجزاء من الثانية (مللي ثانية).



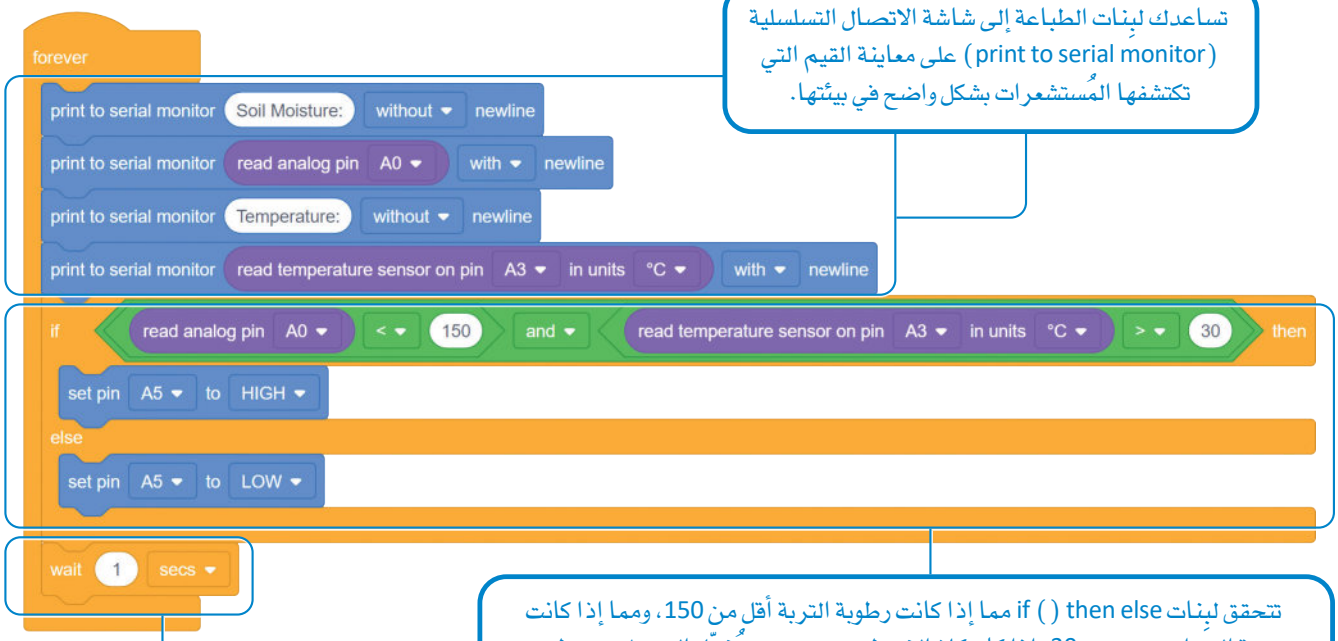
شكل 3.26: استخدام شاشة الاتصال التسلسلي

يبعث مُستشعر درجة الحرارة إشارات تناظرية، ولكنه يستخدم لبنات خاصة به ليتمكن من تحديد وحدة درجة الحرارة المطلوبة (مئوية أو فهرنهايت). لقراءة درجة الحرارة بالدرجة المئوية من الطرف التناظري A3، استخدم اللبنات أدناه:



تنفيذ التعليمات البرمجية Code Implementation

يقوم البرنامج أولًا بطباعة قِيم رطوبة التربة ودرجة الحرارة على شاشة الاتصال التسلسلي، ثم يتحقق مِما إذا كان كلا الشرطين صحيحين، فإذا كان الأمر كذلك، يقوم بتشغيل المحرك، وإلا فإنه يوقف تشغيله. في النهاية، ينتظر البرنامج ثانية واحدة لتجنب امتلاء الشاشة التسلسلية بالمخرجات في نفس الوقت.



تساعدك لبِنات الطباعة إلى شاشة الاتصال التسلسلية (print to serial monitor) على معاينة القيم التي تكتشفها المُستشعرات بشكل واضح في بيئتها.

تتحقق لبِنات if ( ) then else مما إذا كانت رطوبة التربة أقل من 150، ومما إذا كانت درجة الحرارة تزيد عن 30. إذا كان كلا الشرطين متحققين، يُشغّل المحرك عن طريق ضبط طرفه بقيمة HIGH، وإلا فإنه يوقف تشغيله عن طريق ضبط طرفه بقيمة LOW.

تساعد لبِنات wait ( ) في تخفيف ازدحام شاشة الاتصال التسلسلية بالمُخرجات وذلك بإيقاف التنفيذ مؤقتًا.

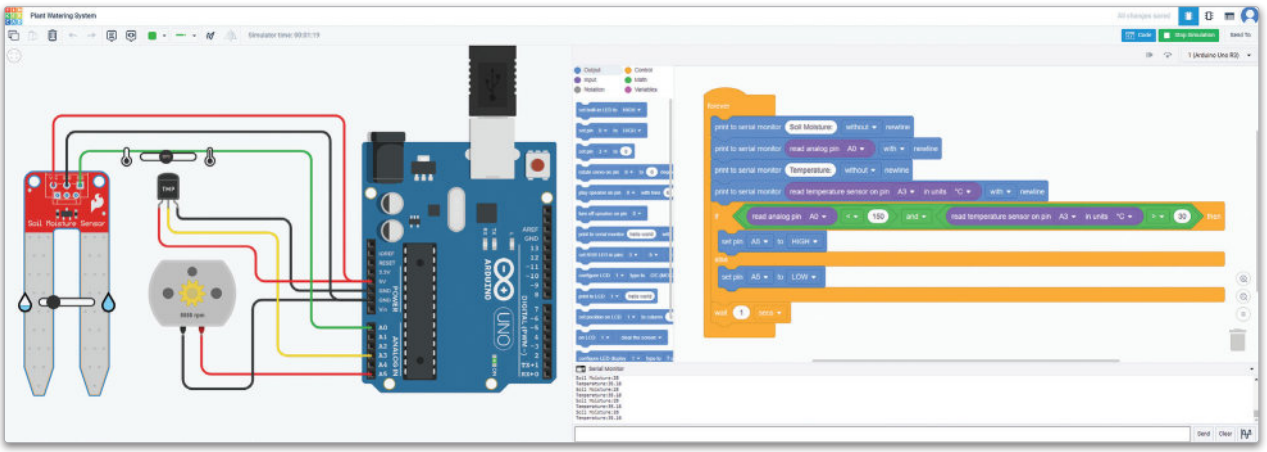
قد يكون الشرط المراد التحقق منه أحيانًا داخل لبنة if ( ) then else أكثر تعقيدًا من مُجرد مقارنة بسيطة بين القيم.

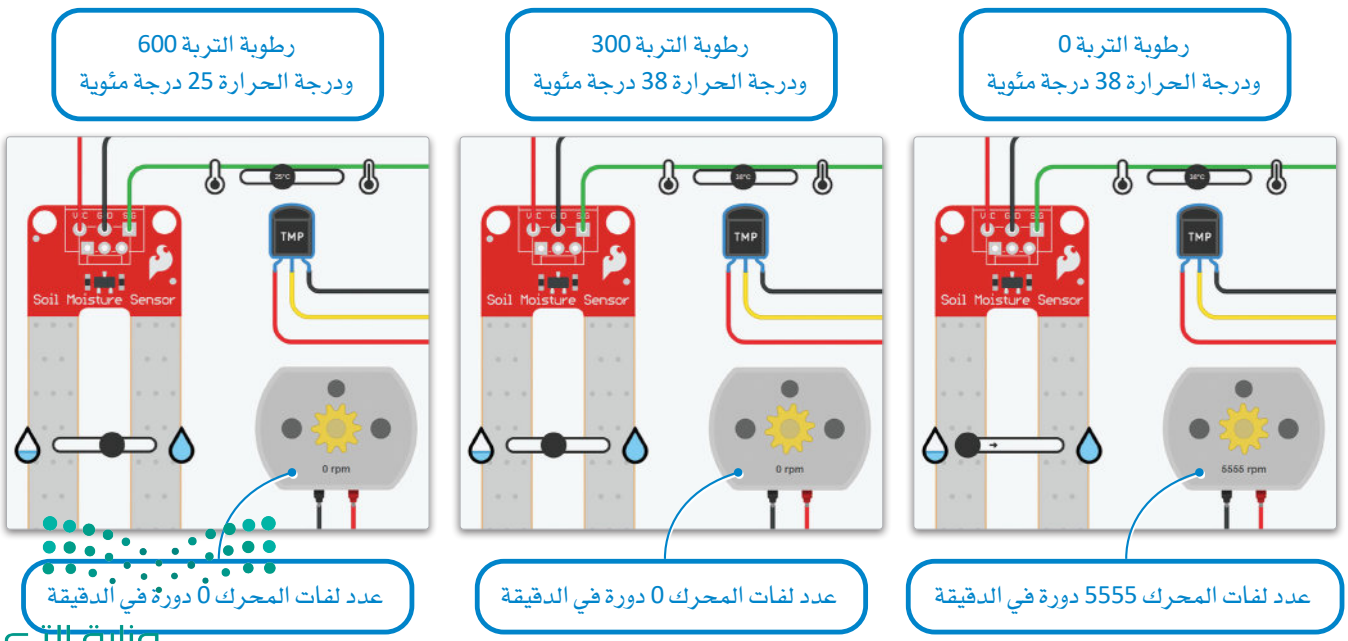
شكل 3.27: تنفيذ البرنامج

الجدول 3.2: قيم مُدخلات النظام وحالات المُخرجات

مستوى رطوبة التربة (الطرف A0)	درجة الحرارة (الطرف A3)	المُخرج إلى المحرك (الطرف A5)
≥ 150	≤ 30	منخفض LOW
≥ 150	> 30	منخفض LOW
< 150	≤ 30	منخفض LOW
< 150	> 30	مرتفع HIGH

الآن وبعد أن انتهيت من إضافة اللبنات البرمجية المناسبة، حان الوقت لتنفيذ البرنامج. ابدأ المحاكاة بالضغط على زر بدء المحاكاة (Start Simulation). يمكنك تجربة الحالات المختلفة للدائرة ومعاينتها عن طريق تمرير شريط قِيم المُستشعر وتحديد قيمته:





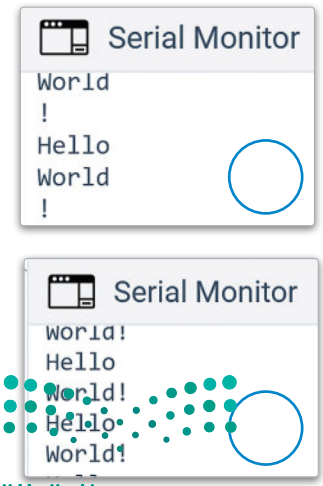
شكل 3.28: سرعة المحرك بقيم رطوبة ودرجات حرارة مختلفة

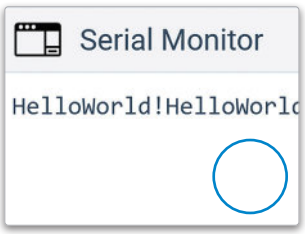
تمرينات

1 إذا أردت إنشاء نظام ري ذكي في منطقة يكون الجوّ فيها أكثر جفافًا، والماءُ أسرع تبخّرًا، فما التغييرات التي يجب أن تقوم بها؟ ضع حلًا ممكنًا واعرض أفكارك أدناه.

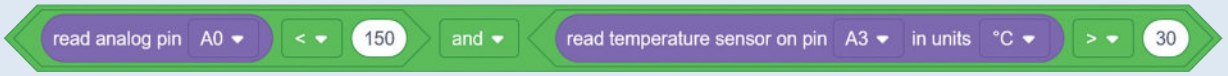
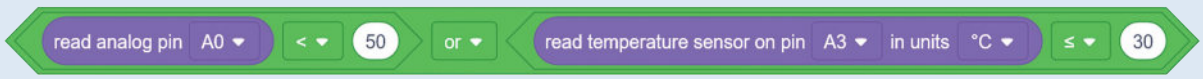
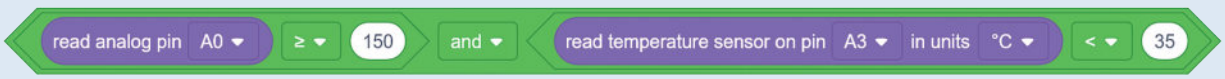
2 لا تتطلب المُستشعرات والمُشغل في مشروع هذا الدرس توصيل لوحة توصيل الدوائر مع الأردوينو على عكس المشاريع الأخرى. وضِّح أسباب ذلك.

3 تحقق من مُخرجات اللبِنات الآتية بوضع إشارة ✔ بجوار المخرج الصحيح:





4 قيّم الشروط الآتية للبِنات التعليمات البرمجية إما بصواب أو خطأ مع الأخذ بالاعتبار قِيم الأطراف التناظرية المُعطاة:

حدد الجملة الصحيحة والجملة الخاطئة فيما يلي:	صحيحة	خاطئة
		
1. مُدخلات مُستشعر رطوبة التربة على الطرف A0 : 180، ومُدخلات مُستشعر درجة الحرارة على الطرف A3:32	○	○
2. مُدخلات مُستشعر رطوبة التربة على الطرف A0 : 167، ومُدخلات مُستشعر درجة الحرارة على الطرف A3:43	○	○
3. مُدخلات مُستشعر رطوبة التربة على الطرف A0 : 255، ومُدخلات مُستشعر درجة الحرارة على الطرف A3:35	○	○
		
4. مُدخلات مُستشعر رطوبة التربة على الطرف A0 : 58، ومُدخلات مُستشعر درجة الحرارة على الطرف A3:41	○	○
		
5. مُدخلات مُستشعر رطوبة التربة على الطرف A0 : 150، ومُدخلات مُستشعر درجة الحرارة على الطرف A3:35	○	○

5 وسّعْ نظام الري الذكي بحيث يتمّ إعلام المستخدم برسالة عند ريّ النبات "Watering Plant Now!" وذلك عندما تكون قيمة الرطوبة أقل من 150، ولا تزيد درجة الحرارة عن 30 درجة مئوية.

لا تنسَ إضافة سطر جديد في الرسالة المعروضة لزيادة وضوح الشاشة.

6 وسّع نظام الري الذكي بحيث يعمل مُحرك التيار المستمر لفترة أطول إذا كانت مستويات رطوبة التربة منخفضة للغاية (أقل من 50). استخدم لبنات الانتظار لجعل مُحرك التيار المستمر يعمل لفترة أطول.

رابط الدرس الرقمي

www.ien.edu.sa

الدرس الثالث

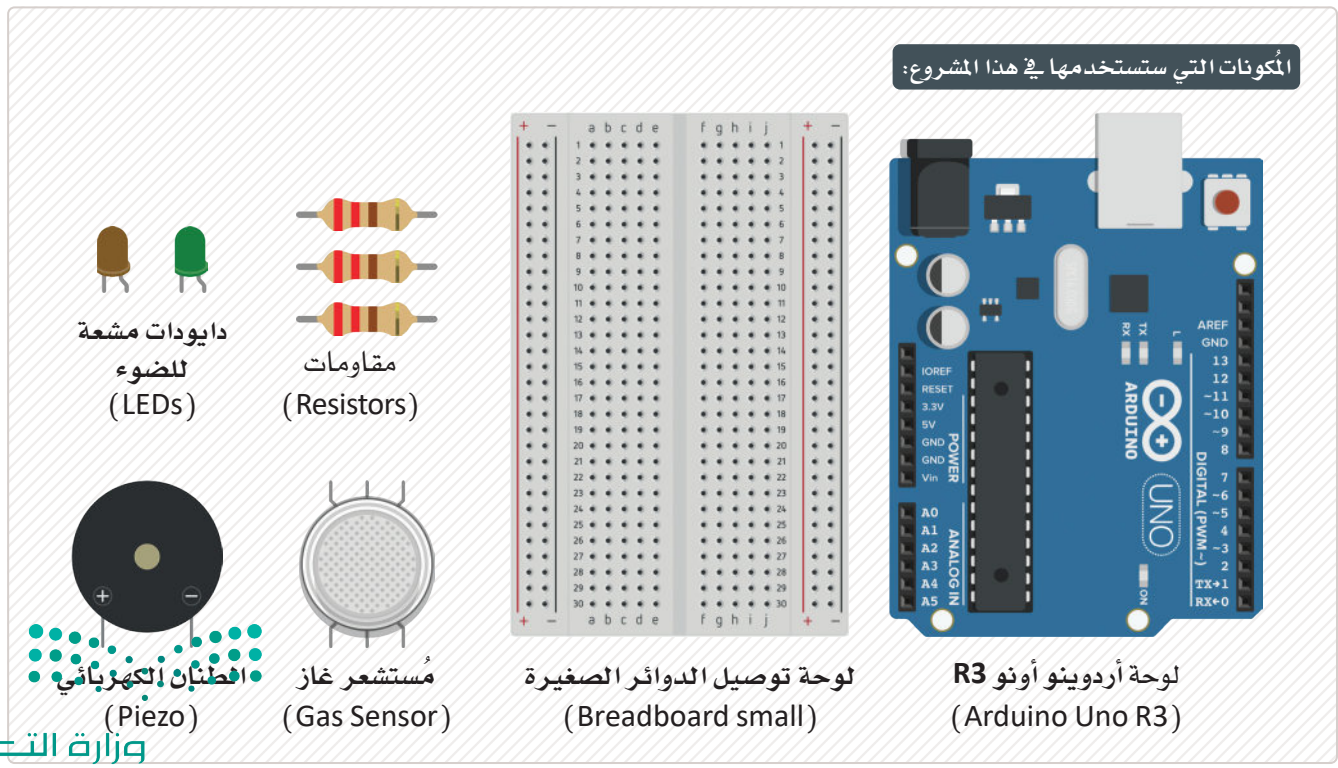
إنشاء نظام تَسرُّب الغاز

إنشاء نظام إنذار تَسرُّب الغاز Build a Gas Leak Alarm System

تُواجه العديد من المرافق مثل المنازل والمصانع خطر تَسرُّب الغاز من الأجهزة أو المواد القابلة للاشتعال، ما قد يتسبب في اندلاع الحرائق، وتتمثل إحدى طرائق منع مثل هذه الحوادث في تزويد تلك المرافق بأنظمة ذكية تكشف عن تَسرُّب الغاز. تستخدم هذه الدائرة لوحة **الأردوينو أونو R3** (Arduino UNO R3) لمحاكاة نظام إنذار تَسرُّب الغاز، فعندما تستقبل لوحة الأردوينو إشارة من مُستشعر الغاز تشير إلى ارتفاع تركيز انبعاثات الدخان، فإنها تبعث إشارة إلى مجموعة من الدايودات المشعة للضوء لتومض بالتناوب، وكذلك إلى طنّانٍ كهربائي لإصدار صوت صفير متقطع.

ستستخدم في هذا المشروع المكونات الآتية:

- طنان كهربائي.
- مُستشعر غاز.
- دايودان مشعان للضوء.
- ثلاثة مقاومات.
- لوحة أردوينو أونو **R3** (Arduino Uno R3).
- لوحة توصيل الدوائر الصغيرة.



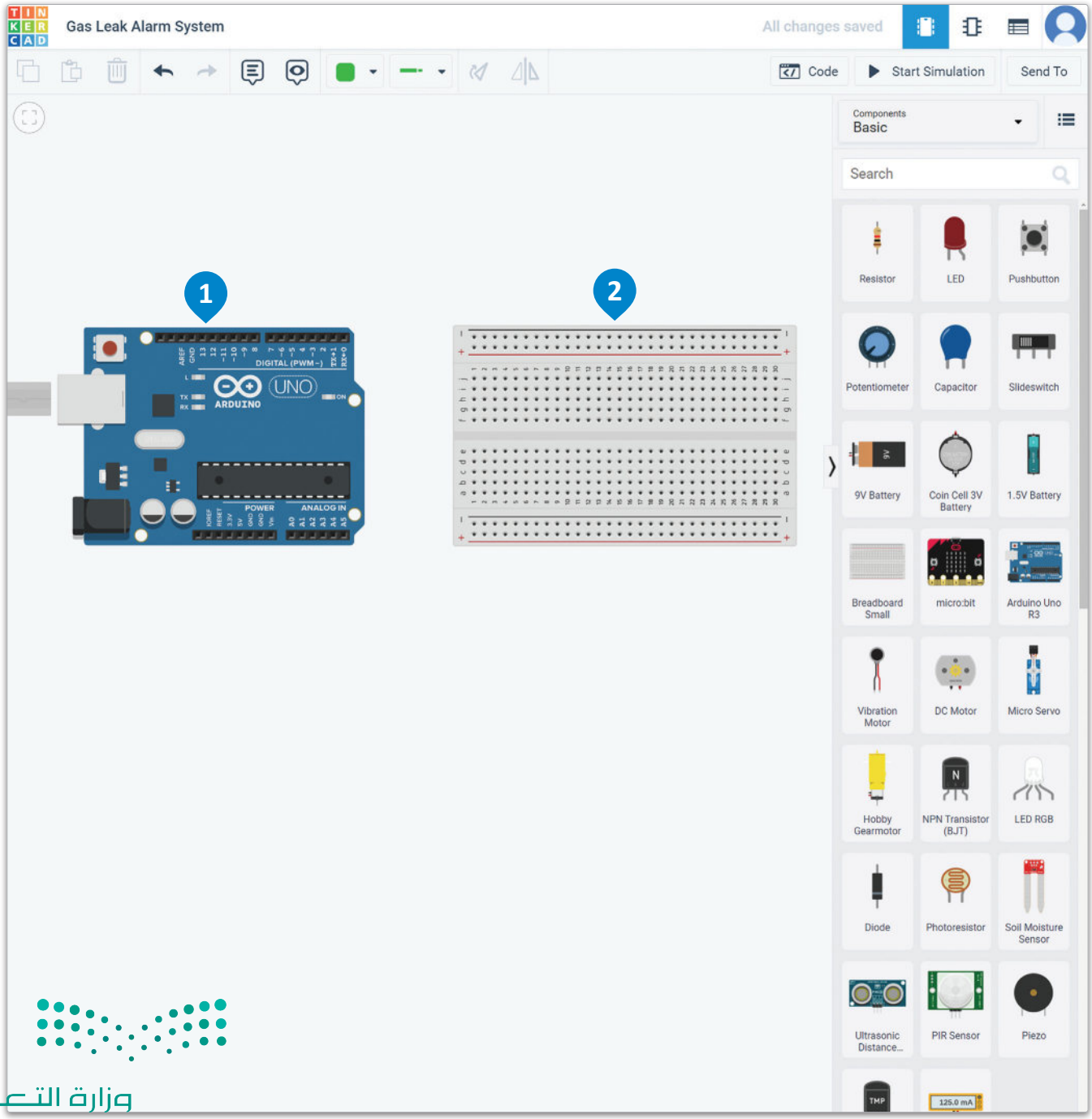
شكل 3.29: مكونات المشروع

ابدأ بإضافة المكونات التي ستحتاج إليها في هذا المشروع إلى مساحة العمل. أولًا، ابحث عن **لوحة توصيل الدوائر الصغيرة** (Breadboard small) ولوحة أردوينو أونو **R3** (Arduino Uno R3).

لإضافة لوحة توصيل الدوائر الصغيرة (Breadboard small) ولوحة الأردوينو أونو R3 (Arduino Uno R3):

> اسحب وأفلت Arduino Uno R3 (لوحة الأردوينو أونو R3) من مكتبة Components (المكونات) إلى مساحة العمل. 1

> اسحب وأفلت Breadboard small (لوحة توصيل الدوائر الصغيرة) من مكتبة Components (المكونات) إلى مساحة العمل. 2

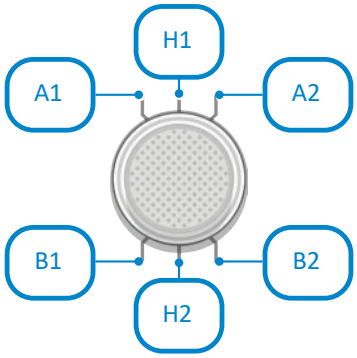


شكل 3.30: إضافة لوحة الأردوينو أونو ولوحة توصيل الدوائر الصغيرة

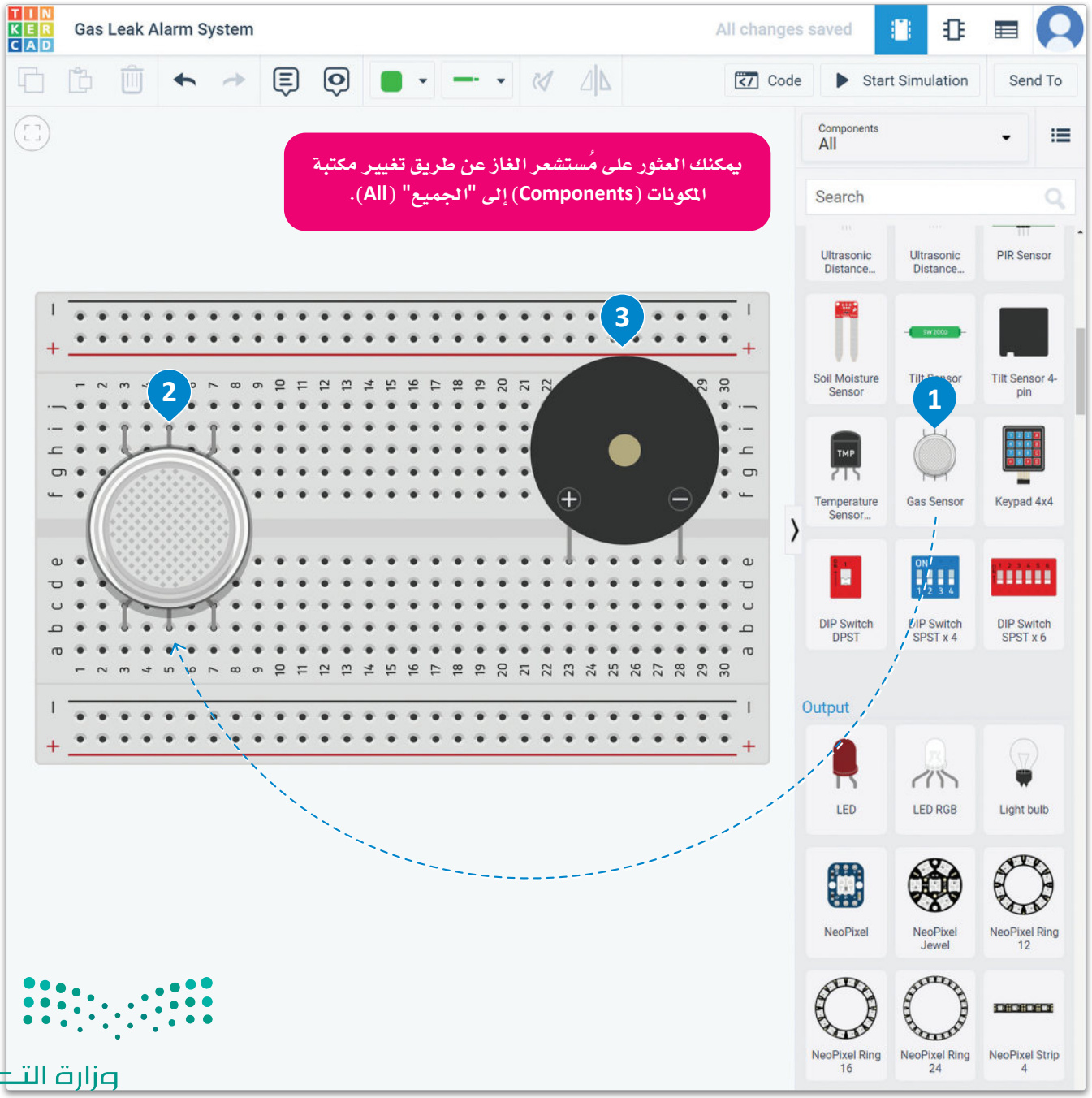
توصيل مُستشعر الغاز والطنان الكهربائي في لوحة توصيل الدوائر الصغيرة.

لإضافة المكونات إلى لوحة توصيل الدوائر الصغيرة:

- ابحث عن Gas sensor (مُستشعر الغاز) من مكتبة Components (المكونات) 1 واسحبه وأفلته في Breadboard Small (لوحة توصيل الدوائر الصغيرة). 2
- اسحب وأفلت Piezo (الطنان الكهربائي) من مكتبة Components (المكونات) في Breadboard Small (لوحة توصيل الدوائر الصغيرة). 3



شكل 3.31: مُستشعر الغاز



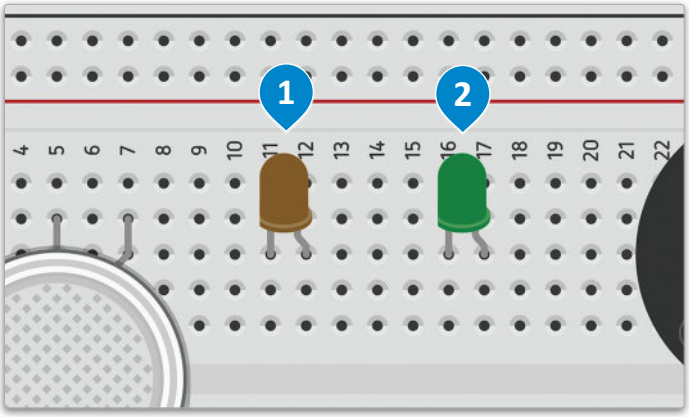
شكل 3.32: إضافة المكونات إلى لوحة توصيل الدوائر الصغيرة

استمر بالعمل وقم بإضافة دايودين مشعين للضوء في لوحة توصيل الدوائر الصغيرة وغيّر ألوانهما.

لإضافة الدايودات المشعة للضوء (LED):

> اسحب وأفلت LED (الدايود المشع للضوء) من مكتبة Components (المكونات) في لوحة توصيل الدوائر الصغيرة وغير لونه إلى اللون orange (البرتقالي). 1

> اسحب وأفلت دايود مشع للضوء آخر في لوحة توصيل الدوائر الصغيرة وغيّر لونه إلى اللون green (الأخضر). 2



شكل 3.33: إضافة الدايودات المشعة للضوء

تحتوي جميع الدايودات المشعة للضوء على حالتين إما تشغيل (On) أو إيقاف (Off). وعندما تكون في وضع الإيقاف، يكون لونها أغمق. على سبيل المثال يكون لون الدايود المشع للضوء البرتقالي أقرب إلى اللون البُني.

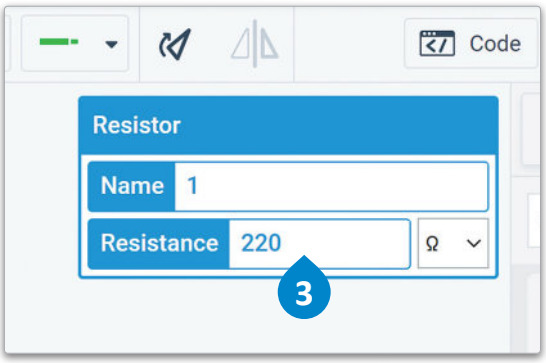
أضف المقاومات إلى الدائرة وغيّر قيمها إلى 220 Ω. ستحتاج إلى ثلاث مقاومات؛ واحدة لكل دايود مشع للضوء، وواحدة لمُستشعر الغاز.

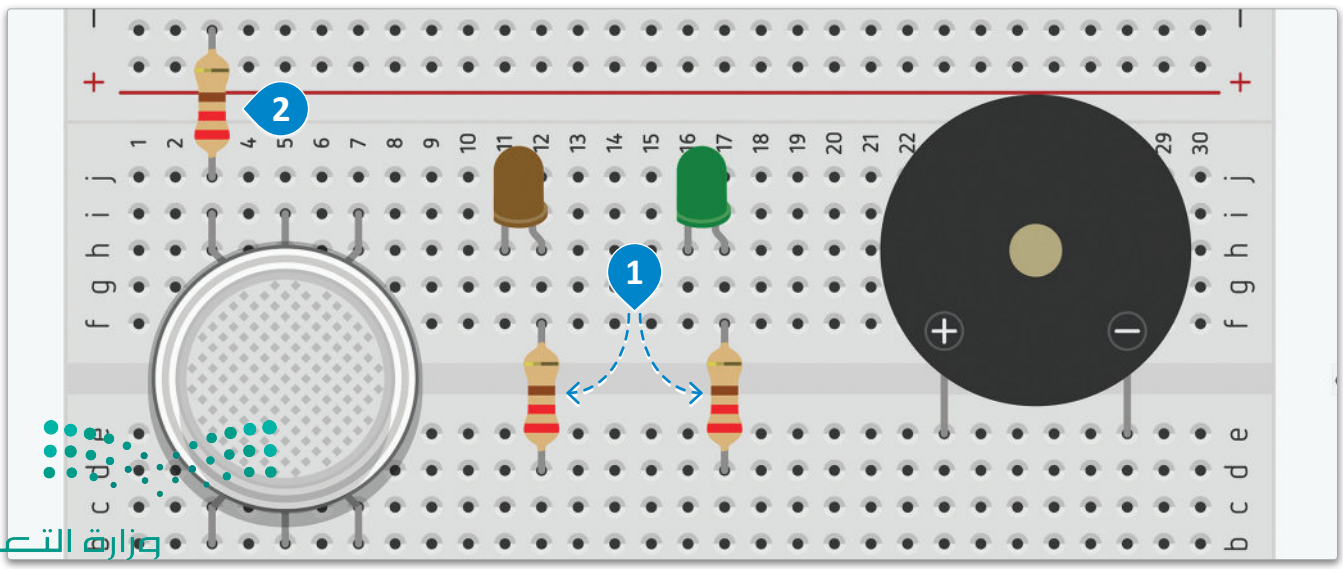
إضافة المقاومات (Resistors):

> اسحب وأفلت resistors (مقاومتين) من مكتبة Components (المكونات) ووصلهما بالدايودات المشعة للضوء. 1

> اسحب وأفلت resistor (مقاومة) أخرى من مكتبة Components (المكونات) ووصلها بالعمود السالب في Breadboard Small (لوحة توصيل الدوائر الصغيرة) وبالطرف A1 لـ Gas sensor (مستشعر الغاز). 2

> اضبط قيمة كل مقاومة على 220 Ω. 3



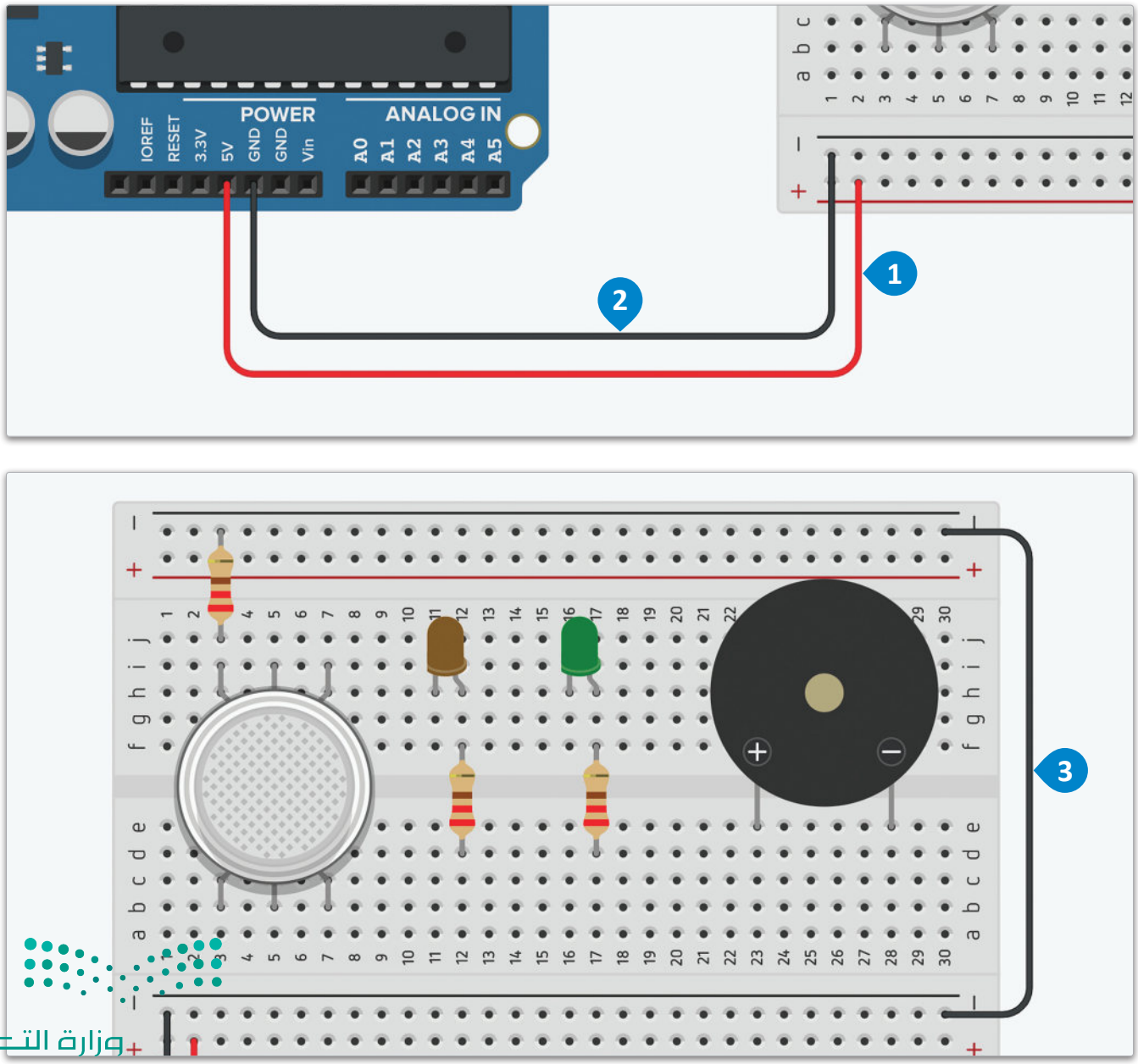


شكل 3.34: إضافة المقاومات

استمر في توصيل أسلاك الدائرة. أولًا ستقوم بتوصيل لوحة الأردوينو بلوحة توصيل الدوائر الصغيرة.

لتوصيل لوحة الأردوينو أونو R3 (Arduino UNO R3):

- قُم بتوصيل الطرف 5V (جهد 5 فولت) في لوحة الأردوينو بالعمود الموجب من لوحة توصيل الدوائر الصغيرة، وغيّر لون السلك إلى red (الأحمر). 1
- قُم بتوصيل GND (الطرف الأرضي) من لوحة الأردوينو أونو R3 بالعمود السالب من لوحة توصيل الدوائر الصغيرة وغيّر لون السلك إلى black (الأسود). 2
- قُم بتوصيل العمودين السالبين من Breadboard Small (لوحة توصيل الدوائر الصغيرة) وغيّر لون السلك إلى black (الأسود). 3



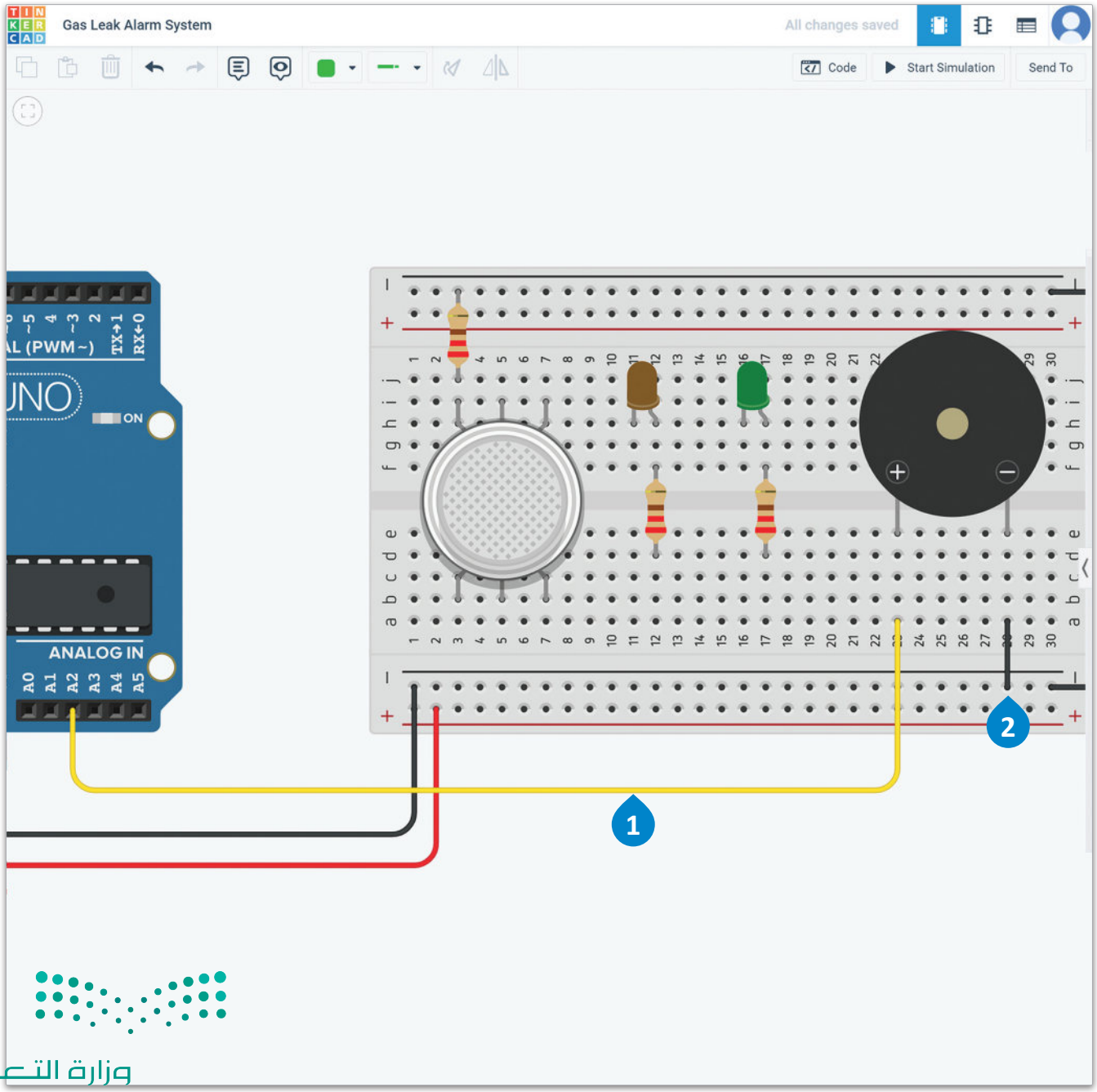
شكل 3.35: توصيل جهاز تحكم الأردوينو الدقيق

استمر بتوصيل أسلاك الطنان الكهربائي.

لتوصيل الطنان الكهربائي:

> قُم بتوصيل الطرف الموجب من Piezo (الطنان الكهربائي) بالطرف التناظري A2 للوحة الأردوينو وغيّر لون السلك إلى yellow (الأصفر). 1

> قُم بتوصيل الطرف السالب من Piezo (الطنان الكهربائي) بالعمود السالب من Breadboard Small (لوحة توصيل الدوائر الصغيرة) وغيّر لون السلك إلى black (الأسود). 2



شكل 3.36: توصيل الطنان الكهربائي

ستوصل بعد ذلك الدايودات المشعة للضوء.

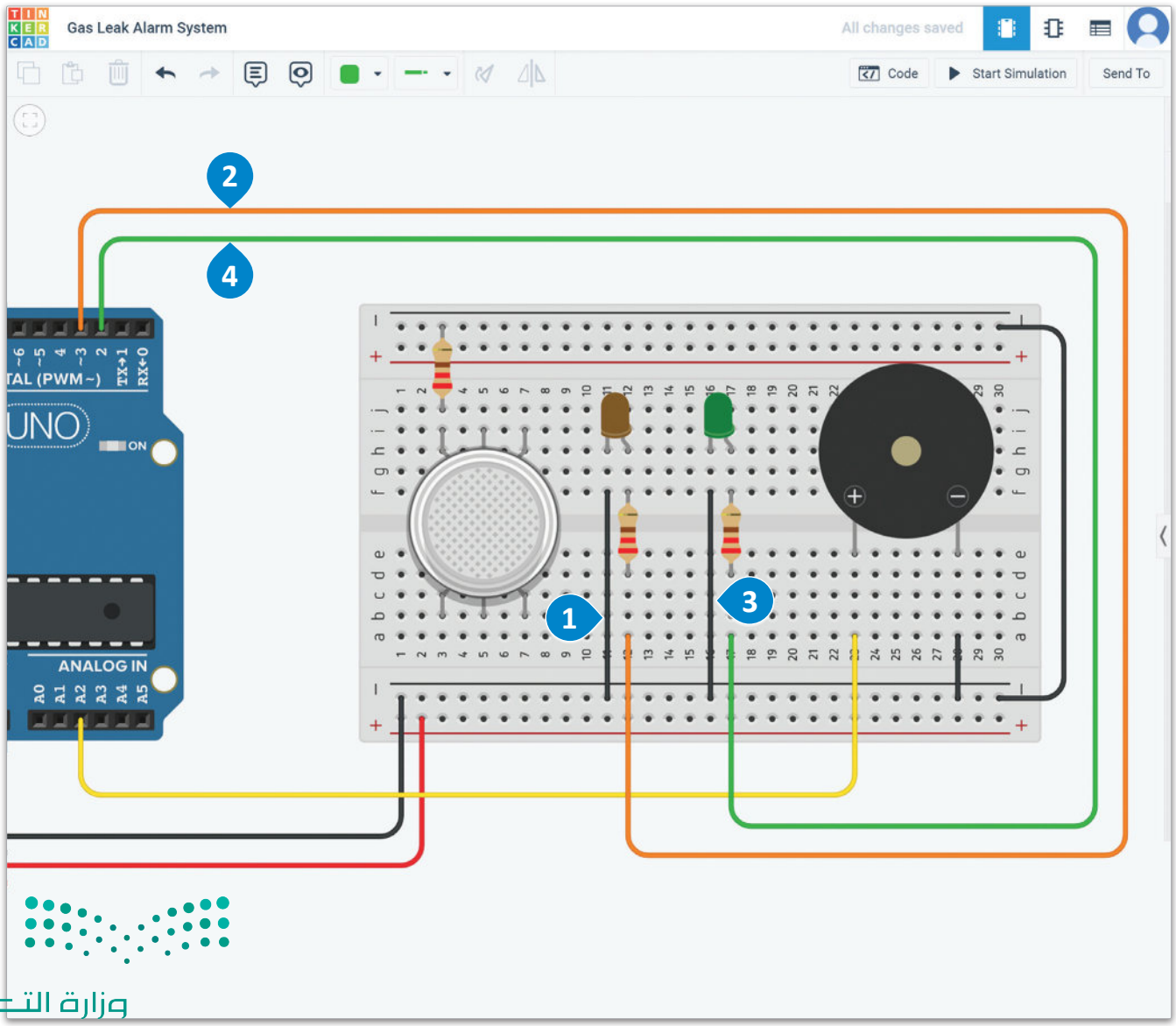
لتوصيل الدايودات المشعة للضوء:

> قُم بتوصيل مِهبط الدايود المشع للضوء البرتقالي بالعمود السالب في Breadboard Small (لوحة توصيل الدوائر الصغيرة)، وغيّر لون السلك إلى black (الأسود). 1

> قُم بتوصيل مقاومة الدايود المشع للضوء البرتقالي إلى Digital pin 3 (الطرف الرقمي 3)، وغيّر لون السلك إلى اللون orange (البرتقالي). 2

> قُم بتوصيل المِهبط الخاص بالدايود المشع للضوء الأخضر بالعمود السالب للوحة توصيل الدوائر الصغيرة (Breadboard Small) وغيّر لون السلك إلى black (الأسود). 3

> قُم بتوصيل مقاومة الدايود المشع للضوء الأخضر بـ Digital pin 2 (الطرف الرقمي 2)، وغيّر لون السلك إلى green (الأخضر). 4



شكل 3.37: توصيل الدايودات المشعة للضوء

في الختام، ستقوم بتوصيل مُستشعر الغاز.

لتوصيل مُستشعر الغاز:

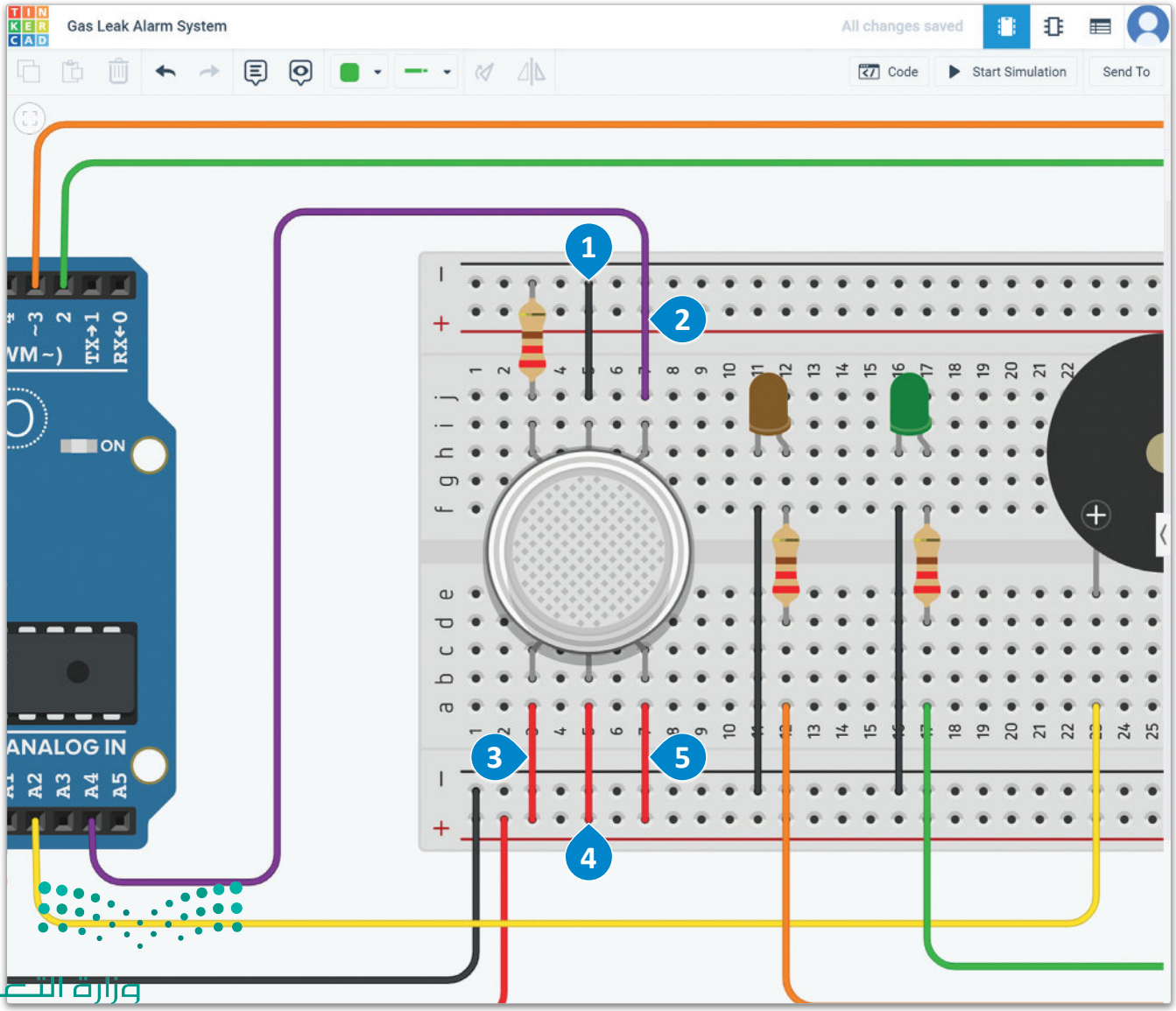
> قُم بتوصيل طرف مُستشعر الغاز H1 بالعمود السالب في Breadboard Small (لوحة توصيل الدوائر الصغيرة) وغيّر لون السلك إلى black (الأسود). 1

> قُم بتوصيل طرف مُستشعر الغاز A2 بـ Arduino Analog pin A4 (طرف الأردوينو الرقمي 4)، وغيّر لون السلك إلى اللون purple (البنفسجي). 2

> قُم بتوصيل طرف مُستشعر الغاز B1 بالعمود الموجب في Breadboard Small (لوحة توصيل الدوائر الصغيرة) وغيّر لون السلك إلى اللون red (الأحمر). 3

> قُم بتوصيل طرف مُستشعر الغاز H2 بالعمود الموجب للوحة توصيل الدوائر الصغيرة. 4

> قُم بتوصيل طرف مُستشعر الغاز B2 بالعمود الموجب للوحة توصيل الدوائر الصغيرة. 5



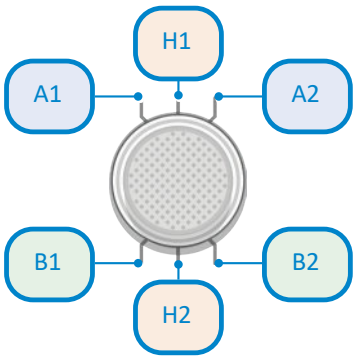
شكل 3.38: توصيل مُستشعر الغاز

كيف يعمل مُستشعر الغاز How the Gas Sensor Works

يحتوي مُستشعر الغاز على ستة أطراف؛ طرفان بحرف A، وطرفان بحرف B، وآخران بحرف H. يعمل المستشعر من خلال الكشف عن جزيئات الغاز وتحويل تركيز الغاز المُستشعَر إلى جُهد كهربائي مختلف.

أما الغرض من الأطراف ذات الحرف H فهو تسخين ملف السخان، والذي بدوره يُنشط المُستشعر الكهروكيميائي. يجب توصيل طرف H واحد بمصدر جهد (VCC)، على سبيل المثال **5 فولت** (5V) أو **3.3 فولت** (3.3V) وطرف H الآخر إلى الأرضي.

لنقل البيانات من مُستشعر الغاز إلى لوحة الأردوينو، يجب استخدام زوجي الطرفين A أو زوجي الطرفين B، حيث يُوصل أحد أطراف الزوج المستخدم بمصدر الجهد (VCC)، ويُوصل الطرف الآخر بالأرضي من خلال المقاومة، وذلك حتى يُمكن ضبط حساسية المُستشعر. يجب توصيل الأطراف غير المستخدمة بمصدر الجهد (VCC).



شكل 3.39: مُستشعر الغاز

الطنان الكهربائي The Piezo Buzzer

يمكن أن يُنتج الطنان الكهربائي مجموعة واسعة من نغمات الأصوات وبمدةٍ مختلفة لكل منها.

لجعل السماعة المتصلة بالطرف A2 تصدر نغمة بتردد 110 هرتز لمدة ثانية واحدة، استخدم اللبنة البرمجية الآتية:

تُكتب المدة بالثواني، ولكن قد لا يمكنك التعرف على نغمات الطنان ووحداتها. يوجد بالجدول أدناه مجموعة من القيم تتوافق مع ترددات النوتات الموسيقية المقاسة **بالهيرتز** (Hz). يمكنك تجربة بعض النوتات الموسيقية ونغماتها كما يعرض هنا:

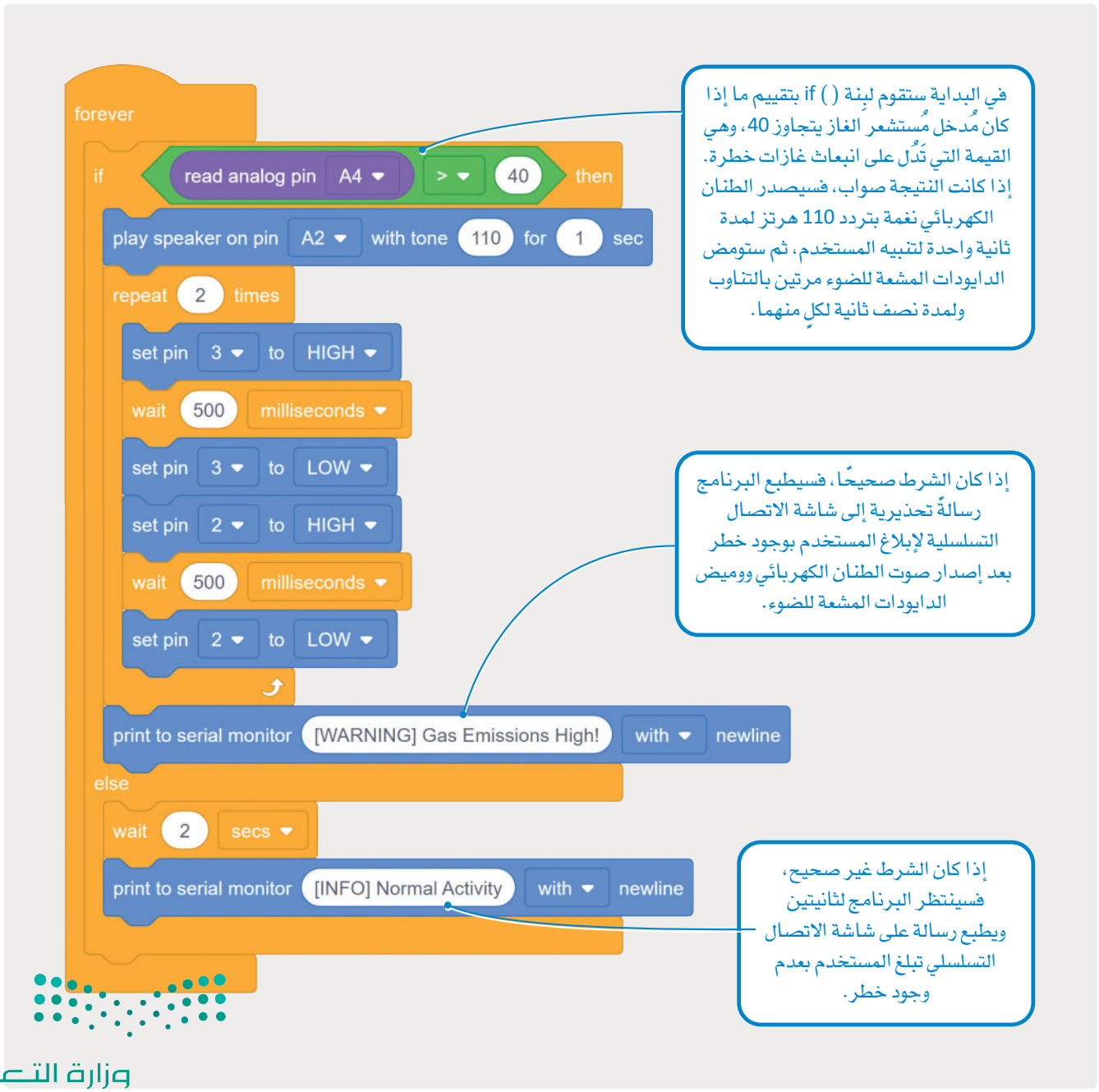
جدول 3.3: النغمات الصوتية

النغمة	الترددات
لا (LA)	110 Hz
دو (DO)	131 Hz
ري (RE)	147 Hz
فا (FA)	175 Hz

برمجة نظام إنذار تسرب الغاز Gas Leak Alarm System Code

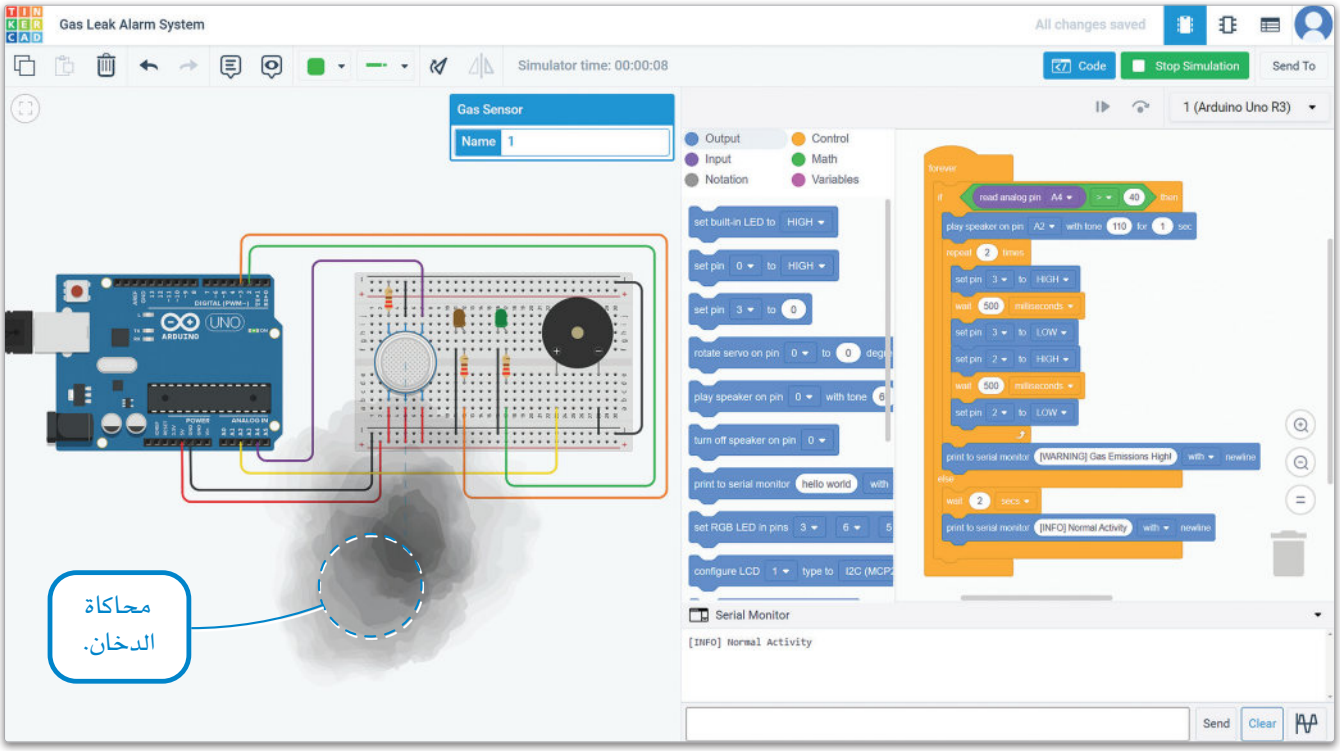
بعد أن قمت بعمل التوصيلات اللازمة للنظام، وتعرفت على طريقة عمل مُستشعر الغاز والطنان الكهربائي، فإن الخطوة التالية هي برمجة نظام الإنذار الذي قمت بتصميمه. يراقب البرنامج مخرجات مُستشعر الغاز للتحقق من وجود خطر حدوث حريق. وإذا اكتُشف خطر، فسيصدر صوت إنذار من الطنان الكهربائي مع وميضٍ من الدايودات المشعة للضوء، أما إذا لم يكن الأمر كذلك، فإن البرنامج سينتظر بعض الوقت.

أنشئ البرنامج الآتي في منطقة البرمجة، ثم اضغط على بِدء المحاكاة (Start Simulation) لمحاكاة تشغيل نظام إنذار تسرب الغاز.



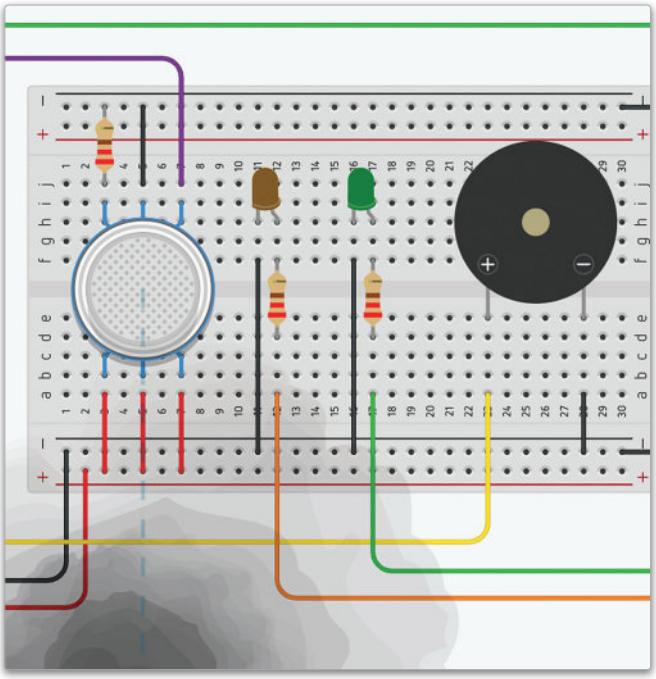
شكل 3.40: برنامج نظام إنذار تسرب الغاز

قم بتشغيل البرنامج لاختباره.



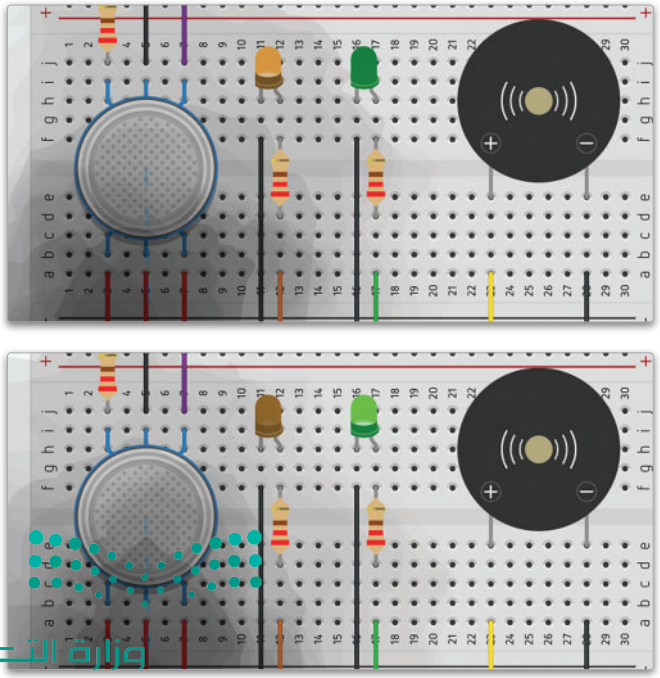
شكل 3.41: تنفيذ البرنامج

عندما يكون الدخان بعيدًا عن مُستشعر الغاز، فلن يصدر الطنان الكهربائي أي صوت، ولن تومض الدايودات المشعة للضوء.



شكل 3.42: لم يتم اكتشاف تسرب الغاز

عندما يكون الدخان قريبًا من المُستشعر، فسيبدأ الطنان الكهربائي في إصدار صوت صفير، وستومض الدايودات المشعة للضوء البرتقالية والخضراء بالتناوب.



شكل 3.43: تم اكتشاف تسرب الغاز

تمرينات

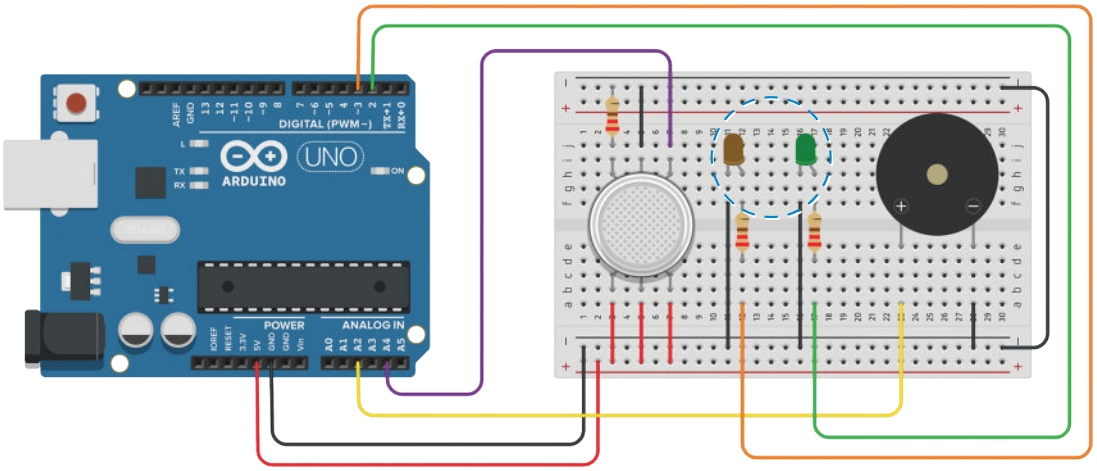
1 ابحث في الإنترنت عن أنواعٍ مختلفة لمخاطر الغاز التي يمكن لمُستشعر الغاز اكتشافُها وتحليلها.
اعرض نتائج بحثك أدناه.

2 كيف يُمكن للطنان الكهربائي إصدار نغماتٍ مختلفة كإنذاراتٍ للأنواع المختلفة من المخاطر.
اعرض أفكارك أدناه.

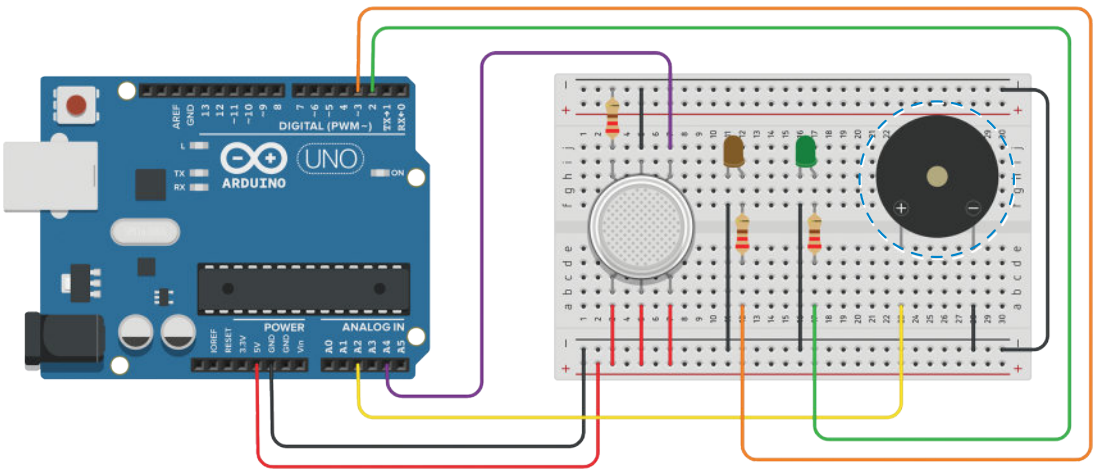
3 صِف الاستخدامات الممكنة للطنان الكهربائي بخلاف نظام الإنذار.

4 يفوق عدد أطراف مُستشعر الغاز والتي تُوصل وتُربط بجهاز التحكم الدقيق عدد الأطراف المُستخدمة في المُستشعرات الأخرى. راجع المعلومات والإشارات التي يُصدرها مُستشعر الغاز وعلل سبب اختلافه عن الأنواع الأُخرى من المُستشعرات.

5 قم بتغيير نمط وميض الدايودات المشعة للضوء، فبدلًا من تشغيل كل دايود مشع للضوء وإيقافه مرتين لمدة 500 مللي ثانية، سيتم تشغيل وإيقاف تشغيل الدايود المشع للضوء الأخضر فقط ثلاث مرات ولمدة ثانية واحدة كل مرة.



6 قم بتغيير نمط صفير الطنان الكهربائي بحيث يُصدر صفيرًا بنغمة بتردد 220 هرتز لمدة 700 مللي ثانية بدلًا من إصدار نغمة بتردد 110 هرتز لمدة ثانية واحدة.



7 قم بتوسيع الدائرة بحيث يصدر الطنان تحذيرًا إضافيًا عند زيادة قيمة الغاز عن 100.

المشروع

تحتل المحميات الزراعية أهمية في مجال الزراعة، لاسيما في المناخ والظروف التي تُعيق الاستثمار الزراعي. يجب مراقبة المحميات الزراعية لضبط الظروف بداخلها وحمايتها.

1 صَمِّم ونفِّذ دائرةً في بيئة محاكاة تينكركاد باستخدام جهاز تحكم الأردوينو الدقيق والدايودات المشعة للضوء والمستشعرات لمحاكاة وحدة مراقبة محمية زراعية لإشعار المستخدِم بوجود تغيرات في بيئتها مثل: الحركة، وتغير درجة الحرارة، ورطوبة التربة، ووجود الدخان.

2 استخدم ألوانًا مختلفة للدايودات المشعة للضوء الخاصة بكل مُستشعر لتتيح للمستخدِم تمييز التغير المحدد.

3 قم بتوسيع التصميم بحيث يقوم أيضًا بإصدار الرسائل في وحدة التحكم عند استيفاء الشروط. على سبيل المثال، عندما يكتشف مُستشعر الغاز دخانًا، فقد تظهر رسالة **خطر الحريق** (Fire Hazard!).

وزارة التعليم
Ministry of Education
2023 - 1445

ماذا تعلّمت

- التعرف على مكونات جهاز التحكم الدقيق وبرمجته.
- قياس البيانات التي جُمعت من مُستشعرات الإدخال المختلفة.
- فهم كيفية عمل بيانات المُستشعرات والخوارزميات معًا في البرمجة.
- استخدام تنبيهات التشغيل والاستجابات الآلية.
- تصميم دوائر إنترنت الأشياء باستخدام جهاز تحكم الأردوينو الدقيق في بيئة محاكاة تينكركاد.
- برمجة جهاز تحكم الأردوينو الدقيق باستخدام لغة لِبنات التعليمات البرمجية في بيئة محاكاة تينكركاد.

المصطلحات الرئيسة

مُستشعر الحرارة	Temperature Sensor
مُستشعر لاسلكي	Wireless Sensor Networks

مُستشعر الغاز	Gas Sensor
تضمين عرض النبضة	Pulse-Width Modulation
مُستشعر رطوبة التربة	Soil Moisture Sensor

وزارة التعليم
Ministry of Education
2023 - 1445

4. إنشاء تطبيق سحابي لإنترنت الأشياء

ستتعلم في هذه الوحدة خطوات إنشاء تطبيق لإنترنت الأشياء يراقب البيئة المحيطة، ويقوم بجمع البيانات وإرسالها عبر الإنترنت إلى منصة سحابية، كما ستُنشئ دائرة باستخدام جهاز تحكم الأردوينو الدقيق، وستُبرمجها باستخدام لغة بايثون.

أهداف التعلُّم

بنهاية هذه الوحدة سيكون الطالب قادرًا على أن:

- يستخدم بايثون لبرمجة جهاز تحكم الأردوينو الدقيق مع بروتوكول PyFirmata.
- يصمم دائرة لتطبيق إنترنت الأشياء باستخدام جهاز تحكم الأردوينو الدقيق.
- يراقب البيئة المحيطة ويجمع بيانات المُستشعر في الزمن الفعلي.
- يستخدم خدمة الويب لإرسال بيانات مُجمعة إلى منصة سحابية.
- يراقب بيئة بعيدة من خلال بيانات على مِنصة سحابية.
- يتعرّف على طريقة توظيف بيانات المستشعر والبيانات السحابية في اتخاذ قرارات وفق تلك البيانات المُجمّعة.
- يتعرّف على طريقة توسيع نطاق تطبيقات إنترنت الأشياء لتشمل حلول معقدة.

الأدوات

- بيئة التطوير المتكاملة للأردوينو (Arduino IDE)
- أداة جيت برينز باي تشارم (JetBrains PyCharm)
- مَنصة الحوسبة السحابية Binary IoT Cloud

وزارة التعليم
Ministry of Education
2023 - 1445

رابط الدرس الرقمي

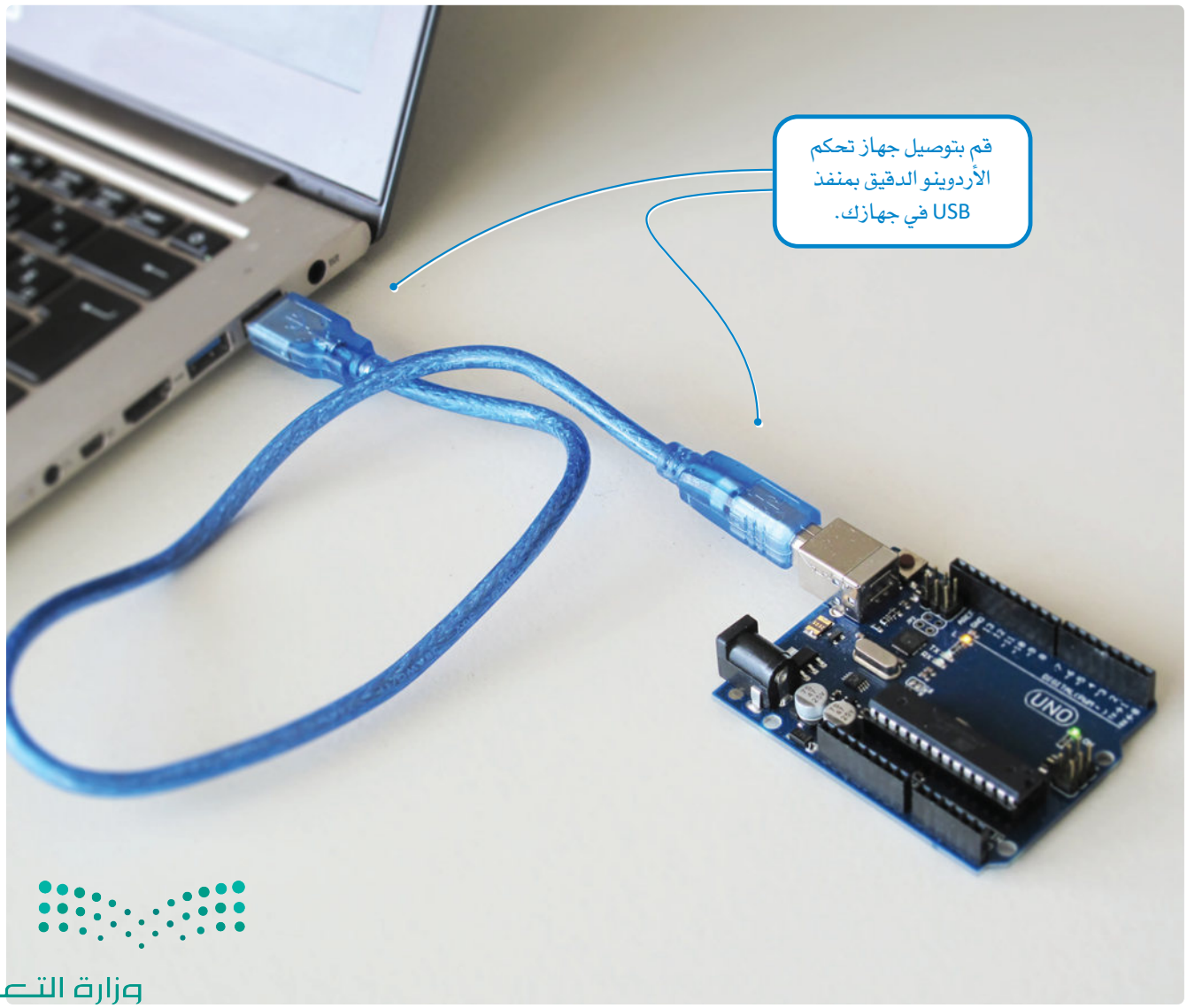
www.ien.edu.sa

الدرس الأول

إعداد بيئة تطوير الأردوينو

استخدام لغة بايثون في برمجة لوحة الأردوينو Using Python with Arduino

تعتبر لغة C++ بمثابة لغة البرمجة الرسمية لجهاز تحكم الأردوينو الدقيق، ولكن يُمكن استخدام لغةٍ أخرى مثل بايثون لبرمجته وذلك من خلال بروتوكول **Firmata**. تُعدُّ بايثون لغة برمجة عالية المستوى، وتكمُن قوتها في العدد الكبير من المكتبات التي يُمكن استخدامها لكي تدعم هذه اللغة وتجعلها شاملة للأغراض المختلفة والمتعددة، ويقوم بروتوكول **Firmata** بتوفير الاتصال بين جهاز التحكم الدقيق وبين الأوامر التي تزوده بها لغة البرمجة. ستستخدم هنا لغة بايثون مع مكتبة **PyFirmata**، والتي تُشكِّلُ واجهة بروتوكول **Firmata**.



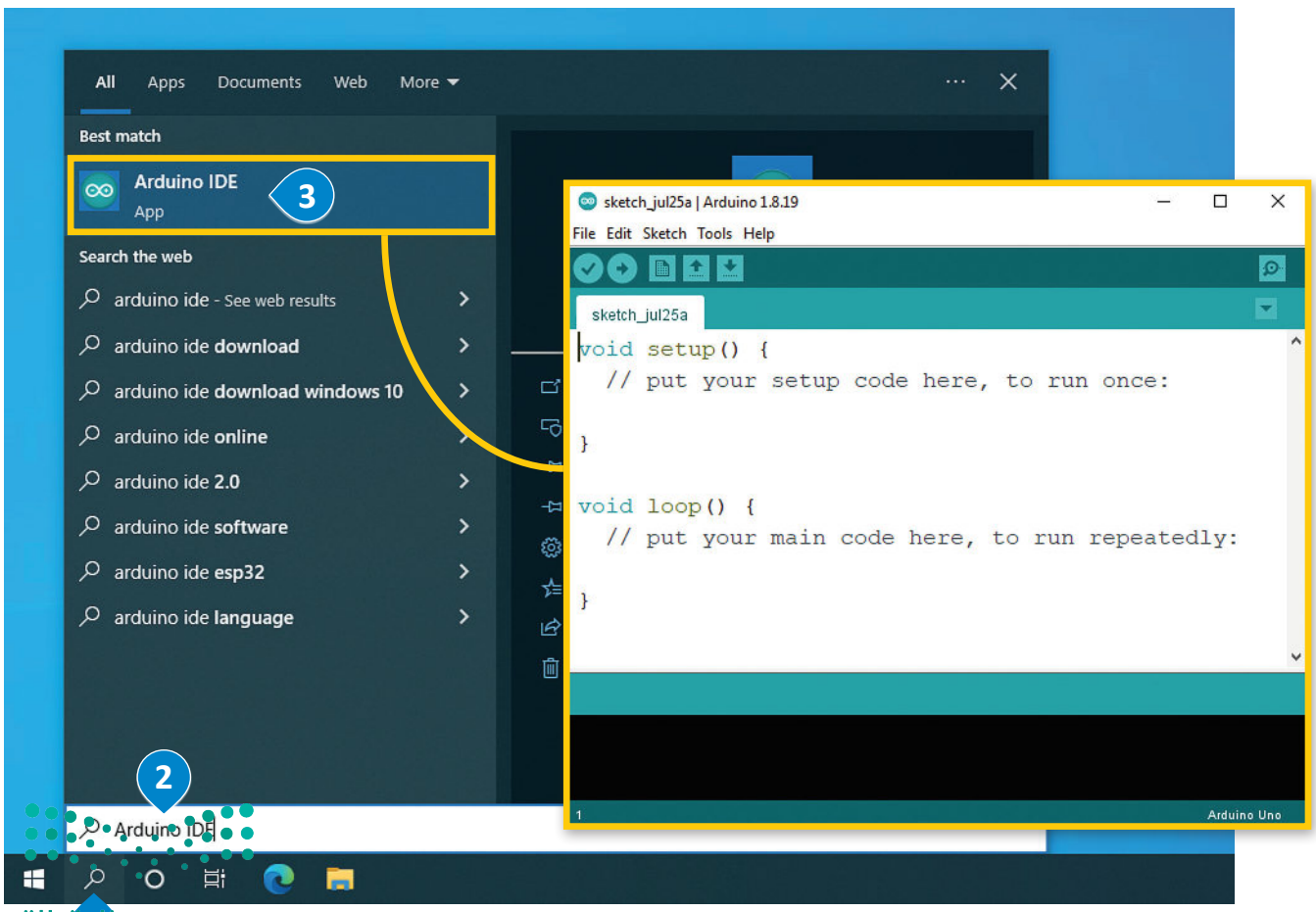
شكل 4.1: توصيل لوحة الأردوينو بِمنفذ USB للحاسب المحمول

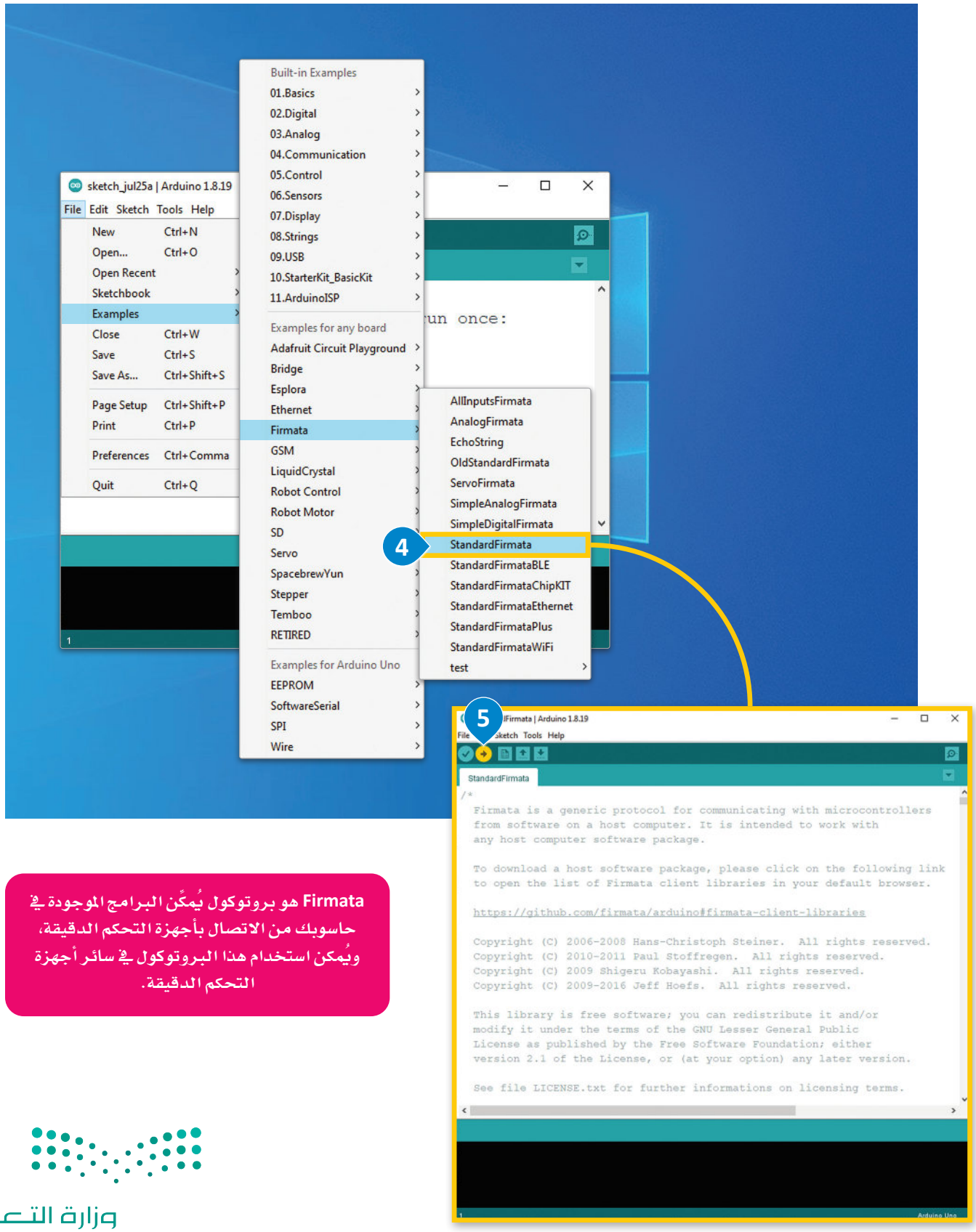
تُعدُّ **بيئة التطوير المتكاملة للأردوينو** (Arduino IDE) بمثابة مُحرر نصي صُمم خصيصًا لأجهزة التحكم الدقيقة في الأردوينو، وتُزوَّد هذه البيئة بأدوات إضافية للتفاعل مع الأردوينو، وتحتوي على برامج مُعدّة مُسبقًا لِأداء مهامٍ مُحدّدة في الأردوينو. لتثبيت بيئة التطوير المتكاملة للأردوينو، انتقل إلى الموقع الإلكتروني **https://www.arduino.cc/en/software** وقم بتنزيل أحدث إصدار، ثم قم بعملية التثبيت من خلال تشغيل برنامج التثبيت. وبعد ذلك ستظهر بيئة التطوير المتكاملة للأردوينو لتقوم بتحميل المكتبة القياسية StandardFirmata لإجراء عملية الاتصال بين الأردوينو وبرنامجك في بايثون. تأكد من أن إصدار بيئة التطوير المتكاملة للأردوينو الذي تقوم بتنزيله متوافق مع جهاز تحكم الأردوينو الدقيق الخاص بك.

يمكنك العثور على **بيئة التطوير المتكاملة للأردوينو** (Arduino IDE) بواسطة أيقونة البحث في حاسوبك.

لتحميل StandardFirmata:

- اضغط على أيقونة البحث في ويندوز 1 واكتب Arduino IDE. 2
- افتح Arduino IDE (بيئة التطوير المتكاملة للأردوينو). 3
- حمّل StandardFirmata في Arduino IDE (بيئة التطوير المتكاملة للأردوينو) بالضغط على File (ملف) > Examples (أمثلة) > Firmata > StandardFirmata. 4
- قم بتحميل المكتبة إلى الأردوينو. 5

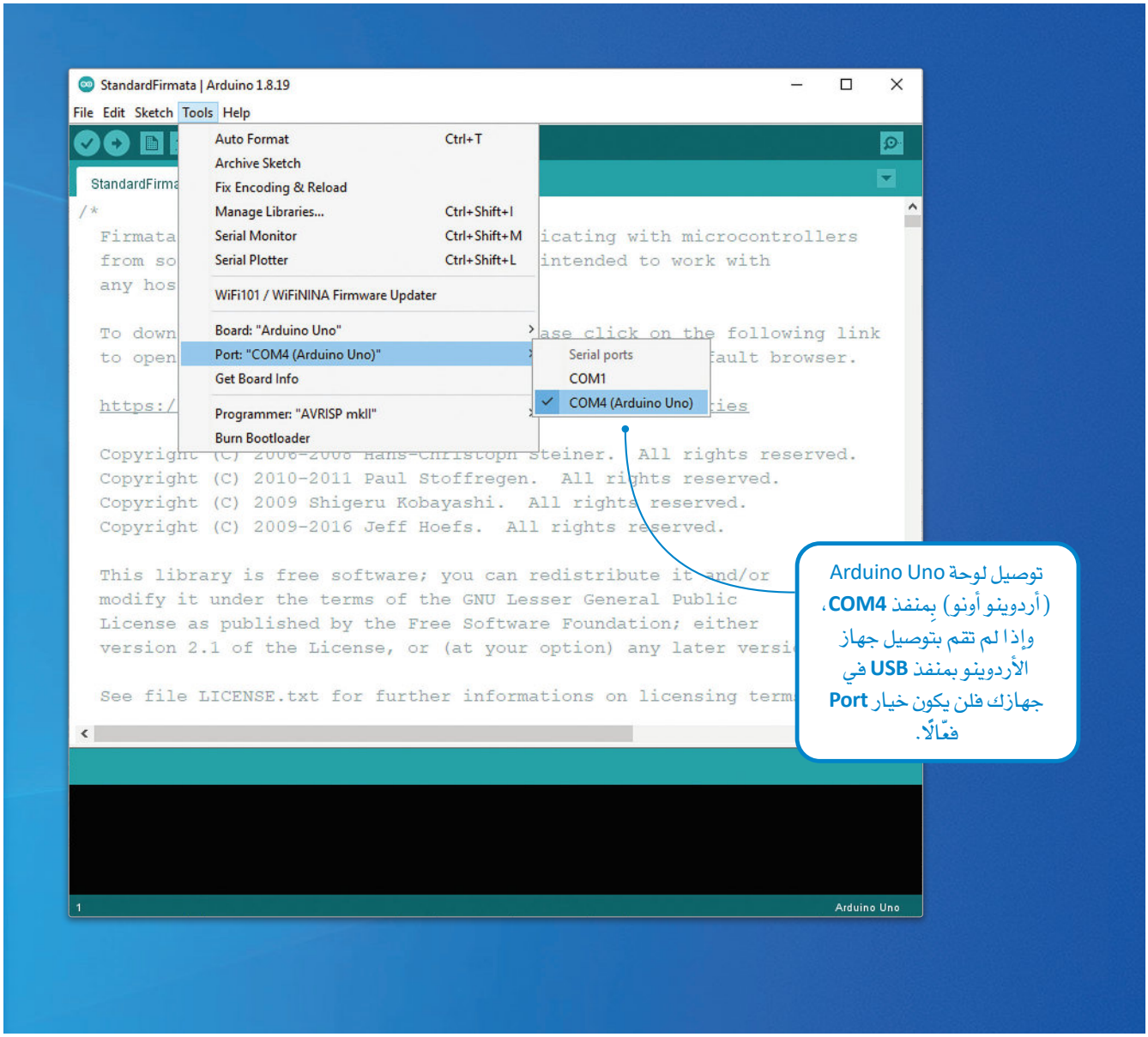




Firmata هو بروتوكول يُمكّن البرامج الموجودة في حاسوبك من الاتصال بأجهزة التحكم الدقيقة، ويُمكن استخدام هذا البروتوكول في سائر أجهزة التحكم الدقيقة.

شكل 4.2: تحميل مكتبة Firmata

يمكنك الوصول إلى مَنفذ الاتصال من حاسوبك إلى لوحة الأردوينو بالضغط على **Tools** (أدوات) ثم **Port** (منفذ) ثم **Serial Ports** (منافذ تسلسلية) كما هو موضح أدناه. عُيّن مَنفذ الاتصال في هذا المثال إلى COM4. قد يختلف المنفذ في حاسوبك، فعلى سبيل المثال قد يكون COM3 أو COM5. تذكَّر أن تُدوِّن مَنفذ الاتصال، حيث ستستخدمه في برنامج بايثون للاتصال بلوحة الأردوينو.



شكل 4.3: عرض منفذ اتصال الأردوينو

عند تشغيل جهاز تحكم الأردوينو الدقيق باستخدام لغة البايثون، عليك إبقاء مكتبة StandardFirmata قيد التشغيل لكي يتمكن برنامج البايثون الذي تكتبه من الاتصال بالأردوينو.

الآن وبعد أن قمت بتحميل StandardFirmata على جهاز تحكم الأردوينو الدقيق، عليك اتباع الخطوات الآتية مع كل مشروع تقوم بتنفيذه باستخدام لغة البايثون:

افتح **باي تشارم** (PyCharm) وقم بتثبيت حِزمة pyfirmata من خلال نظام مدير الحِزَم (pip). في باي تشارم، افتح الواجهة **الطرفية** (Terminal) في مُجلد عملك، وقُم بإدخال الأمر الآتي:

```
pip install pyfirmata
```

أنشئ ملف بايثون جديد، وفي بداية تعليماتك البرمجية، استدعِ حِزمة pyfirmata البرمجية بالسطر البرمجي أدناه:

```
import pyfirmata
```

أنشئ مُتغيرًا باسم *communication-port* (مَنفذ الاتصال)، يقوم بتخزين اسم مَنفذ الاتصال بحاسوبك حيث يتم توصيل لوحة الأردوينو:

```
communication_port = “COM4”
```

استخدم الأوامر الآتية لإجراء الاتصال بين برنامج البايثون ولوحة الأردوينو الخاصة بك، وللوصول إلى أطراف لوحة الأردوينو:

```
# Set the Arduino port to read from
board = pyfirmata.Arduino(communication_port)

# Set up access to the inputs of the circuit
it = pyfirmata.util.Iterator(board)
it.start()
```

يتعين عليك استخدام تكرار لا نهائي تُنفذ من خلاله أوامرك بصورة مستمرة في الأردوينو.

```
while True:
    # write your code here
```

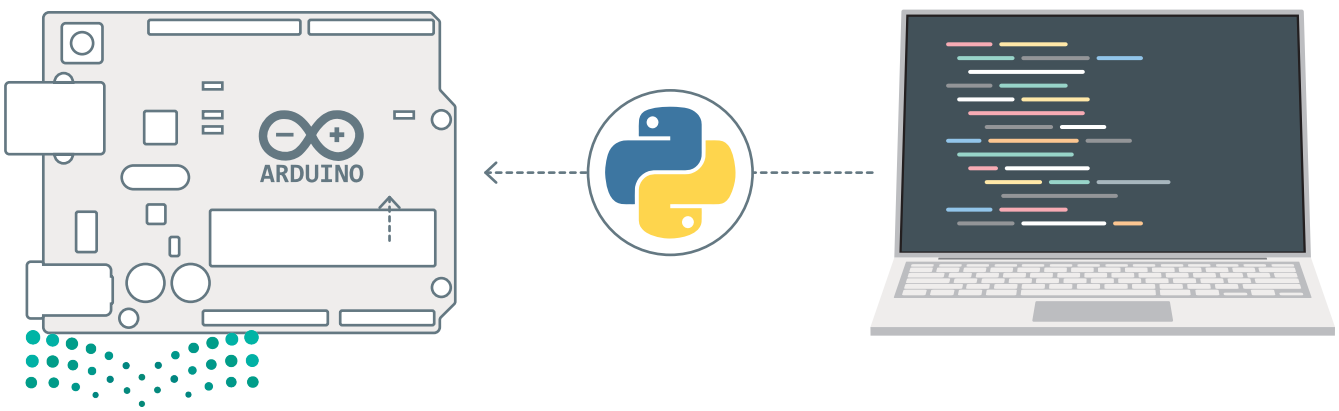
شكل 4.4: برمجة الأردوينو في البايثون من خلال pyfirmata

على الرغم من أن الاتصال يبدو من خلال منفذ USB، إلا أن الواقع هو استخدام ويندوز لمنفذ تسلسلي قياسي لتبادل البيانات مع جهاز الأردوينو، حيث يقوم نظام التشغيل بإنشاء مَنفذ افتراضي تسلسلي.

تبدأ كافة برامج بايثون للوحة الأردوينو بما يلي:

```
import pyfirmata

communication_port = “COM4”

# Set the Arduino port to read from
board = pyfirmata.Arduino(communication_port)

# Set up pyfirmata to access the status of the inputs of the circuit
it = pyfirmata.util.Iterator(board)
it.start()

while True:
    # write your code here
```

يُشغّل البرنامج الذي تقوم بتطبيقه على حاسوبك، وليس على الأردوينو، مما يعني أنه يمكنك الوصول إلى كافة الوظائف التي قد لا تتوافر في جهاز تحكم الأردوينو الدقيق.



شكل 4.5: برمجة الأردوينو في بايثون من خلال pyfirmata

التعامل مع PyFirmata Working with PyFirmata

الوظائف الرئيسة التي تحتاج إلى تنفيذها في بايثون بواسطة PyFirmata هي قراءة القيم وكتابتها لكافة الأطراف التناظرية والرقمية لجهاز تحكم الأردوينو الدقيق. يُنفّذ الإجراء أعلاه في pyfirmata بواسطة الدالة board.get_pin( )، والتي تَستقبل **مُعامِلات** (Parameters) مُكونة من ثلاثة أحرف تفصل بينَ كلٍّ منهما نقطتان رأسيتان.

المُعامِل الأول هو "a" أو "d" ويعني طرف **تناظري** (analog) أو **رقمي** (digital).

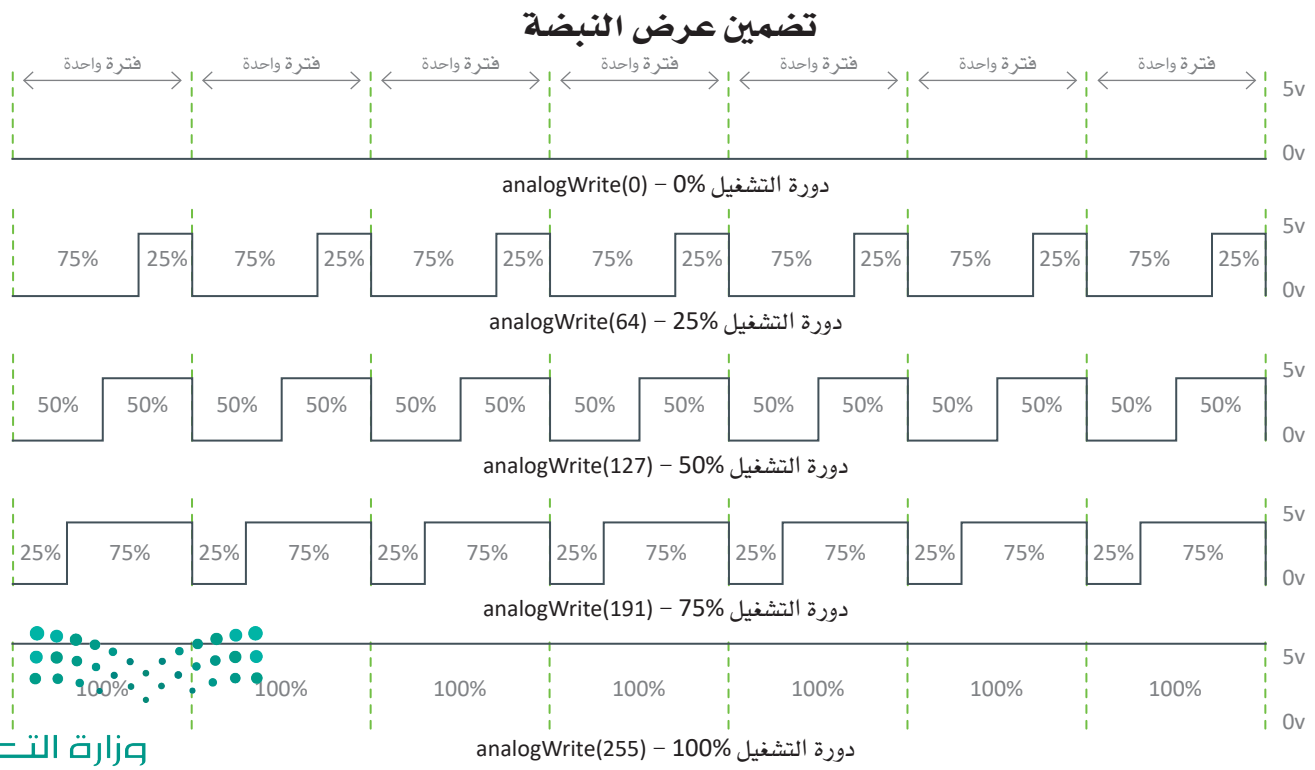
- المُعامِل الثاني هو رقم الطَرف الذي تريده.
- تُرقّم الأطراف الرقمية من 0–12.
- وتُرقّم الأطراف التناظرية من A0–A5.

المُعامِل الثالث هو طريقة التفاعل مع جهاز تحكم الأردوينو الدقيق.

- بالنسبة للأطراف الرقمية، يُشير الحرف "i" إلى **input** (مُدخلات)، والحرف "o" إلى **output** (مُخرجات).
- بالنسبة للأطراف التناظرية، يُشير الحرف "i" إلى **input** (مُدخلات) والحرف "p" إلى **Pulse Width Modulation** (تضمين عرض النبضة).

تضمين عرض النبضة (Pulse Width Modulation-PWM):

هو عملية تعديل تستخدم الإخراج الرقمي لإصدار إشارة تناظرية بقوة متغيرة.



شكل 4.6: تضمين عرض النبضة

التفاعل مع الأطراف الرقمية Interacting with Digital Pins

قراءة قيمة من **الطرف الرقمي 10** (digital pin 10).

```
digital_pin = board.get_pin("d:10:i")
pin_value = digital_pin.read()
```

كتابة قيمة إلى **الطرف الرقمي 10** (digital pin 10).

```
digital_pin = board.get_pin("d:10:o")
digital_pin.write(1)
digital_pin.write(0)
```

التفاعل مع الأطراف التناظرية Interacting with Analog Pins

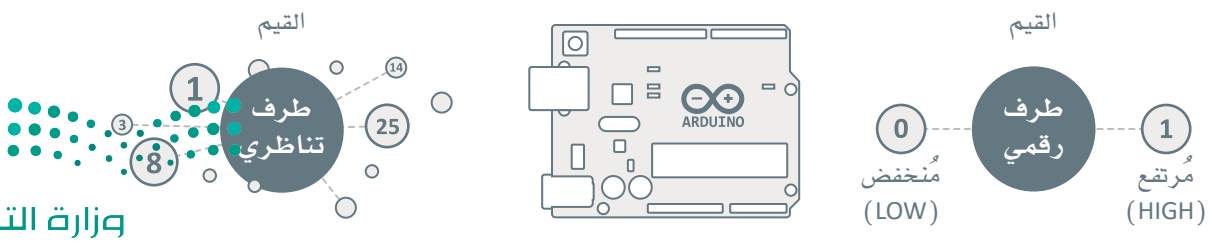
قراءة قيمة من **الطرف التناظري 4** (analog pin 4).

```
analog_pin = board.get_pin("a:4:i")
pin_value = analog_pin.read()
```

كتابة قيمة إلى **الطرف التناظري 4** (analog pin 4).

```
analog_pin = board.get_pin("a:4:p")
analog_pin.write(0.75)
```

يتطلب كل مُستشعر أو مُشغل قِيمًا مختلفة ليعمل بصورة صحيحة.



شكل 4.7: الاختلاف بين قراءة البيانات من الأطراف الرقمية والأطراف التناظرية

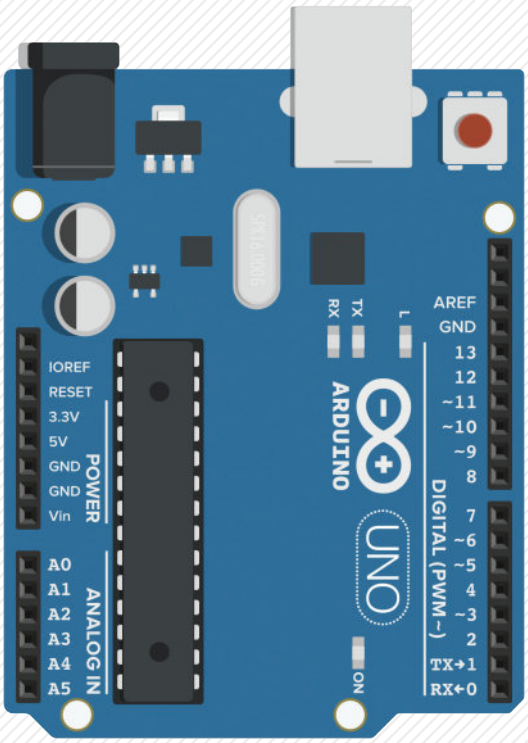
مشروع أردوينو مُبسط مع PyFirmata

A Simple Arduino Project with PyFirmata

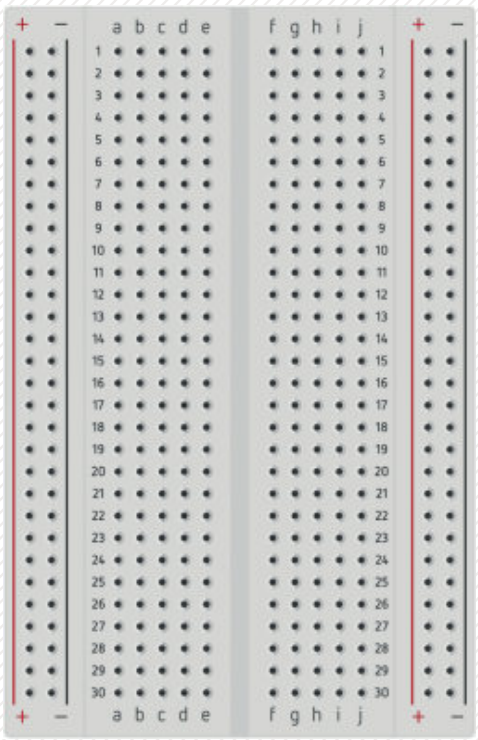
لكي تتعرف على طريقة استخدام مكتبة PyFirmata، ستُنشئ مشروع أردوينو مبسطًا يستخدم دايودًا خارجيًا مشعًا للضوء، بالإضافة إلى طرف الدايود المشع للضوء المُدمج في الأردوينو. ستستخدم بايثون لبرمجة كُل دايود مشع للضوء ليومض بالتناوب. ستنشئ أولًا محاكاة للدائرة في **دوائر تينكركاد** (Tinkercad Circuits)، وستحتاج في هذا المشروع إلى المكونات الآتية:

- لوحة أردوينو أونو R3.
- لوحة توصيل الدوائر الصغيرة.
- دايود مشع للضوء.
- مقاومة.

المُكونات التي ستستخدمها في هذا المشروع:



لوحة أردوينو أونو R3
(Arduino Uno R3)



لوحة توصيل الدوائر الصغيرة
(Breadboard Small)

مقاومة (Resistor)

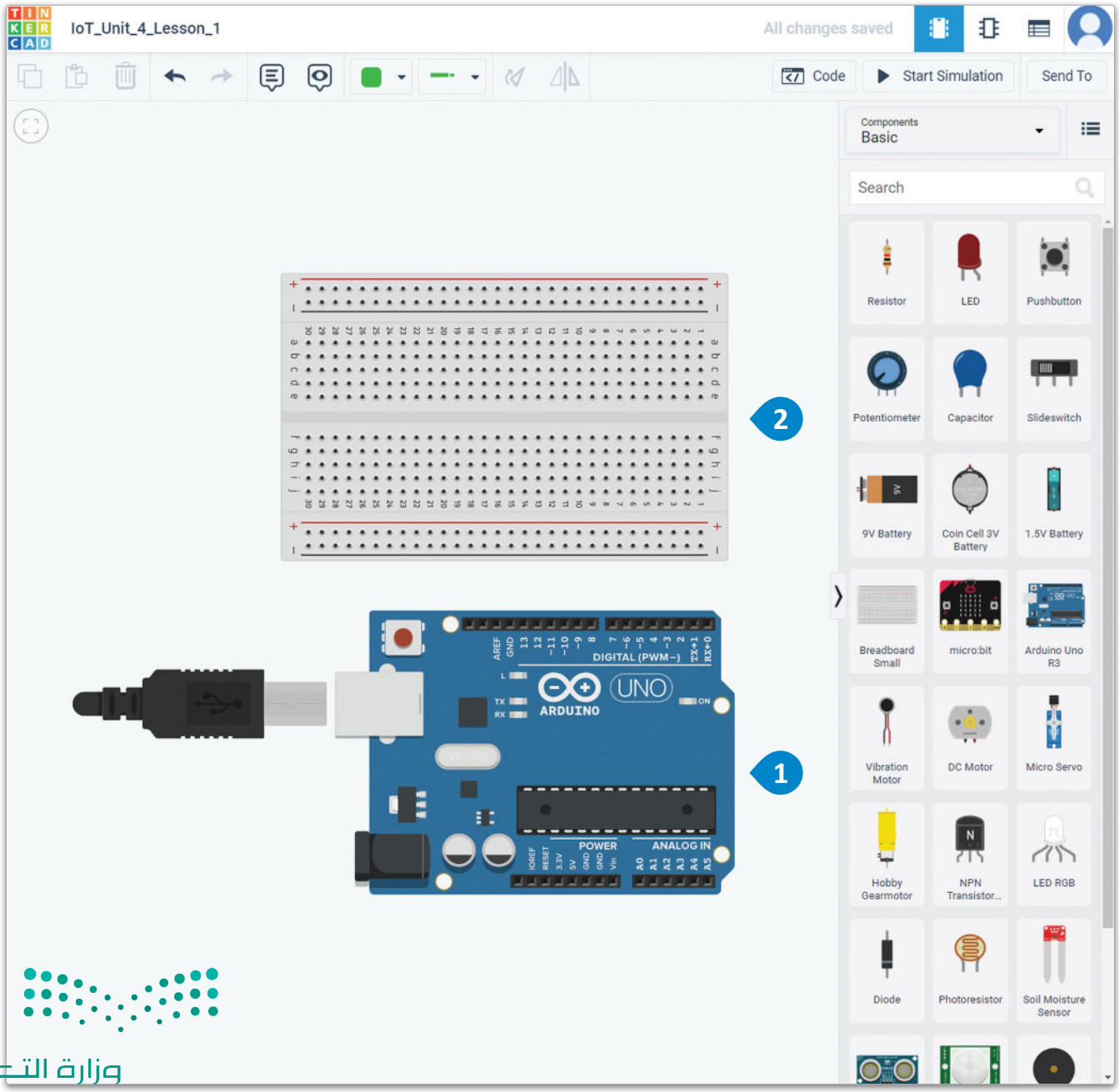
دايود مشع للضوء (LED)

شكل 4.8: مُكونات المشروع

ابدأ بوضع المكونات المطلوبة في مساحة عمل محاكي دوائر تينكركاد.

لتحميل المكونات:

- ابحث عن Arduino Uno R3 (لوحة أردوينو أونو R3) من مكتبة Components (المكونات)، واسحبها وأفلتها في مساحة العمل. 1
- ابحث عن Breadboard Small (لوحة توصيل الدوائر الصغيرة) من مكتبة Components (المكونات)، واسحبها وأفلتها في مساحة العمل. 2

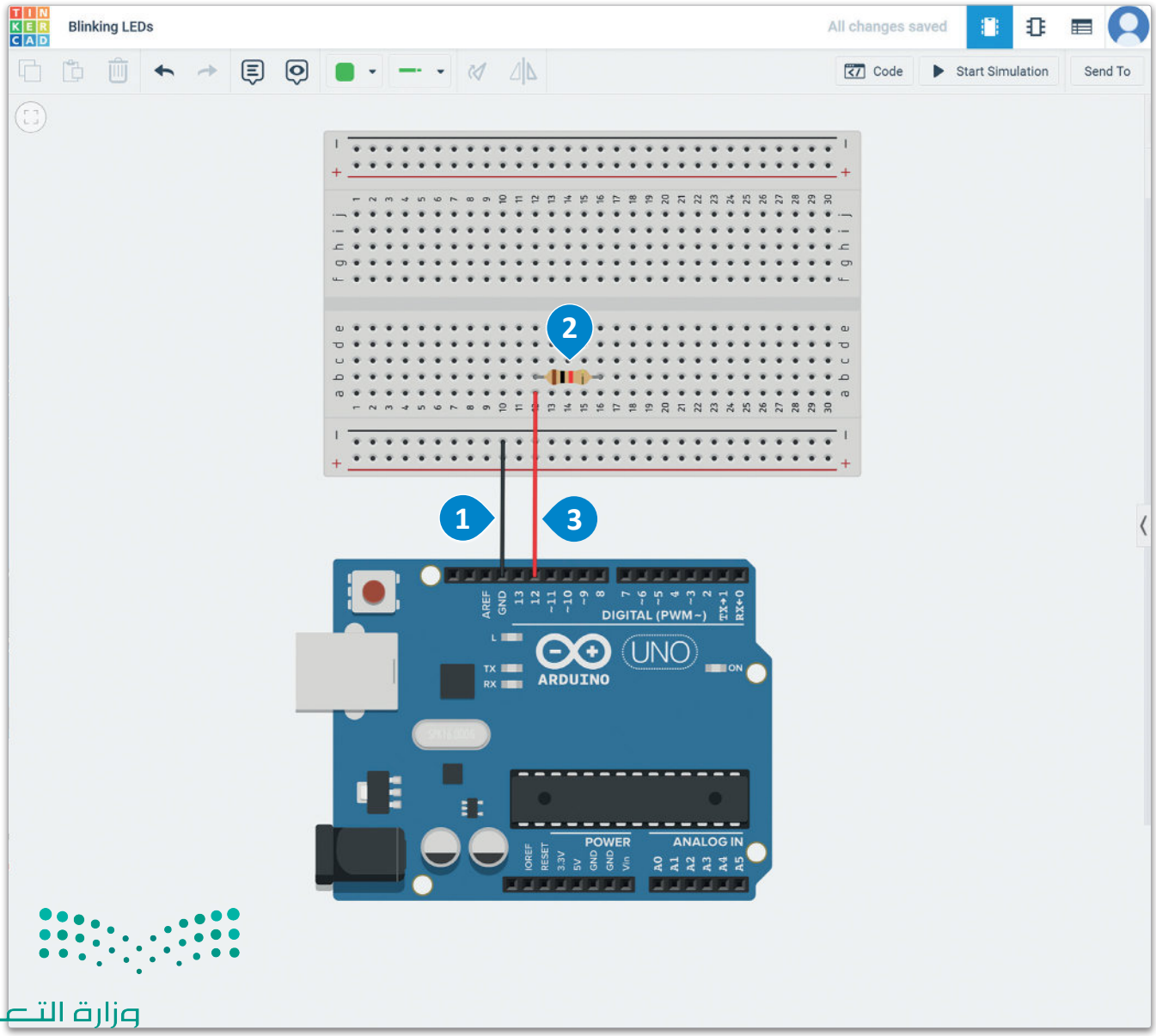


شكل 4.9: تحميل المكونات

ستكمل الآن توصيل الأردوينو بدايود خارجي مشع للضوء.

لتوصيل الأردوينو:

- وصّل GND (الطرف الأرضي) للوحة الأردوينو بالعمود السالب من Breadboard Small (لوحة توصيل الدوائر الصغيرة)، وغيِّر لون السلك إلى black (الأسود). 1
- ابحث عن Resistor (المقاومة) من مكتبة Components (المكونات)، وضعها على Breadboard Small (لوحة توصيل الدوائر الصغيرة)، 2 ثم قم بتوصيل Digital pin 12 (الطرف الرقمي 12) بـ Terminal 1 (الطرف 1) من المقاومة وغيِّر لون السلك إلى red (الأحمر). 3



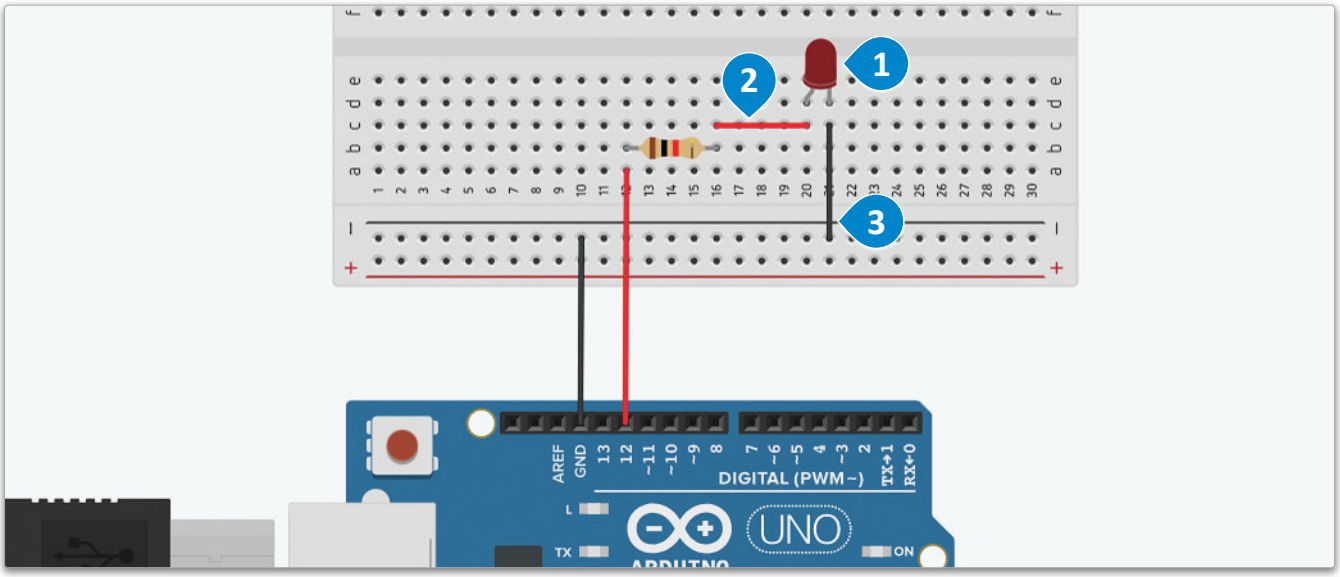
شكل 4.10: توصيل الأردوينو

لتوصيل الدايود المشع للضوء:

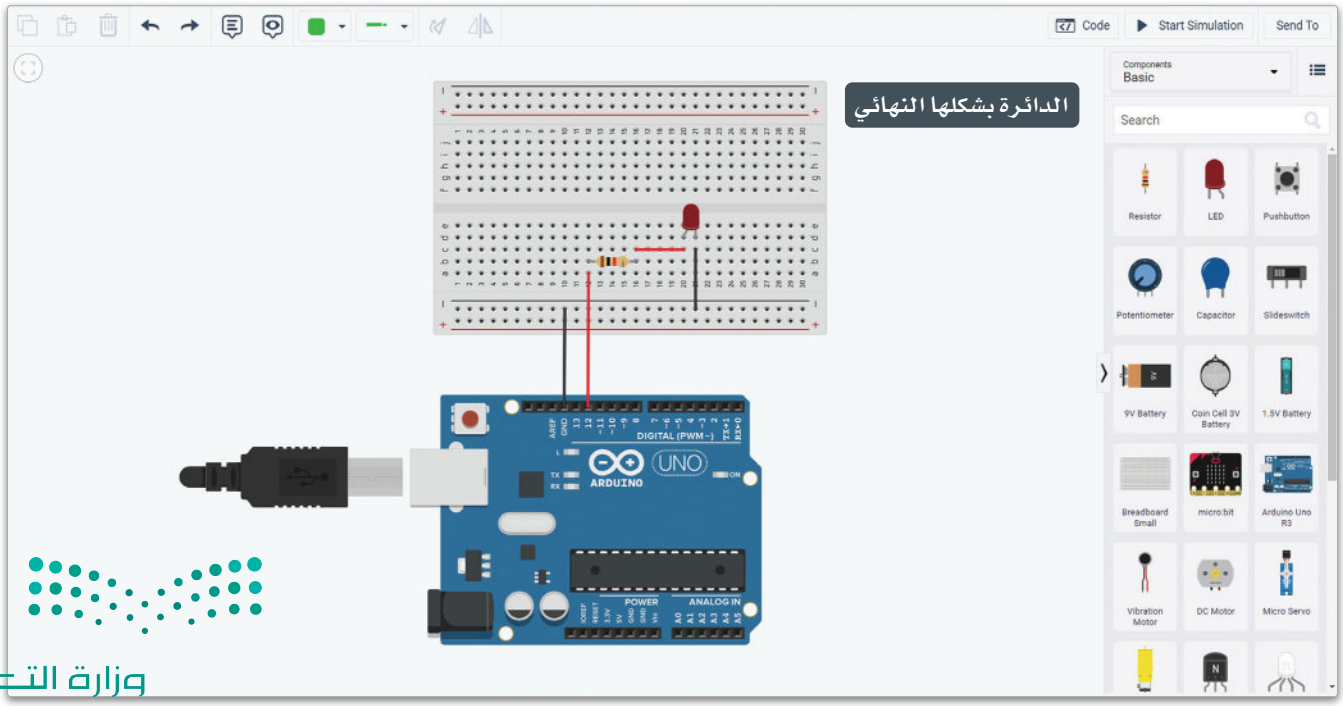
> ابحث عن LED (الدايود المشع للضوء) من مكتبة Components (المكونات)، وضعه على Breadboard Small (لوحة توصيل الدوائر الصغيرة). 1

> قم بتوصيل Terminal 2 (طرف 2) من المقاومة بالدايود المشع للضوء. 2

> قم بتوصيل مهبط الدايود المشع للضوء بالعمود السالب في Breadboard Small (لوحة التوصيل الصغيرة). 3



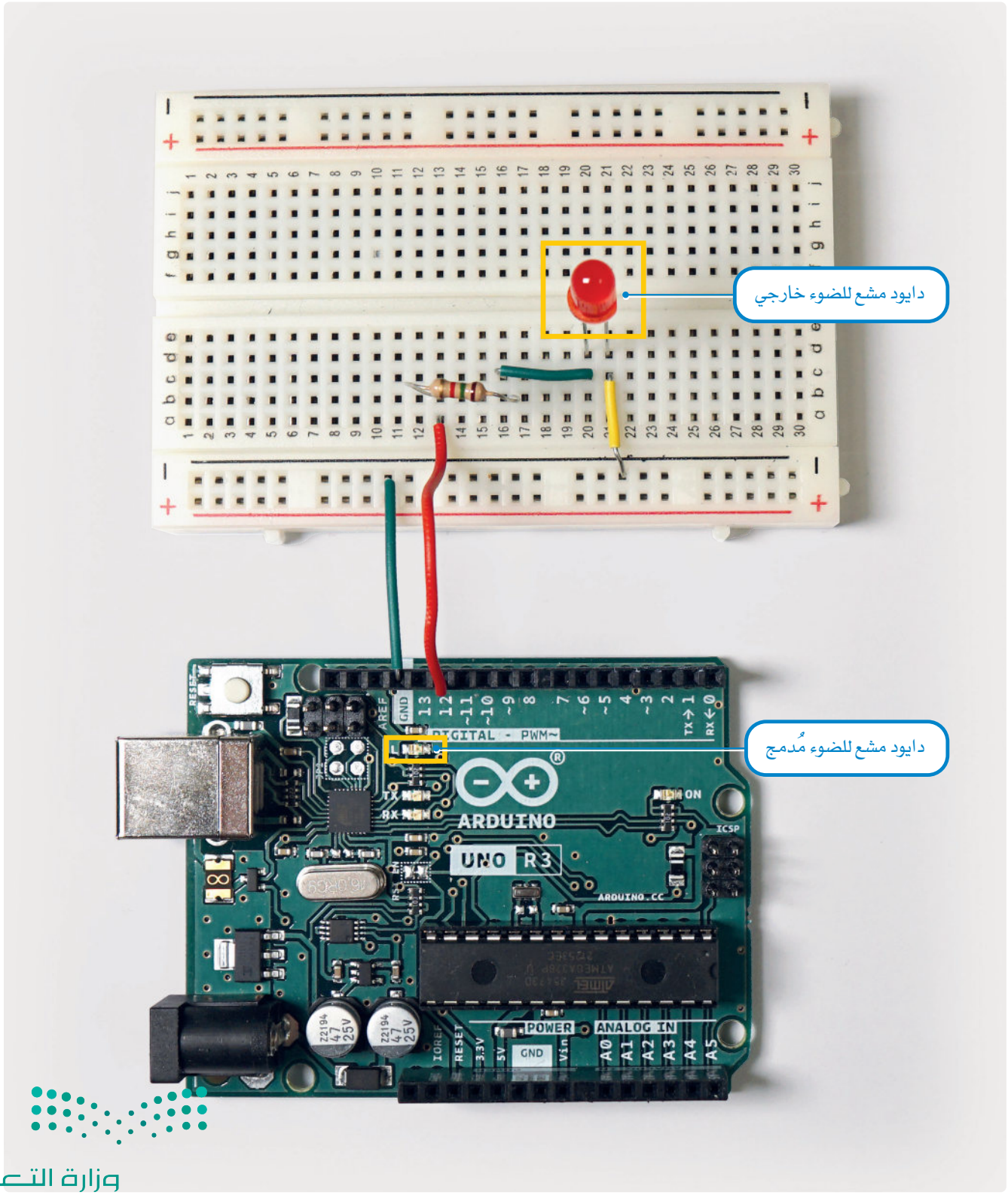
شكل 4.11: توصيل الدايود المشع للضوء الخارجي



شكل 4.12: الدائرة بشكلها النهائي في دوائر تينكركاد

التركيب الفعلي للدائرة Physical Circuit

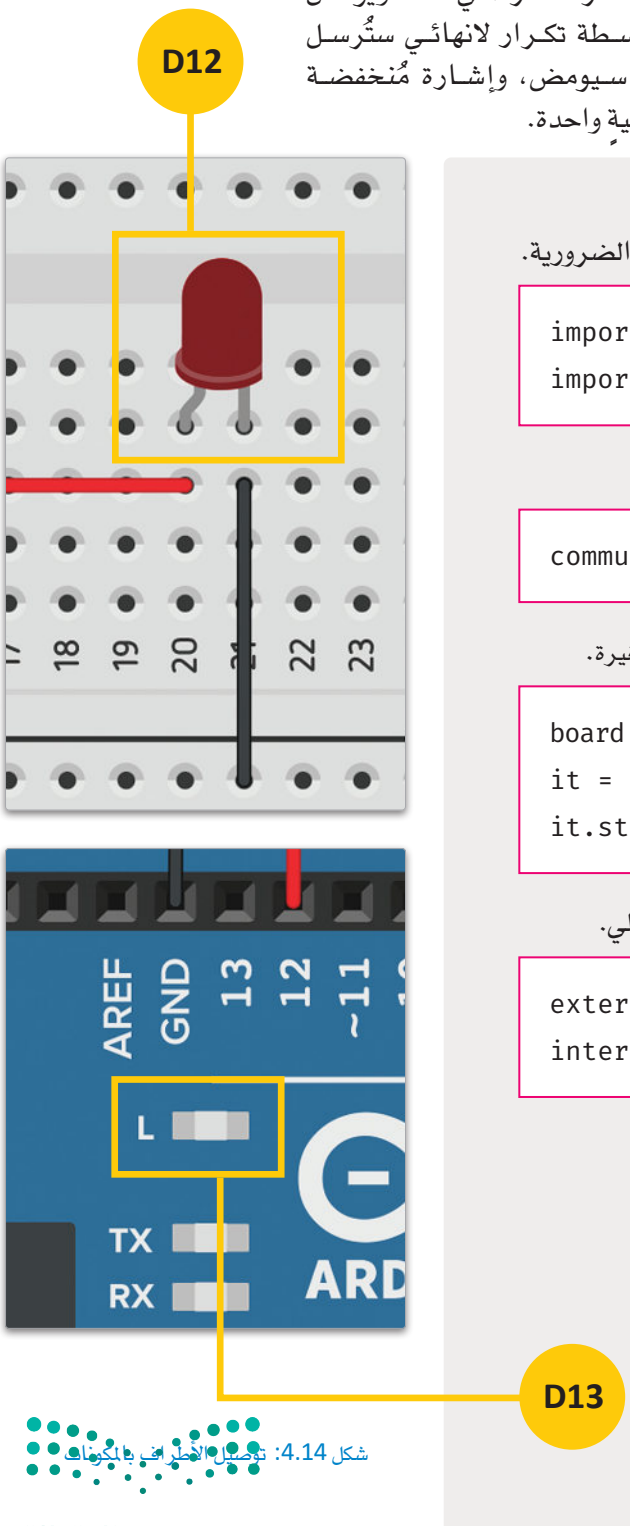
تمثل هذه الصورة شكل الدائرة الفعلية.



شكل 4.13: صورة الدائرة الفعلية

برمجة الأردوينو للوميض Programming the Arduino to Blink

ستبرمج الدايودين المشعين للضوء ليومضا واحدًا تلو الآخر، وذلك بفارق زمني ثانية واحدة. يُوصل الدايود المشع للضوء المُدمج في الأردوينو بالطرف الرقمي 13، ويُوصل الدايود المشع للضوء الخارجي بالطرف الرقمي 12. ثم بواسطة تكرار لانهائي ستُرسل إشارة مُرتفعة (1) HIGH إلى الدايود المشع للضوء الذي سيومض، وإشارة مُنخفضة (0) LOW إلى الدايود المشع للضوء الآخر. ستُعكس الإشارات بعد ثانيةٍ واحدة.



شكل 4.14: توصيل الأطراف بالمكونات

افتح باي تشارم، وأنشئ ملف بايثون جديد، واستورد المكتبات الضرورية.

```
import pyfirmata
import time
```

قم بإعداد منفذ الاتصال.

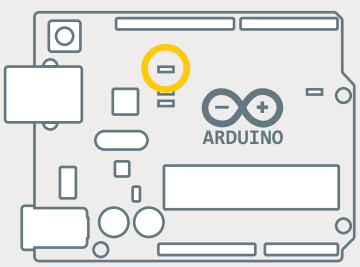
```
communication_port = 'COM4'
```

قم بإعداد الاتصال بين PyFirmata ولوحة توصيل الدوائر الصغيرة.

```
board = pyfirmata.Arduino(communication_port)
it = pyfirmata.util.Iterator(board)
it.start()
```

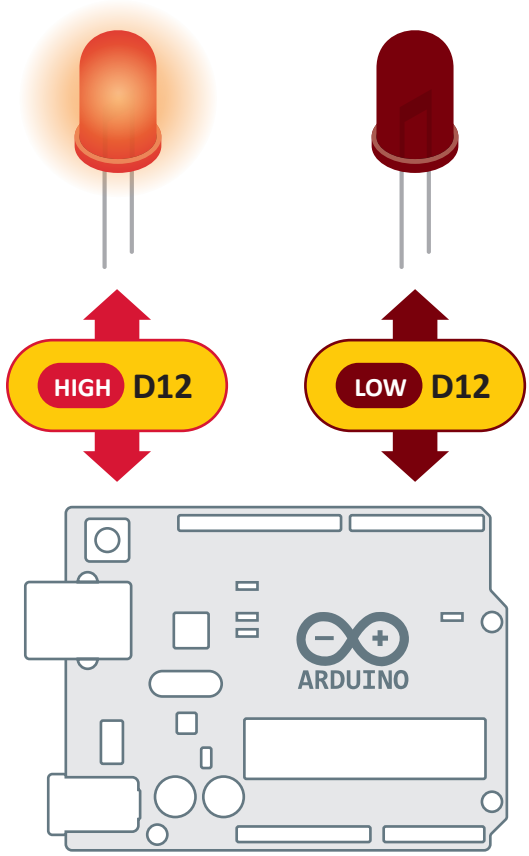
عيِّن الأطراف الخاصة بالدايود المشع للضوء الخارجي والداخلي.

```
external_led = board.get_pin("d:12:o")
internal_led = board.get_pin("d:13:o")
```

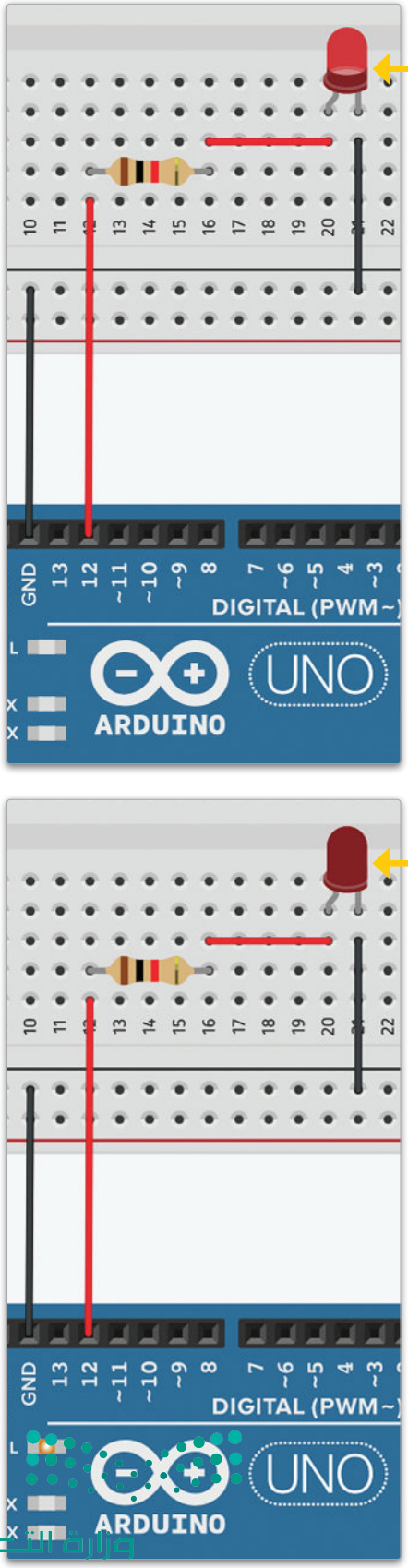


كتابة منطق الدائرة لتشغيل وميض الدايودات المشعة للضوء.

```
while True:
    external_led.write(1)
    internal_led.write(0)

    time.sleep(1)

    external_led.write(0)
    internal_led.write(1)

    time.sleep(1)
```



شكل 4.15: إرسال إشارة رقمية من الأردوينو إلى الأطراف



شكل 4.16: وميض الدايودات المشعة الضوء بالتناوب

البرنامج الكامل Complete Code

```
import pyfirmata
import time

communication_port = 'COM4'

board = pyfirmata.Arduino(communication_port)
it = pyfirmata.util.Iterator(board)
it.start()

external_led = board.get_pin("d:12:o")
internal_led = board.get_pin("d:13:o")

while True:
    external_led.write(1)
    internal_led.write(0)

    time.sleep(1)

    external_led.write(0)
    internal_led.write(1)

    time.sleep(1)
```

وزارة التعليم
Ministry of Education
2023 - 1445

تمرينات

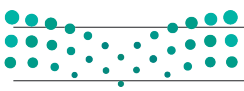
1

حدد الجملة الصحيحة والجملة الخاطئة فيما يلي:	صحيحة	خاطئة
1. يمكن برمجة جهاز تحكم الأردوينو الدقيق بواسطة لغة C++ فقط.	○	○
2. يُعدّ بروتوكول Firmata أحد بروتوكولات التشفير.	○	○
3. تُوظّف مكتبة PyFirmata باستخدام لغة البايثون فقط.	○	○
4. لإنشاء اتصال بين الأردوينو والحاسب، ستحتاج إلى تحميل مكتبة ServoFirmata.	○	○
5. تتعرف برمجة PyFirmata تلقائيًا على منفذ الاتصال الذي تستخدمه لوحة الأردوينو.	○	○
6. تستخدم الأطراف التناظرية تضمين عرض النبضة بدلًا من طريقة الإخراج القياسية.	○	○
7. يُمكن للدايودات المشعة للضوء الخارجية أن تضيء بإشارات رقمية وتناظرية.	○	○
8. يتصل الدايود المشع للضوء المُدمج داخل الأردوينو بالطرف الرقمي 10.	○	○
9. يعمل برنامج البايثون مع PyFirmata على جهاز تحكم الأردوينو الدقيق.	○	○
10. تتركز أهمية استخدام لوحة تجارب حقيقية في أنه إذا رُكّبت المكونات بشكل غير صحيح على اللوحة، فيمكن نقلها ببساطة إلى مكانها الصحيح على اللوحة.	○	○

2 صِفْ عمليّة إعداد بيئة تطوير الأردوينو، واذكر بيئات البرامج والأجهزة المطلوبة؟

3 ما مزايا برمجة الأردوينو بواسطة بايثون؟ علل إجابتك.

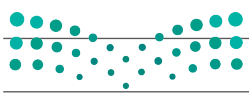
4 ما سلبيات التعامل مع بروتوكول Firmata ومكتبة PyFirmata معًا؟



5 ما وظيفة التعليمات البرمجية الآتية؟

```
pin = board.get_pin("a:4:p")
pin.write(0.75)
```

6 اذكر مثالين على مُستشعرات أو مُشغِّلات تعمل بصورة أفضل مع الإشارات الرقمية، ومثالين آخرين على مُستشعرات تعمل بصورة أفضل مع الإشارات التناظرية، آخذًا في اعتبارك المستشعرات والمشغلات التي تعلمتها.



رابط الدرس الرقمي

www.ien.edu.sa

الدرس الثاني

برمجة الأردوينو في البايثون

مشروع الحديقة الذكيّة بالأردوينو

Smart Garden with an Arduino

نظرًا للتغير المناخي في أنحاء الكرة الأرضية، فقد ازداد الطَلب على البستنة الذكية كطريقة للزراعة المُستدامة والقابلة للتطوير. أصبحت هناك حاجة ماسة لتلبية الاحتياجات الزراعية لدى العدد المتزايد من السكان، وبالتالي ضرورة وجود طرائق زراعة أكثر كفاءة مثل البَستنة الذكية. ستقوم بمحاكاة دائرة أردوينو تراقب حديقة ذكية، وتُرسل البيانات إلى منصة سحابية عبر الإنترنت. سيرسل الأردوينو البيانات باستمرار إلى التخزين السحابي، وعند استيفاء مجموعة معينة من الظروف المتعلقة بدرجة الحرارة والرطوبة، سيتم محاكاة تشغيل نظام للري. ستقوم أولًا بمحاكاة الدائرة في دوائر تينكركاد لاستعراض توصيلات الدائرة بوضوح، ثم ستستخدم المحاكاة لإرشادك في إنشاء الدائرة نفسها باستخدام جهاز تحكم أردوينو حقيقي.

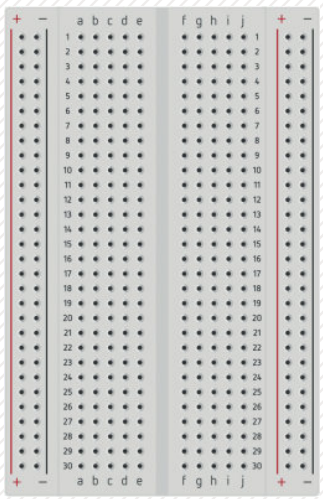
المنصة السحابية (Cloud platform):

المنصة السحابية هي خادم في مركز بيانات قائم على الإنترنت، يُمكّن خدمات البرامج والأجهزة من العمل معًا عن بُعد وفي توزيعات كبيرة.

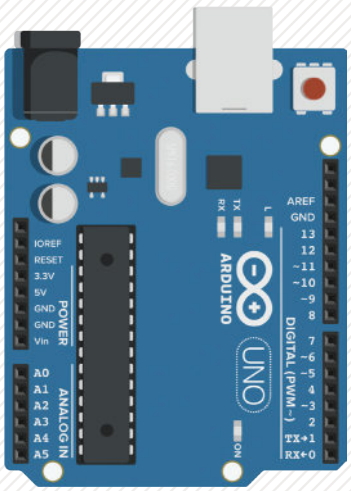
سوف تحتاج إلى المكونات الآتية:

- لوحة أردوينوأونو R3 (Arduino Uno R3).
- لوحة توصيل الدوائر الصغيرة.
- مُستشعر رطوبة التربة.
- مُستشعر درجة الحرارة.
- مُحرك تيار مُستمر.

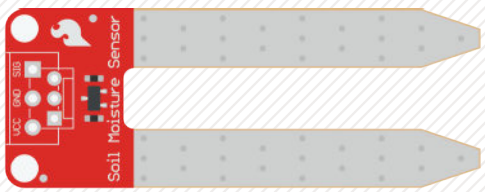
المُكونات التي ستستخدمها في هذا المشروع:



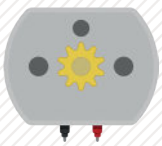
لوحة توصيل الدوائر الصغيرة
(Breadboard Small)



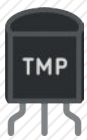
لوحة أردوينو أونو R3
(Arduino Uno R3)



مُستشعر رطوبة التربة
(Soil Moisture Sensor)



مُحرك تيار مستمر
(DC Motor)



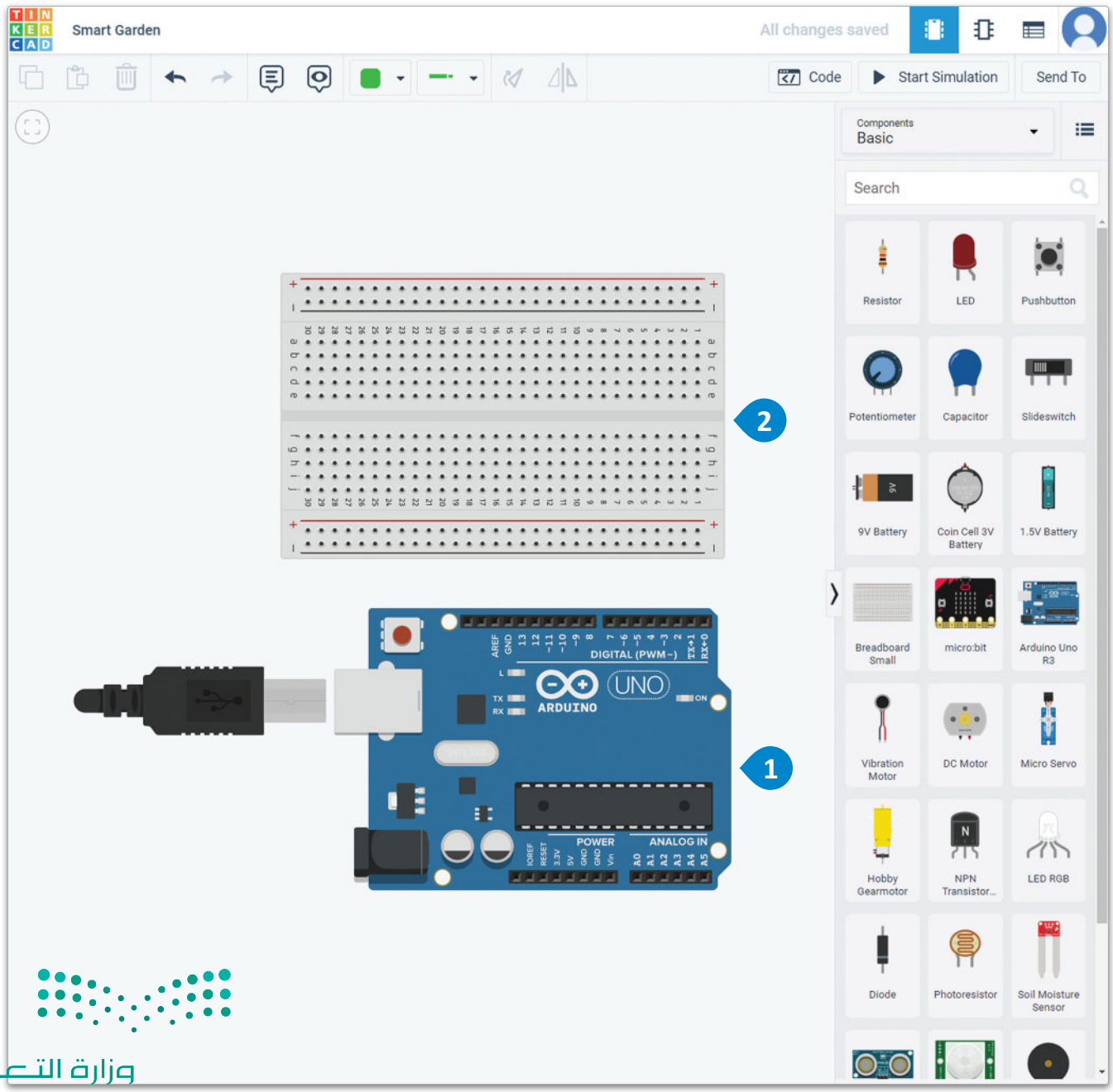
مُستشعر الحرارة TMP
(TMP Temperature Sensor)

شكل 4.17: مكونات مشروع الحديقة الذكية

لتحميل المكونات:

> ابحث عن Arduino Uno R3 (لوحة أردوينو أونو R3) من مكتبة Components (المكونات)، واسحبها وأفلتها في مساحة العمل. 1

> ابحث عن Breadboard Small (لوحة توصيل الدوائر الصغيرة) من مكتبة Components (المكونات)، واسحبها وأفلتها في مساحة العمل. 2



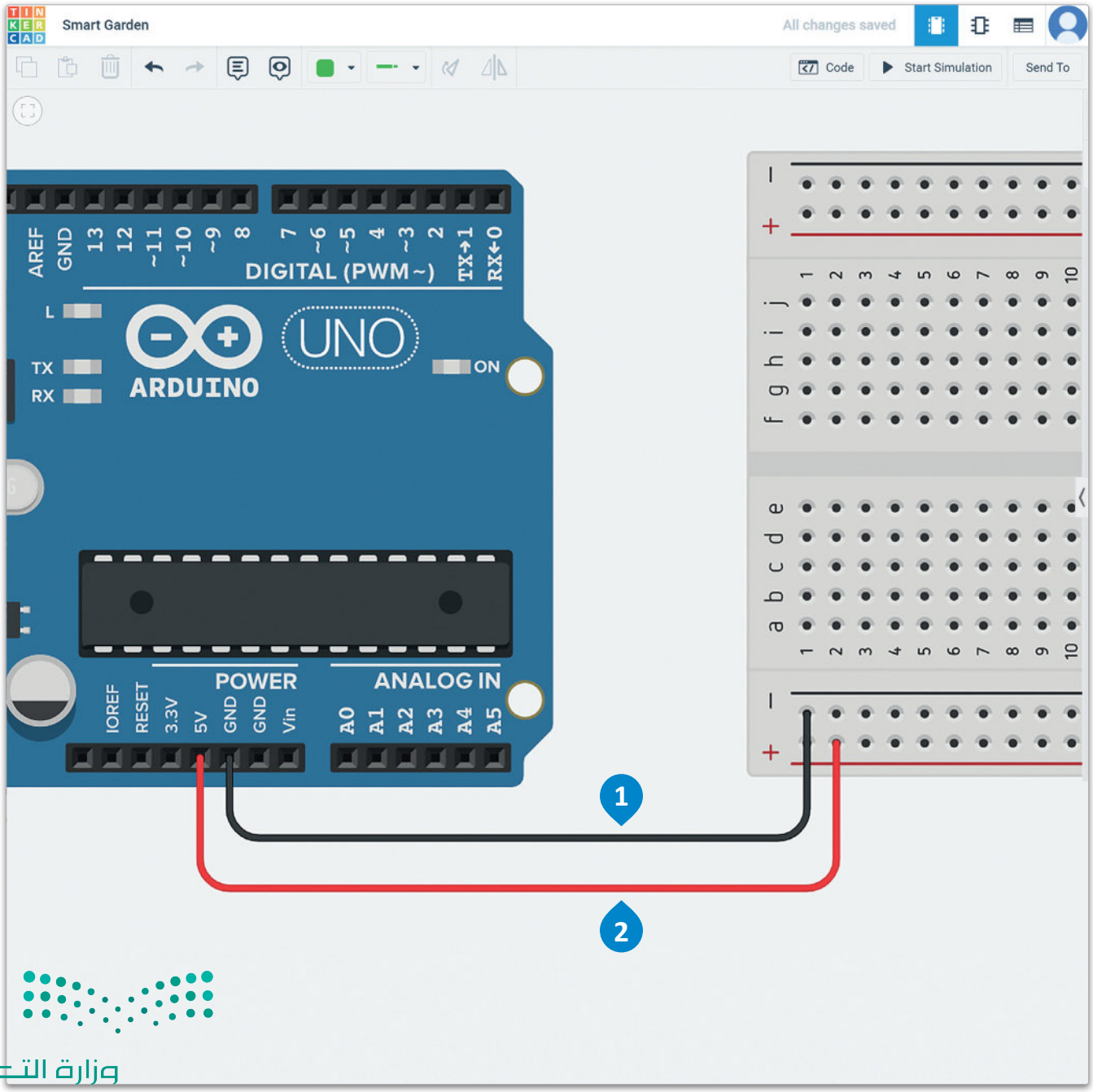
شكل 4.18: تحميل المكونات

ستقوم الآن بتوصيل لوحة الأردوينو بلوحة توصيل الدوائر الصغيرة.

لتوصيل لوحة الأردوينو بلوحة توصيل الدوائر الصغيرة:

> قم بتوصيل GND (الطرف الأرضي) للوحة الأردوينو بالعمود السالب من لوحة توصيل الدوائر الصغيرة، وغيِّر لون السلك إلى black (الأسود). 1

> قم بتوصيل طرف الجهد 5V (5 فولت) للوحة الأردوينو بالعمود الموجب من لوحة توصيل الدوائر الصغيرة وغيِّر لون السلك إلى red (الأحمر). 2



شكل 4.19: توصيل الأردوينو بلوحة توصيل الدوائر

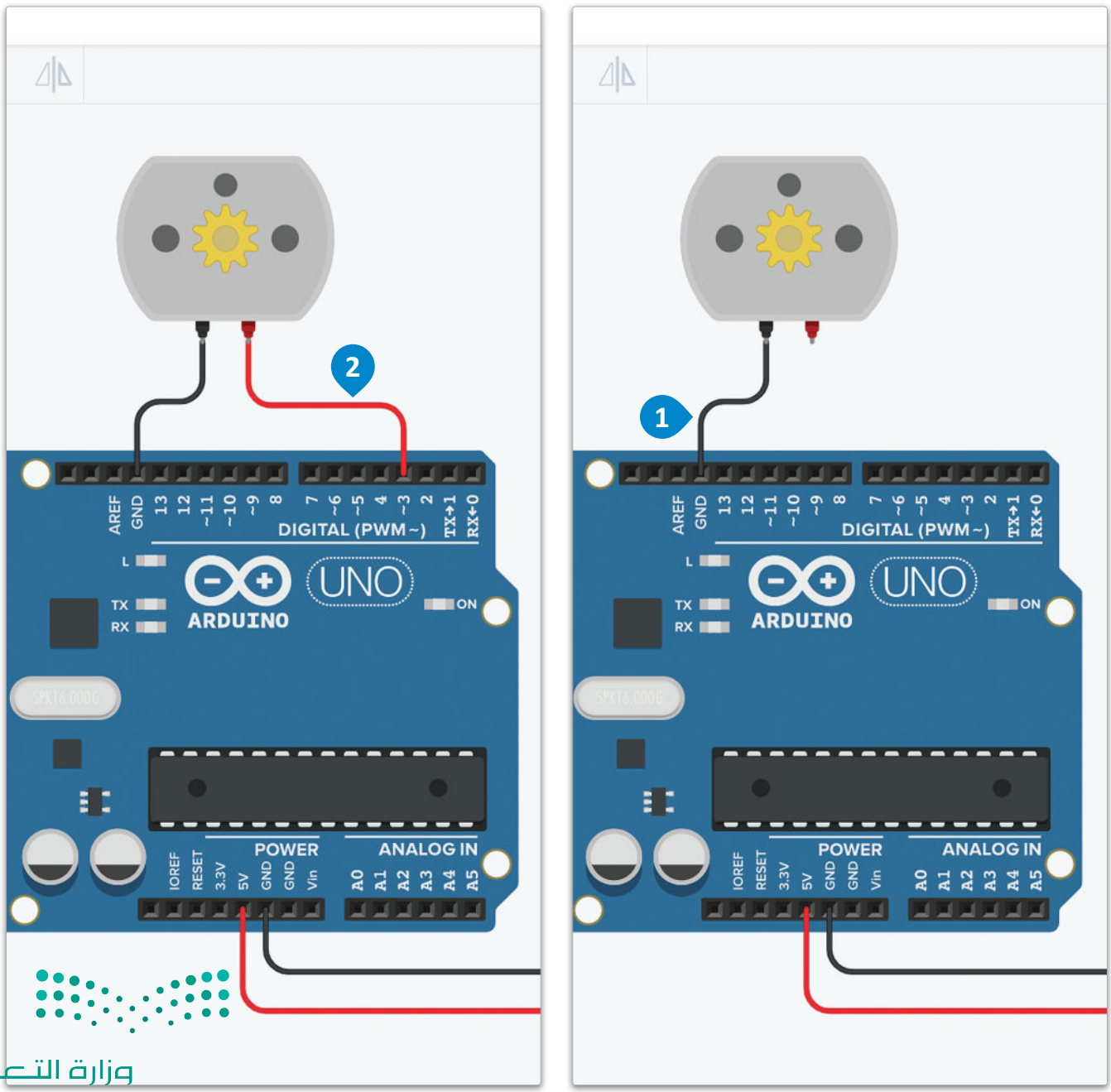
ستقوم الآن بتوصيل مُحرك التيار المستمر بمَنفذ رقمي في لوحة الأردوينو.

لتوصيل مُحرك التيار المستمر:

> قم بتوصيل GND (الطرف الأرضي) للوحة الأردوينو بالطرف 1 لمحرك التيار المستمر وغيِّر لون السلك إلى black (الأسود). 1

> قم بتوصيل Digital pin 3 (الطرف الرقمي 3) بـ Terminal 2 (الطرف 2) لمُحرك التيار المستمر، وغيّر لون السلك إلى red (الأحمر). 2

سيحاكي محرك التيار المستمر عملية فتح صمام نظام الري والذي سيُنشّط عند استيفاء مجموعة معينة من الشروط.

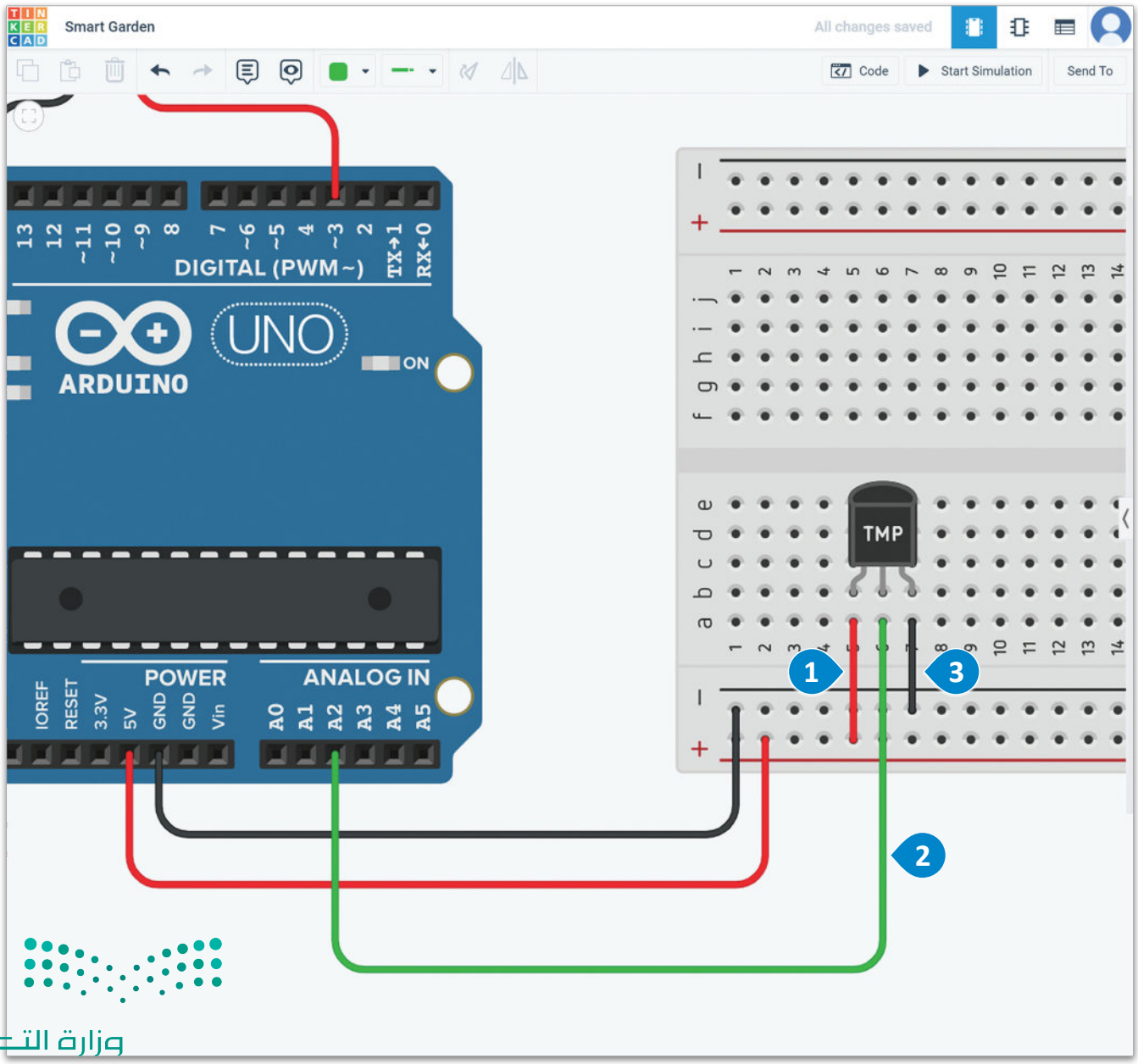


شكل 4.20: توصيل محرك التيار المستمر

ستقوم الآن بتوصيل مستشعر درجة الحرارة بمَنفذ تناظري في الأردوينو.

لتوصيل مُستشعر درجة الحرارة:

- قم بتوصيل طرف Power (الطاقة) لمُستشعر درجة الحرارة بالعمود الموجب من لوحة توصيل الدوائر الصغيرة وغيِّر لون السلك إلى red (الأحمر). 1
- قم بتوصيل طرف مخرج Vout (الجهد) لمُستشعر درجة الحرارة بالطرف التناظري A2 في لوحة الأردوينو، وغيِّر لون السلك إلى green (الأخضر). 2
- قم بتوصيل GND (الطرف الأرضي) لمُستشعر درجة الحرارة بالعمود السالب من لوحة توصيل الدوائر الصغيرة وغيِّر لون السلك إلى black (الأسود). 3



شكل 4.21: توصيل مستشعر درجة الحرارة

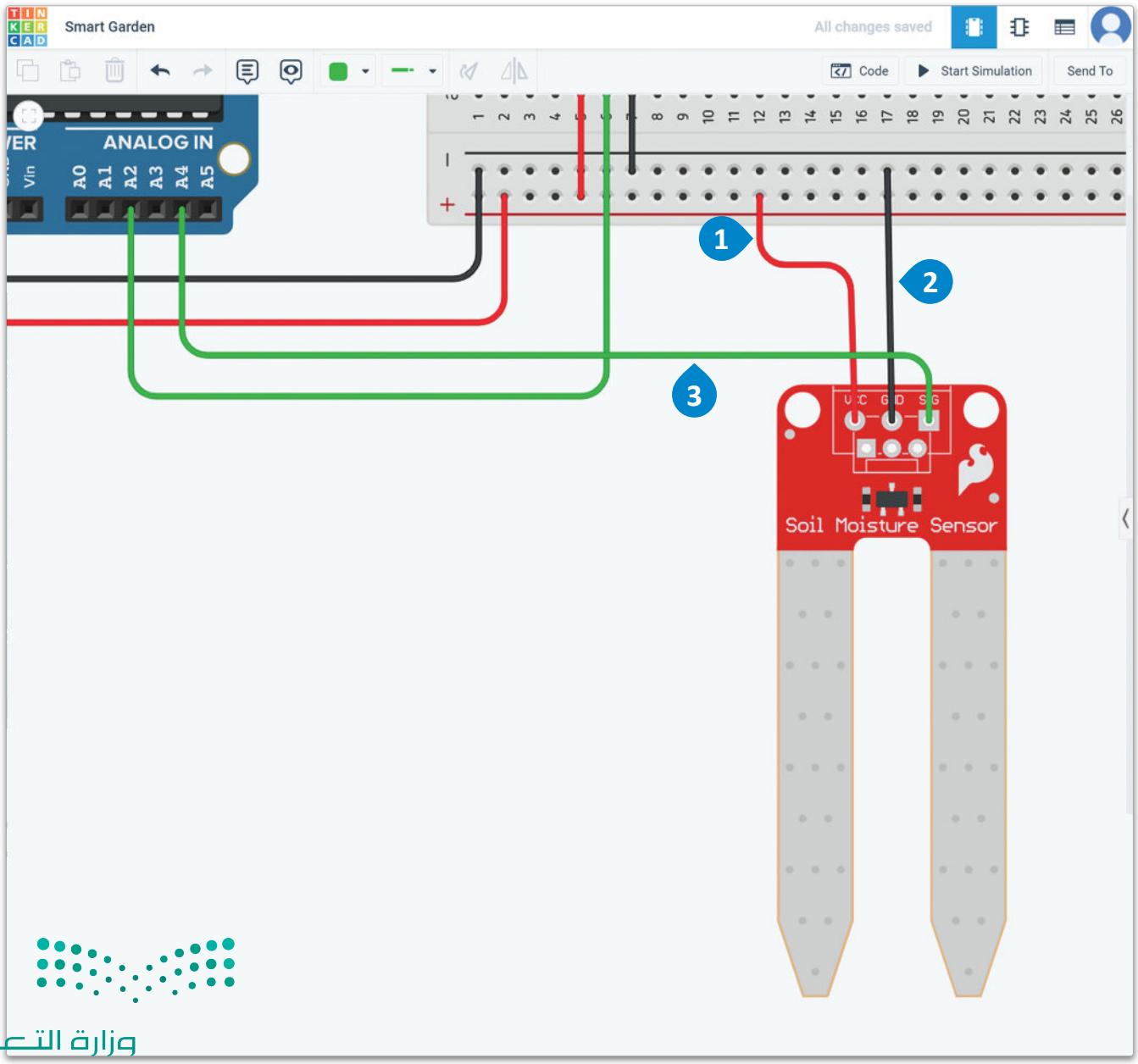
ستقوم في الختام بتوصيل مستشعر رطوبة التربة بمنفذ تناظري في الأردوينو.

لتوصيل مُستشعر رطوبة التربة :

> قم بتوصيل طرف Power (الطاقة) لمُستشعر رطوبة التربة بالعمود الموجب من لوحة توصيل الدوائر الصغيرة وغيِّر لون السلك إلى red (الأحمر). 1

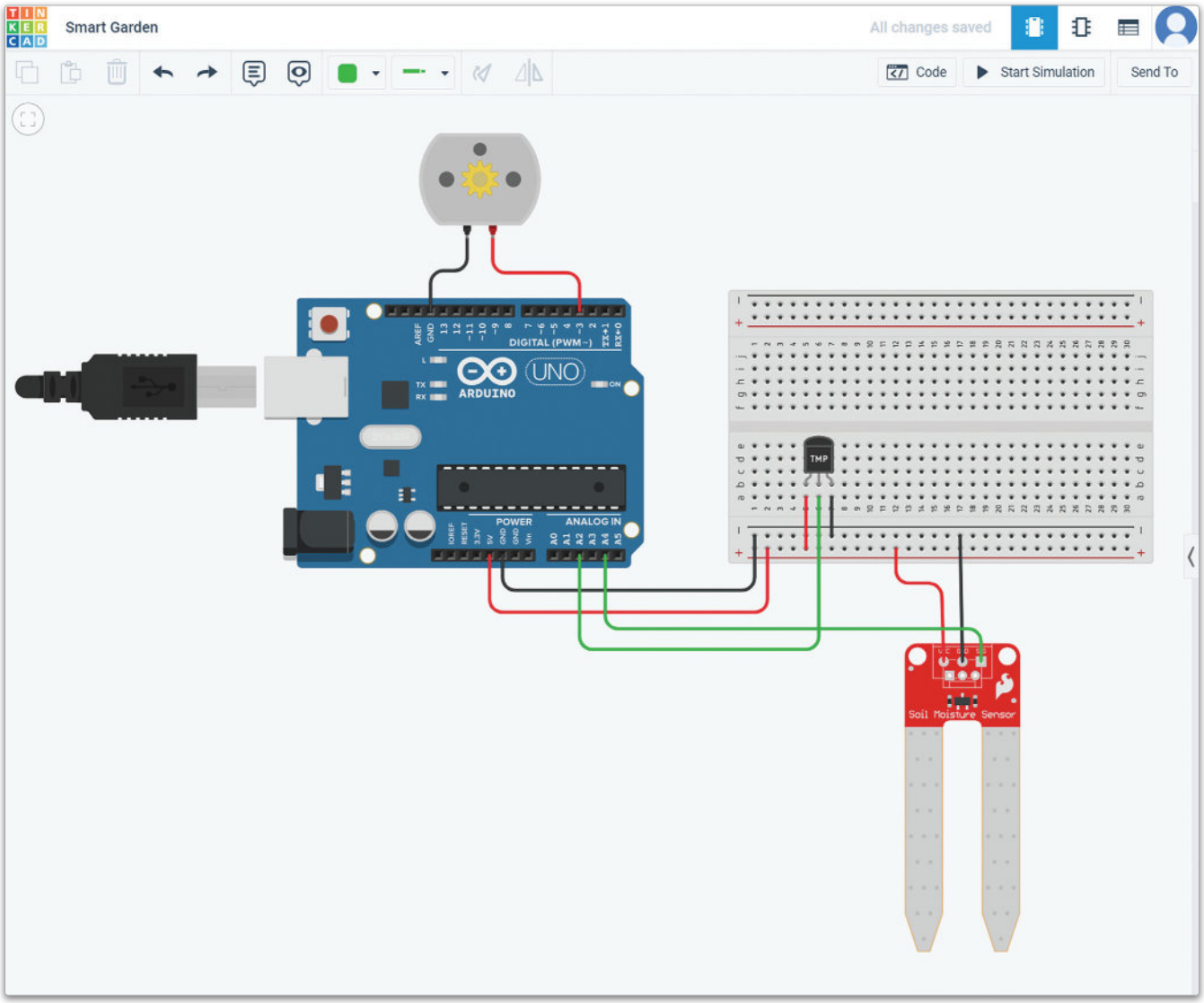
> قم بتوصيل GND (الطرف الأرضي) لمُستشعر رطوبة التربة بالعمود السالب من لوحة توصيل الدوائر الصغيرة وغيِّر لون السلك إلى black (الأسود). 2

> قم بتوصيل طرف Signal (الإشارة) لمُستشعر رطوبة التربة بالطرف التناظري A4 في لوحة الأردوينو، وغيِّر لون السلك إلى green (الأخضر). 3

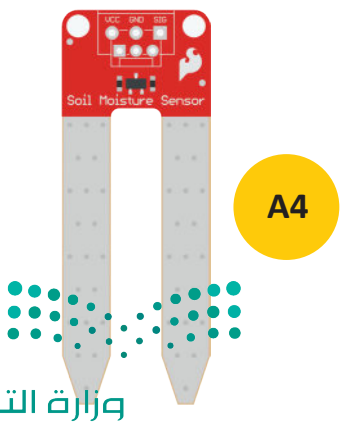


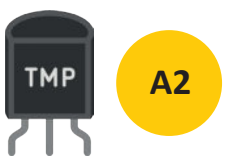
شكل 4.22: توصيل مُستشعر رطوبة التربة

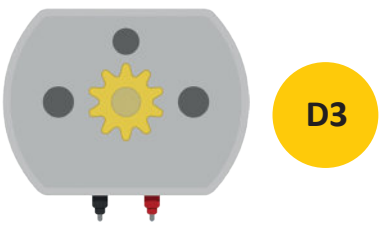
الدائرة بصورتها النهائية Complete Circuit



تتصل المكونات بالأطراف الآتية:



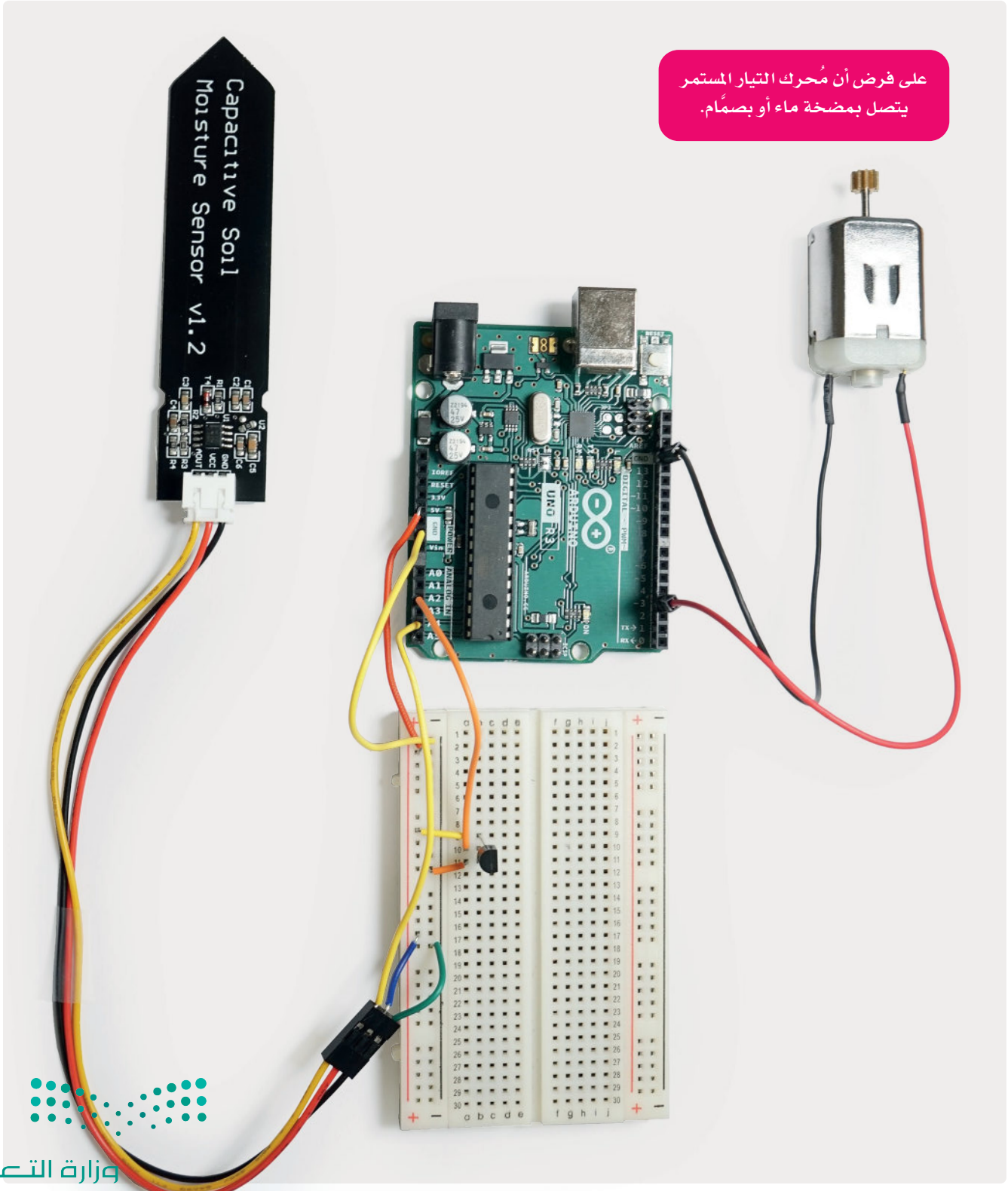




شكل 4.23: توصيل الأطراف بالمكونات

الدائرة المادية Physical Circuit

تُمثِّل هذه الصورة الشكل الذي تبدو عليه الدائرة الحقيقية.



شكل 4.24: صورة الدائرة بمكوناتها المادية

برمجة مُستشعرات ومُحرك ري الحديقة الذكية في الأردوينو

Programming the Arduino Smart Garden Sensors and Motor

ستقوم الآن ببرمجة الأردوينو لقراءة قِيم أطراف مستشعرات درجة الحرارة ورطوبة التربة. عند الوصول إلى تكوينٍ مُحددٍ لقيم درجة الحرارة والرطوبة، سيُنشّط محرك التيار المستمر وذلك باستخدام الدالة، ليعمل لمدة 5 ثوانٍ، ثم يتوقف، وذلك في محاكاةٍ لعملية المراقبة والري التلقائيين للنباتات في الحديقة الذكية.

افتح باي تشارم، وأنشئ ملف بايثون جديد، واِسْتَدْعِ المكتبات المطلوبة.

```
import pyfirmata
import time
```

قم بتكوين مَنفذ الاتصال والأطراف المطلوبة.

```
communication_port = 'COM4'
dc_motor_pin = board.get_pin('d:3:o')
temperature_sensor_pin = board.get_pin('a:2:i')
moisture_sensor_pin = board.get_pin('a:4:i')
```

قم بإعداد الاتصال بين PyFirmata واللوحة.

```
board = pyfirmata.Arduino(communication_port)
it = pyfirmata.util.Iterator(board)
it.start()
```

نفِّذ الدالة الآتية للتحكم في مُحرك التيار المستمر.

```
def water_plant(dc_motor_pin):
    print("--- Watering plant ---")
    dc_motor_pin.write(1)
    time.sleep(5)
    dc_motor_pin.write(0)
```

ترسل هذه الدالة إشارة رقمية مُرتفعة (HIGH) لمدة 5 ثوانٍ إلى محرك التيار المستمر، ثم ترسل إشارة رقمية مُنخفضة (LOW) لإيقاف حركته.

أنشئ تكرارًا لا نهائيًا، وقم بكتابة الأوامر البرمجية أدناه.

```
while True:
    # write your code here
```

اقرأ قيم إدخال درجة الحرارة والرطوبة غير المُعالجة التي تتلقاها من الأطراف التناظرية.

```
temperature_value = temperature_sensor_pin.read()
moisture_value = moisture_sensor_pin.read()
```

تحقق مِما إذا كانت قيم الإدخال من الأطراف فارغة. يُنفّذ منطق البرنامج أدنى هذا الشرط.

```
if (temperature_value is not None) and (moisture_value is not None):
```

تُرسل مُستشعرات جهاز التحكم الدقيق قيمًا فارغة في بعض الأحيان، ولذلك علينا إضافة طريقة تحقق لتجنب حدوث الأخطاء في البرنامج.

أنشئ المتغيرات الآتية التي تُعين قيم المُدخلات غير المُعالجة لقيم درجة الحرارة والرطوبة المناسبة باستخدام الصِيغ الرياضية. بالنسبة لدرجة الحرارة، تُستخدم أول 3 أرقام عشرية للإشارة التناظرية القادمة من مُستشعر درجة الحرارة، ثم تُحوّل هذه القيمة إلى الجهد المُطبق بواسطة مستشعر درجة الحرارة على طرف الإشارة. تقوم الخطوة الآتية بتحويل الجهد إلى درجات مئوية، وفق الصيغة الخاصة بهذا النوع من المُستشعرات.

```
temperature_value  = float(temperature_value) * 1000
voltage = (temperature_value / 1024) * 5
temperature = (voltage - 0.5) * 100
moisture = (1.0 - float(moisture_value)) * 100
```

تحويل درجة الحرارة إلى درجات مئوية، وتحويل مستوى الرطوبة إلى نسبة مئوية.

إدخال شروط درجة الحرارة والرطوبة اللازمة لري النبات.

```
if (temperature >= 24.0) and (moisture <= 40.0):
     water_plant(dc_motor_pin)
```

إذا كانت درجة الحرارة أعلى من 24 درجة مئوية، وكان مستوى رطوبة التربة أقل من 40٪، فستُنشّط آلية ري النبات.

إنشاء رسائل تقارير تُعرض من خلال الواجهة الطرفية عند تشغيل البرنامج وجمع البيانات.

```
temperature_report = "Temperature : " + str(temperature) + " C"
moisture_report = "Moisture : " + str(round(moisture, 2)) + "%"

print(temperature_report)
print(moisture_report)
```

البرنامج بشكله النهائي Complete Code

```
import time
import pyfirmata

board = pyfirmata.Arduino('COM4')
it = pyfirmata.util.Iterator(board)
it.start()

dc_motor_pin = board.get_pin('d:3:o')
```

```
temperature_sensor_pin = board.get_pin('a:2:i')
moisture_sensor_pin = board.get_pin('a:4:i')

def water_plant(dc_motor_pin):
    print("--- Watering plant ---")
    dc_motor_pin.write(1)
    time.sleep(5)
    dc_motor_pin.write(0)

while True:
    temperature_value = temperature_sensor_pin.read()
    moisture_value = moisture_sensor_pin.read()

    if (temperature_value is not None) and (moisture_value is not None):

        temperature_value  = float(temperature_value) * 1000
        voltage = (temperature_value / 1024) * 5
        temperature = (voltage - 0.5) * 100
        moisture = (1.0 - float(moisture_value)) * 100

        if (temperature >= 24.0) and (moisture <= 40.0):
            water_plant(dc_motor_pin)

        temperature_report = "Temperature : " + str(temperature) + " C"
        moisture_report = "Moisture : " + str(round(moisture, 2)) + "%"

        print(temperature_report)
        print(moisture_report)

        time.sleep(10)
```

تمرينات

1 هل تعتقد أن استخدام الإخراج التناظري بواسطة تضمين عرض النبضة (PWM) أكثر كفاءةً في التحكم بمحرّك التيار المستمر؟ فسِّر إجابتك أدناه.

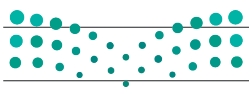
2 بالنظر إلى عدد أطراف الإدخال والإخراج في الأردوينو والمايكروبت (micro：bit)، أيٌّ منهما يُعدُّ أفضل كجهاز تحكم دقيق في نظام الحديقة الذكية؟ اعرض أفكارك أدناه.

3 اشرح سبب معالجة قيم إدخال المستشعر المأخوذة من الأطراف التناظرية باستخدام صيغة رياضية مختلفة اعتمادًا على المستشعر؟

4 ما أهمية إجراء عمليات فحص البيانات المُجمَّعة لمَعرفة ما إذا كانت المُستشعرات ترسل بيانات فارغة أو تالفة؟

5 هل تعتقد أنّ استخدام مُحرك السيرفو (servo motor) هو الأفضل لري النباتات بشكل أكثر دقة وكفاءة. اعرض أفكارك أدناه.

6 قم بإعادة كتابة برنامج البايثون باستخدام أوامر الطباعة لإنشاء تقرير حول قراءات البيئة المحيطة كل 30 ثانية.



رابط الدرس الرقمي

www.ien.edu.sa

الدرس الثالث

التفاعل مع خدمات الويب السحابية

التعامل مع خدمات الويب Interacting with Web Services

ستتابع العمل على مشروع الدرس السابق وستعمل على تطويره في هذه المرحلة لتُرسل البيانات عبر خدمة الويب السحابية من منصة الحوسبة السحابية Binary IoT Cloud. تتيح هذه المَنصة التعليمية السحابية عبر الإنترنت عرض البيانات عن البيئة المحيطة المُجمعة بواسطة دائرة الأردوينو التي أنشأتها في الدرس السابق.

شكل 4.25: منصة الحوسبة السحابية Binary IoT Cloud مع بيانات مُجمَّعة من الأردوينو

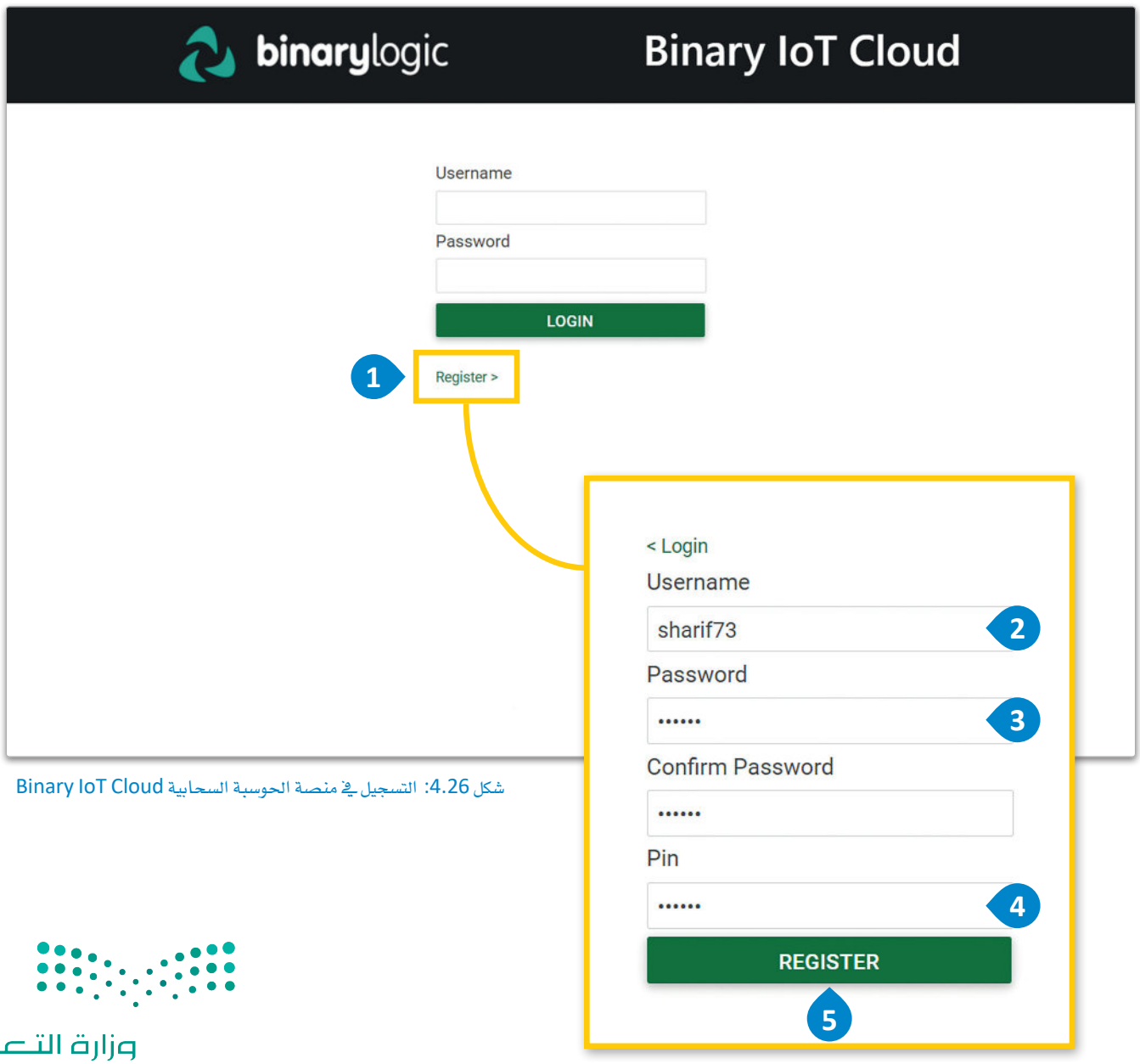
ستحتاج في البداية إلى التسجيل في المنصة السحابية وإنشاء مستخدم جديد.

للتسجيل في منصة الحوسبة السحابية Binary IoT Cloud عبر الإنترنت:

> انتقل إلى منصة الحوسبة السحابية Binary IoT Cloud من خلال الموقع الإلكتروني: https://ksa-iot.azurewebsites.net/Login.aspx، ومن صفحة الترحيب اضغط على Register (تسجيل). 1

> اكتب في Username اسم المستخدم، 2 وفي Password كلمة مرور من اختيارك، 3 وفي PIN (رقم التعريف الشخصي) اكتب: 174563. 4

> اضغط على زر Register (تسجيل). 5



شكل 4.26: التسجيل في منصة الحوسبة السحابية Binary IoT Cloud

استدعاء واجهة برمجة تطبيقات الويب باستخدام البايثون
Calling a Web API with Python

ستقوم بإنشاء كائن **جسون** (JSON) يحتوي على بيانات البيئة المحيطة، ويُرسلها إلى منصة الحوسبة السحابية Binary IoT Cloud من خلال دالة تتفاعل مع **واجهة برمجة تطبيقات الويب** (Web API).

قم بتثبيت حزمة **طلبات البايثون** (Python requests) من خلال نظام إدارة **الحزم** (pip). افتح الواجهة **الطرفية** (Terminal) في مُجلد العمل الخاص بك، وأدخل الأمر الآتي:

```
pip install requests
```

في بداية برنامج بايثون الخاص بك، استدعِ حزمة **الطلبات** (requests) بالسطر البرمجي الآتي:

```
import requests
```

احصل على التاريخ والوقت الحالي، وحولهما إلى نص باستدعاء الوحدة **القياسية** (datetime) بالطريقة الآتية:

```
from datetime import datetime

date_time = str(datetime.now())
```

أنشئ متغيرين باسم username (اسم المستخدم) و password (كلمة المرور)، بحيث يتطابقان مع البيانات التي استخدمتها مُسبقًا للتسجيل في منصة الحوسبة السحابية Binary IoT Cloud.

```
username = "your_username_here"
password = "your_password_here"
```

أنشئ دالة جديدة باسم send_data() تستقبل المُعامِلات الآتية:

```
def send_data(username,password,temperature,moisture):
```

واجهة برمجة تطبيقات الويب (Web API):

واجهة برمجة تطبيقات الويب هي نقطة وصل تسمح لبرنامج ما بالوصول إلى خدمة من برنامج آخر موجود على خادم على شبكة الإنترنت.

جسون أو ترميز الكائنات باستعمال جافا سكريبت (JavaScript Object Notation - JSON):

هي نوع مفتوح لتنسيق البيانات يُستخدم لنقل البيانات بين الخدمات. الكائنات في جسون هي أزواج تتكون من (مفتاح – قيمة) يُمكنها تخزين أنواع البيانات مثل السلاسل النصية، والأعداد الصحيحة، والأعداد العشرية، والمصفوفات وكائنات أخرى.

شكل 4.27: تطبيق يتصل بمنصة سحابية من خلال واجهة برمجة تطبيقات الويب

وزارة التعليم
Ministry of Education
2023 - 1445

أنشئ متغيرًا جديدًا باسم api_url بواسطة الرابط الآتي حيث يعد نقطة وصل واجهة برمجة تطبيقات الويب للمنصة السحابية:

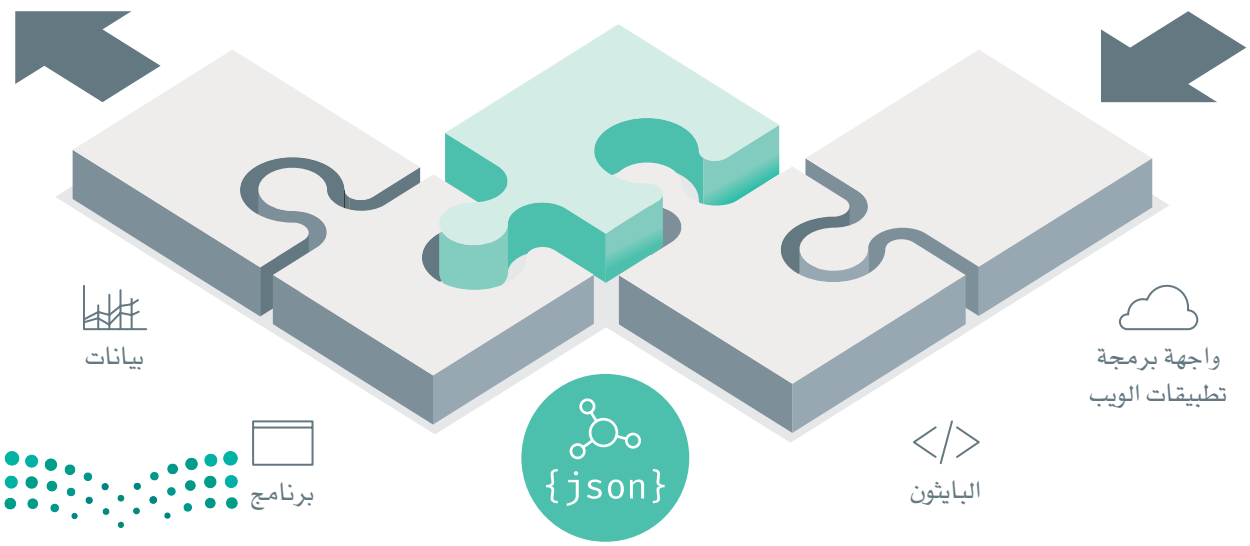
```
api_url = "https://ksa-iot-api.azurewebsites.net/api/readings"
```

أنشئ كائنًا بمتغيرات الدالة التي ستُمرّر إلى واجهة برمجة تطبيقات الويب كجزء الطلب الرئيس:

```
reading = {
      "username": username,
      "password": password,
      "temperature": temperature,
      "moisture": moisture,
      "datetime": date_time
}
```

استدعِ واجهة برمجة تطبيقات الويب باستخدام دالة ( )request.post والتي تحتوي على مُعامِلات كلٍ من **رابط واجهة برمجة تطبيقات الويب** (API URL) و**كائن جسون الرئيس** (JSON body object):

```
response = requests.post(api_url, json=reading)
```



شكل 4.28: جسون هو نوع تنسيق بيانات مفتوح لنقل البيانات بين الخدمات

تُعرض هنا كافة التعليمات البرمجية للدالة send_data( ) :

```
def send_data(username, password, temperature, moisture):
    api_url = "https://ksa-iot-api.azurewebsites.net/api/readings"
    date_time = str(datetime.now())

    reading = {
        "username": username,
        "password": password,
        "temperature": temperature,
        "moisture": moisture,
        "datetime": date_time
    }
    response = requests.post(api_url, json=reading)
```

أضف دالة send_data( ) في البرنامج الرئيس لإرسال البيانات عن البيئة المحيطة التي جُمعت كل 30 ثانية.

```
send_data(username, password, temperature, moisture)
time.sleep(30)
```

البرنامج بشكله النهائي Complete Code

```
from datetime import datetime

import time
import requests
import pyfirmata

board = pyfirmata.Arduino('COM4')
```

```
it = pyfirmata.util.Iterator(board)
it.start()

dc_motor_pin = board.get_pin('d:3:o')
temperature_sensor_pin = board.get_pin('a:2:i')
moisture_sensor_pin = board.get_pin('a:4:i')

username = "your_username_here"
password = "your_password_here"

def send_data(username, password, temperature, moisture):
    api_url = "https://ksa-iot-api.azurewebsites.net/api/readings"
    date_time = str(datetime.now())

    reading = {
        "username": username,
        "password": password,
        "temperature": temperature,
        "moisture": moisture,
        "datetime": date_time
    }

    response = requests.post(api_url, json=reading)

def water_plant(dc_motor_pin):
    print("--- Watering plant ---")
    dc_motor_pin.write(1)
    time.sleep(5)
    dc_motor_pin.write(0)

while True:
    temperature_value = temperature_sensor_pin.read()
```

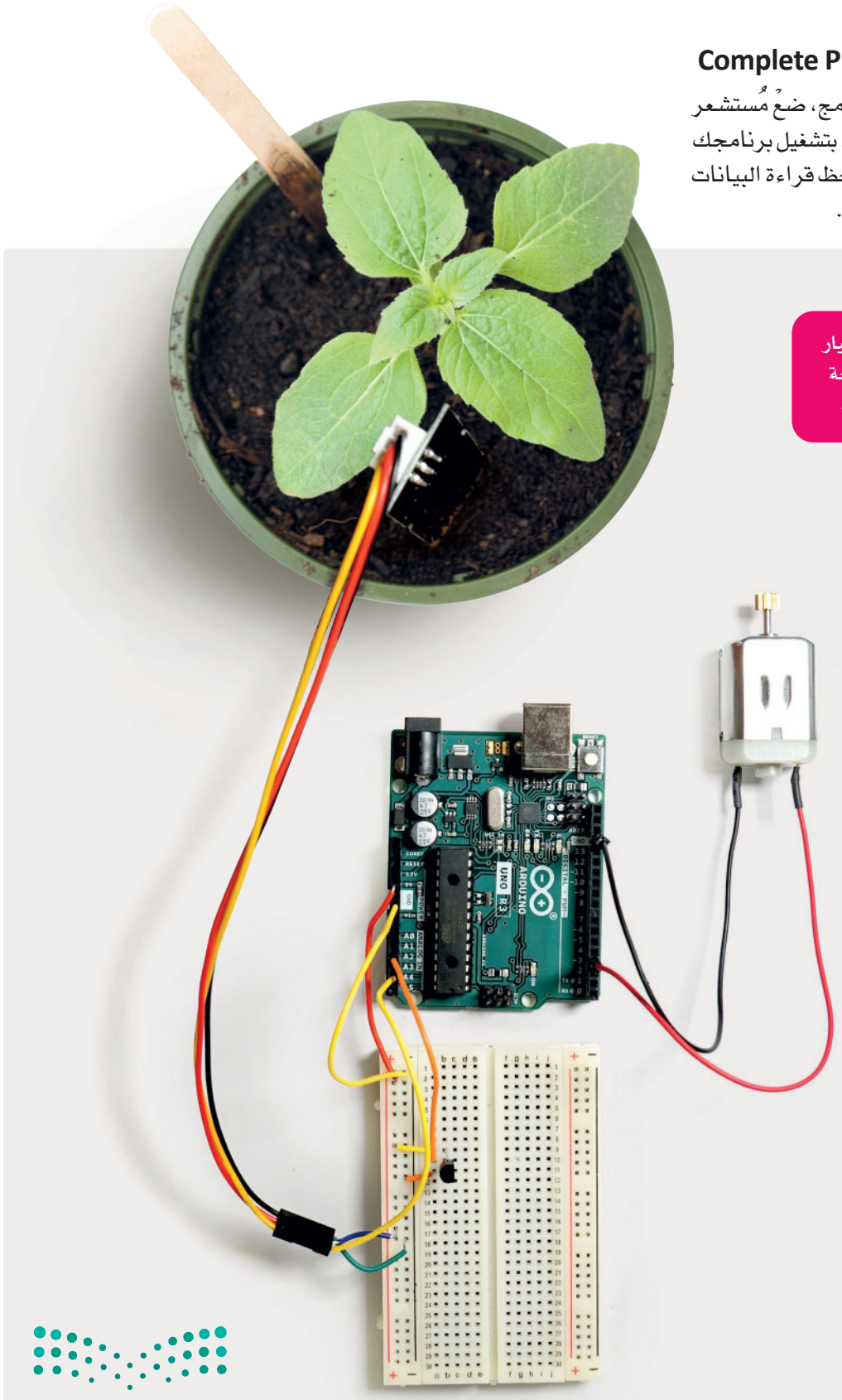
```
moisture_value = moisture_sensor_pin.read()

if (temperature_value is not None) and (moisture_value is not None):

    temperature_value  = float(temperature_value) * 1000
    voltage = (temperature_value / 1024) * 5
    temperature = (voltage - 0.5) * 100
    moisture = (1.0 - float(moisture_value)) * 100

    if (temperature >= 24.0) and (moisture <= 40.0):
        water_plant(dc_motor_pin)

    temperature_report = "Temperature : " + str(temperature) + " C"
    moisture_report = "Moisture : " + str(round(moisture, 2)) + "%"

    print(temperature_report)
    print(moisture_report)

    send_data(username, password, temperature, moisture)

    time.sleep(30)
```

شكل 4.29: تواصل الأردوينو مع الحاسب الشخصي ومَنصة الحوسبة السحابية Binary IoT Cloud

استكمال المشروع Complete Project

بعد الانتهاء من كتابتك للبرنامج، ضعْ مُستشعر رطوبة التربة في تربة النبتة، وقم بتشغيل برنامجك في بايثون من حاسوبك، وستلاحظ قراءة البيانات المُخرجة من البيئة إلى الحاسب.

افترض أن محرك التيار المستمر يتصل بمضخة ماء أو بصمام للري.



شكل 4.30: الحديقة الذكية باستخدام الأردوينو

عرض بيانات الحديقة الذكية Viewing the Smart Garden Data

قم بتشغيل برنامجك بأكمله في بايثون، ودَعْهُ يعمل لبضع دقائق ليجمع بعض البيانات التي سيتمّ تحميلها بعد ذلك إلى منصة الحوسبة السحابية Binary IoT Cloud.

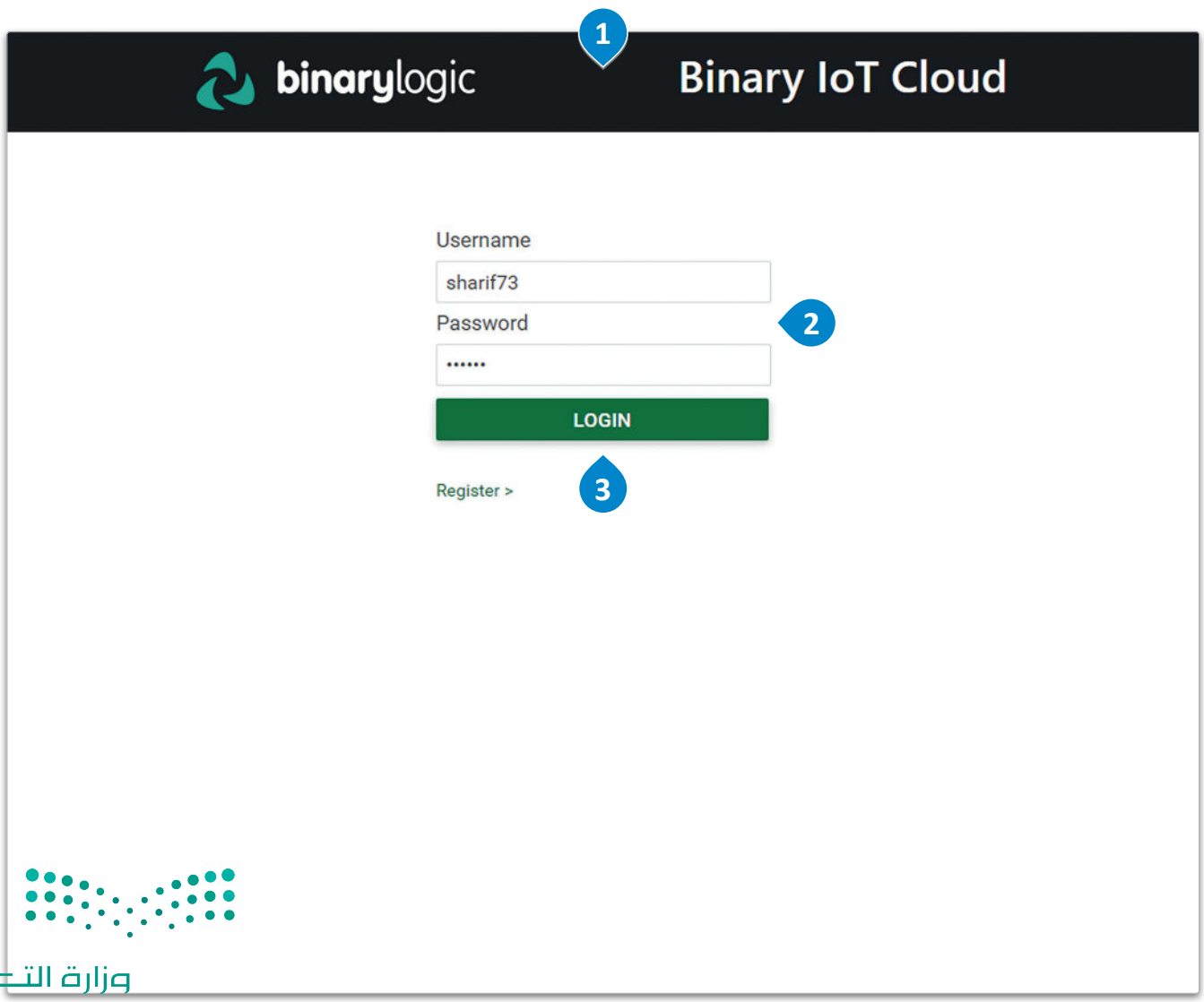
ولتتمكّن من عرض البيانات المجمّعة حول بيئتك النّباتيّة؛ سجّل دخولك إلى المنصّة باستخدام بياناتك.

لتسجيل الدخول إلى منصة الحوسبة السحابية Binary IoT Cloud عبر الإنترنت:

> اذهب إلى منصة الحوسبة السحابية Binary IoT Cloud عبر الإنترنت من الرابط: https://ksa-iot.azurewebsites.net/Login.aspx. 1

> أدخل Username (اسم المستخدم) و Password (كلمة المرور) التي أنشأتها سابقًا. 2

> اضغط على زر Login (تسجيل الدخول). 3



شكل 4.31: تسجيل الدخول إلى منصة الحوسبة السحابية Binary IoT Cloud عبر الإنترنت

شكل 4.32: عرض قراءات البيئة في منصة الحوسبة السحابية Binary IoT Cloud عبر الإنترنت

قيم مفصولة بفواصل (Comma Separated Values - CSV)

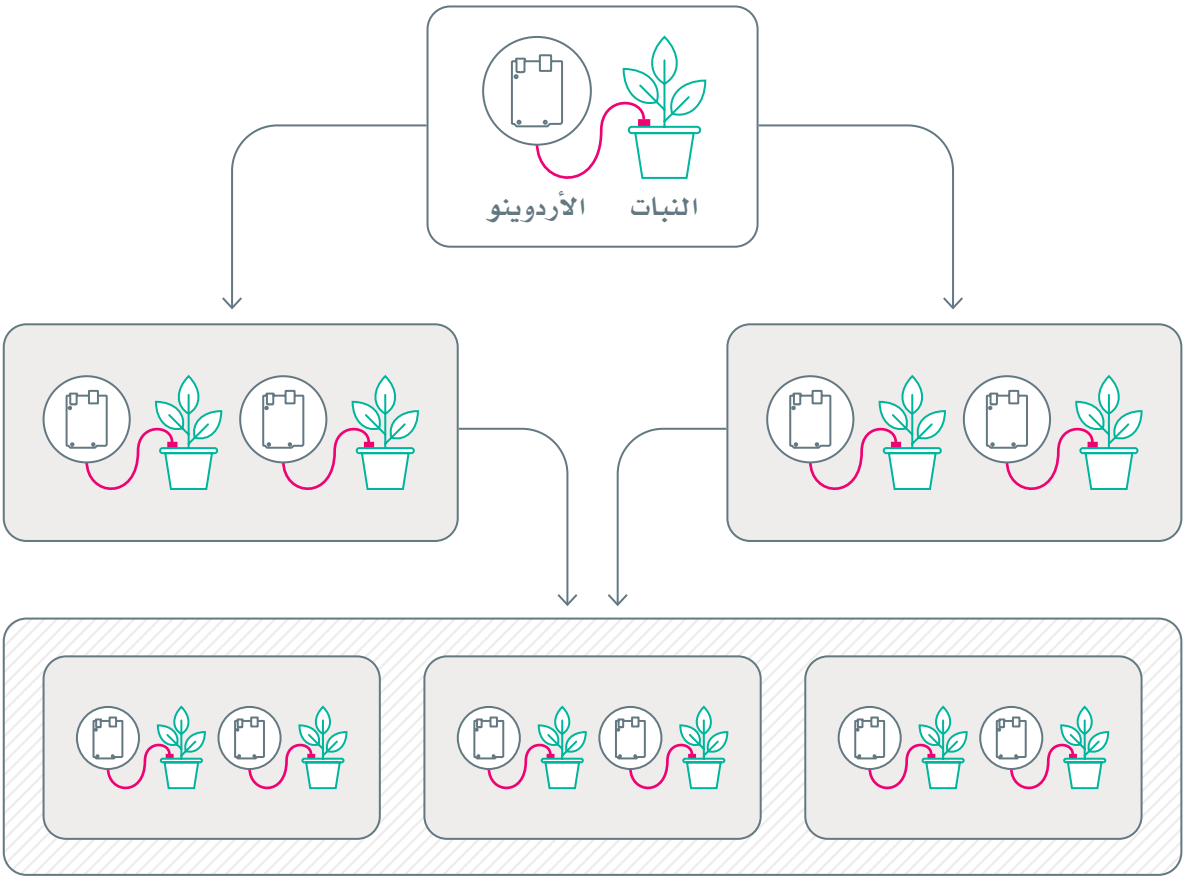
هو ملف نصي يتم الفصل بين القيم المُدرجة فيه بالفواصل، ويُمثِّل كل سطر في الملف سِجل بيانات.

استخدام بيانات المُستشعرات ومجموعات البيانات الضخمة لاتخاذ القرارات وفق تلك البيانات المُجمعة

Using Sensor Data and Big Datasets for Data-Driven Decision Making

لقد كانت مراقبة الحديقة الذكية هي محور المشروع السابق، وهو يُشكِّل مثالًا مُصغرًا على عملية المراقبة الذكية في الزراعة، حيث يتكون من جهاز واحد فقط يجمع البيانات ويرسلها إلى منصة عبر الإنترنت، ولكنّ وجود حديقة كبيرة يتطلب المزيد من الأجهزة لتكون المراقبة شاملة لكافة النباتات والبيئات المختلفة، حيث يتطلب كل نوع وصنف من النباتات إعداد بيئة مختلفة لمراقبته. كما أنّ عملية جمع البيانات المختلفة قد تكشف عن وجود أنماط مُختلفة في بيئة الحديقة تُشكِّلُ معًا نظامًا مُتكاملًا.

افترض الآن وجود عدة حدائق في أنحاء مختلفة من المدينة. يتعين على إدارة الحدائق في البلدية مراقبة ورعاية كل من هذه الحدائق من خلال أنظمة مختلفة. تجمع هذه الحدائق كميات هائلة من البيانات، والتي يمكن تحليلها بدورها لتوفير رؤى وأنماط شاملة حول إدارة وتشغيل جميع تلك الحدائق من خلال إجراء المقارنة بين كل منها مثلًا.



شكل 4.33: توسيع نطاق مشروع الحديقة الذكية

إن الإدارة الذكية للحدائق باستخدام هذه التقنيات تساهم بشكلٍ فاعل في تحسين الحدائق الذكية وتطويرها، وذلك من خلال زيادة فاعلية تلك النُظم، والمحافظة على توازنها واستمراريتها.

تمرينات

1 ما الغرض من استخدام واجهة برمجة تطبيقات الويب؟

2 لماذا يُعدُّ اسم المستخدم وكلمة المرور ضروريان للاتصال بواجهة برمجة تطبيقات الويب؟ اذكر سببين لذلك.

3 ما الأسطُر البرمجية التي يتعين عليك إجراء تغييرات عليها في البرنامج لتعديل الشروط المطلوبة لِبدء ري النبات ومدة الري؟

4 صِفْ البيانات التي يمكن جمعها - بخلاف درجة الحرارة والرطوبة - في بيئة الحديقة الذكية.

5 على فَرَضْ أنه توفرت لك بيانات وقراءات تتعلق بمراقبة البيئة في الحديقة الذكية على المدى الطويل من الماضي، إضافة إلى البيانات الحالية التي تحصل عليها من المنصة السحابية. اشرح كيف يُمكن لهذه البيانات مساعدتك في ري حديقتك الذكية بكفاءة أكبر، واعرض أفكارك أدناه.

6 قارن بين أنواع تنسيقات البيانات JSON وCSV.

المشروع

تتصف الحدائق الذكية بأنها أنظمة مُعقدة تتضمن أنواعًا مختلفة من النباتات في البيئة نفسها، وتتم مراقبتها بنظام واحد.

1 صَمم دائرة باستخدام الأردوينو في محاكي دوائر تينكركاد قائمة على الدائرة التي أنشأتها مُسبقًا لمراقبة وري عدد أكبر من النباتات.

2 استخدم الأطراف التناظرية المتبقية لتوصيل ثلاث نباتات أخرى باستخدام مُستشعرات رطوبة التربة.

3 قم بتطوير برنامج البايثون للحصول على قيم الرطوبة من النباتات الجديدة، وإنشاء ظروف بيئية مختلفة لري كل نبات.

4 أنشئ الدائرة الفعلية وأعد كتابة برنامج البايثون ليتضمن النباتات الجديدة.

وزارة التعليم
Ministry of Education
2023 - 1445

ماذا تعلّمت

- تثبيت مكتبة pyfirmata واستخدام البايثون لبرمجة أجهزة تحكم الأردوينو.
- إنشاء تطبيقات عملية لإنترنت الأشياء باستخدام جهاز تحكم الأردوينو الدقيق.
- استخدام المنصات السحابية لإرسال البيانات المُجمعة و تقييمها .
- تمييز مدى تأثير البيانات المُجمعة من حلول إنترنت الأشياء في عمليات صُنعْ القرار.
- التعرف على طريقة إنشاء حلول إنترنت أشياء موسعة من تطبيقات بسيطة.

المصطلحات الرئيسة

بروتوكول	Protocol
تضمين عرض النبضة	Pulse Width Modulation
واجهة برمجة تطبيقات الويب	Web API
طلب عبر الويب	Web Request
خدمة ويب	Web Service

ملف قيم مفصولة بفواصل	Comma Separated Values
منفذ الاتصال	Communication Port
مكتبة فيرماتا	Firmata
ترميز الكائنات باستعمال جافا سكريبت	JavaScript Object Notation

الجزء الثاني

الوحدة الخامسة
تطبيقات إنترنت الأشياء المتقدمة

الوحدة السادسة
برمجة إنترنت الأشياء باستخدام C++

الوحدة السابعة
الرسائل في إنترنت الأشياء

الوحدة الثامنة
محاكاة شبكة مُستشعرات إنترنت الأشياء اللاسلكية

وزارة التعليم
Ministry of Education
2023 - 1445

وزارة التعليم

Ministry of Education

2023 - 1445

5. تطبيقات إنترنت الأشياء المتقدمة

سيتعرف الطالب في هذه الوحدة على التطبيقات المستخدمة لحلول إنترنت الأشياء في مجال الزراعة ومجال الرعاية الصحية. وسيتعرف أيضًا على هيكليات إنترنت الأشياء، ويكتشف بروتوكولات الشبكات المختلفة. وفي الختام سيتعرف على مفاهيم الأمان والخصوصية في أنظمة إنترنت الأشياء.

أهداف التعلُّم

بنهاية هذه الوحدة سيكون الطالب قادرًا على أن:

- يصف كيفية استخدام تقنيات إنترنت الأشياء في مجال الرعاية الصحية (IoHT).
- يحدِّد تطبيقات الرعاية الصحية الذكية المختلفة.
- يصف مساهمة تقنيات إنترنت الأشياء في تحسين قطاع الزراعة.
- يصنِّف طبقات إنترنت الأشياء الأحادية من آلة إلى آلة (M2M).
- يشرح وظائف طبقات الهيكلية العالمية لإنترنت الأشياء.
- يحدِّد الخصائص الرئيسة لتقنية تحديد الترددات الراديوية (RFID) وتقنية الاتصال قريب المدى (NFC).
- يحدِّد التقنيات والبروتوكولات المستخدمة في شبكات المنطقة الشخصية اللاسلكية (WPANS).
- يميِّز التحديات الأمنية في شبكة الجيل الخامس من أنظمة إنترنت الأشياء.
- يعرِّف المخاوف المتعلقة بالخصوصية الكامنة في إنترنت الأشياء وحلولها المُمكنة.

وزارة التعليم
Ministry of Education
2023 - 1445

رابط الدرس الرقمي

www.ien.edu.sa

الدرس الأول

مجالات تطبيق إنترنت الأشياء

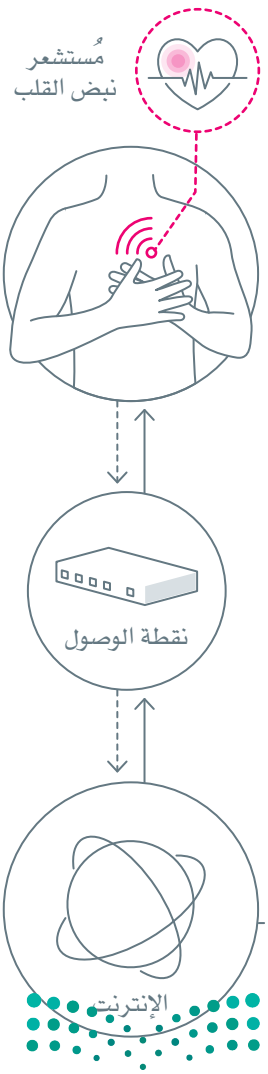
لقد سبق لك وأن تعرفت في كتاب إنترنت الأشياء 1-1 على المفاهيم الأساسية **لإنترنت الأشياء** (IoT) بصفتها إحدى أهم التقنيات الناشئة، وتعرفت أيضًا على بعض التطبيقات الأساسية التي تُستخدم بها تقنية إنترنت الأشياء، وستوسع الآن معرفتك السابقة بتعلم تطبيقات جديدة لإنترنت الأشياء.

الرعاية الصحية الذكية Smart Healthcare

إن تطبيق إنترنت الأشياء في مجال الرعاية الصحية له أثرٌ كبيرٌ في المجتمع، فأجهزة إنترنت الأشياء المختلفة القابلة للارتداء مثل: المُستشعرات، وأجهزة مراقبة الصحة عن بُعد، وتنبيهات الطوارئ، بالإضافة إلى أجهزة مراقبة المؤشرات الصحية، جميعها تشتمل على أدوات لمراقبة صحة الأشخاص وتتبعها من خلال تقنيات مبتكرة تهدف لتعزيز جودة حياة الإنسان ورفاهيته، وذلك بدءًا من مراقبة الأطفال المرضى إلى تشخيص الأمراض المزمنة ومراقبتها لدى كبار السن، مما يتيح توفير خدمات رعاية صحية فعّالة لجميع الأعمار.

تطور الرعاية الصحية The Evolution of Healthcare

أدّت الزيادة المُطردة في عدد السكان إلى ظهور تحديات جديدة يمكن التغلب عليها من خلال الرعاية الصحية الذكية. يشير مصطلح الرعاية الصحية الذكية إلى تطبيق التقنية لتحسين نوعية الحياة. ورغمَ أنّ القصور العام في المعرفة الرقمية لدى بعض العاملين في مجال الرعاية الصحية أدّى إلى إبطاء عملية الانتقال إلى الرعاية الصحية الذكية، إلا أنّ الحكومات والمؤسسات الخاصة تستثمر بشكل واسع في مشاريع دمج التقنيات لتحسين نظام الرعاية الصحية، ويعتمد النظام التقليدي للخدمات الطبية والصحية على مبدأ زيارة المريض لعيادة الطبيب أو المركز الطبي المحلي أو المستشفى عند الحاجة، أما النظام القائم على الرعاية الصحية الذكية فيتيح للمرضى التعامل مع الظروف الطارئة باستقلالية، وبذلك يتحول التركيز على الرعاية الصحية الفردية من العلاج التقليدي في المستشفى إلى الرعاية المنزلية الذكية. توفر الرعاية الصحية الذكية باستخدام أجهزة إنترنت الأشياء إمكانية مراقبة الصحة عن بُعد، وعرض تنبيهات الطوارئ، وتوفير خدمات علاجية بتكلفة معقولة، وكذلك ضمان توافر الخدمات الطبية للجميع بغض النظر عن الموقع أو البعد والقرب من المراكز الطبية والمستشفيات. تتنوع أجهزة مراقبة الصحة هذه، بدءًا من أجهزة مراقبة اللياقة البدنية وأجهزة التتبع التي تقيس المؤشرات الصحية، مرورًا بالتقنيات المتطورة القابلة للارتداء التي تجمع العديد من المؤشرات الحيوية معًا.



شكل 5.1: مراقبة الطوارئ الصحية

إنترنت أشياء الرعاية الصحية Internet of Healthcare Things

إنترنت أشياء الرعاية الصحية (IoHT) هو أحد حلول إنترنت الأشياء التي تستخدم تلك التقنية لربط الأشخاص بخدمات الرعاية الصحية المختلفة، ويُمكن للأطباء الأخصائيين من خلال استخدام هذه التقنية القيام بمراجعة التقارير والسجلات الطبية للمرضى عن بعد، وتقديم التشخيص والتوصيات دون الوجود الفعلي في نفس الموقع مع المريض. يتكون إنترنت أشياء الرعاية الصحية من شبكة متصلة من التقنيات الطبية تشمل التصوير الطبي، وتقارير المختبرات الطبية وأجهزة مراقبة الرعاية الصحية عن بُعد، ويشمل التصوير الطبي التصوير بالأشعة السينية، **والتصوير بالرنين المغناطيسي** (Magnetic Resonance Imaging – MRI)، **والتصوير المقطعي المحوسب** (Computerized Tomography – CT)، والأنواع الأخرى من التصوير، ويشمل إنترنت أشياء الرعاية الصحية أيضًا خدمات الطوارئ كسيارات الإسعاف الذكية والعيادات الذكية.

الأجهزة القابلة للارتداء Wearables

الأجهزة القابلة للارتداء هي أشياء ذكية توضع على جسم الإنسان، ويُمكنها جمع البيانات المتعلقة بصحة الشخص وتخزينها ومعالجتها وتحليلها؛ لتوفير المعلومات المطلوبة وإرسال التنبيهات في سيناريوهات الطوارئ، ويُعدّ المرضى الذين يعانون من إعاقات مؤقتة أو دائمة، وكذلك كبار السن والأطفال من المستخدمين الأساسيين لهذه الأجهزة، حيث تقوم المستشعرات الحيوية المُدمجة في ملابس المريض بالتقاط البيانات وإنتاج مخرجات كهربائية رقمية يمكن استخدامها لمراقبة المؤشرات الصحية الخاصة بذلك المريض، ويُعدّ المُستشعر البيولوجي أداة تحليلية مُصغرة مُدمجة مع مُكون حيوي يتعرف على إشارات معينة، وتختلف المُستشعرات والمُشغلات حسب طبيعة أنظمة المراقبة المنوطة بها، ويمكنها جمع البيانات ونقلها كالإشارات الحيوية، ودرجة حرارة الجسم، ومستوى تشبع الأكسجين في الدم (قياس التأكسج النبضي)، وحركة الإنسان، والموقع الجغرافي للشخص. توجد العديد من الإشارات الحيوية المتولدة من الجسم مثل: **مخطط كهربية القلب** (Electrocardiogram – ECG)، و**مخطط كهربية الدماغ** (Electroencephalogram – EEG)، و**مخطط كهربية العضل** (Electromyography – EMG). يمكن للمُستشعرات مراقبة المؤشرات الفسيولوجية أو الميكانيكية الحيوية للإنسان مثل مُعدَّل ضربات القلب، ونشاط العضلات، ومُعدَّل التنفس، ودرجة حرارة الجسم، وضغط الدم، ووضع الجسم، والحركة، والتسارع، وعادةً ما تكون مُخرجات المستشعرات الذكية وأجهزة إنترنت الأشياء عالية التعقيد، مما يستلزم تطبيق الذكاء الاصطناعي وتحليلات البيانات وتقنيات أخرى مثل الحوسبة السحابية.

مخطط كهربية الدماغ (Electroencephalogram - EEG):

جهاز تخطيط كهربيّة الدماغ هو أداة لتشخيص النشاطات غير الطبيعية في الإشارات الكهربية في الدماغ.

شكل 5.2: شبكات مُستشعرات الجسم المتصلة بشبكات إنترنت أشياء الرعاية الصحية

شبكة مُستشعرات الجسم Body Sensor Network

شبكة مُستشعرات الجسم (BSN) هي **شبكة مستشعرات لاسلكية** (Wireless Sensor Network – WSN) تستخدم لمراقبة جسم الإنسان، فهي عبارة عن شبكة عقدية حساسة يمكن ارتداؤها، ويمكنها الاتصال بالعُقد والكائنات الذكية الأخرى. تحتوي عُقد الاستشعار على قدرات الحوسبة والتخزين والإرسال اللاسلكي بالإضافة إلى الاستشعار. وكما يظهر في الشكل 5.2، يُرسل مُستشعر تدفق الدم بيانات المريض إلى جهاز ذكي يتصل بالإنترنت، ويرسل هذه البيانات إلى المستشفى الذكي. وعلى الرغم من أن الأنظمة القائمة على شبكة مستشعرات الجسم تضم مجموعة متنوعة من التطبيقات، إلا أنه يمكن استخدامها للمراقبة المستمرة وغير الجراحية للمؤشرات الحيوية، حيث تُوضع مُستشعرات لاسلكية صغيرة على الجلد، وقد تدمج في بعض الحالات بالملابس، وهذا من شأنه أن يُسهّل التعرف المبكر على المرض وتشخيصه. عادةً ما تكتشف هذه المستشعرات بيانات عن حركة جسم الإنسان ودرجة حرارة الجسم ومعدل ضربات القلب ومعدل مُوصلية الجلد ووظائف العضلات.

تطبيقات الرعاية الصحية الذكية Smart Healthcare Applications

مراقبة ضغط الدم Blood pressure monitoring

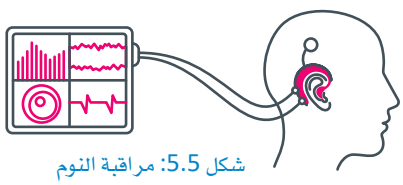
شكل 5.3: مراقبة ضغط الدم

يرتبط الاختلاف في المعدل النموذجي لضخ القلب للدم بارتفاع ضغط الدم لدى البشر، ويُعدّ ارتفاع ضغط الدم مشكلة صحية عالمية ناجمة عن ارتفاع ضغط الدم في الشرايين، ويتسبب ارتفاع ضغط الدم المُزمن في العديد من المشاكل الصحية بما فيها قصور عضلة القلب، وأمراض الكلى المزمنة، وتلف العصب البصري وبالتالي فقدان البصر. تستخدم الساعات الذكية بصفتها أجهزة إنترنت أشياء قابلة للارتداء في تتبع بيانات لياقة المستخدم وتسجيلها وقياس معدل ضربات القلب وكذلك في مراقبة مؤشراتٍ حيوية أخرى كضغط الدم، حيث تقوم بإرسال تلك البيانات ومعالجتها، وقد أصبحت أنظمة إنترنت أشياء الرعاية الصحية القائمة على الحوسبة السحابية شائعة على نطاقٍ واسع، مما يسمح للمرضى بمراقبة ضغط الدم والسيطرة عليه باستخدام أجهزة إنترنت الأشياء.

مراقبة الألم Pain monitoring

شكل 5.4: مراقبة الألم

يُعدُّ التعرف على المشاعر والأنواع المختلفة من الألم لدى البشر أمرًا ضروريًا لأجل تقديم رعاية صحية جيدة للمرضى. يكتسب التعرف على المشاعر والألم أهمية خاصة عند التعامل مع صغار السن والمسنين وأولئك المصابين بالأمراض العقلية أو الذين يفتقدون القدرة على التعبير بشكلٍ لفظي أو وصف مشاعرهم وآلامهم بشكل واضح. تُعدُّ تعابير الوجه مؤشرًا سلوكيًا للألم نظرًا لأن الشعور بالألم يولِّد تغيرات في تعابير الوجه، فيمكن استخدامها كأسلوب تلقائي لتشخيص انزعاج الإنسان. توفر القدرة على قياس تعابير الوجه بديلًا عن الأساليب القياسية لقياس المشاعر والألم، ويمكن استخدامها مع من لا يستطيعون التعبير كمرضى العناية المركزة والرُضَّع، وفي الواقع يلاحظ الكثير من الآباء تعبيرات أوجه أطفالهم لأنها تنقل معلومات حول صحتهم، ويُعدُّ تطوير نظام آلي للتعرف على الألم باستخدام مُدخلات فسيولوجية من مستشعرات إنترنت الأشياء وتحليل البيانات مهمًا في تقييم أنواع مختلفة من المشاعر والألم.



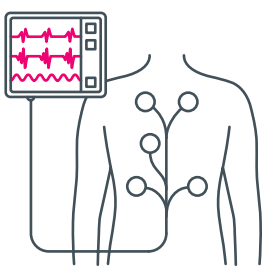
شكل 5.5: مراقبة النوم

مراقبة النوم Sleep monitoring

النوم هو حالة طبيعية ودورية من الراحة النفسية والجسدية، وقد يعاني العديد من الأفراد من اضطرابات النوم، والتي تشمل الأرق وتوقف التنفس وانقطاع النفس الانسدادي أثناء النوم، ويُعدّ **انقطاع النفس الانسدادي النومي** (Obstructive Sleep Apnea – OSA) مرضًا تنفسيًا قاتلًا أثناء النوم يُؤثر سلبًا على حياة الشخص على الصعيد النفسي والجسدي. تتوفر أنظمة عديدة للكشف عن هذا المرض، مثل أجهزة **مخطط كهربية الدماغ** (EEG) في الأذن القابل للارتداء والمتصل بشبكة إنترنت الأشياء في الغرفة. تُوفِّر هذه الأنظمة طريقة مستمرة وغير مزعجة لمراقبة النوم على مدار الساعة طوال أيام الأسبوع وذلك لتقييم جودة النوم. تُستخدم البيانات المُجمَّعة للتنبؤ بمراحل النوم باستخدام خوارزميات الذكاء الاصطناعي.

مراقبة مُخطط كهربية القلب Electrocardiogram monitoring

تلتقط المُستشعرات التي توضع على الجلد الإشارات الكهربائية الناتجة عن ضربات القلب، وتوضع الأقطاب الكهربائية الخاصة بجهاز تخطيط كهربية القلب على مواضع محددة من صدر المريض في العيادات والمستشفيات، حيث لا يمكن للمرضى استخدام مثل تلك الأجهزة في المنزل، ولذلك فقد تم تطوير العديد من الأشياء الذكية التي تُستخدم لفحوصات تخطيط القلب عن بُعد، بحيث يُمكن للأطباء معاينة بيانات المرضى من خلال هذه الأجهزة القابلة للارتداء. تحتوي بعض هذه الأشياء الذكية على تطبيقات للتنبيه والتحذير في حالات النوبات القلبية وكذلك لتقديم توصيات لصحة القلب للأشخاص المعنيين.



شكل 5.6: مراقبة مخطط كهربية القلب

مخطط كهربية القلب (Electrocardiogram - ECG):

مخطط كهربية القلب هو اختبار يقيس النشاط الكهربائي للقلب لتحديد ما إذا كان القلب يعمل بشكل صحيح.

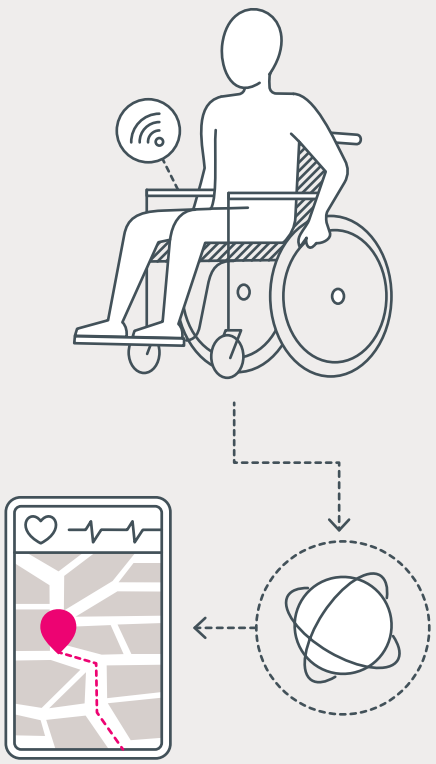
مراقبة علم الأمراض Pathology monitoring

يوصف علم الأمراض بأنه الدراسة العلمية لأصول وتأثيرات المرض والإصابة، ويقوم جهاز **مخطط كهربية الدماغ** (EEG) بتشخيص الأمراض من خلال توصيل أقراص معدنية صغيرة بأسلاك رفيعة بفروة الرأس تُرسل إشارات إلى جهاز حاسب لتخزين النتائج، ويُستخدم جهاز مخطط كهربية الدماغ على نطاقٍ واسع لتشخيص الاضطرابات المتعلقة بالدماغ بسبب تكلفته المنخفضة وطبيعته غير الجراحية، ويُمكن لجهاز تخطيط كهربية الدماغ تشخيص بعض الاضطرابات المتعلقة بالدماغ، مثل الصرع والسكتة الدماغية. يحتاج المرضى الذين يعانون من هذه الحالات إلى عناية فورية لأن أي تأخير قد يؤثر على حياتهم. وهكذا يُمكن أن يكون نظام إنترنت الأشياء الذي يراقب حالة المريض مُنقذًا للحياة في مثل هذه المواقف.

شكل 5.7: مراقبة علم الأمراض

مراقبة الأشخاص ذوي الإعاقة Disabled persons monitoring

تُعدُّ **الكراسي المتحركة الذكية** (Smart wheelchairs – SMW) المتصلة بأنظمة إنترنت الأشياء موضوعًا بحثيًا جديدًا، و يتكون تصميم هذه الأنظمة من عنصرين: خدمة الخرائط المستخدمة للملاحة، والكرسي المتحرك للمستخدم، وتتضمن الكراسي المتحركة الذكية تقنية **قياس المسافات ثلاثية الأبعاد** (3D LIDAR)، وذلك لرسم خرائط للمحيط الخارجي وحركتها المستقلة دون الحاجة إلى **نظام تحديد المواقع العالمي** (Global Positioning System – GPS). تستخدم هذه التقنية كلًا من هندسة التحكم للكرسي المتحرك المزود بمحرك، ونظامًا مُدمجًا لمراقبة المرضى ذوي الحالات الحرجة، ويستخدم النظام المُضمَّن أيضًا الخصائص الحيوية للمستخدم لاكتشاف المواقف الخطرة المحتملة. يُصدر الكرسي المتحرك تحذيرًا عن طريق تنشيط تنبيه عند قياسه لنبضات القلب وارتفاع ضغط الدم بشكلٍ دوري.



شكل 5.8: مراقبة ذوي الإعاقة

تقنية اكتشاف الضوء والمدى
(Light Detection And Ranging - LIDAR):

هي تقنية لقياس المسافات عن طريق توجيه الليزر إلى عنصر أو سطح وقياس الوقت اللازم لانعكاس الضوء إلى المُرسل.

مثال

أطلقت شركة اتصالات سعودية مشروع العيادة الافتراضية، وتُستخدم هذه العيادة من قبل الأطباء لتشخيص مرضاهم عن بُعد، وتستخدم هذه الخدمات أنظمة شبكات إنترنت الأشياء من خلال الأجهزة القابلة للارتداء لمساعدة الأطباء لجمع البيانات الضرورية، والتي تُرسل للمستشفيات والمراكز الطبية المحلية لمتابعة حالات المرضى.

الزراعة الذكية Smart Agriculture

لقد قُمت في الوحدة السابقة بأولى خطواتك في مجال الزراعة الذكية، وذلك من خلال إنشاء نظام لري النبات. يمكن تحسين وتطوير القطاع الزراعي وسير عمله من خلال استخدام العديد من تقنيات إنترنت الأشياء، حيث يتيح تطبيق إنترنت الأشياء في القطاع الزراعي مزايا خاصة مثل: ترشيد استخدام الموارد كالأرض والمياه والأسمدة ومبيدات الآفات؛ وكذلك تحسين الأرباح وتحقيق الاستدامة، وسلامة الغذاء وحماية البيئة، وخفض تكاليف الإنتاج.

شكل 5.9: الزراعة الذكية باستخدام الطائرات دون طيار

تطبيقات الزراعة الذكية Smart Agriculture Applications

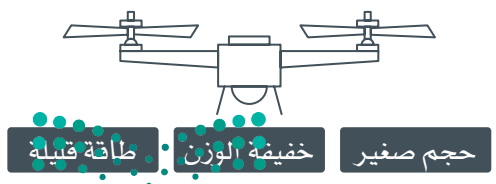
الزراعة الدقيقة Precision farming

تتضمن الزراعة الدقيقة ري النباتات حسب موقعها ووفق كميّات المياه التي تحتاج إليها الأنواع المختلفة من النباتات، ويتطلب هذا النوع من الزراعة جمع وتحليل البيانات من خلال العديد من المُستشعرات مثل موقع النبات والرطوبة ودرجة حرارة الأرض، والتي يمكن الحصول عليها من خلال المسح والمراقبة الجوية، وقد اكتسبت الطائرات التي يُتحكم فيها عن بُعد، والتي تُعرَف غالبًا باسم **المركبات الجوية دون طيار** (Unmanned Aerial Vehicles – UAVs) أو **الطائرات المسيرة** (Drones)، شيوعًا كبيرًا في عمليات المراقبة الجوية. فعلى مدار السنوات الماضية، استُخدمت الطائرات دون طيار على نطاق واسع لمراقبة الحقول والمزروعات، ولتقديم حلول زراعية دقيقة وفعّالة. لقد أضحى من الممكن من خلال استخدام الاستشعار عن بُعد متابعة مجموعة متنوعة من المقاييس المتعلقة بالمحاصيل والغطاء النباتي، وذلك بالاستعانة بصور ذات أطوال موجية متفاوتة حلّت بديلًا عن صور الأقمار الصناعية التي كان يُعتمد عليها في الماضي، وقد أثبتت أنظمة الطائرات دون طيار فعاليتها في العديد من تطبيقات الزراعة الدقيقة، بما فيها رش مبيدات الآفات، والتعرف على نقص المياه، وتحديد أمراض النباتات. وهكذا أصبح بالإمكان اتخاذ العديد من القرارات بناءً على البيانات المُلتقطة من الطائرات دون طيار لتقدير تكاليف معالجة المشاكل المحددة وتحسين جودة الإنتاج.

يتمثل دور الطائرات دون طيار في التقاط البيانات بتفاصيل مكانية دقيقة، حيث تُستخدم العديد من المُستشعرات اعتمادًا على المؤشرات الزراعية التي يجب مراقبتها، و يجب أن تفي مستشعرات الطائرات دون طيار بثلاثة متطلبات أساسية: استهلاك مُنخفض للطاقة، وخفة الوزن، وصِغَر الحجم. تعمل هذه التقنيات على إنشاء خرائط بيئية تصور طبيعة التربة، مما يسمح بتخطيط أنظمة ري أكثر كفاءة لجميع المحاصيل، وتُستخدم تقنيات **نظام تحديد المواقع العالمي** (GPS) على نطاق واسع للمساعدة في تحديد المواقع والإسناد الجغرافي للأشياء المُلتقطة بواسطة الاستشعار عن بُعد، ونظرًا لأن معلومات الاستشعار عن بعد تُعدُّ مصدرًا رئيسًا للبيانات البيئية؛ فإنّه يتم في العادة استيرادها إلى **أنظمة المعلومات الجغرافية** (Geographic Information Systems – GISs) ودمجها مع مجموعات البيانات الأخرى.

مركبة جوية دون طيار (Unmanned Aerial Vehicle - UAV):

يتم تسيير "المركبات الجوية دون طيار" دون طيارين أو طاقم أو ركاب.



شكل 5.10: المتطلبات الأساسية للمركبات الجوية دون طيار

جدول 5.1: أنواع المُستشعرات الهامة المُستخدمة في المركبات الجوية دون طيار (UAVs)

نوع المُستشعر	الوصف
مستشعرات الإضاءة المرئية	تلتقط الصور في ظروف مختلفة، بما في ذلك الطقس المُشمس والغائم، وتعتمد جودة الصور على ظروف الإضاءة.
مُستشعرات الأشعة تحت الحمراء الحرارية	تقيس مُستشعرات الأشعة تحت الحمراء الحرارية درجات حرارة السطح. فباستخدام مستشعرات الأشعة تحت الحمراء وعدسة بصرية، تجمع الكاميرات الحرارية طاقة الأشعة تحت الحمراء. تركز كاميرات التصوير الحراري على الإشعاع وتكتشفه عند نفس الأطوال الموجية، ثم تحوله إلى صور ذات تدرجات رمادية تمثل الحرارة، ويمكن لأجهزة استشعار التصوير الحراري المتعددة إنشاء صورٍ ملونة أيضًا.
مُستشعرات التصوير متعددة الأطياف	تجمع المستشعرات متعددة الأطياف الأطوال الموجية المرئية وكذلك الأطوال الموجية التي تقع خارج الطيف المرئي، بما في ذلك **الأشعة تحت الحمراء القريبة** (Near-Infrared Radiation – NIR) و**الأشعة تحت الحمراء قصيرة الموجة** (Short-Wave Infrared Radiation – SWIR) وغيرها. تقوم الطائرات دون طيار المزودة بمُستشعرات متعددة الأطياف أو فائقة الطيف بجمع معلومات امتصاص المحاصيل للمياه، وعلى الرغم من تكلفتها العالية، إلا أنّ البيانات الطيفيّة يمكن أن تكون ذات قيمة كبيرة لتقييم العديد من الخصائص البيولوجية والفيزيائية للمحاصيل.

الري الدقيق Precision irrigation

تُعدُّ تقنية الري الدقيق تقنية زراعية تحافظ على العناصر الغذائية وتُحسِّن كمية المياه التي تتطلبها النباتات من خلال تزويد جذور النباتات بقطرات الماء ببطء تحت سطح التربة أو فوقه، كما يتم زيادة إنتاجية المحاصيل باستخدام تقنيات إنترنت الأشياء الدقيقة للري، حيث تحدد المستشعرات الثابتة الخصائص الفيزيائية والكيميائية للأراضي الزراعية، بما فيها الطقس ودرجة الحرارة والرطوبة، وصحة النبات، ورطوبة وحموضة التربة، ومغذيات التربة. يتم تحليل البيانات التي تُجمع لإبلاغ المزارعين بالتعديلات التي يتعين عليهم إجراؤها، كما يساعد تحليل البيانات في تحديد العناصر الغذائية المناسبة وكمياتها، وكذلك تحديد كمية المياه اللازمة للري.

شكل 5.11: تطبيق الري الدقيق

وزارة التعليم
Ministry of Education
2023 - 1445

الزراعة العمودية Vertical farming

يتم في الزراعة العمودية زراعة النباتات بنطاق رأسي وليس أفقي، مما يسمح بإنتاج المزيد من المحاصيل في مساحاتٍ صغيرة، وكذلك زراعة أنواع متعددة من المحاصيل في ذات الوقت. يمكن التعامل مع الأجهزة باستخدام تقنيات إنترنت الأشياء عن بُعد باستخدام تقنيات الاتصال مثل البلوتوث والواي فاي وتقنية الاتصال اللاسلكي (RFID)، وتهدف الزراعة العمودية إلى زراعة المحاصيل في البيئات الحضرية، ويتمتع نظام الزراعة العمودية الداخلي بمناخ مثالي بعيدًا عن القلق من المؤثرات البيئية الخارجية، وتُعدُّ تقنيات إنترنت الأشياء ضرورية في بيئة الزراعة لمراقبة صحة النبات والري، حيث تتطلب الزراعة العمودية معالجة كميات هائلة من البيانات وتحليلها للمساهمة في تطور المحاصيل بشكل فعّال، كما يمكن تحسين الإنتاجية الزراعية بالزراعة العمودية مثل أتمتة العملية برمتها من وضع البذور إلى حصاد المحصول في بيئة مغلقة.

شكل 5.12: تطبيق الزراعة العمودية

مثال

من المخطط أن تكون مدينة نيوم العملاقة في المملكة العربية السعودية مدينة عمودية تستخدم فيها أحدث التقنيات لحل مشاكل التلوث والنقل واستدامة الغذاء. ستحتوي مدينة نيوم على مبنيين يبلغ ارتفاع كل منهما 500 متر، ويبعدان عن بعضهما مسافة 200 متر، ويمتدان بالتوازي لمسافة 170 كيلومترًا. تقع المدينة العمودية المتطورة في المنطقة الواقعة بين هذين المبنيين، وتهدف نيوم إلى إنشاء أول نظام متكامل للاكتفاء الذاتي الغذائي الصحراوي ومع ندرة المياه المتاحة، ستكون هناك حاجة لأنظمة زراعة ذكية لإنشاء مجتمعات مُكتفية ذاتيًا. تُعزّز تقنيات الزراعة المائية والزراعة العمودية من خلال تقنيات إنترنت الأشياء والذكاء الاصطناعي لتحسين استخدام الموارد وتعزيز الإنتاج الزراعي. تعرف أكثر على مشروع ذا لاين بمدينة نيوم من هنا: https://www.neom.com/ar-sa/about.

تمرينات

1

حدد الجملة الصحيحة والجملة الخاطئة فيما يلي:	صحيحة	خاطئة
1. لا تسهم تقنيات إنترنت الأشياء في تحسين مجال الرعاية الصحية.	○	○
2. يُعدُّ إنترنت أشياء الرعاية الصحية امتدادًا لإنترنت الأشياء.	○	○
3. تتصل كافة الأجهزة الطبية القابلة للارتداء بصورة مستمرة بشبكة الإنترنت.	○	○
4. يمكن لشبكات مُستشعرات الجسم أن تكون أنظمة إنترنت أشياء مستقلة.	○	○
5. يتضمن الكرسي المتحرك الذكي نظامًا مدمجًا يستخدم الخصائص الحيوية لمستخدمه لاكتشاف المواقف الخطرة المحتملة.	○	○
6. يُمكن للمركبات الجوية دون طيار إجراء نوع واحد فقط من المسح للأراضي الزراعية.	○	○
7. تكتشف مُستشعرات الأشعة تحت الحمراء الحرارية أي إشعاع حراري.	○	○
8. يستخدم الري الدقيق لتحسين استخدام الموارد اللازمة للأنظمة الزراعية.	○	○
9. لا يحتاج نظام الري الدقيق إلى الكثير من المُستشعرات في عمله.	○	○
10. تستخدم الزراعة العمودية لإتاحة الاستخدام الأفضل للأراضي الزراعية.	○	○

2 وضِّح المقصود بإنترنت أشياء الرعاية الصحية.

3 قارن بين أنواع البيانات التي يمكن جمعها بواسطة الأشياء الذكية القابلة للارتداء.

4 مِمَ تتكون شبكة مُستشعرات الجسم؟

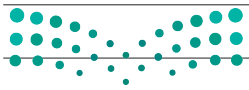
5 كيف يُمكن استخدام تقنيات الذكاء الاصطناعي لحلول إنترنت الأشياء في مراقبة المشاعر والألم؟

6 صِف كيفية استخدام المركبات الجوية دون طيار في الزراعة الدقيقة لتطبيقات إنترنت الأشياء.

7 صنِّف الأنواع المختلفة لمُستشعرات المركبات الجوية دون طيار.

8 صِفْ كيفية مساهمة أنظمة إنترنت الأشياء في تطبيقات الري الدقيق.

9 ما مدى اعتماد الزراعة العمودية على حلول إنترنت الأشياء الفعّالة؟



رابط الدرس الرقمي

www.ien.edu.sa

الدرس الثاني

تقنيات شبكات إنترنت الأشياء

المقارنة بين هيكلية شبكة oneM2M وهيكلية أنظمة إنترنت الأشياء العالمي

oneM2M Architecture Versus IoT World Forum Architecture

أدى التطور السريع للاتصالات من **آلة إلى آلة** (M2M) إلى إنشاء هيكليات إنترنت أشياء مختلفة. تساعد هذه الهيكليات في تسريع اعتماد تطبيقات وأجهزة (M2M) بما فيها إنترنت الأشياء، وتُعدُّ هيكلية (oneM2M) وهيكلية أنظمة إنترنت الأشياء العالمي من هيكليات إنترنت الأشياء الأكثر شيوعًا على نطاق واسع. تصمم هيكلية (oneM2M) حلول إنترنت أشياء تختص بالأجهزة وتطبيقاتها فقط، بينما تراعي هيكلية أنظمة إنترنت الأشياء العالمية تقنيات أخرى مثل تخزين البيانات ومعالجتها والاتصال بالشبكة والحوسبة المتطورة.

آلة إلى آلة

(Machine To Machine - M2M):

يصف مصطلح **آلة إلى آلة** (M2M) أي تقنية تُمكِّن الأجهزة المتصلة بالشبكة من تبادل البيانات وتنفيذ المهام دون أي تدخل بشري.

هيكلية oneM2M oneM2M Architecture

يُعدُّ التعامل مع مجموعة متنوعة من الأجهزة والبرامج وطرق الوصول أحد أكبر التحديات التي تواجهها عملية تطوير هيكلية إنترنت الأشياء. تقوم هيكلية (oneM2M) من خلال إنشاء تصميم منصة أفقية بإنشاء معايير التشغيل البيني على جميع مستويات مراحل إنترنت الأشياء. بناءً على هيكلية (oneM2M)، يتم تقسيم وظائف إنترنت الأشياء إلى ثلاث طبقات: طبقة التطبيقات، وطبقة الخدمات، وطبقة الشبكة. قد تبدو هذه الهيكلية للوهلة الأولى أساسية وعامة نسبيًا؛ ولكنها رغم ذلك غنية جدًا وداعمة للتشغيل البيني عبر واجهات برمجة لتطبيقات تقنية المعلومات، وتدعم مجموعة واسعة من تقنيات إنترنت الأشياء.

الشبكات المُعرفة بالبرمجيات

(Software-Defined Networks - SDN):

الشبكة المعرفة بالبرمجيات هي إحدى هيكليات الشبكات، والتي يُتحكّم بها من خلال وحدات تحكم قائمة على البرامج أو **واجهات برمجة التطبيقات** (Application Programming Interfaces – APIs) عوضًا عن استخدام المعدات أو الأجهزة المتخصصة.

طبقة التطبيقات Applications layer

تعطي هيكلية (oneM2M) الأولوية للاتصالات بين الأجهزة والتطبيقات الخاصة بها. يحتوي هذا المجال على بروتوكولات طبقة التطبيق والتكامل مع أنظمة **ذكاء الأعمال** (Business intelligence – BI).

طبقة الخدمات Services layer

يتم تمثيل هذه الطبقة أفقيًا عبر التطبيقات الخاصة بكل نوع من الصناعات المحددة. تتكون الوحدات الأفقية في هذا المستوى من الشبكة المادية التي تعمل عليها تطبيقات إنترنت الأشياء، وبروتوكولات الإدارة الأساسية، والأجهزة. من الأمثلة المهمة هنا: الاتصالات الخلوية، **والتبديل متعدد الاتفاقيات باستخدام المؤشرات التعريفية** (MPLS) **والشبكات الافتراضية الخاصة** (Virtual Private Networks – VPNs) **والشبكات المُعرَّفة بالبرمجيات** (SDNs) وغيرها، وتُعدُّ طبقة الخدمات المشتركة أعلى طبقة هنا.

التبديل متعدد الاتفاقيات باستخدام المؤشرات التعريفية (Multiprotocol Label Switching - MPLS):

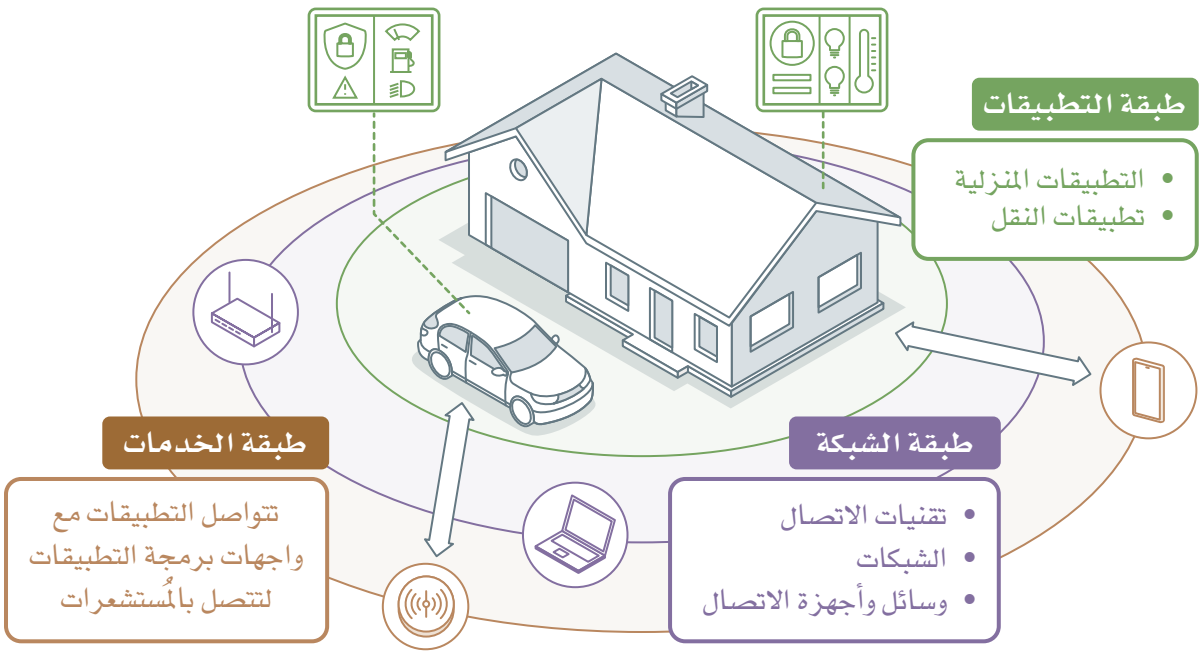
يعمل التبديل متعدد الاتفاقيات باستخدام المؤشرات التعريفية على توجيه البيانات بين العُقد بناءً على المؤشرات التعريفية والوسوم وليس عناوين الشبكة.

طبقة الشبكة Network layer

تُشكل هذه الطبقة مجال الاتصال بين أجهزة إنترنت الأشياء والنقاط النهائية، وتتكون طبقة الشبكة من كافة الأجهزة وشبكة الاتصالات التي تربط أنواعًا مختلفة من الشبكات، مثل الشبكات المتداخلة اللاسلكية وأنظمة النقطة إلى عدة نقاط.

نظام من نقطة إلى عدة نقاط (Point-to-multipoint system):

يوفر نظام نقطة إلى عدة نقاط مسارات مختلفة من عُقدة شبكية واحدة إلى عُقد وجهات متعددة.



شكل 5.13: طبقات هيكلية oneM2M

تتواصل الآلات الذكية وغير الذكية مع بعضها البعض بشكل متكرر، وفي بعض الحالات، يكون الاتصال من آلة إلى آلة غير ضروري، حيث تتصل الأجهزة فقط بتطبيقات خاصة بالاستخدام في مجال تطبيق إنترنت الأشياء عبر **شبكة منطقة ميدانية** (Field Area Network – FAN). تُعدُّ هذه الشبكة أكثر العناصر تعقيدًا في شبكة الاتصالات نظرًا لكونها مسؤولة بشكل أساسي عن توفير اتصالات **الميل الأخير** (last-mile) للأجهزة الطرفية. يتكون نطاق الجهاز أيضًا من جهاز البوابة الذي يوفر اتصالات بالشبكة الأساسية ويعمل كحد بين نطاقات الجهاز والشبكة.

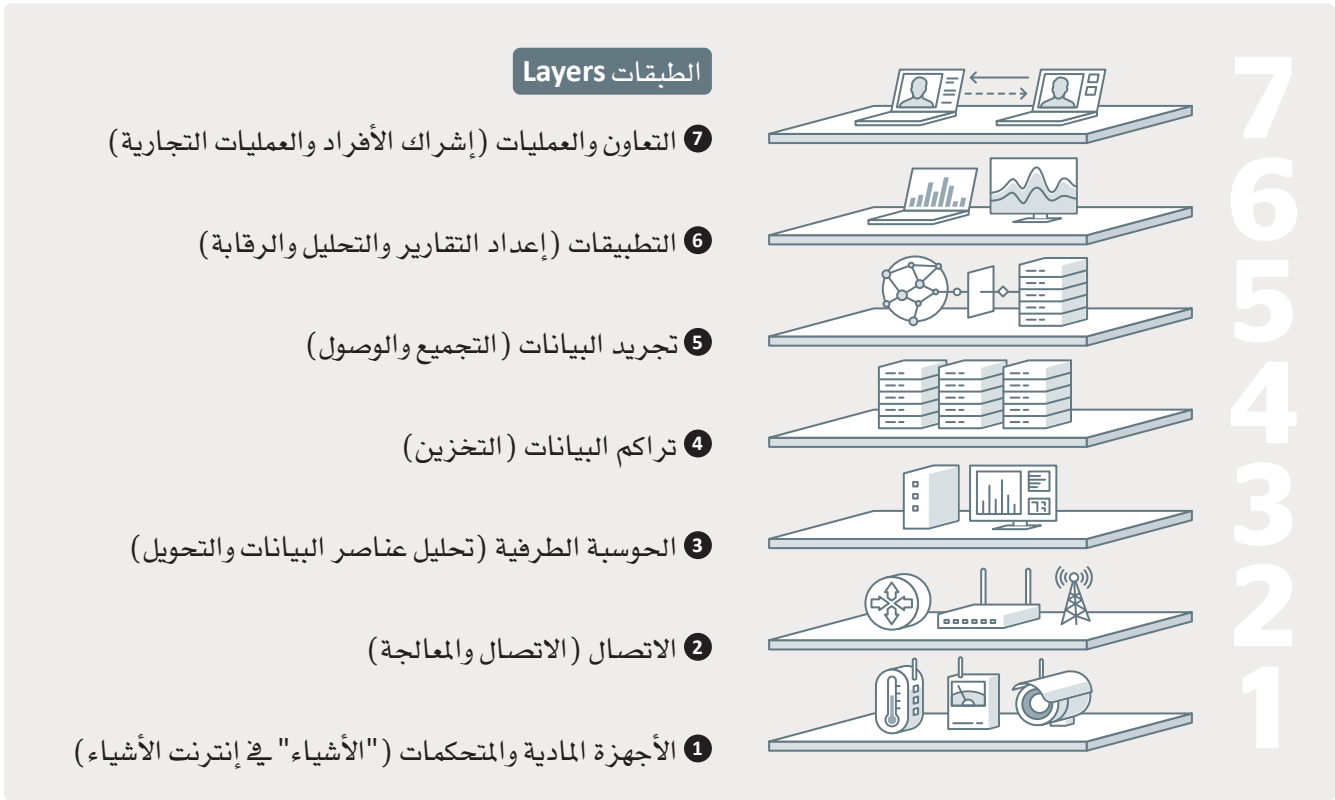
هيكلية أنظمة إنترنت الأشياء العالمي IoT World Forum Architecture

يُحدد النموذج المرجعي لإنترنت الأشياء –الذي قُدّم في المنتدى العالمي لإنترنت الأشياء – سلسلةً من المستويات مع تحكم رئيس من نقطة مركزية إلى طبقات الحافة، والتي تتكون من المستشعرات والأجهزة والآلات وعُقد النهاية الذكية الأخرى. يمكن القول إنه بشكلٍ عام، تنتقل البيانات من الطبقات الطرفية إلى المركز.

يشبه النموذج المرجعي لإنترنت الأشياء نموذج شبكات الربط البيني للأنظمة المفتوحة (OSI).

يمكنك باستخدام هذا النموذج المرجعي تحقيق ما يلي:

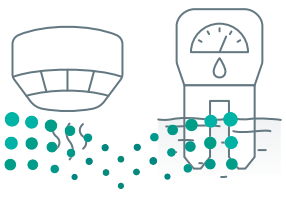
- تقسيم التحديات التي تواجه إنترنت الأشياء إلى مشاكل فرعية.
- تحديد التقنيات المختلفة في كل طبقة وطبيعة العلاقة بينها.
- تعريف نظام متكامل قائم على مكونات متعددة من مزودين مختلفين.
- تحديد **الواجهات** (Interfaces) بطريقة تعزز إمكانية التشغيل البيني.
- تحديد نموذج حماية متعدد الطبقات يُفرض في نقاط الانتقال لكل مستوى.



شكل 5.14: طبقات هيكلية إنترنت الأشياء العالمي

الطبقة الأولى: طبقة الأجهزة المادية والمتحكمات
Layer 1: Physical Devices and Controllers Layer

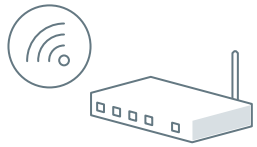
إن أول طبقة في النموذج المرجعي لإنترنت الأشياء هي طبقة الأجهزة المادية والمتحكمات. تحتوي هذه الطبقة على "الأشياء" الخاصة بإنترنت الأشياء، مثل الأجهزة الطرفية والمُستشعرات التي ترسل البيانات وتستقبلها. يمكن أن تتراوح هذه "الأشياء" في حجمها من مُستشعرات صغيرة للغاية إلى آلات تصنيع ضخمة، والمهمة الرئيسة لهذه الطبقة هي إنتاج البيانات والسماح بالتحكم عبر الشبكة.



شكل 5.15: طبقة الأجهزة والمتحكمات

وزارة التعليم
Ministry of Education
2023 - 1445

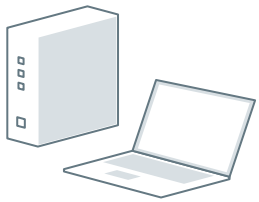
الطبقة الثانية: طبقة الاتصال Layer 2: Connectivity Layer



شكل 5.16: طبقة الاتصال

يتمثل دور طبقة الاتصال في نقل البيانات بطريقة موثوقة وفي الوقت المناسب، ويشمل هذا عمليات النقل بين أجهزة الطبقة الأولى والشبكة، وكذلك عمليات النقل بين الشبكة وطبقة الحوسبة الطرفية (معالجة معلومات الطبقة الثالثة). تشتمل طبقة الاتصال على جميع أجزاء الشبكات في إنترنت الأشياء، ولا تميز بين شبكة (الميل الأخير) – الشبكة بين المستشعر أو نقطة النهاية وبوابة إنترنت الأشياء، والتي سيتم تناولها لاحقًا في هذا الفصل– وشبكة البوابة، والشبكة الرئيسة.

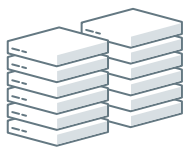
الطبقة الثالثة: طبقة الحوسبة الطرفية Layer 3: Edge Computing Layer



شكل 5.17: طبقة الحوسبة الطرفية

تلعب الحوسبة الطرفية دور الطبقة الثالثة في الهيكلية. تركز هذه الطبقة على تقليل البيانات وتحويل تدفقات بيانات الشبكة إلى معلومات جاهزة للتخزين والمعالجة بمستويات أعلى، وتتمثل إحدى الأهداف الأساسية لهذا النموذج المرجعي في بدء معالجة المعلومات بالقرب من حافة الشبكة بقدر الإمكان وبأسرع ما يمكن، كما تقوم الطبقة الثالثة أيضًا بفحص البيانات لمعرفة ما إذا كان يمكن تصفيتها أو تجميعها قبل نقلها إلى طبقة أعلى. يسمح هذا أيضًا بإعادة تنسيق البيانات أو فك تشفيرها، مما يُسهِّل المعالجة الإضافية بواسطة الأنظمة الأخرى.

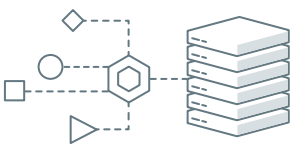
الطبقة الرابعة: طبقة تراكم البيانات Layer 4: Data Accumulation Layer



شكل 5.18: طبقة تراكم البيانات

يتم في هذه الطبقة التقاط وحفظ البيانات حتى تتمكن البرامج من الوصول إليها عند الضرورة، كما تُحوّل البيانات المستندة على الأحداث إلى تنسيقات يمكن الاستعلام عنها بواسطة خدمات أخرى.

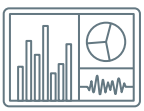
الطبقة الخامسة: طبقة تجريد البيانات Layer 5: Data Abstraction Layer



شكل 5.19: طبقة تجريد البيانات

في هذه الطبقة يتم التوفيق بين تنسيقات البيانات المتنوعة وضمان اتساق الدلالات من المصادر المتنوعة، وفي هذه الطبقة يتم التحقق باستخدام الحوسبة والمحاكاة الافتراضية من أن مجموعة البيانات تحتوي على بيانات كاملة، كما تُدمج البيانات في موقع واحد أو عدة مخازن بيانات.

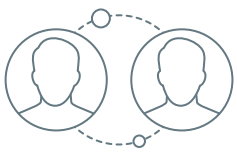
الطبقة السادسة: طبقة التطبيقات Layer 6: Applications Layer



شكل 5.20: طبقة التطبيقات

في هذه الطبقة، تُستخدم البرامج لتفسير البيانات، حيث تتيح البرامج والتطبيقات مراقبة التقارير، وإنشاءها، وتنظيمها اعتمادًا على تحليل البيانات.

الطبقة السابعة: طبقة التعاون والعمليات Layer 7: Collaboration and Processes Layer



شكل 5.21: طبقة التعاون والعمليات

يتم هنا "استهلاك" وتوزيع بيانات التطبيق، وتتبع فائدة إنترنت الأشياء من حقيقة أن المشاركة في بيانات إنترنت الأشياء تتضمن في كثير من الأحيان العديد من الخطوات، ويمكن من خلال هذه الطبقة الحصول على بيانات تسهم في تغيير عمليات الشركة وتحسينها، وذلك بالاستفادة من مزايا إنترنت الأشياء.

بروتوكولات وشبكات الاتصالات قصيرة المدى
Short Range Communication Network and Protocols

تحديد الترددات الراديوية والاتصال قريب المدى RFID and NFC

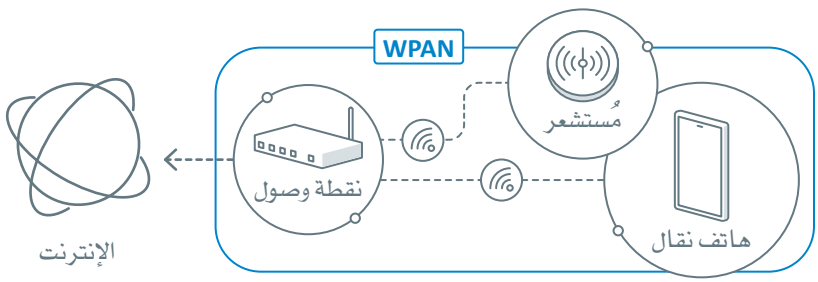
تُعدّ تقنية **تحديد الترددات الراديوية** (Radio Frequency Identification – RFID) وتقنية **الاتصال قريب المدى** (Near Field Communication – NFC) من تقنيات الاتصال التي تسمح بالاتصالات قصيرة المدى بين أجهزة إنترنت الأشياء والشبكة، ويتم استخدام تقنيات (RFID) و(NFC) لتخزين البيانات واستردادها عن بُعد، وتشتمل هذه التقنيات على جهاز إرسال وجهاز استقبال لاسلكي، حيث تستخدم الحقول الكهرومغناطيسية للتعرف تلقائيًا وتتبع الرقاقات المُدمجة بالأشياء الذكية. ترسل الرقاقة البيانات الرقمية وتستقبلها عندما يتم تنشيطها بواسطة نبضة كهرومغناطيسية تصدر من قارئ (RFID) أو (NFC) قربها. تتيح (RFID) تتبع الأدوات والمعدات والمواد في المخازن والمركبات والأشخاص، وذلك من خلال الرقاقات المُرفقة بها. يُمكن لأجهزة قراءة الرقاقات قراءة الرقاقة القريبة منها، حتى لو لم تكن مرئية، كما يمكن قراءة عدد كبير من الرقاقات في ذات الوقت سواء كانت ظاهرة أو مخفية داخل صندوق أو حاوية مثلًا، وذلك خلافًا للرموز الشريطية – الباركود (barcodes)، والتي يجب أن تكون ظاهرةً أمام جهاز القراءة ولا يمكن قراءتها إلا واحدة تلو الأخرى. تُستخدم تقنية (NFC) على نطاق واسع لتبادل البيانات بين الأجهزة في نطاق يبلغ حوالي 4 سنتيمترات، وتُستخدم هذه التقنية في عمليات الدفع غير التلامسية ببطاقات الائتمان، وكبديل لمفاتيح المكاتب التقليدية وغرف الفنادق، وفي ربط إعداد بعض الأجهزة مثل سماعات الرأس، ويتمثل الاختلاف الرئيس بين تقنيتي (RFID) و (NFC) في أن (NFC) صُممت لتبادل البيانات بشكل آمن، مما يجعلها مناسبة للمعاملات المالية، بينما تُستخدم (RFID) بصورة أساسية في التطبيقات التي نحتاج فيها إلى تحديد عناصر معينة لاسلكيًا.

جدول 5.2: مقارنة بين (RFID) و(NFC)

الخاصية	RFID	NFC
تردد الاستخدام	125 كيلوهيرتز إلى 2.45 جيجاهيرتز.	13.56 ميجاهيرتز.
نطاق الاتصال	بحد أقصى 100 متر.	في حدود 10 سم (مسافة قصيرة).
الاتصال	اتصال أحادي الاتجاه.	اتصال ثنائي الاتجاه.
الميزة	التعرف خلال مسافات بعيدة.	أمان عالي.

شبكات المنطقة الشخصية اللاسلكية وبروتوكولاتها
Wireless Personal Area Networks (WPANS) and Protocols

تتطلب المستشعرات والكائنات الأخرى المتصلة بالإنترنت وسيلة معينة لنقل البيانات واستقبالها. سيتم في هذا الدرس تناول **شبكات المنطقة الشخصية** (PAN) وتقنية الاتصالات بعيدة المدى. يمكن للمستشعرات والمُشغلات في بيئة إنترنت الأشياء الاتصال عبر الأسلاك أو من خلال **شبكات المنطقة الشخصية اللاسلكية** (WPAN).



شكل 5.22: شبكة المنطقة الشخصية اللاسلكية

شبكة المنطقة الشخصية (Personal Area Network - PAN):
شبكة المنطقة الشخصية هي إحدى شبكات الحاسب المستخدمة لتوصيل الأجهزة الإلكترونية داخل مساحة عمل المستخدم.

وزارة التعليم
Ministry of Education
2023 - 1445

بروتوكولات شبكات المنطقة الشخصية اللاسلكية غير المستندة إلى عنوان

Non-IP Based WPANS Protocols

زيجبي Zigbee

يُعدُّ بروتوكول زيجبي أحد بروتوكولات (WPAN) القائم على أساس معيار (IEEE 802.15.4) الذي صُمّم لشبكات إنترنت الأشياء التجارية والسكنية ذات التكلفة والطاقة والمساحة المحدودة. يمكن لزيجبي تكوين الشبكات، واكتشاف الأجهزة، وتأمين وإدارة الشبكة، ولكن بروتوكول زيجبي لا يوفر خدمات نقل البيانات أو بيئة لتنفيذ تطبيقات معينة. تُعدُّ زيجبي **شبكة متداخلة** (Mesh Network) ذاتية الإصلاح، ويوضح الجدول الآتي المكونات الرئيسة لهذه الشبكة:

جدول 5.3: المكونات الرئيسة لشبكة زيجبي

المُكون	الوصف
مُتحكم زيجبي (Zigbee Controller – ZC)	جهاز عالي القدرة يستخدم لبناء وظائف الشبكة والبدء بها على شبكة زيجبي، قادر على تعيين عناوين الشبكة المنطقية والسماح للعُقد بالانضمام إلى الشبكة أو مغادرتها.
مُوجه زيجبي (Zigbee Router – ZR)	يعالج هذا المكون الاختياري جزءًا من الشبكة المتداخلة عن طريق تعيين عناوين الشبكة المنطقية والسماح للعُقد بالانضمام إلى الشبكة أو الخروج منها.
جهاز زيجبي طرفي (Zigbee End Device – ZED)	يُعدُّ هذا الجهاز بمثابة نقطة نهاية بسيطة ومباشرة ذات قدرة على التواصل مع الوسيط. من هذه الأجهزة مفتاح الإضاءة ومُنظم الحرارة.

يعالج زيجبي ثلاثة أنواع مُختلفة من حركة البيانات:

مُستشعر

1. البيانات الدورية: يُحدّد معدل التسليم الدوري للبيانات أو إرسالها من خلال التطبيقات (على سبيل المثال المُستشعرات التي تُرسل البيانات بصورة دورية). تُنتج بيانات متقطعة عند حدوث تطبيق أو مُحفزات خارجية بوتيرة عشوائية.

مفتاح الإضاءة

2. البيانات المُتقطعة: يُعدُّ مفتاح الإضاءة مثالًا جيدًا على البيانات المتقطعة المثالية لزيجبي.

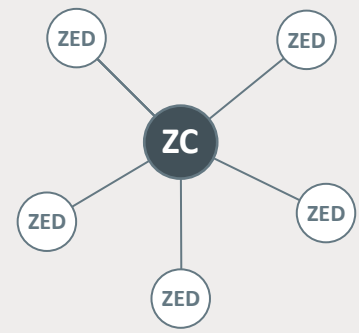
الفأرة

3. بيانات زمن الانتقال المنخفض المتكررة: يُعيّن زيجبي فترات زمنية للإرسال، ويمكن أن يكون بزمن انتقال منخفض جدًا، مما يجعله مناسبًا لأجهزة الفأرة ولوحات المفاتيح.

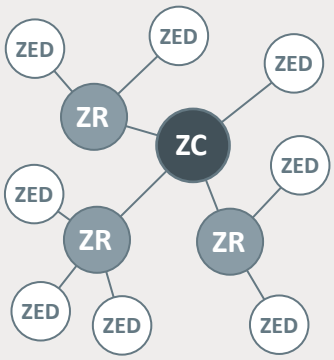
توجد ثلاث هيكليات أساسية لزيجبي:

جدول 5.4: هيكليات زيجبي

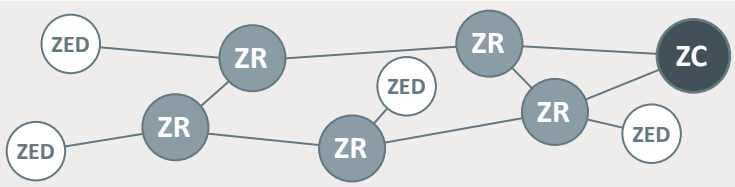
الهيكلية	الوصف
هيكلية النجمة (Star Topology)	يحتوي مُتحكم زيجبي على واحد أو أكثر من أجهزة الزيجبي. يمتد إلى نقطتين فقط، مما يَحد المسافة بين العُقد. ويتطلب أيضًا وسيلة ارتباط يمكن الاعتماد عليها مع نقطة عُطل مفردة في متحكم زيجبي.
هيكلية الشجرة العنقودية (Cluster Tree topology)	وهي شبكة **متعددة النقاط** (Multi-hop) تستخدم أجهزة **الإرشاد** (Beacons) لتوسيع التغطية والمدى. تُعدُّ أجهزة الزيجبي بمثابة نقاط نهاية، ويمكن أن تحتوي عقدة **موجه الزيجبي** (ZR) وعقدة **متحكم الزيجبي** (ZC) على عُقد فرعية. تتواصل العُقد الفرعية مع العُقد الرئيسة فقط، ويمكن للعقد الرئيسة التواصل مع العُقد الفرعية **لأعلى** (Upstream) أو **لأسفل** (Downstream) منها، وتُشكِّل **نقطة العُطل المركزية** (Central failure point) مشكلة في هذه الهيكلية.
هيكلية الشبكة المتداخلة (Mesh Topology)	يمكن توجيه أي جهاز مصدري إلى أي جهاز بصفته وجهة، وذلك باستخدام طرق **التوجيه المستند إلى الأشجار** (Tree-based routing) و **التوجيه المستند إلى الجداول** (Table-based routing). يجب تشغيل موجات متحكمات الزيجبي وموجهاتها طوال الوقت لتنفيذ وظائف التوجيه، مما يؤدي إلى استنزاف عمر البطارية. يُسمح للموجهات الموجودة في نطاق محدد بالتفاعل بشكل مباشر، وتكمن الفائدة الأساسية في هذه الهيكلية في إمكانية توسُّع الشبكة ووجود مسارات متعددة للبيانات.



هيكلية النجمة



هيكلية الشجرة العنقودية



هيكلية الشبكة المتداخلة

شكل 5.23: هيكليات زيجبي

القفزة (Hop):
تحدث القفزة عندما تُمرّر حزمة من قطاع في الشبكة إلى قطاعٍ آخر.

المنارة (Beaconing):
ترمز المنارة في الشبكات إلى ما يشبه جهاز الإرشاد، وذلك عبر القيام بالبث الرقمي بصورة دورية.

البلوتوث Bluetooth

البلوتوث هي تقنية اتصال لاسلكية منخفضة الطاقة تُستخدم على نطاق واسع في الأجهزة الإلكترونية مثل الهواتف المحمولة ووحدات التحكم في الألعاب ولوحات المفاتيح، وقد استُخدم البلوتوث على نطاق واسع في إنترنت الأشياء لإرسال الإشارات عند تشغيلها في وضع **الطاقة المنخفضة** (Low Energy – LE)، وذلك في أجهزة **الإرشاد** (Beacons)، والمُستشعرات اللاسلكية وأنظمة تتبع المركبات والأصول الأخرى، وأجهزة التحكم عن بعد، وأجهزة المراقبة الصحية، وأجهزة الإنذار. تتميز شبكات البلوتوث الشخصية اللاسلكية بحصول ما يطلق عليه اسم: **الأحداث** (Events)، ويُعدّ **الإعلان** (Advertising) و**التوصيل** (Connecting) بمثابة الحدثين الرئيسين في تلك الشبكات.

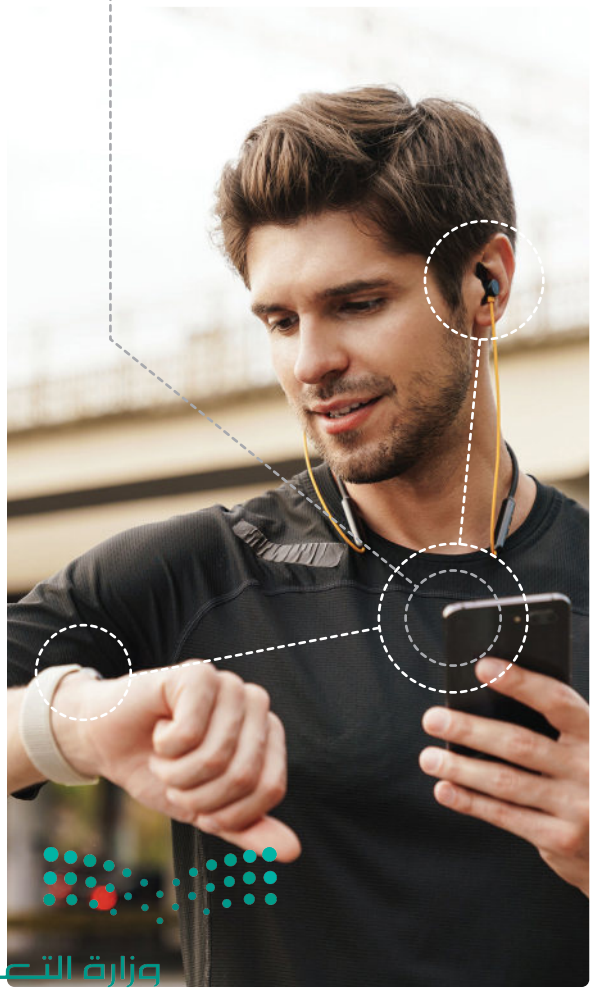
الإعلان Advertising

يبدأ الإعلان بوجود جهاز يطلب القيام بعملية اقتران مع الأجهزة الأخرى التي تقوم بالمسح، أو بإرسال رسالة تحتوي على معلومات التعارف.

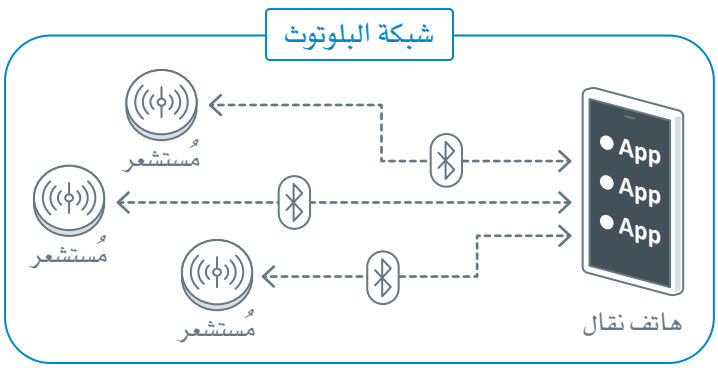
التوصيل Connecting

يصف هذا الحدث عملية اقتران الجهاز بجهازٍ آخر يسمى بالمُضيف.

يمكن للجهاز في وضع الطاقة المنخفضة إجراء اتصال كامل باستخدام قناة الإعلان فقط. قد يكون هذا الاتصال أيضًا اتصالًا رسميًا ثنائي الاتجاه بين الأجهزة، وستبدأ الأجهزة حينها بالقيام بما يسمى بإجراء التكوين لإجراء الاتصال، وذلك من خلال الاستماع إلى حزم الإعلانات. في هذه الحالة، يُعدّ الجهاز المُستمع بادئًا، وإذا أرسل المُعلن حدثًا إعلانيًا قابلًا للاتصال، يجوز للجهاز البادئ إرسال طلب اتصال باستخدام نفس القناة التي استُلمت الحزمة الإعلانية من خلالها. يمكن للمُعلن أن يقرر بعد ذلك ما إذا كان سيُنشئ الارتباط أم لا، وإذا أُنشئ الارتباط، ينتهي الحدث الإعلاني، وحينها يُشار إلى البادئ باسم: رئيسي (Master) والمُعلن باسم: **فرعي** (Slave). يطلق على مثل هذا الاتصال بمصطلحات البلوتوث تسمية **بيكونيت** (piconet)، حيث تحدث أحداث الاتصال بين الجهاز الرئيس والفرعي على نفس قناة البداية، وبعد نقل البيانات وانتهاء حدث الاتصال، يمكن تغيير التردد لتحديد قناة جديدة للمُرسل والمُستقبل.



شكل 5.25: توصيل البلوتوث



شكل 5.24: شبكة البلوتوث

بروتوكولات شبكات المنطقة الشخصية اللاسلكية IP Based WPANS Protocols

الاصدار السادس لبروتوكول الإنترنت عبر شبكات المنطقة الشخصية اللاسلكية منخفضة الطاقة 6LoWPAN

تُصمّم شبكات (IP) عبر أنظمة اتصالات ترددات لاسلكية منخفضة الطاقة لتعمل مع الأجهزة ذات الطاقة والقدرات المحدودة التي لا تتطلب خدمات شبكات ذات نطاق ترددي عالٍ. يتوافق هذا البروتوكول مع العديد من اتصالات شبكات (WPAN) بما فيها معايير (IEEE.802.15.4)، وتقنيات **البلوتوث** (Bluetooth)، وتقنيات **الترددات اللاسلكية الفرعية واحد جيجاهرتز** (sub-1 GHz RF)، بالإضافة إلى الاتصالات عبر خطوط الكهرباء (Power Line Controller – PLC). تتمثل الميزة الأساسية لبروتوكول (6LoWPAN) في أن معظم المستشعرات الأساسية تعمل بتوافق مع نظام عنونة (IP) وبذلك يمكنها أن تعمل كعناصر في الشبكة عبر موجهات الشبكة المحلية أو الواي فاي أو شبكات الجيل الثالث وشبكات(LTE)، وشبكات الجيل الرابع. يمكن لعنونة (IPV6) تغطية ما يصل إلى 50 مليار جهاز متصل بالإنترنت، مما يسمح لها بالاستمرار كنظام للعنونة في المستقبل، وبالتالي إتاحة التوسع المطلوب في نشر إنترنت الأشياء.

تُعدّ شبكات (6LoWPAN) شبكات متداخلة تبنى على جوانب شبكات أكبر. تتميز هذه الشبكات بهيكلياتها المرنة مما يسمح بوجود شبكات **مخصصة** (Ad hoc) و**مفككة** (Disjoined) دون اشتراط الارتباط بالإنترنت أو بأنظمة أخرى. يمكن لهذه الشبكات الارتباط بالشبكة الرئيسة أو بالإنترنت من خلال ما يسمى **بموجهات طرفية** (Edge routers). يمكن للموجهات الطرفية المختلفة توصيل شبكات (6LoWPAN) متعددة من خلال ما يُعرف باسم **التوجيه المتعدد** (Multi-homing)، ويمكن إنشاء الشبكات المخصصة دون الحاجة إلى الوصول إلى الإنترنت من الموجه الطرفي، حيث تُنشئ الموجهات الطرفية شبكات (6LoWPAN) متداخلة على محيط الشبكات التقليدية الأكبر حجمًا، ويمكنها أيضًا تسهيل تبادلات عناوين (IPV6) إلى (IPV4) عند الضرورة. يتم التعامل مع حزم البيانات بشكل مشابه لشبكة عناوين(IP)، والتي تُقدِّم بعض المزايا مقارنة بالبروتوكولات المعروفة الأخرى. تشترك جميع العُقد داخل شبكة (6LoWPAN) في بادئة (IPv6) التي أنشأها الموجه الطرفي. يُسجّل العُقد مع الموجهات الطرفية بشكل مستمر خلال مرحلة اكتشاف الشبكة. تحكم مرحلة اكتشاف الشبكة التفاعل بين المضيفين والموجهات في شبكة (6LoWPAN) المحلية. وتُمكّن خاصية **التوجيه المتعدد** (Multi-homing) عدة موجهات (6LoWPAN) من تشغيل الشبكة؛ على سبيل المثال، عندما يتطلب تجاوز العطل أو التجاوز عن الخطأ وسائط مختلفة (الجيل الرابع و واي فاي).

بروتوكول التشعب Thread

التشعب هو بروتوكول لشبكات إنترنت الأشياء يعتمد على (6LoWPAN) IPV6. الهدف الأساسي لهذا البروتوكول هو إتاحة أتمتة المنازل والشبكات المنزلية، ويمكن وصف التشعب بأنه عنوان (IP) يستند إلى معايير وهيكلية (IEEE 802.15.4) و(6LoWPAN). يتشابه التشعب مع زيجبي ونسخ 802.15.4 الأخرى، ولكنه يختلف من حيث قابليته لعنونة(IP). يعتمد هذا البروتوكول على البيانات والطبقات المادية للمعايير التقنية 802.15.4 وخصائص الأمان والتوجيه لشبكات(6LoWPAN). يعتمد التشعب أيضًا على هيكلية **الشبكة المتداخلة** (Mesh Network) مما يجعله خيارًا عمليًا لأنظمة الإضاءة المنزلية الذكية، وذلك بسعة تصل إلى 250 جهاز لكل شبكة. يتميز بروتوكول التشعب بقابلية عنونة (IP) في الأجهزة المختلفة بما فيها المُستشعرات الصغيرة للغاية، وأنظمة تشغيل الأتمتة المنزلية، كما يتميز بتوفيره في استهلاك الطاقة ؛ لأن البروتوكول لا يتطلب استمرار تنفيذه في طبقة الشبكة. يعني هذا أيضًا أن الموجه الطرفي الذي يستضيف الشبكة المتداخلة لا يحتاج إلى التعامل مع بروتوكولات طبقة التطبيقات، مما يحد من متطلبات الطاقة والمعالجة، ويُعدُّ هذا البروتوكول آمنًا جدًا نظرًا لكونه متوافقًا مع (IPV6) ولكون جميع الاتصالات مُشفرة باستخدام **معيار التشفير المتقدم** (Advanced Encryption Standard – AES).

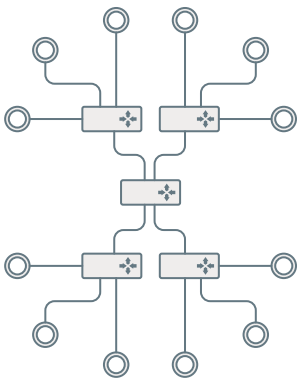
شكل 5.28 شبكات المنطقة الشخصية اللاسلكية

شبكات وبروتوكولات الاتصالات بعيدة المدى
Long Range Communication Networks and Protocols

تربط **شبكات المنطقة الشخصية اللاسلكية** (WPAN) و**شبكات المنطقة المحلية اللاسلكية** (Wireless Local Area Networks – WLAN) المُستشعرات بشبكة محلية، ولكن ليس بالضرورة بشبكة الإنترنت أو بأنظمة الشبكات الأخرى. تشمل بيئة إنترنت الأشياء المُستشعرات والمُشغلات والكاميرات والأدوات الذكية المُدمجة والمركبات، وحتى الروبوتات التي تعمل في الأماكن النائية. لقد أصبح من المُسلَّم به أنّ علينا الاعتياد على التعامل مع **شبكة المنطقة الواسعة** (Wide Area Network – WAN) على المدى الطويل.

تقنية لوراوان LoRaWAN

تُعدُّ التقنيات اللاسلكية **منخفضة الطاقة واسعة النطاق** (Low Power Wide Area – LPWA) مثالية لنقاط النهاية (الأجهزة المختلفة) طويلة المدى التي تعمل بالبطارية. عادةً ما يشار إلى هيكلية (LoRaWAN) باسم هيكلية **نجمة النجوم** (Star of Stars). تقوم نقاط النهاية بتبادل الحزم عبر بوابات تعمل كجسور، وذلك بوجود خادم شبكة (LoRaWAN) مركزي. تتصل نقاط النهاية مباشرة بواحدة أو بأكثر من البوابات، بينما تتصل المداخل **بالشبكة الخلفية** (Backend Network) عبر اتصالات (IP) العادية. يمكن في هذه التقنية استلام الحِزم نفسها ونقلها بواسطة العديد من البوابات، وفي حال تلقي حزم مُكررة، يكون خادم الشبكة مسؤولًا عن إلغاء التكرار، وتوفر تقنيات (LPWA) المفتوحة المتوفرة خيارات جديدة لشبكات الشركات الخاصة والبث ومُقدِّمي الخدمات المتنقلة وغير المتنقلة لنشر البُنى التحتية لإنترنت الأشياء وحلولها. تتوسع بيئة نقاط النهاية بسرعة، وستكون بلا شك العامل الحاسم بين تقنيات وحلول التقنيات اللاسلكية **منخفضة الطاقة واسعة النطاق** (LPWA) المختلفة مثل (LoRaWAN)، ويُعدّ بناء البنى التحتية وتطويرها المحلية منها والإقليمية أمرًا حيويًا لتفعيل استخدام إنترنت الأشياء على نطاق استهلاكي أوسع، ويتحمل مسؤولية ذلك الأشخاص المسؤولون عن المدن الذكية، وهيئات تنظيم البث والإذاعة، ومقدمو خدمات الاتصالات الخلوية والعادية.



شكل 5.27: هيكلية LoRaWAN "نجمة النجوم"

الشبكات الخلوية (الجيل الخامس)
Cellular Networks (5G)

من أكثر أنواع الاتصالات شيوعًا استخدام الترددات الخلوية خاصةً البيانات الخلوية. فقبل تطور التقنية الخلوية، كانت تغطية أجهزة الاتصالات المحمولة محدودة، واستُخدمت مساحة ترددات مشتركة، فالأجهزة كانت بمثابة أجهزة إرسال لاسلكي ثنائية الاتجاه. ثم أصبحت الشبكات الخلوية ممتازة في نقل البيانات في كلا الاتجاهين بسرعات عالية، ولكن على حساب النطاق واستهلاك البطارية. يُعدُّ **الجيل الخامس** (5G) الجيل التالي من تقنية الاتصالات القائمة على بروتوكول الإنترنت والتي يتم تطويرها لتحل محل شبكات الجيل الرابع الخلوية، بالإضافة إلى ذلك تعمل شبكات الجيل الخامس على تحسين النطاق الترددي ووقت الاستجابة والكثافة وتقليل نفقات المستخدم، وتهدف إلى أن تكون معيارًا شاملًا واحدًا يشمل جميع الخدمات والفئات الخلوية، بدلًا من إنشاء خدمات وتصنيفات مميزة لكل حالة استخدام.

جدول 5.5: السمات الرئيسة لشبكات الجيل الخامس الحديثة

الميزات	الوصف
	النطاق العريض المتنقل المحسّن (Enhanced Mobile Broadband - eMBB)
	اتصالات فائقة الموثوقية وذات زمن انتقال منخفض (Ultra-Reliable and Low-Latency Communications - URLLC)
	اتصالات نوع الآلة الضخمة (Massive Machine Type Communications - mMTC)

تمرينات

1

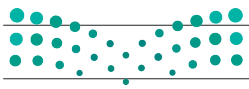
حدد الجملة الصحيحة والجملة الخاطئة فيما يلي:	صحيحة	خاطئة
1. تحتوي هيكلية شبكة oneM2M على طبقة بيانات.	○	○
2. يمكن استخدام خدمات الشبكة الافتراضية الخاصة (VPN) في طبقة الخدمات لهيكلية (oneM2M).	○	○
3. يمكن أن تحتوي طبقة التطبيقات على خدمات المراقبة في أنظمة إنترنت الأشياء العالمي.	○	○
4. تستخدم تقنيات (NFC) للاتصالات بعيدة المدى بين الأجهزة.	○	○
5. يتصل بروتوكول زيجبي عبر قنوات شبكة ببروتوكول (UDP).	○	○
6. يُعدُّ موجه زيجبي مسؤولًا عن خصائص الإصلاح الذاتي للشبكات المتداخلة.	○	○
7. يُرسل الحدث الإعلاني لاتصالات البلوتوث حزم بيانات إلى الأجهزة المجاورة.	○	○
8. لا يُعدّ التشعب (Thread) بروتوكولًا قائمًا على الشبكة.	○	○
9. لا تحتاج أنظمة شبكات المدن الذكية إلى شبكات وبروتوكولات اتصالات بعيدة المدى.	○	○
10. تُصنَّف شبكات الجيل الخامس (5G) بأنها منخفضة استهلاك الطاقة.	○	○

2 صنِّف الطبقات الرئيسة لهيكلية (oneM2M) لأنظمة إنترنت الأشياء.

3 حلل الطبقات الرئيسة لهيكلية أنظمة إنترنت الأشياء العالمي.

4 حدِّد الخصائص الرئيسة لتقنية تحديد الترددات الراديوية (RFID) وتقنية الاتصال قريب المدى (NFC).

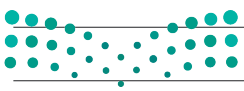
5 صنِّف النوعين الرئيسين لشبكات المنطقة الشخصية اللاسلكية (WPANS)، واعرض بعض الأمثلة على كل نوع.



6 حدِّد المكونات الرئيسة الثلاثة لشبكة زيجبي (Zigbee).

7 قارن بين الحدثين الأساسيين اللذين يحدثان أثناء الاتصال بالبلوتوث.

8 قدِّم وصفًا لبروتوكولي (WPANS) الرئيسين المُستندين إلى عنونة (IP).



9 وضِّح هيكلية "نجمة النجوم" التي تستخدمها شبكات(LoRaWAN).

10 قدِّم تحليلًا لكيفية تطور شبكات الجيل الخامس (5G) من تقنيات شبكات الجيل الرابع (4G).

رابط الدرس الرقمي

www.ien.edu.sa

الدرس الثالث

أمان وخصوصية أنظمة إنترنت الأشياء

الأمان Security

يُشكِّل الإنترنت وأنظمة إنترنت الأشياء والخدمات السحابية والأنظمة الإلكترونية الملموسة (CPSs) والأجهزة المحمولة ملامح الحياة الحديثة في القرن الحادي والعشرين، فقد أسهمت التقنية في التواصل بين أطراف العالم بما يعود بالفائدة على كافة المجتمعات. ولكن هذا التطور التقني أدّى إلى تَمكُّن مجرمي الإنترنت من استغلال العديد من نقاط الضعف لتهديد مستخدمي هذه التقنيات. يتزايد تأثير إنترنت الأشياء على المؤسسات ونماذج الأعمال، ويعتمد إنترنت أشياء الشركات على ثقة المستهلك. ومع ذلك تُقدّم العديد من المنتجات والخدمات التقنية إلى الأسواق بصورة متسارعة مع اهتمام غير كافٍ بأمان وخصوصية المستخدمين، فالأمان يُعدُّ جزءًا مُهمًا من عملية التصميم من أدنى المستويات، إلى أعلاها؛ ولهذا يجب إنشاء السياسات والبروتوكولات والمعايير الأمنية بموازاة أي تطور تقني لدعمه وحمايته.

يعرض الجدول الآتي أسس الأمان في إنترنت الأشياء:

النظام الإلكتروني الملموس (Cyber Physical System - CPS):
هو نظام محوسب يتحكم أو يراقب آلية معينة باستخدام خوارزميات محوسبة.

جدول 5.6: أُسس الأمان في إنترنت الأشياء

الأساس	الوصف
الثقة	السماح للمستخدمين أو الخدمات المُصرَّح لها فقط بالوصول إلى الجهاز أو البيانات.
التحقق من الهوية	التحقق من هوية الأفراد والخدمات و "الأشياء".
الخصوصية	الحفاظ على خصوصية جهاز المستخدم ومعلوماته الشخصية وبياناته الحساسة.
الحماية	حماية الأجهزة والمستخدمين من الأضرار المادية والمالية والمتعلقة بالسمعة.

تحديات أنظمة إنترنت الأشياء المُرتكزة على المستخدم

User-Centered Challenges of IoT Systems

شكل 5.28: المساءلة في إنترنت الأشياء

أصبحت التدابير الأمنية التقليدية غير كافية لتوفير الأمان الكافي للعالم الحديث المتطور تقنيًا والمُتصل معًا، فعلى النقيض من الأجهزة الإلكترونية التقليدية، فإنّ أجهزة إنترنت الأشياء تتفاعل معًا ومع الخدمات على الإنترنت. لا يمكن تحقيق الفوائد المرجوة من تطبيق الأنظمة والتقنيات الحديثة دون الحصول على ثقة المستخدمين بهذه التقنيات الحديثة؛ لذلك تُعدّ المساءلة أمرًا بالغ الأهمية لزرع الثقة بين المستخدمين ومُنشئي أنظمة إنترنت الأشياء، ويسهم تعقيد تدفق البيانات الموزعة، وآليات التوافق غير الكافية، ونقص المعلومات بالنسبة للمستخدمين في ظهور الحاجة إلى وجود نظام للمساءلة في إنترنت الأشياء.

الأمان في إنترنت الأشياء والجرائم الإلكترونية

IoT Security and Cybercrime

تُعدُّ البُنية التحتية للإنترنت بمثابة المُنشآت الحيوية داخل الحدود الإقليمية للدول ذات السيادة. تَمرُّ البيانات المتدفقة عبر هذه البنية التحتية عبر العديد من الدول الأخرى، مما يشكِّل مصدرًا للقلق فيما يتعلق بسلامة البيانات، فالجرائم الإلكترونية لا تعرف الحدود، بل وتتجاوز الحدود الجغرافية بسهولة. ومن الملاحظ أن القوانين المتعلقة بحماية البيانات وأمن المعلومات تتباين بين الدول المختلفة؛ لذا، تمثِّل الفجوة الواسعة بين التشريعات القانونية والتقنية عقبة رئيسة في مكافحة الجريمة الإلكترونية، ويواجه نظام العدالة لمكافحة هذه الجرائم تحديات كثيرة ويتسم بالبطء وعدم القدرة على تنظيم هذا الفضاء الإلكتروني. كما أن سرعة تبني التقنية في المجتمعات تفوق السرعة التي تُوضع بها السياسات والتشريعات لتنظيم وضبط هذه التقنية؛ لهذا السبب، يُتحكّم في الفضاء الإلكتروني والتقنية من خلال دمج مجموعة من القوانين غير المُلائمة والقديمة والمتناقضة أحيانًا، كما يصعب تحقيق توافق دولي أو إقليمي في الآراء حول معايير وقوانين السلامة الإلكترونية نظرًا لأن لكل دولة معاييرها ومعتقداتها وممارساتها المستقلة، مما يقرر رؤى مختلفة لعملية تنظيم الفضاء الإلكتروني. تعزز بعض الدول سيادتها على الفضاء الإلكتروني بحجة أن سياساتها الوطنية العامة تنطبق على الفضاء الإلكتروني أيضًا، وبأن الدولة يجب أن تكون قادرة على تنظيم كيفية استخدام الأفراد والشركات للإنترنت داخل حدودها.

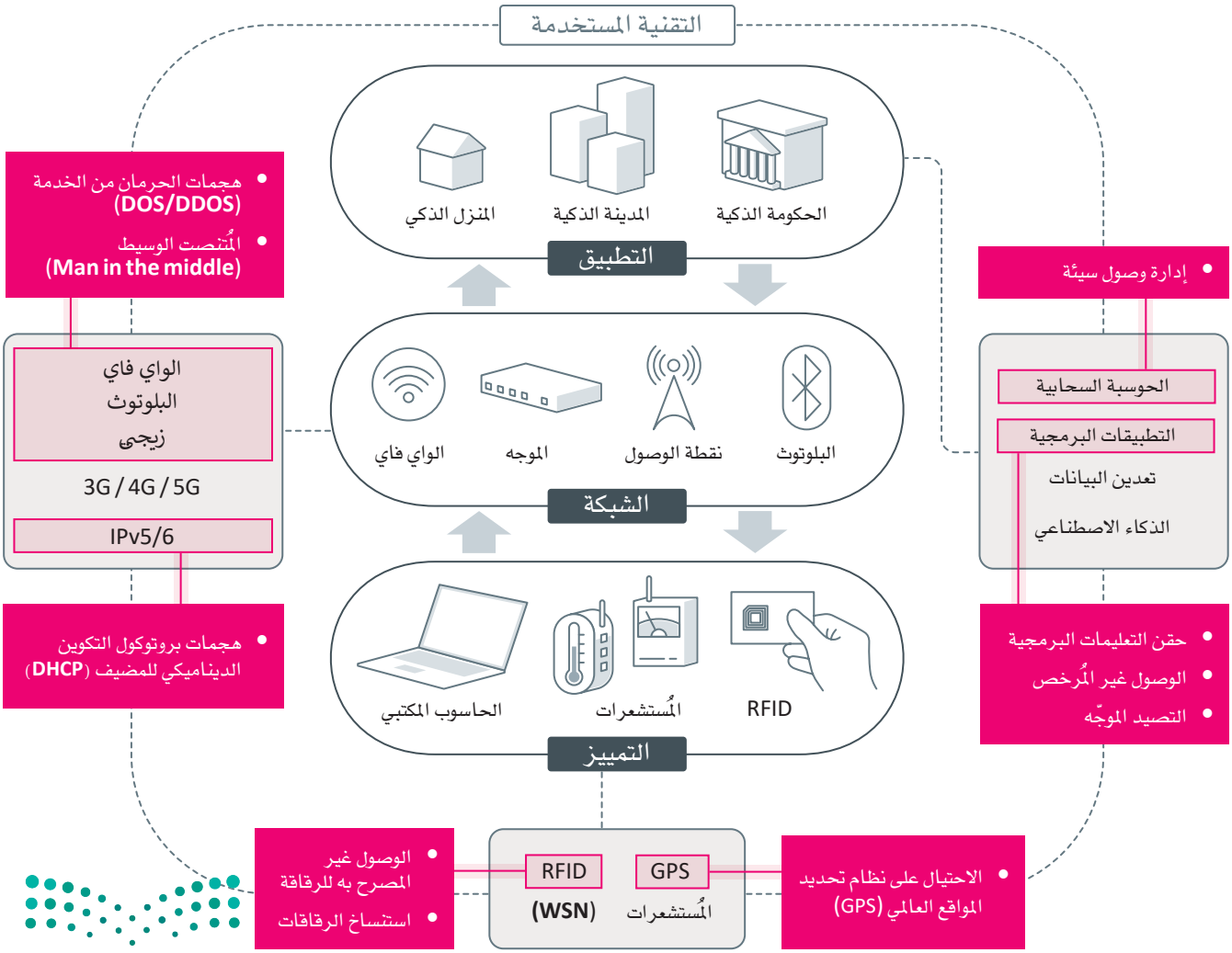
شكل 5.29: هجوم واختراق برامج الفدية الضارة

التحديات الهيكلية للأمان في إنترنت الأشياء

Architectural Challenges of IoT Security

يتطلب إنترنت الأشياء مجموعة من المعايير ووجود هيكلية مُحددة جيدًا مع واجهات ونماذج بيانات وبروتوكولات تناسب التنوع في الأجهزة والبروتوكولات والخدمات المَعنية. يمكن حدوث العديد من الهجمات عند اتصال أجهزة إنترنت الأشياء بالخدمات السحابية وتبادلها للبيانات لأول مرة، كما قد تتسبب خصائص أجهزة إنترنت الأشياء المختلفة في وجود مخاطر ومشكلات أمنية، حيث تقدم قابلية التنقل والاعتماد المتبادل والخصائص المماثلة الأخرى تحديات ومخاطر مختلفة، وتشمل هذه التحديات والمخاطر نقاط الضعف في **البرامج الثابتة** (Firmware)، والتخزين، وقوة المعالجة، وهجمات الشبكة والقواعد والمعايير، والتي تتطلب المزيد من الدراسات الإضافية. يتطلب إنترنت الأشياء المزيد من أجهزة النقل بين عنونة (IPv4) إلى (IPv6)، مما يستلزم زيادة **عرض النطاق الترددي** (Bandwidth) للشبكات، حيث يؤدي اعتماد عنونة (IPv6) وتقنية الجيل الخامس، واستخدام الجيل الجديد من الاتصالات فائق السرعة إلى ظهور مخاطر وصعوبات إضافية.

توضِّح الرسوم التوضيحية الآتية كيفية تطور هيكلية نظام بسيط إلى نظام مُعقَّد. وكما تلاحظ فإن إضافة كل طبقة من التعقيد تتسبب في ظهور نقاط ضعف جديدة لمكونات النظام.



شكل 5.30: الثغرات الأمنية في أنظمة إنترنت الأشياء

شبكات الجيل الخامس وأمان إنترنت الأشياء

5G Networks and IoT Security

تُعدّ تقنية **الجيل الخامس** (5G) تقنية واعدة بصفتها المستقبل الواعد للتطور العالمي للاتصالات المتنقلة. إن تقنية الجيل الخامس هي المكون الرئيس لاتصال المجتمع بالشبكات وبأنظمة **إنترنت الأشياء واتصال آلة إلى آلة** (IoT / M2M)، مما يُتيح الوصول السريع إلى المعلومات والخدمات، وتهدف تقنية الجيل الخامس إلى تحقيق الاتصالات المحمولة بين البشر في كل مكان باستخدام أي جهاز أو تطبيق محوسب يمكنه أن يتصل بالإنترنت، مثل **إنترنت الأشياء** (IoT)، و**ويب الأشياء** (Web of Things – WoT)، ونظرًا لتطور شبكات الجيل الخامس، أصبح من الطبيعي ظهور مشكلات تتعلق بتأثير الجيل الخامس على الأمان في اتصالات أجهزة إنترنت الأشياء، وأصبحت هناك حاجة إلى برمجيات وسيطة لإنترنت الأشياء، ومعايير أمنية لتنفيذ طرق جديدة لربط مختلف الشبكات والأجهزة المُعرّفة. وهكذا ومع وجود بُنية تحتية للشبكة أفضل وأسرع، سيكون هناك تفاعل أكبر بين الأشياء، لا سيما مع توزيع المعالجة عبر الخدمات السحابية، مما سيؤدي إلى إحداث تأثير كبير فيما يتعلق بأمن البيانات وتمكين تطوير تطبيقات جديدة تعمل على تحسين حياة البشر.

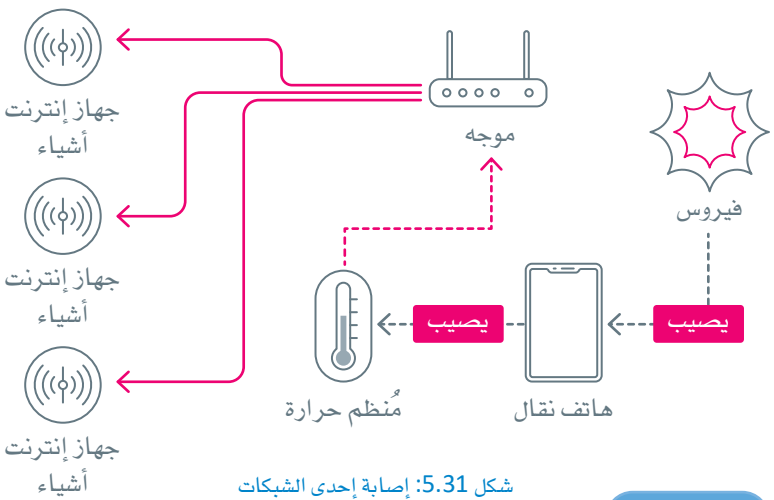
يوضِّح الجدول الآتي المخاوف الأمنية الرئيسة لشبكات الجيل الخامس الخاصة بأنظمة إنترنت الأشياء.

جدول 5.7: المخاوف الأمنية لشبكات الجيل الخامس لأنظمة إنترنت الأشياء

المخاوف	الوصف
أمن البيانات الضخمة	تنشئ أنظمة إنترنت الأشياء كميات كبيرة من البيانات غير المتجانسة وبصورة مستمرة. تتوسع متطلبات حركة البيانات للاتصالات المتنقلة في أنظمة إنترنت الأشياء بشكل كبير، ولذلك يُعدُّ ابتكار طريقة فعّالة لإدارة هذا الكم الكبير من البيانات التي أنشأتها أنظمة إنترنت الأشياء أمرًا ضروريًا، وتوفر تقنيات شبكات الجيل الخامس إمكانية نقل البيانات بتكلفة أقل بكثير لكل بت من البيانات مقارنةً بالشبكات السابقة، ولكنها تخلق الحاجة إلى معايير بروتوكولات آمنة لإدارة وتنظيم هذا الكم الكبير من البيانات بشكل صحيح، وذلك من خلال وضع حلول أمنية تشمل إنترنت أشياء قائم على الجيل الخامس.
حماية الأجهزة والتطبيقات	تمثل حماية العديد من الأجهزة والتطبيقات صعوبة إضافية. تتميز أنظمة إنترنت الأشياء القائمة على الجيل الخامس بقدرتها على دعم عدد أكبر بكثير من الأجهزة والتطبيقات مما هو موجود الآن، حيث ستؤدي الاتصالات بين ملايين الأجهزة والتطبيقات الإضافية إلى بروز مخاوف أمنية جديدة، فمثلًا قد يتسبب حدوث هجوم إلكتروني بسيط في منع السكان من مغادرة منازلهم وسياراتهم وغيرها من الأشياء المرتبطة بالشبكة.
حماية قنوات الاتصال	الحفاظ على خصوصية جهاز المستخدم والمعلومات الشخصية والبيانات الحساسة.

الخصوصية Privacy

تُشكِّل مسألة الأمان عبر الإنترنت مصدر قلقٍ وتحديًا كبيرًا في بيئة إنترنت الأشياء. ومن ناحيةٍ أخرى فإن الحفاظ على خصوصية بيانات المستخدمين يُشكِّل تحديًا كبيرًا أيضًا يتطلب اهتمامًا إضافيًا. قد تتعرض خصوصية المستخدمين لإنترنت الأشياء للخطر إذا تم تسريب البيانات الشخصية إلى جهات غير مُصرح لها، ونظرًا لتنوع الأجهزة المتصلة بإنترنت الأشياء ونقاط الضعف الكامنة في الأجهزة والبرامج، فإن حماية خصوصية المستخدم النهائي تُمثل العديد من التحديات الأمنية. يسمح الكم الهائل من البيانات الشخصية المُجمَّعة من أنظمة البيانات الضخمة للمؤسسات بدمج مجموعات البيانات المختلفة، مما يزيد من القدرة على تحديد الأفراد، وتزداد القدرة على استخراج مجموعات البيانات وتحليل حجمها وتغيرها بشكلٍ يومي. وللتغلب على هذا، فإن من الحِكمة التأكد من إخفاء البيانات التي يمكن أن تدل على شخصية صاحبها وجعل تلك **البيانات مجهولة المصدر** (Anonymized Data)، كما يجب على المؤسسات التي تستخدم البيانات مجهولة المصدر إجراء تقييم شامل للمخاطر وتطبيق تقنيات أمنية فعّالة، ويشمل ذلك مجموعة متنوعة من الضمانات التقنية، مثل إخفاء البيانات، والتسمية المستعارة، فضلًا عن الضمانات القانونية والتنظيمية.



شكل 5.31: إصابة إحدى الشبكات

مثال

يمكن للقراصنة اختراق شبكة لإنترنت الأشياء، وجمع البيانات الخاصة عن طريق استغلال آلية التوصيل والتشغيل العالمية (UPnP)، والتي تتطلب تكوينًا بسيطًا دون الحاجة إلى وجود مصادقة للاتصال، ويستغل المتسللون هذه الميزة لإصابة جهاز، ومن ثَم إصابة شبكة إنترنت الأشياء. على سبيل المثال، يمكن للهاتف المحمول المصاب بفيروس أن يتصل بمنظم الحرارة في المنزل الذكي عبر شبكة الواي فاي. يُوصل منظم الحرارة هذا من خلال (UPnP) بالموجه الخاص بالمنزل الذكي، وبالتالي تصاب شبكة إنترنت الأشياء المنزلية بأكملها بهذا الفيروس، مما يؤدي إلى خرقٍ كامل لبيانات المعلومات الخاصة.

إخفاء البيانات (Data masking):

يتم في عملية إخفاء البيانات تغيير البيانات الحساسة. وهكذا فإن البيانات تصبح عديمة الجدوى بالنسبة للمتطفلين غير المصرح لهم، ولكن لا يزال بالإمكان استخدامها من قِبل البرامج والموظفين المعتمدين للمزيد من التحليل.

الأسماء المستعارة (Pseudonymization):

تُستخدم الأسماء المستعارة لإدارة البيانات واستبدال محددات الهوية من تلك البيانات، وبالتالي تحل هذه الأسماء مكان حقول معلومات التعريف الشخصية في سجل البيانات، وذلك باستخدام قيم وأسماء مستعارة.

الخصوصية التفاضلية (Differential Privacy):

يُضاف مقدار عشوائي من الخصوصية، وهو عبارة عن مجموعة بيانات غير ذات أثر على دقة مجموعة البيانات. تُستخدم هذه التقنية لمنع تحديد أي معلومات شخصية للأفراد في مجموعة البيانات.

التوصيل والتشغيل العالمي (Universal Plug and Play - UPnP):

هي خدمة تُمكّن الأجهزة الموجودة على نفس الشبكة المحلية من البحث والاتصال ببعضها تلقائيًا باستخدام بروتوكولات الشبكات القياسية. تُعدُّ الطابعات والموجهات والأجهزة المحمولة وأجهزة التلفاز الذكية من أنواع أجهزة (UPnP).

وزارة التعليم
Ministry of Education
2023 - 1445

شكل 5.32: الأسماء المستعارة وإخفاء البيانات

تُعدُّ حماية البيانات وأمنها أمرًا غير سهل في بيئة إنترنت الأشياء، حيث يعتمد جوهر النظام على وجود واجهة اتصال بين الكائنات الذكية دون تدخُّل بشري. ونظرًا للمعدل المتسارع لتطور مثل هذه الأنظمة، فإن التأخير الملحوظ في أنظمة حماية البيانات وكذلك في وعي المُشرِّعين بالمخاطر العملية المتعلقة بالحماية والأمان ليس مُستغربًا. يوضِّح الجدول الآتي مخاوف الخصوصية الحالية في إنترنت الأشياء والحلول الممكنة لها.

جدول 5.8: مخاوف خصوصية إنترنت الأشياء والحلول الممكنة لها

مخاوف الخصوصية	الحلول المقترحة
جمع البيانات من مصادر مختلفة دون التحقق الدقيق من المُلاءمة أو الدقة.	استخدام الذكاء الاصطناعي للتحقق من دقة البيانات التي يتم جمعها.
تُمكِّن أنظمة البيانات الضخمة المؤسسات من دمج مجموعات بيانات متعددة، مما يعزز احتمال أن تحدد البيانات الأفراد الأحياء.	استخدام مجموعة متنوعة من الإجراءات الأمنية، مثل إخفاء البيانات وإخفاء الهوية والتسمية المستعارة والتجميع، بالإضافة إلى الضمانات القانونية والتنظيمية.
قد يسهم الغموض السائد في عمليات معالجة البيانات والتعقيدات المتعلقة بتحليلات البيانات الضخمة في انعدام الثقة.	تحسين مستوى الشفافية من خلال توفير معلومات حول سياسة الخصوصية قبل معالجة أي بيانات يتم الحصول عليها.
صعوبة تحديد ما إذا كانت الاستخدامات الفعلية للبيانات تتوافق مع الغرض الأصلي الذي تم جمعها لأجله.	قد تقوم المؤسسة بجمع البيانات الشخصية لغرض واحد ثم تحليلها لاحقًا لغرض مختلف تمامًا. في مثل هذه الحالة، يجب إبلاغ المستخدمين بالتغيير وعند الضرورة يجب الحصول على الموافقة.
إن أي انتهاكات أو تهديد لخصوصية المستخدمين سيشكل ضررًا لمصداقية المُنشئين، وتتسبب في فقدان المستخدمين للثقة في المؤسسة والنظام ككل.	تُستخدم الأساليب التقنية مثل بروتوكولات التشفيروتقنية **سلسلة الكُتل** (Blockchain)، ويمكن أيضًا الاستعانة بأنظمة الأمان المادية كأنظمة التحكم في الوصول والمراقبة بالفيديو والسجلات الأمنية.
مراعاة حماية الخصوصية عند تصميم الأنظمة.	إن إجراء تقييم مخاطر الخصوصية يعطي تحذيرات مبكرة لاكتشاف مشكلات الخصوصية.
عدم وجود سياسات وأطر تنظيمية وطنية وإقليمية وعالمية ذات صلة بإنترنت الأشياء، والتي إن وجدت قد تتعارض مع التطور التقني أيضًا.	من الضروري اشتراك الدول والمنظمات الدولية والشركاء المعنيين وخبراء الأمن وإنترنت الأشياء من الصناعة والأوساط الأكاديمية في تطوير حلول لحماية البيانات الشخصية الناتجة عن إنترنت الأشياء.

تمرينات

1

حدد الجملة الصحيحة والجملة الخاطئة فيما يلي:	صحيحة	خاطئة
1. النظام الإلكتروني المادي هو نظام يراقب آلية محددة فقط.	○	○
2. يشمل مبدأ حماية إنترنت الأشياء القيام بالحماية المادية لأجهزة إنترنت الأشياء.	○	○
3. يتم تطبيق قوانين الأمان الإلكتروني بنفس الطريقة في كل الدول.	○	○
4. تُعدّ تقنيات عنونة (IPv6) والجيل الخامس آمنة تمامًا.	○	○
5. يمكن إنشاء تقنيات آلة إلى آلة (M2M) دون أي تدخل بشري.	○	○
6. تُشكِّل الكائنات الذكية (أجهزة إنترنت الأشياء) المُخترقة خطرًا على مستخدميها.	○	○
7. تُعدُّ أنظمة البرمجيات الوسيطة للاتصال بين شبكات الجيل الخامس عرضةً للهجمات الإلكترونية.	○	○
8. تُشفّر البيانات الشخصية التي تُنشأ بواسطة أي كائن ذكي بشكلٍ تلقائي.	○	○
9. تقدم تقنيات إخفاء الهوية بيانات مزيفة لحماية البيانات الحقيقية.	○	○
10. يمكن أن تساعد تقنيات سلسلة الكُتل (Blockchain) في حماية البيانات في أنظمة إنترنت الأشياء الموزعة.	○	○

2 ما المسألة الأكثر إلحاحًا بشأن التطور والانتشار السريع لأنظمة إنترنت الأشياء؟

3 صنِّف المبادئ الأساسية لأمن إنترنت الأشياء.

4 صِف التحدي الرئيس للأمان في إنترنت الأشياء وطبيعة الجرائم الإلكترونية على الإنترنت، وكيف يُمكن التغلب على مثل هذه التحديات؟

5 ميِّز بين الأنواع المختلفة للهجمات المحتملة على كل طبقة من هيكلية إنترنت الأشياء البسيطة.

6 ما التحدي الأمني التقني الأكثر أهمية الذي أسهمت شبكات الجيل الخامس في أنظمة إنترنت الأشياء في ظهوره؟ قدِّم أفكارك أدناه.

7 كيف أسهمت تقنيات البيانات الضخمة في ظهور تحديات جديدة للخصوصية؟

8 صنِّف مخاوف الخصوصية الموجودة في أنظمة إنترنت الأشياء في الوقت الحالي.

المشروع

تُعدّ الرعاية الصحية الذكية من أهم القطاعات التي تعمل على تحسين تقنيات إنترنت الأشياء، حيث ترتبط مجموعة متنوعة من الأجهزة والأنظمة ببعضها وتتبادل كميات كبيرة من البيانات، وتُعدّ البيانات الطبية والحيوية للمرضى من أكثر البيانات خصوصية، والتي يجب على الشركات والحكومات حمايتها بشكلٍ جيد.

1 يستخدم المرضى والأطباء والمراكز الطبية والمستشفيات البيانات الطبية والحيوية داخل تلك المستشفيات والمراكز، أما في الرعاية الصحية الذكية فيمكن الوصول إلى هذه البيانات من أي مكان. دَوِّن أنواع الأجهزة والخدمات والأنظمة التي تنقل البيانات الحيوية الشخصية أو تعالجها أو تُخزنها من خلال أنظمة الرعاية الصحية الذكية.

2 لا تقتصر عملية حماية البيانات الحيوية على شركات التقنية التي تقوم بتطوير أنظمة إنترنت الأشياء، فالحكومات مسؤولة عن توفير التشريعات واللوائح لحماية المواطنين من إساءة استخدام البيانات الشخصية أو اختراقها. ابحث في الإنترنت عن أمثلة للتشريعات التي فرضتها المملكة العربية السعودية لأنظمة الرعاية الصحية الذكية، وعن تشريعات مُشابهة فرضتها دولة أخرى من اختيارك.

3 بعد تدوين ملاحظاتك المتعلقة بالمشكلات المحتملة للأمان والخصوصية في الرعاية الصحية الذكية، والمقارنة بين التشريعات في المملكة العربية السعودية ودولة أخرى، قم بعرضها من خلال إنشاء عرض تقديمي باستخدام باوربوينت (PowerPoint).

ماذا تعلّمت

- كيفية استخدام شبكات مُستشعرات الجسم في تطبيقات الرعاية الصحية الذكية.
- تحديد أنواع مُستشعرات الطائرات دون طيار المستخدمة في الزراعة الذكية باستخدام تطبيقات إنترنت الأشياء.
- تحديد المجالات الرئيسة لهيكلية (oneM2M).
- تمييز الطبقات المختلفة للهيكلية العالمية لأنظمة إنترنت الأشياء.
- تحديد الاختلافات بين تقنية تحديد الترددات الراديوية (RFID) وتقنية الاتصال قريب المدى (NFC).
- تحديد بروتوكولات الشبكة المستخدمة في شبكات المنطقة الشخصية اللاسلكية (WPANS).
- تصنيف الأسس الرئيسة للأمان في إنترنت الأشياء.
- التعرف على تقنيات الأمان المُستخدمة في خصوصية إنترنت الأشياء.

المصطلحات الرئيسة

شبكة المنطقة الواسعة طويلة المدى	LoRaWAN
آلة إلى آلة	Machine To Machine
الاتصال قريب المدى	NFC
هيكلية oneM2M	oneM2M Architecture
شبكة المنطقة الشخصية	Personal Area Network
أسماء مستعارة	Pseudonymization
تحديد الترددات الراديوية	RFID
التشعب	Thread
مركبة جوية دون طيار	UAV
شبكة المنطقة الشخصية اللاسلكية	Wireless Personal Area Network
زيجبي	Zigbee

البلوتوث	Bluetooth
شبكة مُستشعرات الجسم	Body Sensor Network
النظام الإلكتروني الملموس	Cyber Physical System
إخفاء البيانات	Data Masking
الحوسبة الطرفية	Edge Computing
مُخطط كهربية القلب	Electrocardiogram
مخطط كهربية الدماغ	Electroencephalogram
إنترنت أشياء الرعاية الصحية	Internet of Health Things
هيكلية أنظمة إنترنت الأشياء العالمي	IoT World Forum Architecture
IP النسخة السادسة	IPv6

6. برمجة إنترنت الأشياء باستخدام لغة C++

سيتعرف الطالب في هذه الوحدة على تطبيقات الحماية الذكية. وسيتعلم كذلك كيفية برمجة جهاز تحكم الأردوينو الدقيق (Arduino microcontroller) باستخدام لغة C++، وكيفية الانتقال من اللبنات البرمجية إلى هذه اللغة في بيئة محاكاة دوائر تينكركاد (Circuits Tinkercad). وفي الختام سيُنشئ مشروعًا للحماية الذكية بواسطة هذا الجهاز، وسيقوم ببرمجته باستخدام لغة C++.

أهداف التعلُّم

بنهاية هذه الوحدة سيكون الطالب قادرًا على أن:

- يحدد ميزات ومخاطر نظام الأمان في إنترنت الأشياء.
- يتعرف على بعض أجهزة إنترنت الأشياء الأكثر استخدامًا في أنظمة الحماية الذكية.
- يتعرف على أنواع البيانات الشائعة في لغة C++.
- يستخدم المُعامِلات في لغة C++.
- يستخدم الجُمل الشرطية في C++.
- يستخدم التكرارات في C++.
- يُنشئ دالة في C++.
- يحوّل اللبنات البرمجية في بيئة تينكركاد إلى أوامر C++.
- يبرمج نظامًا للحماية الذكية باستخدام لوحة الأردوينو ولغة C++.

الأدوات:

- بيئة محاكاة دوائر أوتوديسك تينكركاد (Autodesk Tinkercad Circuits)

وزارة التعليم
Ministry of Education
2023 - 1445

رابط الدرس الرقمي

www.ien.edu.sa

الدرس الأول

تطبيقات الحماية الذكية ولغة C++

الحماية الذكية Smart Security

يُعدّ نظام الحماية الذكي وسيلة أو عملية لحماية شيء ما باستخدام مجموعة من الأدوات والمكونات التي تعمل معًا. يُمكن لأنظمة إنترنت الأشياء التعامل مع عمليات المراقبة الداخلية والخارجية للبيوت والممتلكات، وتحديد مَنْ يمكنه الوصول إلى البوابات والأبواب من خلال استخدام الأقفال الذكية المُثبتة عليها. على سبيل المثال، يُمكن الاستعانة بأجراس الباب الذكية للتعرف على الزائرين ومخاطبتهم قبل فتح باب المنزل، كما يُمكن دمج كاميرات عالية الدقة يتم تنشيطها بواسطة الحركة في هذه الأدوات لحماية المنزل، وتُوفِّر أنظمة الحماية الذكية تحذيرًا من أي تحركات غير اعتيادية، كما يُمكنها تنشيط إنذار معين أو حتى الاتصال بالشرطة.

الميزات Benefits

توجد العديد من الميزات لتركيب أنظمة الحماية المنزلية الذكية، حيث يتيح إنترنت الأشياء مراقبة المنزل وإدارته عن بُعد عبر تطبيقات الهاتف المحمول. تستخدم أجهزة الحماية الذكية تقنيات الذكاء الاصطناعي لاكتشاف الأخطار مبكرًا لتحذير المستخدمين واتخاذ الإجراءات المحددة كالاتصال بالشرطة مثلًا، ويستثمر الناس في أنظمة الحماية المنزلية الذكية لجعل مساكنهم أكثر أمانًا. توفر هذه التقنيات المتطورة إمكانية الدخول إلى منزلك دون الحاجة إلى المفتاح، وتمنحك تحديثات فورية في حال حصول أي أمور غير اعتيادية.

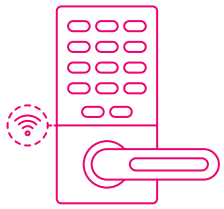
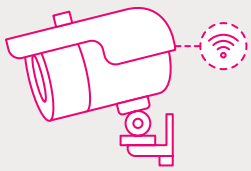
المخاطر Risks

رغم الميزات السابقة، إلا أن انعدام التشريعات الخاصة أو ضعفها باستخدام أجهزة إنترنت الأشياء وتوفير الحماية يُشكل تهديدًا خطيرًا عند تطبيق إنترنت الأشياء في المنزل الذكي. كما تبرُز أخطار الخصوصية وأمن البيانات أثناء استخدام أجهزة إنترنت الأشياء في ظل عدم وجود معايير أمان عالمية. تجمع أدوات وأجهزة إنترنت الأشياء في منزلك البيانات، ولذلك عليك حماية كل نظام يجمع معلوماتكَ الشّخصية ويحتفظ بها إذا أردت الحفاظ على خصوصيتك.

شكل 6.1: استخدام الهاتف الذكي لفتح بوابة أوتوماتيكية في مسكن

ستستكشف بعضًا من أكثر الأجهزة الشائعة القائمة على إنترنت الأشياء والمُستخدمة في أنظمة الحماية الذكية، مع أخذ المخاطر المختلفة بعين الاعتبار.

جدول 6.1: الأجهزة الشائعة التي تدعم إنترنت الأشياء

الأجهزة	الاستخدامات في أنظمة الحماية الذكية
 الأقفال الذكية	تعمل الأقفال الذكية على تحسين الأمان لمنزلك، وتسمح لك بالتحكم في البوابات عن بُعد، كما يمكنك وضع القيود على دخول الزوار في فترات زمنية معينة أو بناء على جدول محدد، وتوفر بعض الأقفال الذكية ميزات أكثر تقدمًا كالتحكم من خلال بصمة الإصبع أو الوجه أو حتى المصادقة بمسح شبكية العين.
 الكاميرات الذكية	لا يكتمل نظام الحماية المنزلية دون استخدام الكاميرات الذكية، حيث تعمل الكاميرات كعيون رقمية لمنزلك، مما يسمح لك بمشاهدة أي نشاط داخل المنزل وخارجه بصورة فورية. توجد العديد من خيارات الكاميرا الذكية المتاحة بما فيها كاميرات بروتوكول الإنترنت (IP) اللاسلكية التي يمكن مراقبتها من أي مكان يتصل بالإنترنت. يمكن التقاط فيديو المراقبة للأماكن في محيط بوابات الدخول بواسطة كاميرات الباب أو البوابة.
مُستشعرات الحرائق والدخان	تلعب أجهزة الكشف عن الحرائق والدخان دورًا مهمًا في الإنذار المبكر وإعلامك على الفور بوجود خطر ما في منزلك. غالبًا ما تُجهز المنازل الذكية بأجهزة كشف لغاز أول أكسيد الكربون حيث توفر تنبيهات عند اكتشاف كميات كبيرة من هذا الغاز بشكل خطير. قد تقوم هذه الأجهزة بتنشيط نظام الإطفاء، أو بإخطار قسم الإطفاء للتأكد من عدم انتشار الحريق بصورة خطيرة مما قد يتسبب بخسائر أو إصابات في الممتلكات.
مُستشعرات الحركة	تُعدُّ أجهزة الكشف عن الحركة مكونًا هامًا في نظام الحماية الذكي. تقوم هذه الأجهزة بتسجيل الاهتزازات والمعلومات وتحليلها من عدة أبعاد بواسطة هذه الأنظمة، والتي بدورها يمكنها أن تشير إلى أي حركة غير طبيعية. يمكن أن تقوم هذه المستشعرات بتنشيط أجهزة الإنذار لإعلام المستخدمين بالأنشطة المشبوهة سواء داخل المنزل أو في محيطه الخارجي.

لغة C++

C++ Language

ليس من السهل تحقيق أمن المعلومات، ولذلك أنت بحاجة إلى استخدام لغات برمجة قوية مثل لغة C++ لبرمجة واجهات البرامج. تُعدّ C++ لغة برمجة تجميعية عالية المستوى تتضمن العديد من ميزات البرمجة الكائنية، إضافةً إلى العديد من الإمكانيات القوية في معالجة الذاكرة، كما تتميز هذه اللغة بكفاءتها وسرعة أدائها. صُمّمت لغة C++ كتطوير للغة برمجة C.

أنواع البيانات الأساسية Basic Data Types

على عكس الكثير من لغات البرمجة الأخرى، يجب تعريف نوع المتغير في لغة C++ قبل استخدامه، لأن نوع المتغير يشير إلى نوع البيانات التي يحملها. يحتاج البرنامج في C++ إلى هذه المعلومات لمعرفة مقدار الذاكرة المطلوب تخصيصها لهذه البيانات.

جدول 6.2: أكثر أنواع البيانات شيوعًا في C++

النوع	الرمز	مثال
الأعداد الصحيحة	(int)	-4 ,5
الأعداد العشرية أو الحقيقية	(float or double)	-7.5 , 3.14
النص	(char)	'c'
البيانات المنطقية	(bool)	bool flag = true;

يمكنك تغيير نوع البيانات باستخدام مغيّر النوع، فعلى سبيل المثال (long int) يعني عدد صحيح طويل. تظهر التركيبات الممكنة لهذه المجموعات في الجدول أدناه:

double	int	char	
	✓	✓	signed
	✓	✓	unsigned
	✓		short
✓	✓		long

يُمكن للمبرمج تعريف أنواع خاصة به من البيانات بناءً على احتياجاته.

هناك بعض قواعد التسمية التي تحتاج إلى اتّباعها عند إنشائك لمتغير.

شروط تسمية المتغيرات الصحيحة:

- يمكن لاسم المتغير أن يحتوي فقط على الحروف الأبجدية (A–Z ، a–z) والأرقام (0–9) والشرطة السفلية (___).
- لا يُمكن أن يبدأ اسم المتغير برقم.
- لا يُمكن أن يكون اسم المتغير هو أحد كلمات لغة البرمجة، ككلمة int مثلًا، والتي هي كلمة أساسية تُستخدم للدلالة على الأعداد الصحيحة.

يُمكن تعريف المتغيرات مع تحديد قيمتها، أو دون ذلك.

المصفوفات Arrays

يُعدُّ هيكل **المصفوفة** (Array) من أكثر هياكل البيانات شيوعًا في C++. المصفوفة هي ببساطة مُتغيِّر يمكنه الاحتفاظ بقيم بيانات متعددة من نفس النوع.

صيغة الإعلان عن المصفوفة:

لا يُمكنك تغيير نوع أو حجم المصفوفة بعد الإعلان عنها، ويمكنك الوصول إلى عناصرها باستخدام الدليل أو ما يسمى **بفهرس المصفوفة** (Array Index).

على سبيل المثال، إذا كنت تريد تخزين 10 قيم صحيحة، يُمكنك إنشاء مصفوفة تخزن فيها هذه القيم. عليك أولًا الإعلان عن نوع وحجم المصفوفة:

```
int values[10];
```

تُمثِّل "int" نوع العناصر المُخزَّنة في المصفوفة، و"values" هو اسم المصفوفة وحجمها هو 10. ولتعبئتها بالقيم يُستخدم الأمر الآتي:

```
values [10] = {0,1,2,3,4,5,6,7,8,9};
```

للوصول إلى أي من عناصر المصفوفة، تحتاج إلى فهرس العنصر، فيكون الأمر:

```
int a = values[3];
```

يُعلن عن متغير باسم a، وهو عدد صحيح، وقيمته تساوي العنصر الرابع من المصفوفة "values" (تبدأ الفهرسة في C++ من 0). وعلى الرغم من إمكانية استخدام مصفوفات بأكثر من بُعد واحد، إلا أن أكثر أنواع المصفوفات شيوعًا هي المصفوفات أحادية وثنائية الأبعاد. لإنشاء مصفوفة ثنائية الأبعاد، تحتاج إلى إعلان حجم كل بُعد من أبعادها. على سبيل المثال:

```
char keys[4][2];
```

يُعلن عن مصفوفة تتكون من أربعة صفوف وعمودين، يمكنها تخزين قيم من نوع "char". ولتعبئة المصفوفة بقيمك، يجب أن تنفذ ذلك كما في المصفوفة أحادية الأبعاد:

```
keys[4][2] =
{{1,2},
{3,4},
{5,6},
{7,8}
};
```

ستحتاج هنا إلى زوج من القيم لكل صف من المصفوفة.

المُعامِلات الأساسية في C++

Basic Operators in C++

تتضمن المُعامِلات الأساسية كلًا من المُعامِلات الرياضية، ومُعامِلات الإسناد، والمُعامِلات العلائقية والمنطقية.

جدول 6.3: المُعامِلات الرياضية

المعامل	العملية
+	الجمع
-	الطرح
*	الضرب
/	القسمة
%	باقي القسمة

في الأعداد العشرية (float، double) يتم استخدام "/" فقط من أجل حاصل القسمة، مثلًا 5.0/2.0=2.5

في الأعداد الصحيحة (int)، يتم استخدام "/" لحساب حاصل القسمة و "٪" لحساب باقي القسمة. مثلًا 5/2= 2, 5%2=1

جدول 6.4: مُعامِلات الإسناد

المعامل	مثال	يكافئ
=	a = b;	a = b;
=+	a += b;	a = a + b;
=-	a -= b;	a = a - b;
=*	a *= b;	a = a * b;
=/	a /= b;	a = a / b;
=%	a %= b;	a = a % b;

جدول 6.5: المُعامِلات العلائقية

المعامل	الوصف	مثال
==	يساوي	5 == 3 يعطي خطأ
!=	لا يساوي	5=! 3 يعطي صواب
>	أكبر من	5 > 3 يعطي خطأ
<	أصغر من	5 < 3 يعطي صواب
>=	أكبر من أو يساوي	5=> 3 يعطي خطأ
<=	أصغر من أو يساوي	5=< 3 يعطي صواب

جدول 6.6: المُعامِلات المنطقية

المُعامل	الوصف	مثال
&&	التعبير الأول && التعبير الثاني	AND (و) المنطقية تكون صائبة إذا كان التعبيران صائبين.
||	التعبير الأول || التعبير الثاني	OR (أو) المنطقية. تكون صائبة إذا كان أحد التعبيرين على الأقل صائبًا.
!	!التعبير	NOT (لا) المنطقية تكون صائبة فقط إذا كان التعبير خاطئًا.

التعليقات في لغة C ++
Comments in C++

تدعم جميع لغات البرمجة ميزة إضافة التعليقات داخل التعليمات البرمجية. لا تُنفّذ هذه التعليقات ضمن البرنامج، ولكنها تُستخدم لتحسين قابلية قراءة البرنامج، مما يُسهِّل على المبرمجين أو مراجعي البرامج فهمَ وظائف البرنامج. توجد طريقتان لإضافة تعليق في C++، وذلك حسب الحاجة إلى إضافة التعليق في سطر واحد أو أسطر متعددة.

استخدم // لإضافة تعليق يتكون من سطر واحد

```
// this is a comment
int y = 10;
cout << y;
```

تعليمات غير نشطة

استخدم */ لبدء تعليق متعدد و/* لإنهائه. تُستخدم هذه الطريقة أيضًا لجعل جزء من التعليمات البرمجية غير نشط أثناء اختبار عمل البرنامج. على سبيل المثال يتم في البرنامج الآتي تخطي الجملة الشرطية if بواسطة مترجم لغة البرمجة.

```
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Enter password:");

bool correctPass = true;
char buttonPressed;
/*
int index = 4;
buttonPressed = keypad.waitForKey();
if(password[index] != buttonPressed){
    correctPass = false;
}
*/
lcd.setCursor(i, 1);
lcd.print(buttonPressed);
```

تعليمات غير نشطة

الطباعة في C++
Printing in C++

لطباعة المتغير x في C++ ، استخدم الأمر الآتي:

```
cout << x;
```

العبارات الشرطية في C++
Conditional Statements in C++

لتنفيذ مجموعة معينة من التعليمات البرمجية بناءً على تحقق شرط ما، يُمكنك استخدام مجموعة من الجُمل الشرطية:

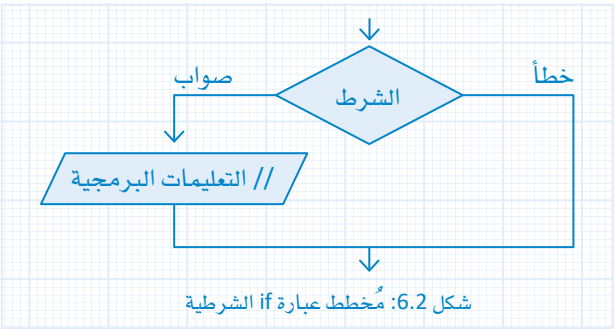
- عبارة **if**
- عبارة **if... else**
- عبارة **if... else if... else**

عبارة if الشرطية

يُستخدم هذا النوع من العبارات الشرطية إذا أردت تنفيذ مجموعة تعليمات برمجية حال تحقق شرط محدد.

صيغة عبارة if البسيطة في C++ كالآتي:

```
if (condition) {
// body of if statement
}
```



شكل 6.2: مُخطط عبارة if الشرطية

يتم أولًا فحص الشرط الموجود بين قوسين، وفي حال كانت قيمته صائبة، تُنفّذ التعليمات البرمجية الموجودة داخل الأقواس { }، أما إذا كانت خاطئة، فإنه يتم تخطي تلك التعليمات البرمجية. تعمل عبارة if كالآتي:

إذا كان الشرط صائبًا

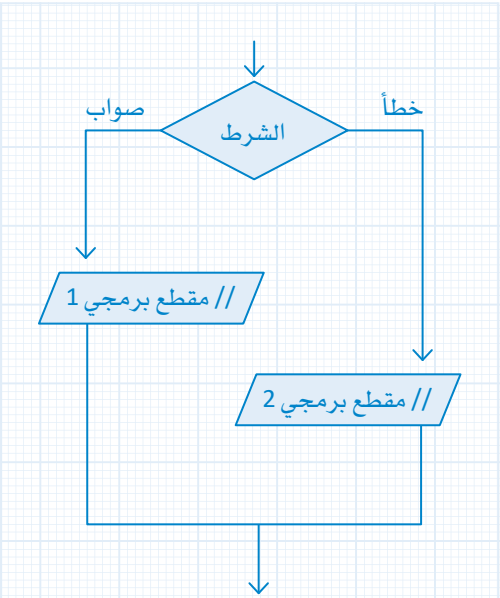
```
int number = 5;
if (number > 0) {
    // code
}
// code after if
```

إذا كان الشرط خطأ

```
int number = 5;
if (number < 0) {
    // code
}
// code after if
```

عبارة if...else الشرطية

في هذا النوع من العبارات الشرطية، تُنفّذ مجموعة التعليمات البرمجية داخل {} if ويتم تخطي التعليمات البرمجية الموجودة داخل {} else، أو يتم تخطي التعليمات البرمجية داخل {} if وتُنفّذ التعليمات البرمجية الموجودة داخل {} else.



شكل 6.3: مُخطط جملة if...else

تركيب عبارة if...else:

```
if (condition) {
// block of code 1 if condition is true
}
else {
// block of code 2 if condition is false
}
```

يتم أولًا تقييم الشرط الموجود بين قوسين وإذا كانت قيمته صائبة، فستُنفّذ التعليمات البرمجية الموجودة داخل {} if، وإذا كان الشرط خطأ، فسيتم تنفيذ التعليمات البرمجية الموجودة داخل {} else.

كيف تعمل عبارة if...else:

إذا كان الشرط خطأ:

```
int number = 5;
if (number < 0) {          يتم تجاوزه
   // code
}
else {                     يتم تنفيذه
   // code
}

// code after if...else
```

إذا كان الشرط صائبًا:

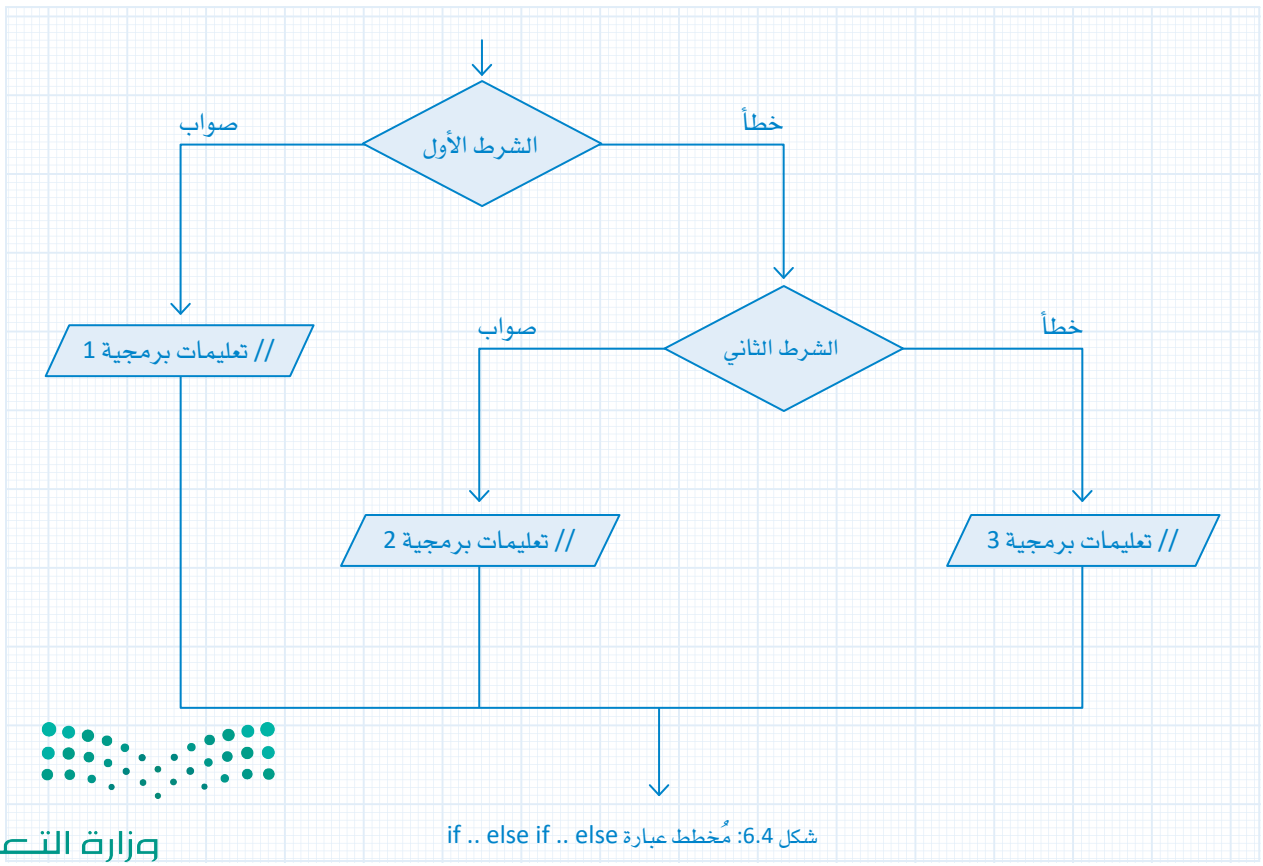
```
int number = 5;
if (number > 0) {          يتم تنفيذه
   // code
}
else {                     يتم تجاوزه
   // code
}

// code after if...else
```

عبارة if..else if..else

يُستخدم النوع الأخير من العبارات الشرطية if... else if... else عندما تحتاج إلى التحقق من أكثر من شرط واحد، أو عندما تحتاج إلى تنفيذ 3 مجموعات أو أكثر من التعليمات البرمجية وفقًا لبعض الشروط.

تركيب عبارة if .. else if .. else هو:

```
if (condition1) {
   // code block 1
}
else if (condition2) {
   // code block 2
}
else {
   // code block 3
}
```



شكل 6.4: مُخطط عبارة if .. else if .. else

كيف تعمل عبارة if .. else if .. else

إذا كان الشرط الأول صائبًا، ستُنفّذ مجموعة التعليمات البرمجية الأولى ويتم تخطي باقي التعليمات البرمجية.

إذا كان الشرط الأول صائبًا:

```
int number = 2;
if (number > 0) {                    يتم تنفيذه
    // code
}
else if (number == 0) {              يتم تجاوزه
    // code
}
else {
    // code
}
// code after if
```

إذا كان الشرط الأول خطأ وكان الشرط الثاني صائبًا، ستُنفّذ مجموعة التعليمات البرمجية الثانية، وتخطي مجموعة التعليمات البرمجية الثالثة.

إذا كان الشرط الثاني صائبًا:

```
int number = 0;
if (number > 0) {                    يتم تجاوزه
    // code
}
else if (number == 0) {              يتم تنفيذه
    // code
}
else {                               يتم تجاوزه
    // code
}
// code after if
```

إذا لم يكن أي من الشرط الأول أو الشرط الثاني صائبًا، ستُنفّذ مجموعة التعليمات البرمجية الثالثة.

كافة الشروط خطأ.

```
int number = 0;
if (number > 0) {                    يتم تجاوزه
    // code
}
else if (number == 0) {              يتم تجاوزه
    // code
}
else {                               يتم تنفيذه
    // code
}
// code after if
```

يمكنك أيضًا تضمين عبارة if داخل مجموعة التعليمات البرمجية لعبارة if أخرى. ولا يُشترط أن تكون من نفس النوع. فمثلًا:

```
// outer if statement
if (condition1) {

    // statements

    // inner if statement
    if (condition2) {
    }
}
// code after if
```

التكرارات Loops

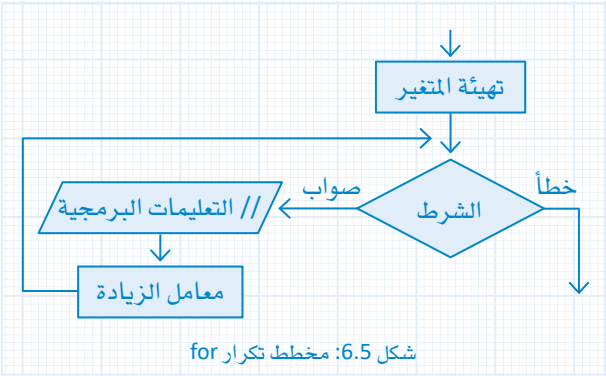
يمكنك في C++ استخدام ثلاثة أنواع من التكرارات البرمجية:

- تكرار **for**
- تكرار **while**
- تكرار **do...while**

تكرار for

صيغة تكرار for هي:

```
for (variable initialization;
condition; increment operation) {
// loop statements;
}
```



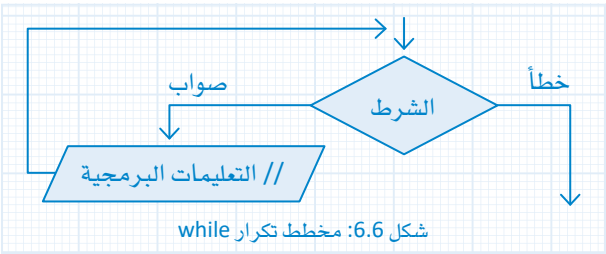
شكل 6.5: مخطط تكرار for

تُنفّذ تهيئة المتغير مرة واحدة فقط قبل بدء التكرار وتعيين قيم البداية للمتغيرات التي تشكل جزءًا من الشرط. يمكنك أيضًا الإعلان عن متغير وتهيئته في هذه الخطوة، وغالبًا يُستخدم عداد لتنفيذ التكرار عدة مرات حسب الشرط. فإذا كانت قيمة الشرط صواب، تُنفّذ جُمل التكرار ثم تتم الزيادة بتحديث قيم المتغيرات التي تمت تهيئتها. يستمر هذا حتى تتغير قيمة الشرط إلى خطأ.

تكرار while

صيغة تكرار while هي:

```
while (condition) {
// loop statements;
}
```



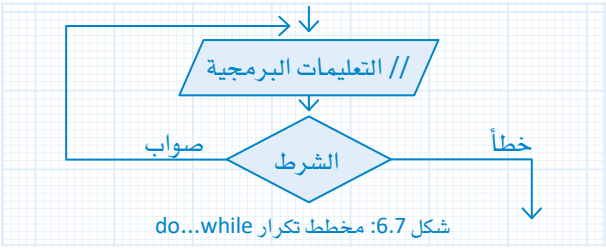
شكل 6.6: مخطط تكرار while

حيث تُنفّذ عبارات التكرار عندما يكون الشرط صائبًا، وعندما يصبح الشرط خطأ، يتوقف التكرار ويتم تخطي عبارات التكرار.

تكرار do...while

النوع الثالث للتكرارات هو do... while، وهو نوع يختلف عن تكرار while وصيغته هي:

```
do {
// statement execution;
} while (condition);
```



شكل 6.7: مخطط تكرار do...while

يختلف هذا التكرار عن تكرار while في أنه في تكرار do... while loop يُفحص الشرط بعد جُمل التكرار، وهذا يعني أن التعليمات البرمجية داخل جسم التكرار ستُنفذ مرة واحدة على الأقل. ويتوقف التكرار عند تحول الشرط إلى خطأ.

عبارات التحكم البرمجية "break" و "continue"

" break" and "continue" Statements

توجد عباراتان مفيدتان جدًا عند التعامل مع التكرارات، وهما break وcontinue واللتان تعملان مع جميع أنواع التكرارات.

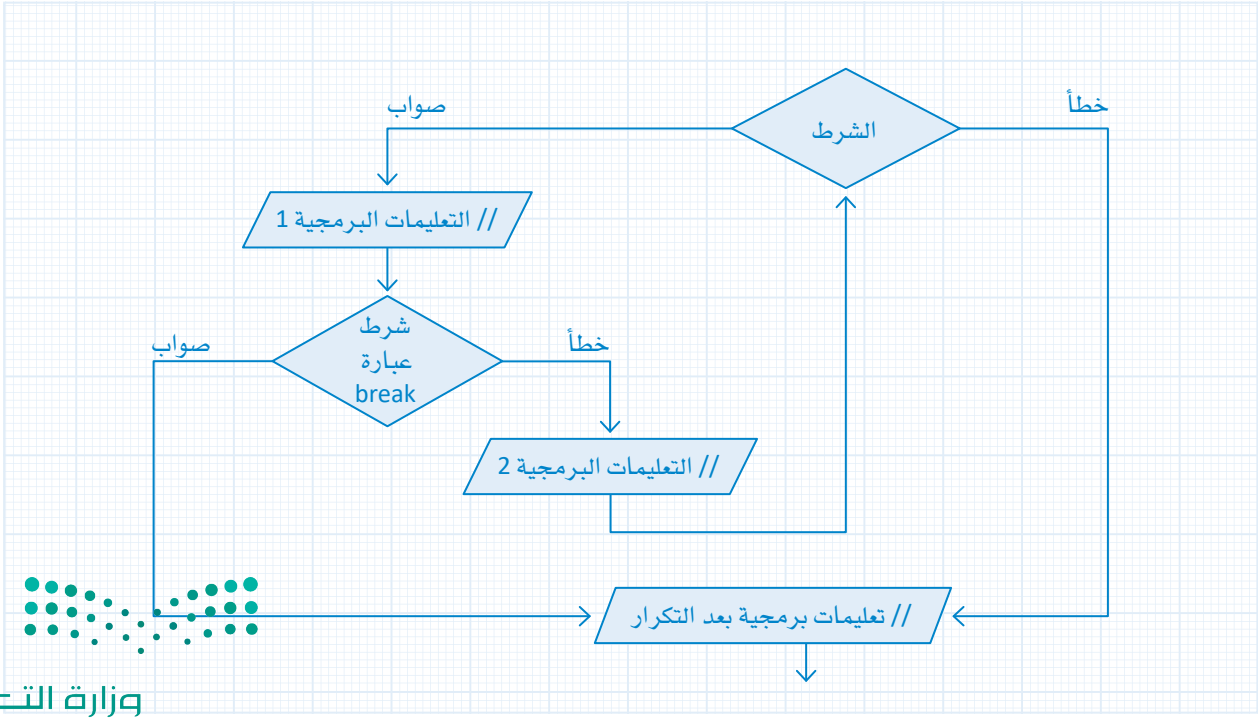
عبارة "break"

تنهي عبارة break التكرار حيث تكون موجودة.

```
while (condition) {
   // code block 1
   if (condition to break) {
      break
   }
   // code block 2
}
 // code after loop
```

```
for (init; condition; update) {
   // code block 1
   if (condition to break) {
      break
   }
   // code block 2
}
 // code after loop
```

إذا عُثر على عبارة break داخل تكرار مُتداخل، فإنها تُنهي التكرار الداخلي.



شكل 6.8: مخطط جملة break

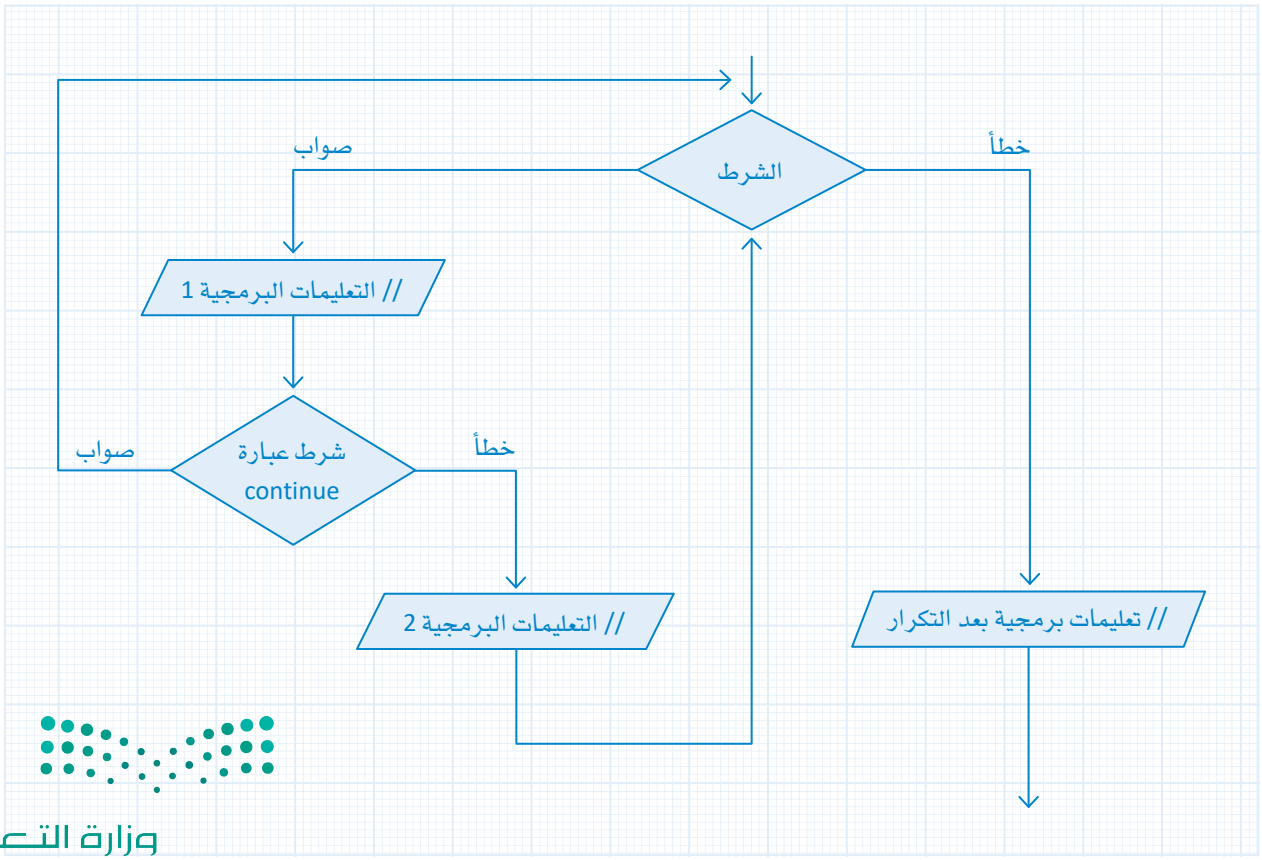
عبارة "continue"

تتخطى عبارة continue بقية التعليمات البرمجية داخل التكرار وتنتقل إلى التكرار التالي.

```
for (init; condition; update) {
    // code block 1
    if (condition to continue) {
        continue
    }
    // code block 2
}
// code after loop
```

```
while (condition) {
    // code block 1
    if (condition to continue) {
        continue
    }
    // code block 2
}
// code after loop
```

إذا وُجدت عبارة continue داخل التكرار المتداخل، سيتم تخطى التكرار الحالي في التكرار الداخلي.



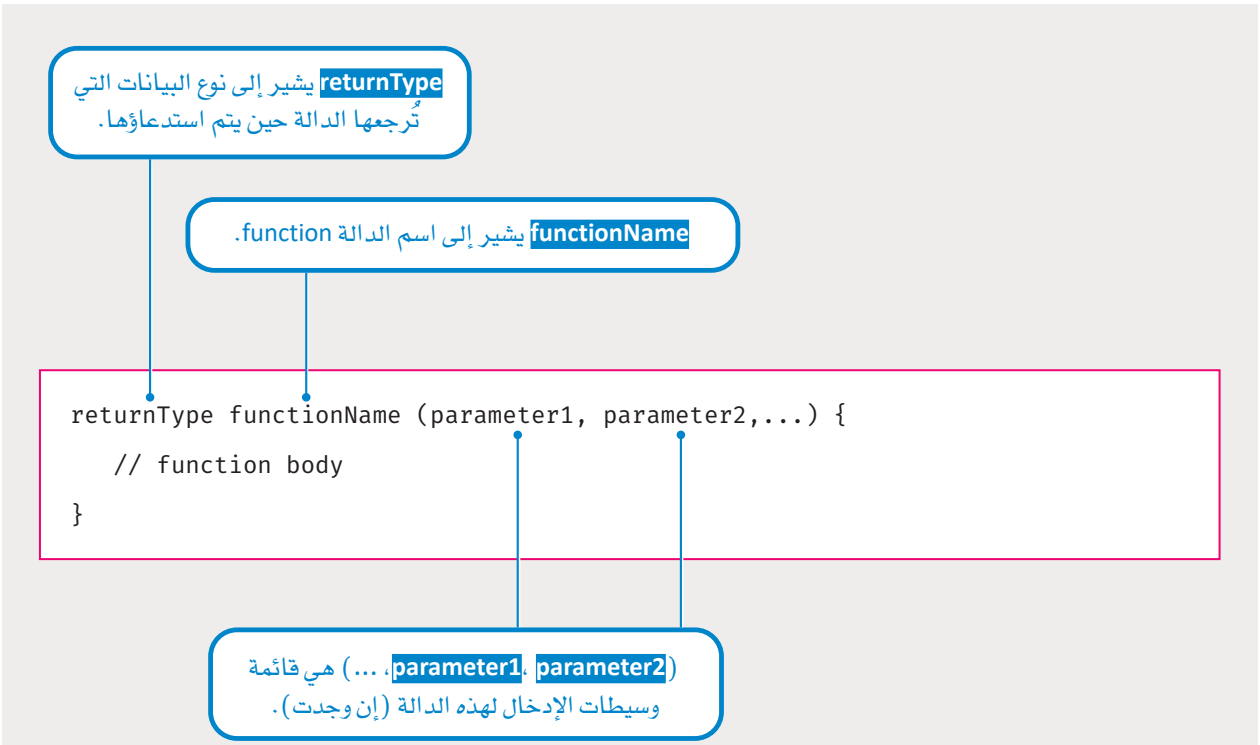
شكل 6.9: مخطط جملة continue

الدوال في C++

Functions in C++

عند القيام بكتابة البرامج، ستلاحظ أن الكثير من المهام البرمجية قد تحتاج إلى تنفيذ عمليات معينة عدة مرات خلال تشغيل البرنامج. يمكنك بالطبع كتابة نفس سطور التعليمات البرمجية كل مرة تحتاج فيها إلى تنفيذ هذه المهام، ولكن الحل الأفضل هو تجميع هذه التعليمات البرمجية وإنشاء دالة لتؤدي هذه المهام. توجد في C++ العديد من الدوال القياسية المُضمنة والتي يُمكن للمبرمجين استخدامها. يُمكن للمبرمجين أيضًا إنشاء دوالهم الخاصة بناءً على احتياجاتهم حيث يمكنهم تسميتها بأنفسهم. يمكن لكل دالة قبول بعض المتغيرات كمُعاملات إدخال، وتنفيذ بعض التعليمات البرمجية المضمنة بين الأقواس {}، ولإنهاء الدالة توجد عبارة إرجاع (return) تُرجع قيمة.

لإنشاء دالة، تحتاج أولًا إلى الإعلان عنها:



مثال على دالة بسيطة تستقبل عددين صحيحين كوسيطين لتعيد مجموعهما:

```
// function declaration
int adding (int a, int b) {
    s = a+b;
    return s
}
```

لاستخدام هذه الدالة في برنامجك الرئيس، يمكنك استدعاؤها من خلال اسمها وتمرير عددين صحيحين لها كمُعامِلات:

```
int main () {
    int a=2;
    int b=5;
    int c;
    //calling the function and passing a, b as arguments
    c = adding(a,b);
    //cout will print the value of c
    cout << c;
    return 0;
}
```

فقط في الدالة ( )main يكون تعبير الإرجاع (return) اختياريًا، ويمكن الاستغناء عنه.

كما تلاحظ فإن الدالة main هي أيضًا دالة تقوم بإرجاع القيمة 0، وهكذا فإن نوع البيانات الذي تُرجعه الدالة هو int (عدد صحيح)، ولكنه لا يقبل أي مُعامِلات إدخال في هذه الحالة ويشار إليها بالأقواس الفارغة ( ). دالة main هي نوع خاص من الدوال في C++، حيث يوجد الجزء الرئيس من البرنامج.

يجب أن يتطابق النوع والعدد والترتيب للوسيطات التي تُمرّر إلى دالة ما مع نوع المُعامِلات الموجود في إعلان الدالة.

من الممكن ألا تُرجع الدالة أي قيمة، وفي مثل هذه الحالة يكون نوع الإرجاع **"void"** (فارغًا).

```
void displyNumber () {
   // code
}
```

دوال ( )Setup و ( )Loop
Setup( ) and Loop( ) Functions

عند كتابة برنامج أردوينو في منصة تينكركاد، توجد دالتان يجب استدعاؤهما لتنفيذ برنامج الدائرة. تُستدعى هذه الدوال تلقائيًا عند بدء تنفيذ البرنامج، وذلك على عكس باقي الدوال التي يجب استدعاؤها يدويًا من خلال تعليماتك البرمجية.

أول دالة تُنفّذ هي ( )void setup، وتُنفّذ هذه الدالة مرة واحدة فقط في البداية، وهي مسئولة عن تكوين أجزاء الدائرة المختلفة مثل ضبط وضع أطراف الأردوينو الرقمية، وإنشاء اتصال مع الطرف التسلسلي وغيرها من الأمور.

بعد تنفيذ دالة ( )setup، تُستدعى الدالة ( )void loop بشكلٍ متكرر أثناء عمل النظام، وهذه الدالة هي التي تنفذ الوظيفة الرئيسة للدائرة.

بشكل عام، يجب أن تكتب برنامج الإعداد داخل دالة ( )void setup، وتكتب منطق البرنامج الرئيس داخل ( )void loop، وأي إعلان عن أي ثوابت أو دوال يكون خارج هاتين الدالتين.

مثال على برنامج أردوينو بلغة C++.

```
void setup() {
    int a = 10;
    int b = 20;
}

void loop() {
    for (int i = 0; i < b; i++) {
        a += i;
        cout << a;
    }
}
```

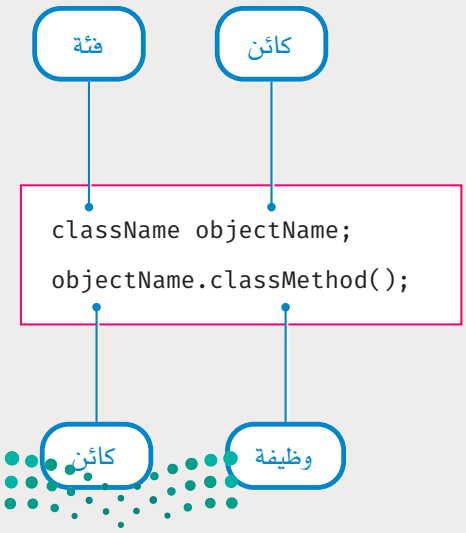
تُشغّل دالة **setup()** مرة واحدة فقط لتكوين المتغيرات والكائنات.

تعمل دالة **loop()** بصورة متكررة في الأردوينو.

الفئات والكائنات والوظائف

Classes, Objects, and Methods

ترتكز **البرمجة الكائنية** (object-oriented programming) على إجراء جميع العمليات البرمجية على أساس "الكائنات". الكائن هو الوحدة الأساسية للبرمجة الكائنية. قد يكون لهذه الكائنات خصائص، كما يمكنها أن تنفذ بعض **الأحداث** (actions) الأساسية. على سبيل المثال، يمكن اعتبار **محرك سيرفو** (servo motor) بمثابة كائن له بعض الخصائص مثل (الاسم والنوع)، ويمكنه تنفيذ بعض الإجراءات الأساسية مثل القراءة من طرف رقمي، وتدوير محركه بعدد مُعين من الدرجات وغيرها. تُسمى هذه الإجراءات التي يمكن لكل كائن تنفيذها **بالوظائف** (Methods)، وهي في لغة C++ بالأساس الدوال التي أُعلن عنها داخل جسم الكائن. من الناحية الفنية، يُعلن عن الخصائص والوظائف داخل جسم **الفئة** (Class) وليس **الكائن** (Object). لفهم الفرق بين الفئة والكائنات، يمكنك اعتبار الفئة كمفهوم، والكائنات على أنها تجسيد لهذا المفهوم. على سبيل المثال، في محاكاة الدائرة حيث سيكون هناك ثلاثة **محركات سيرفو** (servo motors)، فستحتاج أولًا إلى الإعلان عن فئة "Servo"، وسيكون كل من هذه المحركات الثلاثة كائنًا مؤازرًا، ويطلق عليه عادةً تسمية **العينة** (Instance) من فئة " Servo" .



تمرينات

1

حدد الجملة الصحيحة والجملة الخاطئة فيما يلي:	صحيحة	خاطئة
1. يمكن لأجهزة إنترنت الأشياء التحكم في أبواب المنزل، وإغلاقها.	○	○
2. لا يمكنك مراقبة المنزل الذكي باستخدام الهاتف الذكي.	○	○
3. تواكب التشريعات والقوانين القضايا المتعلقة بتطبيقات الحماية الذكية لإنترنت الأشياء.	○	○
4. لا يمكن الوصول إلى أنظمة الكاميرات الذكية إلا من خلال الشبكة المنزلية.	○	○
5. يمكن لأنظمة المنزل الذكي الاتصال تلقائيًا بخدمات الطوارئ.	○	○
6. يمكن لأنظمة القفل الذكية استخدام البيانات الحيوية (البيولوجية) للتعرف على المستخدمين.	○	○
7. تختلف لغة C++ تمامًا عن لغة C.	○	○
8. C++ هي لغة برمجة كائنية.	○	○
9. المصفوفات في لغة C++ مُحددة النوع دائمًا.	○	○
10. ليست هناك أي أهمية خاصة للدالتين ( )setup و ( )loop في برنامج الأردوينو.	○	○

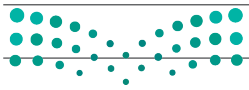
2 عدِّد الفوائد التي توفرها تطبيقات الحماية الذكية في إنترنت الأشياء.

3 وضِّح المخاطر المحتملة للاستخدامات المتقدمة لإنترنت الأشياء للحماية الذكية.

4 صنِّف أكثر الأجهزة المنزلية الذكية الشائعة التي تدعم إنترنت الأشياء.

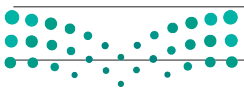
5 حدِّدْ الأنواع الأساسية للبيانات للبرمجة بلغة C++.

6 دوِّن القواعد الأساسية التي يجب مراعاتها عند تسمية متغيرات C++.



7 وضِّح كيفية تنفيذ تكرارات for في لغة C++.

8 صِفْ الفرق بين تكرارات while وdo... while في لغة C++.



9 وضِّح استخدام دوال setup() و loop() في مخطط الأردوينو.

10 وضِّح الخطوات اللازمة لاختزال مُكوِّن إلكتروني يتصل بلوحة الأردوينو إلى فئة وكائن في لغة C++.

رابط الدرس الرقمي

www.ien.edu.sa

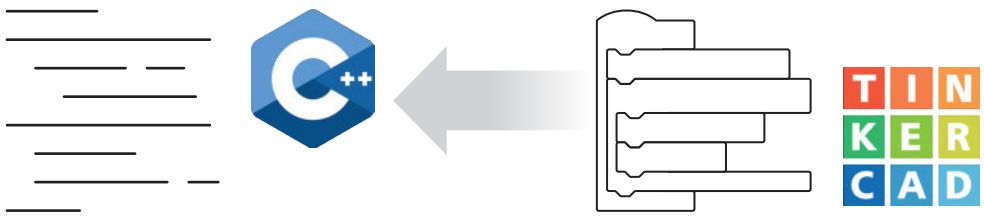
الدرس الثاني

الانتقال من اللبنات البرمجية في تينكركاد إلى لغة C++

الانتقال من برمجة اللبنات المرئية إلى البرمجة في C ++

Migrating from Visual Blocks Programming to C++ Programming

ستتعلم في هذا الدرس كيفية الانتقال من برمجة **الأردوينو** (Arduino) بلبنات **تينكركاد** (Tinkercad) البرمجية إلى برمجته باستخدام لغة البرمجة C++ . تُعدُّ اللبنات البرمجية في تينكركاد مفيدة في تنفيذ النماذج الأولية والمهمات البرمجية البسيطة، إلا أن استخدام C ++ يُعدُّ ضروريًا للاستفادة الكاملة من إمكانات مُتحكم الأردوينو. ستتعلم في هذا الدرس الدوال والجُمل الأساسية لبدء برمجة متحكم الأردوينو باستخدام لغة C++.



شكل 6.10: من اللبنات البرمجية في تينكركاد إلى البرمجة في C++

يوفر تينكركاد بيئة محاكاة لبرمجة الأردوينو بالنمذجة، والتي لا تتطلب وجود أردوينو فعليًا وتوصيله بجهاز الحاسب.

الإعلان عن المتغيرات والعمليات

Variable Assignments and Operations

يتم إعلان المتغيرات وتغييرها في لبنات تينكركاد البرمجية من خلال مجموعتي أوامر **المتغيرات** (Variables) و**الحساب** (Math) . يوضِّح الجدول الآتي أمثلة للأوامر المتاحة.

الإعلان عن متغير باسم x.

C++

```
int x = 0;
```

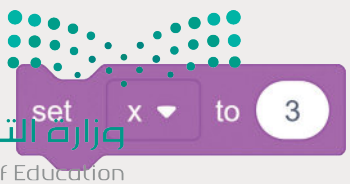
لبنة تينكركاد

تعيين قيمة للمتغير.

C++

```
x = 3;
```

لبنة تينكركاد



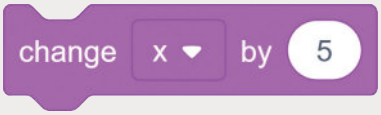
وزارة التعليم
Ministry of Education
2023 - 1445

تغيير قيمة متغير بقيمة محددة.

C++

```
x += 5;
```

لبنة تينكركاد

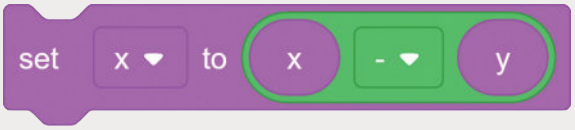


تنفيذ عملية رياضية بين المتغيرين x وy.

C++

```
x = x - y;
```

لبنة تينكركاد

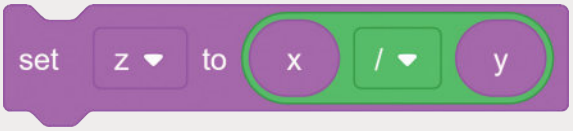


تعيين متغير ثالث z لناتج عملية رياضية بين المتغيرين x وy.

C++

```
z = x / y;
```

لبنة تينكركاد



إجراء مقارنة رياضية بين المتغيرين x وy.

C++

```
x < y
```

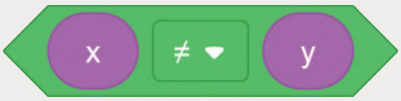
لبنة تينكركاد

إجراء مقارنة منطقية بين المتغيرين x وy.

C++

```
x != y
```

لبنة تينكركاد

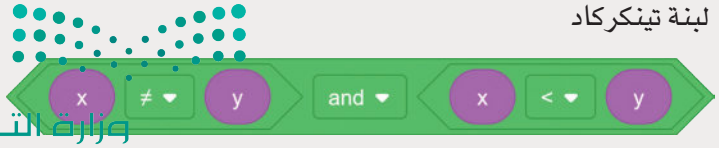


إجراء عملية منطقية بين عبارتين.

C++

```
x != y && x < y
```

لبنة تينكركاد



العبارات الشرطية والتكرارات ورسائل الإخراج
Conditional Statements, Loops and Output Messages

تُنشأ العبارات الشرطية والتكرارات ورسائل الإخراج في لبنات تينكركاد من خلال مجموعات أوامر **التحكم** (Control) و**الإخراج** (Output). يوضح الجدول الآتي أمثلة للأوامر المتوفرة.

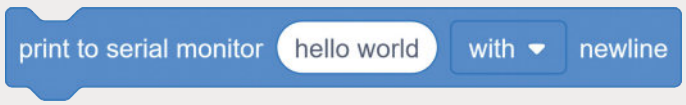
مجموعة لبنات الأوامر المستخدمة:

طباعة رسالة على **الشاشة التسلسلية** (Serial Monitor).

C++

```
Serial.println("hello world");
```

لبنة تينكركاد

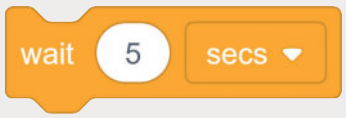


الانتظار 5 ثواني.

C++

```
delay(5000);
```

لبنة تينكركاد

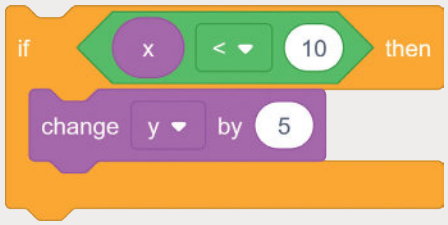


تنفيذ التعليمات البرمجية داخل لبنة if إذا كان الشرط المنطقي صحيحًا.

C++

```
if (x < 10) {
   y += 5;
}
```

لبنة تينكركاد

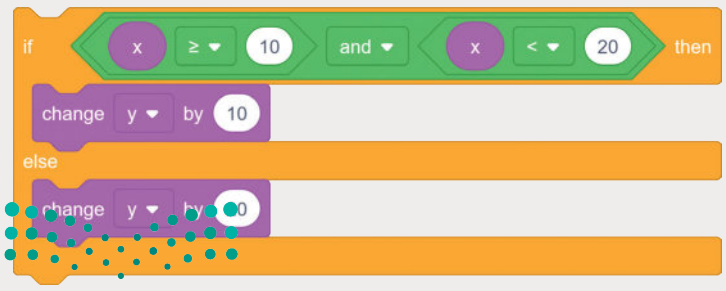


تنفيذ التعليمات البرمجية داخل لبنة if إذا كان الشرط المنطقي صحيحًا.

C++

```
if (x >= 10 && x < 20) {
    y += 10;
}
else {
    y += 20;
}
```

لبنة تينكركاد

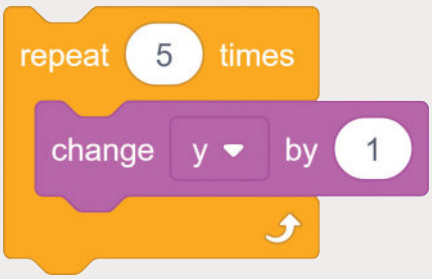


تنفيذ التعليمات البرمجية داخل لبنة for إذا كان الشرط المنطقي صحيحًا.

C++

```
for (counter = 0; counter < 5; ++counter) {
    y += 1;
  }
```

لبنة تينكركاد

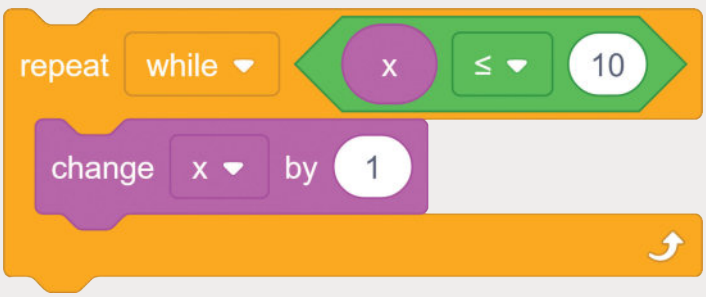


تنفيذ تكرار while في الحالة الآتية.

C++

```
while (x <= 10) {
    x += 1;
}
```

لبنة تينكركاد



مُدخلات ومُخرجات أطراف أردوينو الرقمية والتناظرية

Arduino Digital and Analog Pin I/O

يتم التفاعل مع الأطراف الرقمية والتناظرية للوحة الأردوينو في لبنات تينكركاد من خلال مجموعات أوامر **الإدخال** (Input) **والإخراج** (Output)، و**الحساب** (Math). في كل مرة يُستخدم فيها أحد أطراف الأردوينو التناظرية أو الرقمية، تتعرف لبنات تينكركاد على ما إذا كان سيتم استخدامه للإدخال/الإخراج الرقمي أو التناظري. لاستخدام طرف تحتاج إلى تحديد ذلك في دالة ()setup في الأردوينو لتوضيح ما إذا كان سيستُخدم في **الإدخال** (Input) أو **الإخراج** (Output). يتم استخدام الأطراف 3، 5، 6، 9، 10، 11 مع تعديل قيمة **عرض النبضة** (PWM). يعرض الجدول أدناه أمثلة لبعض الأوامر المتوفرة.

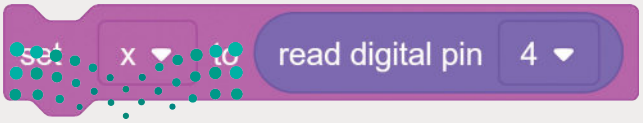
مجموعة لبنات الأوامر المستخدمة:

الحصول على قيمة الطرف الرقمي 4 وتخزينه في المتغير x.

C++

```
pinMode(4, INPUT);
x = digitalRead(4);
```

لبنة تينكركاد

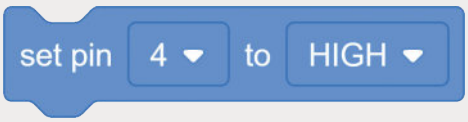


ضبط قيمة الطرف الرقمي 4 على قيمة HIGH (مرتفعة).

C++

```
pinMode(4, OUTPUT);
digitalWrite(4, HIGH);
```

لبنة تينكركاد

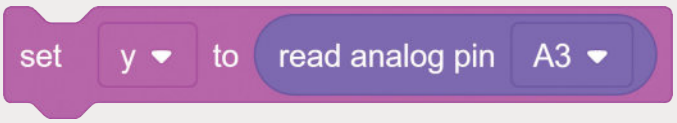


الحصول على قيمة الطرف التناظري A3 وتخزينه في المتغير y.

C++

```
pinMode(A3, INPUT);
y = analogRead(A3);
```

لبنة تينكركاد

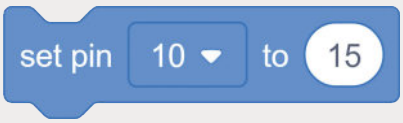


إعداد قيمة الطرف 10 من القيمة التناظرية 15 باستخدام تضمين **عرض النبضة** (PWM).

C++

```
pinMode(10, OUTPUT);
analogWrite(10, 15);
```

لبنة تينكركاد



أمثلة على الانتقال من لبنات تينكركاد البرمجية إلى لغة C++

Examples of Migration from Tinkercad Blocks to C++

ستُنشئ أمثلة بسيطة في تينكركاد للانتقال من برمجة لوحة الأردوينو باستخدام لبنات تينكركاد البرمجية إلى استخدام لغة البرمجة C++.

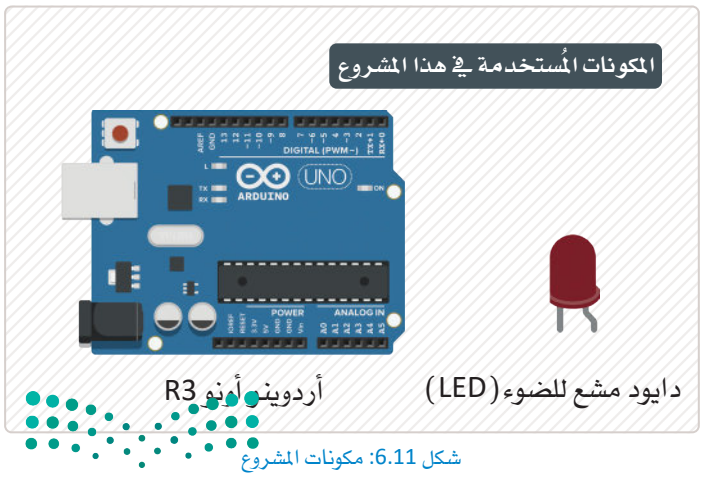
مثال الدايودات المشعة للضوء الوامضة
Blinking LEDs Example

ستُنشئ برنامجًا بسيطًا يحتوي على تكرارين يجعلان دايود مشع للضوء يومض 5 مرات و10 مرات بشدة مختلفة.

المكونات المطلوبة:

- 1 لوحة أردوينو أونو R3.
- 1 **دايود مشع للضوء** (LED).

يمكن البرمجة بلغة C++ في تينكركاد وذلك بتحويل نوع التحرير من لبنات (Blocks) الى نص (Text).

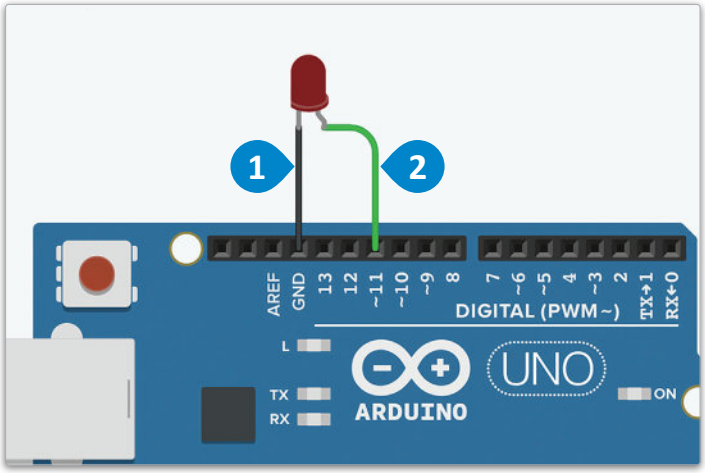


شكل 6.11: مكونات المشروع

لتوصيل الدايود الضوئي:

> قُم بتوصيل Cathode (مهبط) LED (الدايود المشع للضوء) بالطرف GND (الطرف الأرضي) للوحة الأردوينو، وغيّر لون السلك إلى black (الأسود). 1

> قُم بتوصيل Anode (مِصعد) الدايود المشع للضوء (LED) بالطرف Digital (الرقمي) 11 للوحة الأردوينو، وغيّر لون السلك إلى green (الأخضر). 2



شكل 6.12: توصيل الدايود المشع للضوء

برمجة الأردوينو

عند تشغيل البرنامج، سيومض الدايود المشع للضوء 5 مرات في الثانية الواحدة، ثم سيومض الدايود المشع للضوء 10 مرات وذلك بفارق 200 ملي ثانية بين كل منها.

لبنات تينكركاد

C++

```
int counter;
int counter2;
void setup() {
  pinMode(11, OUTPUT);
}
void loop() {
  for (counter = 0; counter < 5; ++counter) {
    digitalWrite(11, HIGH);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(11, LOW);
    delay(1000); // Wait for 1000 millisecond(s)
  }
  for (counter2 = 0; counter2 < 10; ++counter2) {
    digitalWrite(11, HIGH);
    delay(200); // Wait for 200 millisecond(s)
    digitalWrite(11, LOW);
    delay(200); // Wait for 200 millisecond(s)
  }
}
```

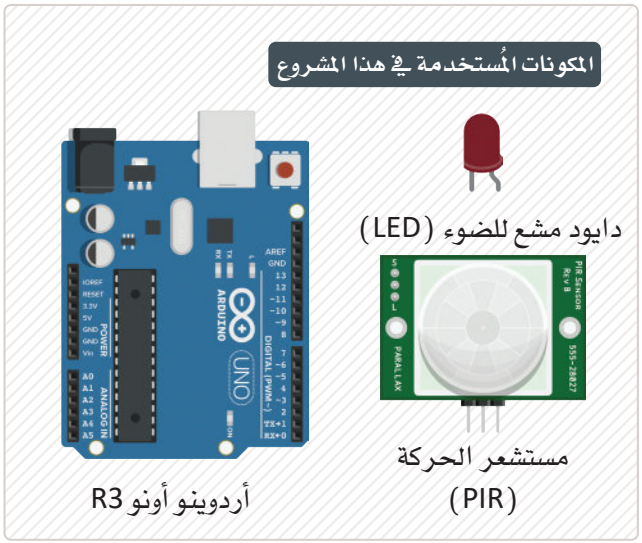
مثال إنذار مُستشعر الحركة

Passive Infrared Sensors (PIR) Alarm Example

سنقوم بتوسيع المشروع السابق بإنشاء جهاز إنذار PIR يرسل إشارة لإضاءة ثلاثة دايودات ضوئية مشعة في تتابع سريع.

المكونات المطلوبة:

- **لوحة أردوينو أونو R3** (Arduino Uno R3).
- **دايود مُشع للضوء** (LED).
- **مُستشعرالحركة** (PIR).



شكل 6.13: المكونات المستخدمة بالمشروع

لإنشاء الدائرة:

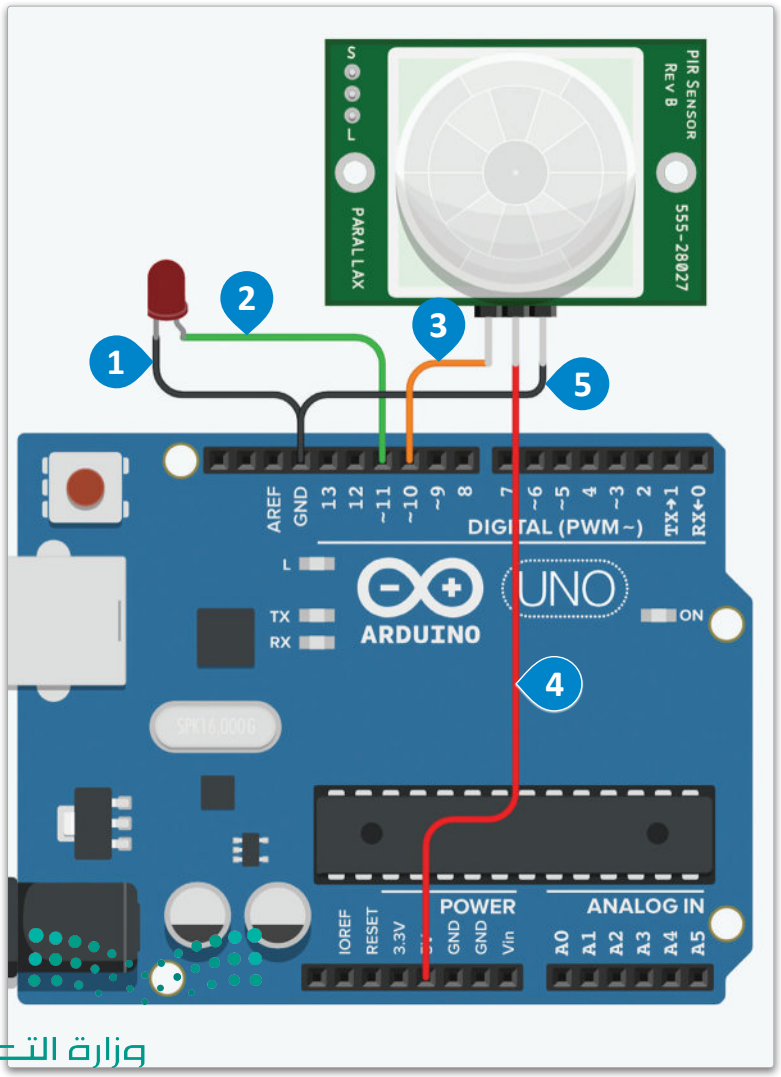
> قم بتوصيل Cathode (مهبط) لـ LED (دايود مُشع للضوء) بالطرف GND (الطرف الأرضي) للوحة الأردوينو، وغيّر لون السلك إلى black (الأسود). 1

> قم بتوصيل Anode (مصعد) لـ LED (لدايود مُشع للضوء) بالطرف Digital (الرقمي) 11 للوحة الأردوينو، وغيّر لون السلك إلى green (الأخضر). 2

> قُم بتوصيل طرف إشارة PIR (مُستشعر الحركة) بالطرف Digital (الرقمي) 10 للوحة الأردوينو، وغيّر لون السلك إلى اللون orange (البرتقالي). 3

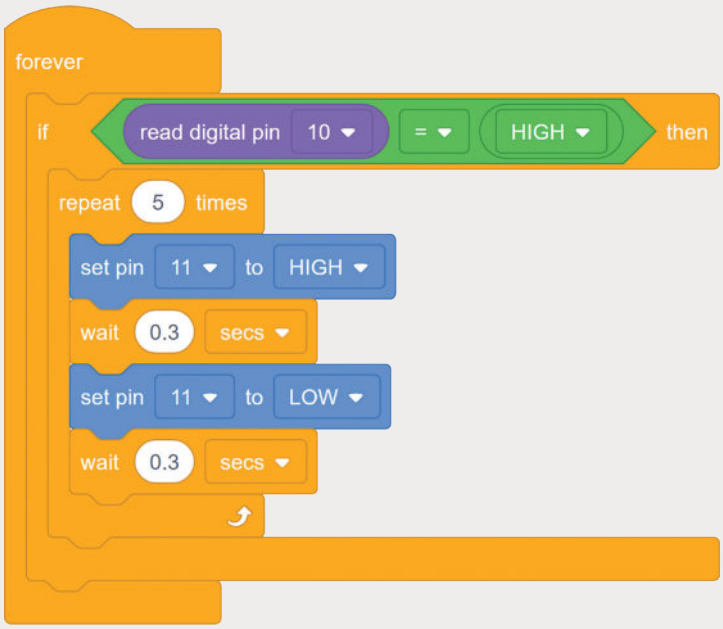
> قُم بتوصيل طرف إشارة PIR (مُستشعر الحركة) بمصدر 5V (5 فولت) من لوحة الأردوينو، وغيّر لون السلك إلى red (الأحمر). 4

> قُم بتوصيل GND (الطرف الأرضي) لـ PIR (مُستشعر الحركة) بالطرف GND (الطرف الأرضي) للوحة الأردوينو، وغيّر لون السلك إلى black (الأسود). 5



شكل 6.14: توصيل الدائرة

لبنات تينكركاد



برمجة الأردوينو

سيتحقق البرنامج مما إذا كان مستشعر (PIR) قد اكتشف كائنًا في مجال رؤيته، وعند اكتشافه لشيء ما، سيرسل إشارة للدايود المشع للضوء ليومض خمس ومضاتٍ سريعة متتابعة.

اختر وضع البرمجة نص (Text) في محرر التعليمات البرمجية لرؤية النص الناتج بلغة C++.

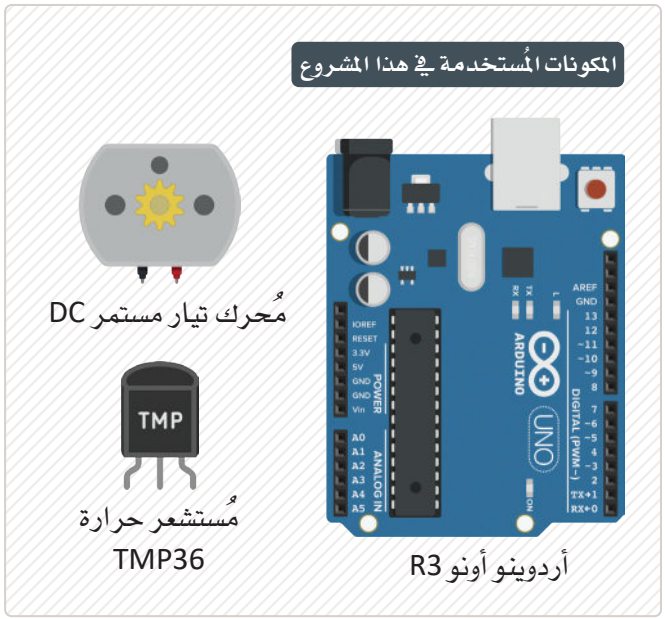
C++

```
int counter;
void setup() {
  pinMode(10, INPUT);
  pinMode(11, OUTPUT);
}
void loop() {
  if (digitalRead(10) == HIGH) {
    for (counter = 0; counter < 5; ++counter) {
      digitalWrite(11, HIGH);
      delay(300); // Wait for 300 millisecond(s)
      digitalWrite(11, LOW);
      delay(300); // Wait for 300 millisecond(s)
    }
  }
}
```

مثال محرك التيار المستمر DC Motor Example

ستُنشئ دائرة بسيطة للتحكم في محرك DC وِفق درجة الحرارة المحيطة به. ستحتاج إلى المكونات الآتية:

- لوحة أردوينو أونو R3 (Arduino Uno R3).
- محرك تيار مستمر (DC motor).
- مُستشعر درجة الحرارة (TMP36).



شكل 6.15: مكونات المشروع

لإنشاء الدائرة:

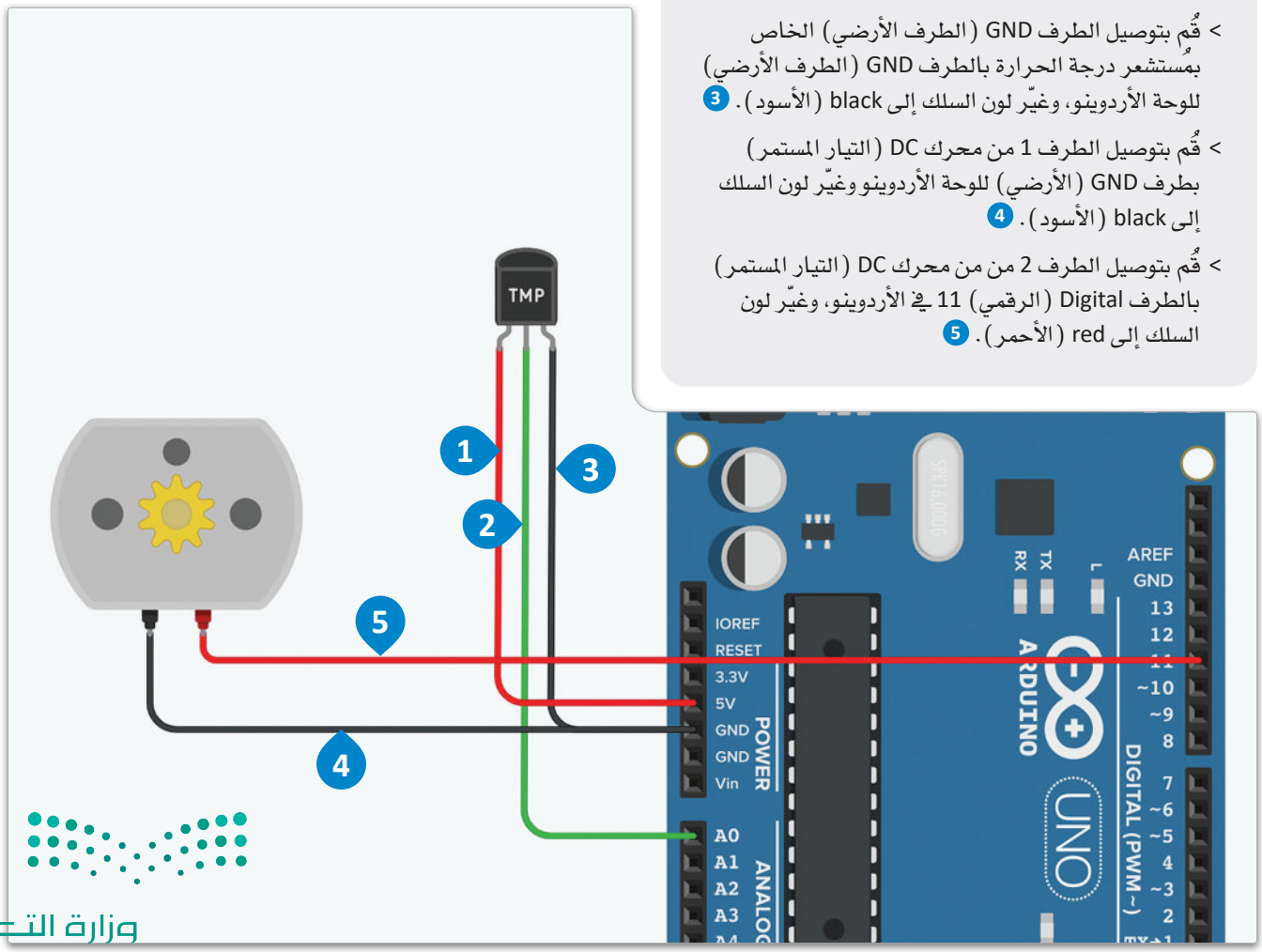
> قُم بتوصيل طرف Power (الطاقة) الخاص بمُستشعر درجة الحرارة بطرف 5V (5 فولت) من لوحة الأردوينو، وغيّر لون السلك إلى red (الأحمر). 1

> قُم بتوصيل طرف مخرج الجهد (Vout) الخاص بمُستشعر درجة الحرارة بالطرف التناظري (A0) للوحة الأردوينو وغيّر لون السلك إلى green (الأخضر). 2

> قُم بتوصيل الطرف GND (الطرف الأرضي) الخاص بمُستشعر درجة الحرارة بالطرف GND (الطرف الأرضي) للوحة الأردوينو، وغيّر لون السلك إلى black (الأسود). 3

> قُم بتوصيل الطرف 1 من محرك DC (التيار المستمر) بطرف GND (الأرضي) للوحة الأردوينو وغيّر لون السلك إلى black (الأسود). 4

> قُم بتوصيل الطرف 2 من من محرك DC (التيار المستمر) بالطرف Digital (الرقمي) 11 في الأردوينو، وغيّر لون السلك إلى red (الأحمر). 5

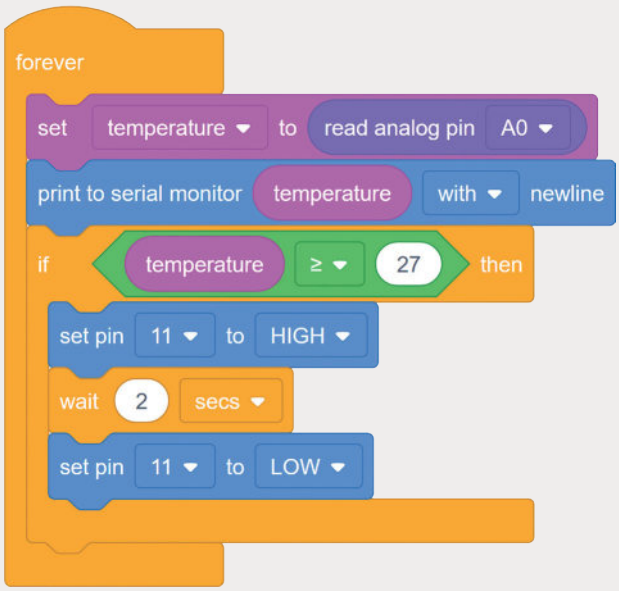


شكل 6.16:توصيل الدائرة

برمجة الأردوينو

سيُنشئ البرنامج متغيّرًا باسم temperature وسيتم توصيله بالطرف التناظري A0 في لوحة الأردوينو لتسجيل درجة حرارة البيئة المحيطة. عندما يصل متغير درجة الحرارة temperature إلى القيمة 27 (درجة مئوية) في محاكي تينكربـاد، يُنشّط محرك التيار المستمر لمدة ثانيتين.

لبنات تينكركاد



اختر وضع البرمجة نص (Text) في محرر التعليمات البرمجية لرؤية النص الناتج بلغة C++.

C++

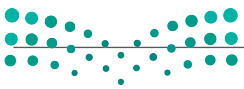
```
int temperature = 0;
void setup() {
  pinMode(A0, INPUT);
  Serial.begin(9600);
  pinMode(11, OUTPUT);
}
void loop() {
  temperature = analogRead(A0);
  Serial.println(temperature);
  if (temperature >= 27) {
    digitalWrite(11, HIGH);
    delay(2000); // Wait for 2000 millisecond(s)
    digitalWrite(11, LOW);
  }
}
```

يُستخدم الكائن التسلسلي (Serial) للطباعة على الشاشة التسلسلية. في دالة ( )setup، تقوم دالة ( )start بتهيئة الشاشة التسلسلية ليُمكن استخدامها لاحقًا. يُمكن للمستخدم بعد ذلك طباعة القيم والرسائل على الشاشة باستخدام دالة ( )print أو دالة ( )println، مع ملاحظة أن دالة ( )println ستضيف سطرًا جديدًا في نهاية الرسالة.

تمرينات

1 اكتب دالة بلغة C++ تستقبل وسيطين عشريين من نوع بيانات float، وإشارة تناظرية، ومُضاعِفًا. مع مراعاة أن تقوم الدالة بتضخيم (مضاعفة) الإشارة ثم إرجاعها.

2 ارسم مُخطط C++ يقرأ إدخال إشارة تناظرية من طرف يُمثل قراءة درجة الحرارة بالفهرنهايت. ثم أنشئ دالة تُحوِّل هذه القيمة إلى درجات مئوية، وترسلها إلى طرف كمُخرج تناظري.



3 ابحث عن خطأ قاعدي (syntax error)، وخطأ منطقي (logical error) في التعليمات البرمجية الآتية:

```
void loop() {
  for (counter = 0; counter < 5; --counter) {
    digitalWrite(11, HIGH);
    // Wait for 1000 millisecond(s)
    delay("1000");
    digitalWrite(11, LOW);
    // Wait for 1000 millisecond(s)
    delay("1000");
  }
}
```

خطأ منطقي _____ خطأ قاعدي _____

```
void loop() {
  temperature = digitalRead(A0);
  Serial.println(temperature);
    if (temperature >= 270) {
    digitalWrite(11, 1);
    // Wait for 2000 millisecond(s)
    delay(2000);
    digitalWrite(11, 0);
    }
}
```

خطأ منطقي _____ خطأ قاعدي _____

4 ارسم مخطط C++ للأردوينو يستخدم الدالة في التمرين الأول ويقرأ مُدخل إشارة تناظري. ثم أنشئ تكرار for يستخدم الدالة في التمرين الأول لتضخيم الإشارة الأصلية 5 مرات. وفي كل مرة تُضخّم الإشارة، تُرسل إلى طرف كمُخرج تناظري.

5 تَوسَّع في المثال السابق الخاص بالدايودات المشعة للضوء الوماضة (LEDs) وقُم بإضافة دايود مشع للضوء آخر بلون مختلف يومض كل مرة يتم فيها إيقاف تشغيل الدايود المشع للضوء الأول.

6 تَوسَّع في المثال السابق الخاص بالإنذار باستخدام مُستشعر الموجات تحت الحمراء، وقُم بإضافة إنذار بواسطة مُستشعر حركة آخر ودايود مشع للضوء بلونٍ آخر. سيُوصل كل مُستشعر حركة بدايود مشع للضوء يومض بناءً على اكتشاف مُستشعر الحركة لشيء ما.

7 قُم بضبط محرك التيار المستمر في المثال الخاص باستخدام المحرك لإرسال إشارة تناظرية إلى المحرك بناءً على درجة الحرارة التي يكتشفها مستشعر درجة الحرارة.

رابط الدرس الرقمي

www.ien.edu.sa

الدرس الثالث

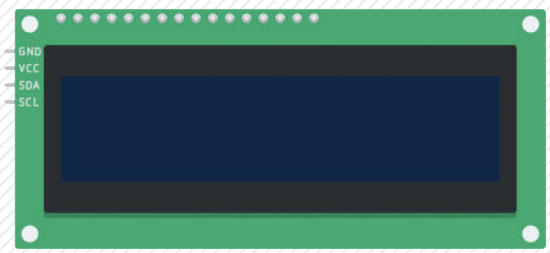
برمجة جهاز التحكم الدقيق باستخدام لغة C++

إنشاء قفل باب ذكي Build a Smart Door Lock

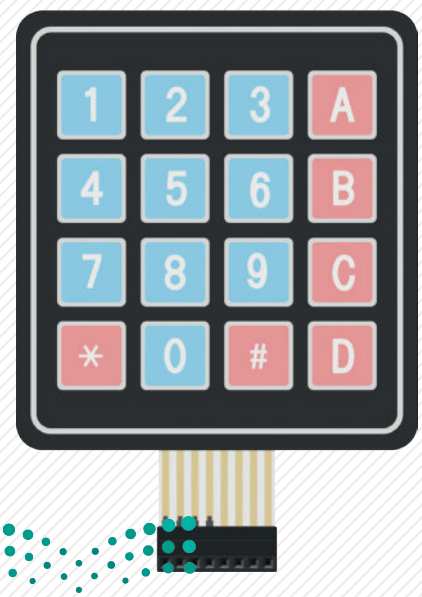
ستستخدم في هذا المشروع المكونات الآتية:

- **لوحة أردوينو أونو** R3 (Arduino Uno R3).
- لوحة مفاتيح (مقاس 4x4).
- شاشة LCD (مقاس 16 × 2 I2C).
- **محرك سيرفو** (Micro Servo).

المكونات المُستخدمة في هذا المشروع

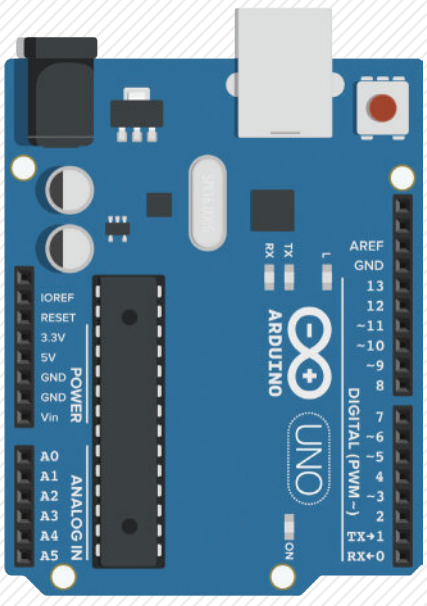


شاشة LCD مقاس (16x2 I2C)



لوحة مفاتيح 4x4

محرك سيرفو



أردوينو أونو R3

شكل 6.17: مكونات المشروع

ستبدأ بإضافة لوحة المفاتيح من فئة Input (الإدخال) من components (المكونات) وتوصيلها بالأردوينو.

لتوصيل لوحة المفاتيح:

> ابحث عن مكون Keypad 4x4 (لوحة المفاتيح) من فئة Input (الإدخال) في Components (المكونات) واسحبه وأفلته في مساحة العمل. 1

> قم بتوصيل السطر الأول من لوحة المفاتيح بالطرف الرقمي 9 الخاص بالأردوينو. 2

> قم بتوصيل السطر الثاني من لوحة المفاتيح بالطرف الرقمي 8 الخاص بالأردوينو. 3

> قم بتوصيل السطر الثالث من لوحة المفاتيح بالطرف الرقمي 7 الخاص بالأردوينو. 4

> قم بتوصيل السطر الرابع من لوحة المفاتيح بالطرف الرقمي 6 الخاص بالأردوينو. 5

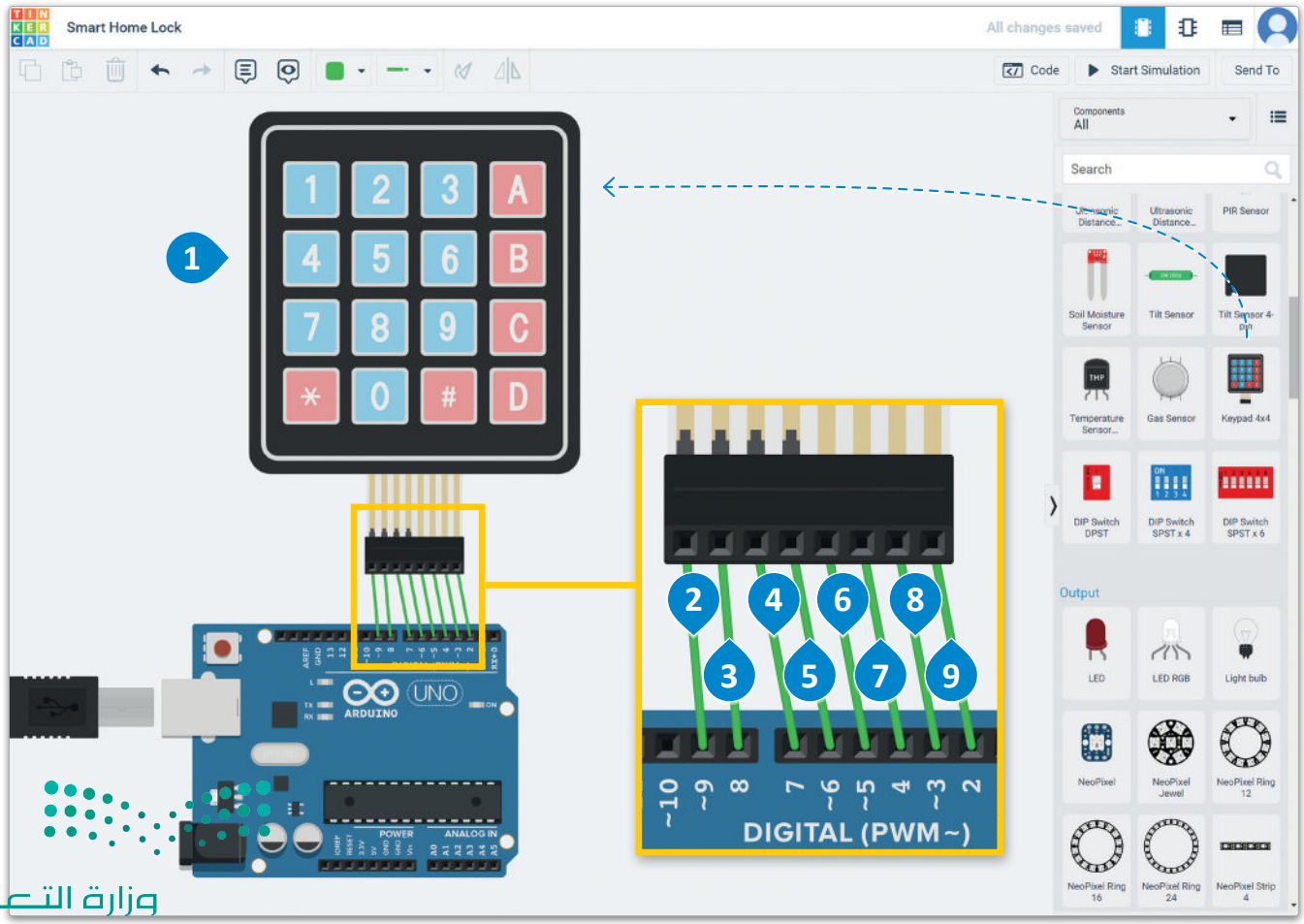
> قم بتوصيل السطر الأول من لوحة المفاتيح بالطرف الرقمي 5 الخاص بالأردوينو. 6

> قم بتوصيل السطر الثاني من لوحة المفاتيح بالطرف الرقمي 4 الخاص بالأردوينو. 7

> قم بتوصيل العمود الثالث من لوحة المفاتيح بالطرف الرقمي 3 الخاص بالأردوينو. 8

> قم بتوصيل العمود الرابع من لوحة المفاتيح بالطرف الرقمي 2 الخاص بالأردوينو. 9

> غيّر كافة الأسلاك إلى اللون green (الأخضر).



شكل 6.18: توصيل لوحة المفاتيح

ابحث الآن عن شاشة LCD من فئة Output (الإخراج) من Components (المُكونات)، ووصلها في breadboard small (لوحة توصيل الدوائر الصغيرة).

لتوصيل شاشة LCD:

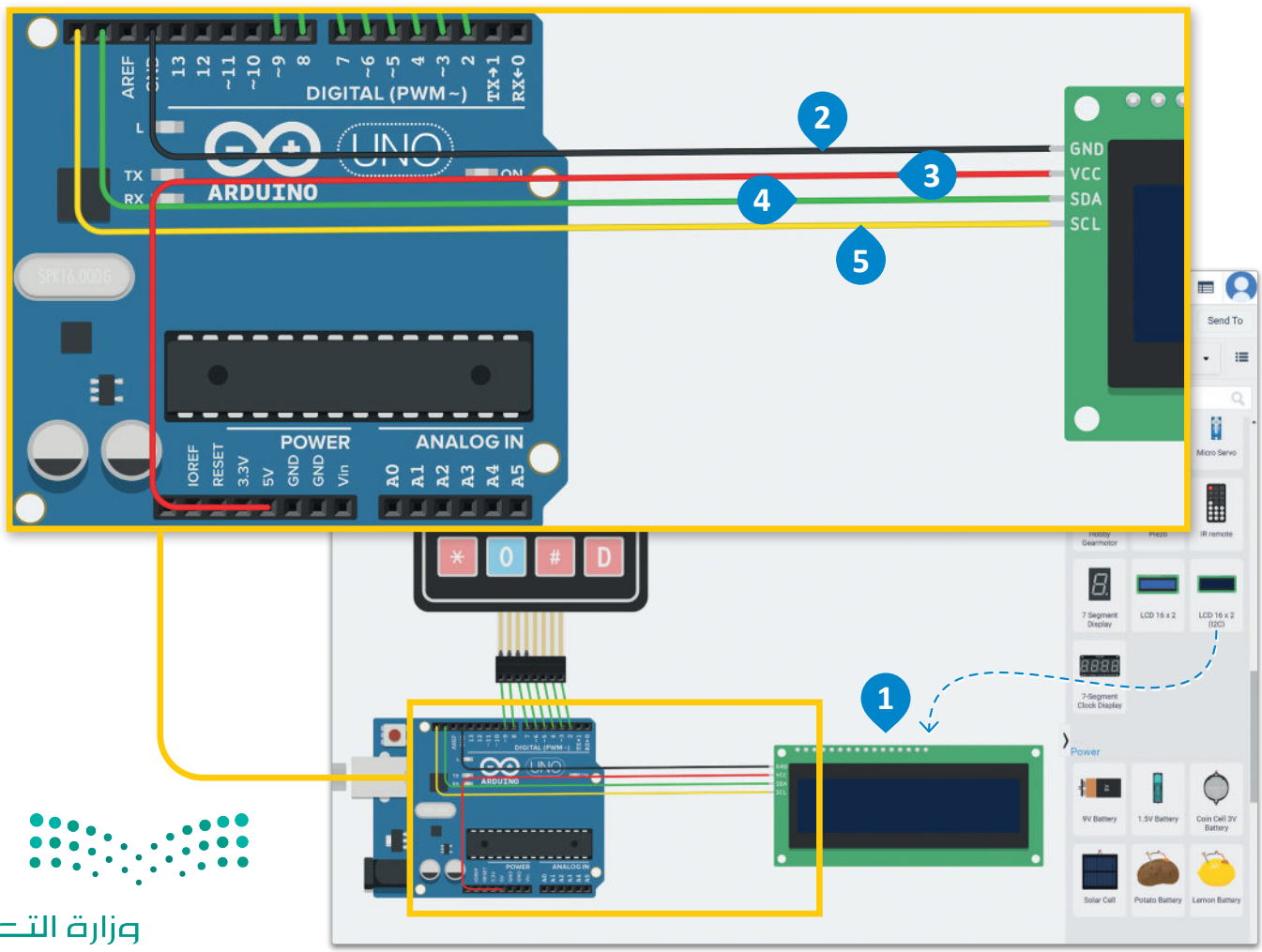
> ابحث عن المكون LCD 16x2 (I2C) من فئة Output (الإخراج) من فئة Components (المُكونات)، واسحبه وأفلته في مساحة العمل. 1

> قم بتوصيل الطرف GND (الأرضي) لشاشة LCD بطرف GND (الأرضي) الخاص بالأردوينو، وغيّر لون السلك إلى black (الأسود). 2

> قم بتوصيل طرف Power (الطاقة) لشاشة LCD بالطرف 5V (5 فولت) بالأردوينو، وغيّر لون السلك إلى red (الأحمر). 3

> قم بتوصيل طرف SDA لشاشة LCD بطرف SDA بالأردوينو، وغيّر لون السلك إلى green (الأخضر). 4

> قم بتوصيل طرف SCL لشاشة LCD بطرف SCL بالأردوينو، وغيّر لون السلك إلى yellow (الأصفر). 5



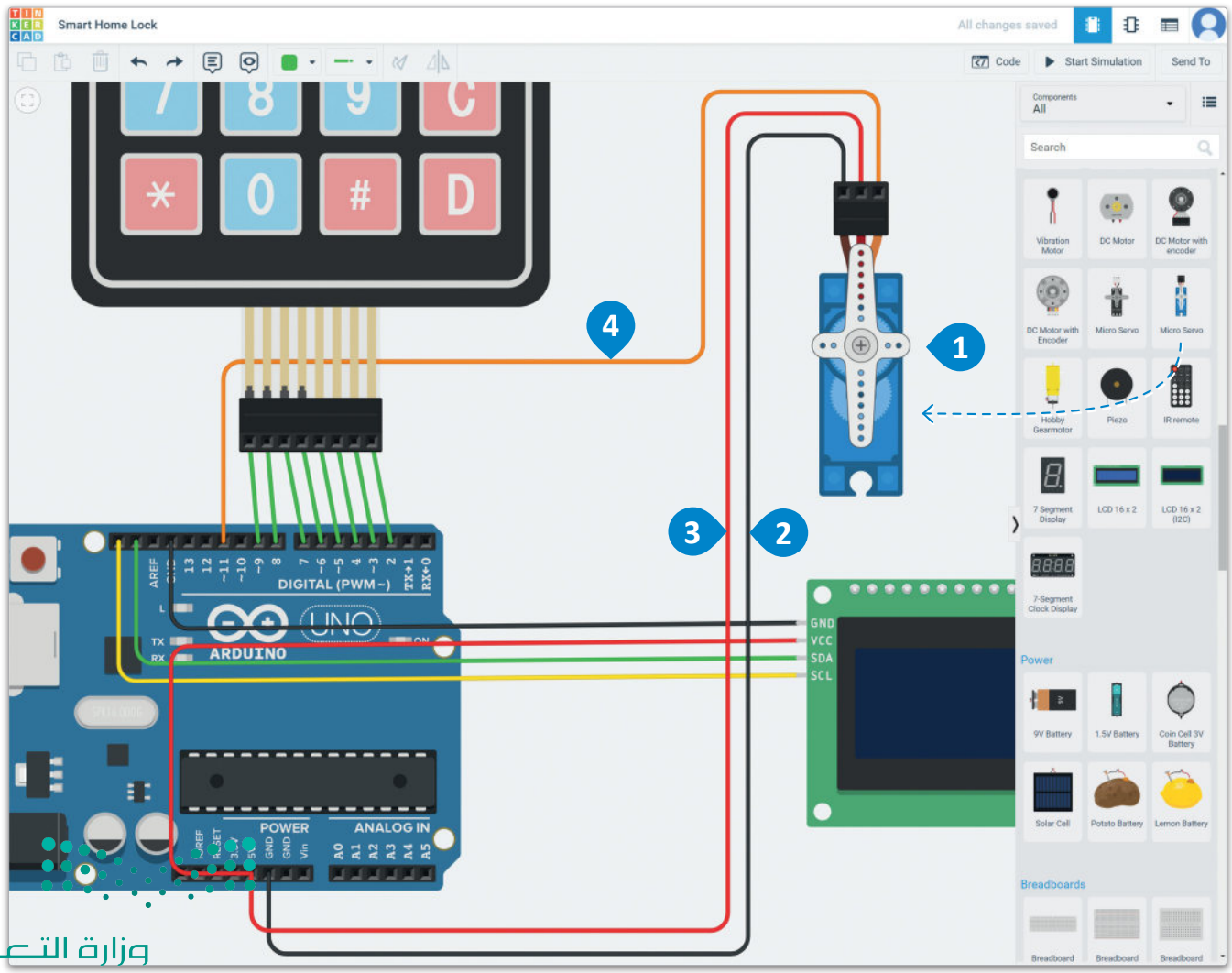
شكل 6.19: توصيل شاشة LCD

ختامًا، ستقوم بتوصيل محرك سيرفو.

ابحث عن Servo motor (محرك سيرفو) من فئة Output (الإخراج) في Components (المُكونات)، ووصله في breadboard small (لوحة توصيل الدوائر الصغيرة).

لتوصيل المحرك المؤازر:

- ابحث عن Servo motor (محرك سيرفو) من فئة Output (الإخراج) في Components (المُكونات)، واسحبه وأفلته في مساحة العمل. 1
- قم بتوصيل الطرف GND (الأرضي) للمحرك بطرف GND (الأرضي) بالأردوينو، وغيّر لون السلك إلى black (الأسود). 2
- قم بتوصيل طرف Power (الطاقة) لشاشة LCD بالطرف 5V (5 فولت) بالأردوينو، وغيّر لون السلك إلى red (الأحمر). 3
- قم بتوصيل طرف Signal (الإشارة) للمحرك بالطرف Digital (الرقمي) 11 بالأردوينو، وغيّر لون السلك إلى orange (البرتقالي). 4



شكل 6.20: توصيل المحرك سيرفو

تضمين المكتبات Include the Libraries

بعيدًا عن وحدة تحكم الأردوينو، ولاستخدام باقي المكونات وبرمجة منطقها بلغة C++، فأنت بحاجة أولًا إلى تضمين مكتباتها في قسم البرمجة في بيئة تينكركاد الأساسية. تُوفِّر هذه المكتبات العديد من الدوال الخاصة بكل مُكوِّن.

ستحتاج إلى كتابة الصيغة الآتية لتضمين مكتبة في C++:

```
#include <library name>
```

بالنسبة للمشروع الحالي، ستحتاج إلى تضمين المكتبات الآتية:

بالنسبة للوحة LCD

```
#include <Adafruit_LiquidCrystal.h>
```

بالنسبة للوحة المفاتيح

```
#include <Keypad.h>
```

بالنسبة للمحرك المؤازر

```
#include <Servo.h>
```

إنشاء الكائنات Create the Objects

بعد تضمينك للمكتبات الضرورية، ستحتاج إلى إنشاء بعض الكائنات وتهيئة بعض الوسيطات.

ستحتاج إلى إنشاء الكائنات الآتية:

- كائن لشاشة LCD.
- كائن **للمحرك المؤازر** (Servo motor).
- كائن للوحة المفاتيح.

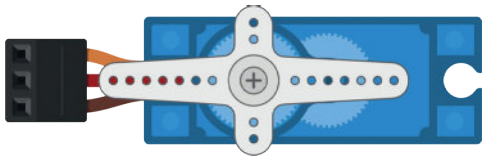
عند إنشاء **كائن** (object) أو **عينة** (instance) **لفئة** (class)، تحتاج أحيانًا إلى تزويد بعض الوسيطات إلى **مُنشئ** (constructor) هذا الكائن. المُنشئ هو وظيفة فئة خاصة تُستدعى عند إنشاء كائن، وتتمثل وظيفتها في تهيئة وسيطات الكائن.

كائن المحرك المؤازر

لإنشاء كائن للمحرك سيرفو:

```
Servo servo;
```

تشير دالة "Servo" إلى نوع الكائن، وتشير دالة "servo" إلى الكائن الفعلي المُستخدم في البرنامج، وهنا لا تحتاج إلى إضافة أي معاملات للتهيئة.



شكل 6.21: محرك سيرفو (كائن تينكركاد)

كائن شاشة LCD

لإنشاء كائن لشاشة LCD:

```
Adafruit_LiquidCrystal lcd(0);
```

يمكنك باستخدام هذا الأمر تهيئة كائن من نوع Adafruit_LiquidCrystal، وتمرير عنوان الأردوينو الابتدائي الخاص به (وهو 0 افتراضيًا) كوسيط إلى مُنشئ الفئة.

شكل 6.22: شاشة LCD (كائن تينكركاد)

كائن لوحة المفاتيح

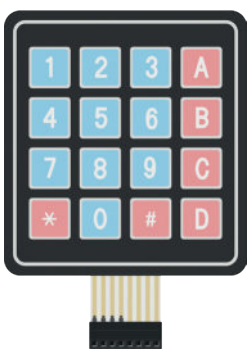
تحتاج عملية الإنشاء والتهيئة لكائن لوحة المفاتيح إلى بعض البرمجة لإعداده. ستحتاج في البداية إلى تحديد عدد الصفوف والأعمدة الموجودة في لوحة المفاتيح.

يتم ذلك بهذه الأوامر:

```
const byte numRows = 4; // number of rows on the keypad
const byte numCols = 4; // number of columns on the keypad
```

تُحدد هنا **عدد الصفوف** (numRows) من النوع "const byte" وقيمته هي 4.

وينطبق الشيء ذاته على **عدد الأعمدة** (numCols).



شكل 6.23: لوحة المفاتيح (كائن تينكركاد)

ستحتاج بعد ذلك إلى تحديد المفتاح المضغوط وفقًا للصف والعمود تمامًا كما يظهر على لوحة المفاتيح. طريقة القيام بذلك هي:

```
// keymap defines the key pressed according to the rows and columns just as
   they appear on the keypad


char keymap[numRows][numCols] =
   {
   {'1', '2', '3', 'A'},
   {'4', '5', '6', 'B'},
   {'7', '8', '9', 'C'},
   {'*', '0', '#', 'D'}
   };
```

ستُنشئ هنا مصفوفة keymap باستخدام numRows و numCols التي قُمت بتعريفها سابقًا، وتحديد المفاتيح الموجودة على لوحة المفاتيح بشكل صريح.

ستحتاج بعد ذلك إلى إعداد اتصالات لوحة المفاتيح بأطراف الأردوينو. يمكنك القيام بذلك عن طريق تحديد متغيرين من نوع byte:

```
// Code that shows the the keypad connections to the arduino terminals
byte rowPins[numRows] = {9,8,7,6}; //Rows 0 to 3
byte colPins[numCols] = {5,4,3,2}; //Columns 0 to 3
```

تتمثل الخطوة الأخيرة في تحديد كائن Keypad عن طريق استدعاء مُنشئه، وتوفير وسيطاته اللازمة.

```
// initializes an instance of the Keypad class
Keypad keypad = Keypad(makeKeymap(keymap), rowPins, colPins, numRows, numCols);
```

لاستكمال برنامج الإعداد، ستُعرّف متغيرًا باسم password يقوم بتخزين كلمة مرور قفل الباب، وهو عبارة عن مجموعة من الأحرف بطول 4.

```
char password[4];
```

إيقاف البرنامج Break down the Code

في هذه المرحلة يكون برنامج الإعداد قد اكتمل. وكما أوضحنا في الدرس الأول، فإن وحدة التحكم في الأردوينو تُنفذ دالة ()setup مرة واحدة فقط عند تشغيلها، ثم تُنفّذ دالة ()loop بصورة مستمرة. دعونا الآن نوقف البرنامج.

التعليمات البرمجية لدالة ()setup هي:

```
void setup()
{
    //servo setup
    servo.attach(11);
    servo.write(0);

    //lcd setup and password set
    lcd.begin(16, 2);
    lcd.setCursor(0, 0);
    lcd.print("Set 4 character");
    lcd.setCursor(0, 1);
    lcd.print("password:");

    for(int i = 0; i < 4; i++) {
       password[i] = keypad.waitForKey();
    }
}
```

تُستخدم دالتي servo من مكتبة Servo كما يلي:

servo.attach(11) تُرفق متغير Servo بالطرف 11.

servo.write(0) تُستخدم لكتابة قيمة إلى servo، في هذه الحالة تُكتب القيمة 0، وتتحكم في عمود الحركة وفقًا لذلك. يحدد هذا زاوية العمود في محرك سيرفو القياسي (بالدرجات)، ثم يُحرّك العمود إلى هذا الاتجاه.

نستخدم بعد ذلك ثلاث دوال لمكتبة Adafruit_LiquidCrystal كالآتي:

lcd.begin(col,row) تهيئ واجهة شاشة LCD، وتحدد أبعاد الشاشة (العرض والارتفاع). يجب استدعاء هذه الدالة ()begin قبل أي أوامر أخرى خاصة بمكتبة LCD. وسيطات هذه الدالة هي:

- cols، وهي عدد الأعمدة الموجودة في الشاشة.
- rows، وهي عدد الصفوف الموجودة في الشاشة؛ ولأن شاشة LCD المُستخدمة16x2 ، فيمكن إعطاء الدالة الوسيطات col=16 وrow=2 وبالتالي تكون صياغتها lcd.begin (16,2) .

الدالة الآتية هي:

تقوم دالة **lcd.setCursor(col,row)** بتحديد الموقع الذي سيُعرض فيه النص المكتوب على شاشة LCD. لعرض عبارة "Set 4 character password" (تعيين كلمة مرور مكونة من 4 أحرف)، فأنت بحاجة إلى كلا الصفين لشاشة LCD. ستُعرض في الصف الأول عبارة "Set 4 character"، وفي الصف الثاني ستُعرض عبارة "password". للقيام بذلك، عليك استدعاء الدالة بالصيغة ;lcd.setCursor(0، 0) قبل عرض العبارة الأولى، ثم استدعاء الدالة مرة أخرى بالصيغة ;lcd.setCursor(0، 1) لعرض العبارة الثانية.

الجزء الأخير من البرمجة في دالة ()setup هو تكرار for الذي يخزن كلمة مرور تتكون من 4 أحرف يكتبها المستخدم على لوحة المفاتيح، في متغير password[4]. للقيام بذلك، تُستخدم دالة مكتبة لوحة المفاتيح.

تُستدعى الدالة **keypad.waitForKey()** والتي ستتعرف على المفتاح الذي تم الضغط عليه، وتخزنه في مصفوفة password (كلمة المرور).

الآن وبالنسبة للمهمة الرئيسة لهذا المشروع، ستُستدعى دالة **التكرار** ( )loop عدة مرات.

أوامر دالة ( )loop هي:

```
void loop()
{
  // clear the screen and display the new message
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Enter password:");

  bool correctPass = true;
  char buttonPressed;

  // this code checks each button pressed against the corresponding password
     digit
  // e.g. it will check the 1st button pressed against the first digit of the
     password and so on
  for (int i = 0; i < 4; i++) {
    buttonPressed = keypad.waitForKey();
    if(password[i] != buttonPressed){
      correctPass = false;
    }
    lcd.setCursor(i, 1);
    lcd.print(buttonPressed);
  }

  delay(1000);

  //this code will be executed if the password is correct
  if (correctPass) {
    // clear the lcd screen
    lcd.clear();
```

```
    // set the cursor to the beginning of the 1st line
    lcd.setCursor(0, 0);
    lcd.print("Correct password!");
    // set the cursor to the beginning of the 2nd line
    lcd.setCursor(0, 1);
    lcd.print("Unlocking...");
    // write the angle by which the servo will rotate
    servo.write(180);
    // wait 5 sec and then rotate the servo to its original angle
    delay(5000);
    servo.write(0);
  }
  else {
    // this code will be executed if the password is wrong
    // clear the lcd screen
    lcd.clear();
    // set the cursor at the 1st col,row
    lcd.setCursor(0, 0);
    // print the message
    lcd.print("Wrong password!");
  }
}
```

إيقاف البرنامج Break this Code down

توجد في البرنامج بعض التعليمات البرمجية لمسح شاشة LCD، ولعرض رسالة تطلب كلمة المرور.

```
  // clear the screen and display the new message
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Enter password:");
```

يتبع ذلك التعليمات البرمجية التي تستقبل كلمة المرور التي يقوم المستخدم بإدخالها وتتحقق من صحتها. يتم ذلك بالمقارنة بين الأزرار التي يتم الضغط عليها واحدًا تلو الآخر بالتتابع مع رقم كلمة المرور الموجود في نفس الموضع.

على سبيل المثال، لنفترض أن كلمة المرور التي ضُبطت في البداية هي "5456" ويقوم المستخدم بكتابة كلمة المرور "5453". نظرًا لأن كل مفتاح يضغط عليه المستخدم سيُقارن بمفتاح كلمة المرور المقابل، فإن ما سيحدث هو:

5 مقارنة مع 5 (نفس الشيء، لا توجد مشكلة)

4 مقارنة مع 4 (نفسها، لا توجد مشكلة)

5 مقارنة مع 5 (لا توجد مشكلة بعد)

3 مقارنة مع 6 (ليستا متطابقتين، لذا فإن كلمة المرور التي تم الضغط عليها غير صحيحة).

عندما يقارن البرنامج بين مفتاحين مختلفين، يجب تحديث المتغير بالمعلومات التي تفيد بأن كلمة المرور غير صحيحة. ولا يُهم ما إذا كان المفتاح الخطأ أول رقم أو آخره أو في أي مكان بينهما، فالنتيجة أن كلمة المرور بأكملها خطأ. لذلك، ولتخزين هذه المعلومات، يمكنك استخدام متغير منطقي يتم تهيئته على أنه **صائب** (true)، وعند الضغط على مفتاح خطأ، تتغير قيمة المتغير إلى **خطأ** (false). وبعد إجراء المقارنة يمكنك التحقق من قيمة هذا المتغير، وإذا كانت هذه القيمة صائبة، فهذا يعني أن المستخدم كتب كلمة المرور الصحيحة، أما إذا كانت هذه القيمة خطأ، فهذا يعني أن المستخدم كتب كلمة مرور خطأ.

يتم تنفيذ الدالة التي تم وصفها أعلاه بواسطة هذا الجزء من البرنامج:

```
bool correctPass = true;
char buttonPressed;


// this code checks each button pressed against the corresponding password
   digit
// e.g. it will check the 1st button pressed against the first digit of the
   password and so on
for (int i = 0; i < 4; i++) {
  buttonPressed = keypad.waitForKey();
  if(password[i] != buttonPressed){
    correctPass = false;
  }
  lcd.setCursor(i, 1);
  lcd.print(buttonPressed);
}
```

والآن نصل إلى الجزء المهم من البرنامج وهو عملية فتح الباب (تدوير محرك سيرفو) إذا كانت كلمة المرور المكتوبة صحيحة، وغلق القفل مرة أخرى بعد فترة من الوقت، أو عرض رسالة تفيد بأن كلمة المرور كانت خطأ.

يتم تنفيذ هذه الدالة بواسطة التعليمات البرمجية الآتية:

```
// this code will be executed if the password is correct
if(correctPass){
   // clear the lcd screen
   lcd.clear();
   // set the cursor to the beginning of the 1st line
   lcd.setCursor(0, 0);
   lcd.print("Correct password!");
   // set the cursor to the beginning of the 2nd line
   lcd.setCursor(0, 1);
   lcd.print("Unlocking...");
   // write the angle by which the servo will rotate
   servo.write(180);
   // wait 5 sec and then rotate the servo to its original angle
   delay(5000);
   servo.write(0);
}
else {
   // this code will be executed if the password is wrong
   // clear the lcd screen
   lcd.clear();
   // set the cursor at the 1st col,row
   lcd.setCursor(0, 0);
   // print the message
   lcd.print("Wrong password!");
   }
```

ختامًا، سيبدو البرنامج بأكمله لمشروع القفل الذكي للباب كالآتي:

البرنامج بأكمله Complete Code

اختر وضع البرمجة نص (Text) في محرر التعليمات البرمجية.

```
// C++ code
//
#include <Adafruit_LiquidCrystal.h>
#include <Keypad.h>
#include <Servo.h>

Adafruit_LiquidCrystal lcd(0);
Servo servo;

const byte numRows = 4; //number of rows on the keypad
const byte numCols = 4; //number of columns on the keypad

// keymap defines the key pressed according to the rows and columns just as they
   appear on the //keypad
char keymap[numRows][numCols] =
{
{'1', '2', '3', 'A'},
{'4', '5', '6', 'B'},
{'7', '8', '9', 'C'},
{'*', '0', '#', 'D'}
};

// Code that shows the the keypad connections to the arduino terminals
byte rowPins[numRows] = {9,8,7,6}; //Rows 0 to 3
byte colPins[numCols] = {5,4,3,2}; //Columns 0 to 3

// initializes an instance of the Keypad class
Keypad keypad = Keypad(makeKeymap(keymap), rowPins, colPins, numRows, numCols);
```

```
char password[4];

void setup()
{
  // servo setup
  servo.attach(11);
  servo.write(0);

  // lcd setup
  lcd.begin(16, 2);
  // lcd print 1st line
  lcd.setCursor(0, 0);
  lcd.print("Set 4 character");
  // lcd print 2nd line
  lcd.setCursor(0, 1);
  lcd.print("password:");

  // gets and stores the password
  for(int i = 0; i < 4; i++){
    password[i] = keypad.waitForKey();
  }
}

void loop() {
  // clear the screen and display the new message
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Enter password:");

  bool correctPass = true;
  char buttonPressed;
```

```
// this code checks each button pressed against the corresponding password digit
// e.g. it will check the 1st button pressed against the first digit of the
   password and so on
for(int i = 0; i < 4; i++) {
  buttonPressed = keypad.waitForKey();
  if(password[i] != buttonPressed) {
    correctPass = false;
  }
  lcd.setCursor(i, 1);
  lcd.print(buttonPressed);
}

delay(1000);

//this code will be executed if the password is correct
if (correctPass){
// clear the lcd screen
lcd.clear();
// set the cursor to the beginning of the 1st line
lcd.setCursor(0, 0);
lcd.print("Correct password!");
// set the cursor to the beginning of the 2nd line
lcd.setCursor(0, 1);
lcd.print("Unlocking...");
// write the angle by which the servo will rotate
servo.write(180);
// wait 5 sec and then rotate the servo to its original angle
delay(5000);
servo.write(0);
}
else {
// this code will be executed if the password is wrong
```

```
  // clear the lcd screen
  lcd.clear();
  // set the cursor at the 1st col,row
  lcd.setCursor(0, 0);
  // print the message
  lcd.print("Wrong password!");
  }
}
```

بعد الانتهاء من كتابة البرنامج، اضغط على زرّ بدء المحاكاة (Start Simulation).

تمرينات

1 أنشئ دائرة في برنامج تينكركاد تتصل بمُستشعر درجة الحرارة وشاشة LCD، ثم قم ببرمجتها باستخدام لغة C++ لعرض درجة الحرارة التي يقوم المستشعر بقراءتها على شاشة LCD.

2 أنشئ دائرة في برنامج تينكركاد تتصل بلوحة مفاتيح 4x4 وشاشة LCD، ثم قم ببرمجتها باستخدام لغة C++ لعرض الأحرف التي يتم الضغط عليها وذلك على شاشة LCD.

3 أنشئ دائرة في برنامج تينكركاد تتصل بلوحة مفاتيح 4x4 ودايودين مشعين للضوء، أحدهما أحمر والآخر أخضر. سيقوم المستخدم بتعيين كلمة مرور ثم سيحاول استخدامها. إذا كان الإدخال صائبًا، سيضيئ الدايود المشع للضوء الأخضر، وإذا كان خطأ، فإنّ الدايود المشع للضوء الأحمر سيومض بصورة متكررة.

4 أنشئ دائرة في برنامج تينكركاد تتصل بمُستشعر رطوبة التربة ومحرك سيرفو، ثم قم ببرمجتها باستخدام لغة C++ لتشغيل محرك سيرفو عند وصول درجة رطوبة التربة إلى قيمة معينة.

المشروع

يُعدُّ نظام الحماية الذكية جزءًا واحدًا فقط من نظام المنزل الذكي الكامل لإنترنت الأشياء. توجد العديد من التطبيقات الأخرى لإنترنت الأشياء المنزلية، من أهمها تنظيم درجة الحرارة.

في هذا المشروع ستقوم بتوسعة الدائرة وبرمجة مشروع قفل الباب الذكي من جديد لإضافة المزيد من المكونات الإلكترونية للتحكم في درجة الحرارة المنزلية.

1

القراءات البيئية التي يجب مراقبتها هي درجة الحرارة والساعة الحالية.

ستُراقب درجة الحرارة بواسطة مستشعر درجة الحرارة، والحصول على الوقت بواسطة ترانزيتور ضوئي (phototransistor) يشير إلى مستويات الضوء خارج المنزل.

2

قم بتوصيل محرك التيار المستمر بالدائرة التي تمثل مُنظم الحرارة وشاشة LCD أخرى. ستعرض شاشة LCD درجة الحرارة الحالية بالدرجات المئوية. سيُنشّط محرك التيار المستمر عن طريق إشارة تناظرية اعتمادًا على القراءات من البيئة المحيطة.

3

أنشئ مستويات مختلفة من درجات الحرارة وظروف الإضاءة التي ستُرسل قيمًا تناظرية مختلفة إلى محرك التيار المستمر.

تحتاج البيئات الأكثر برودة إلى المزيد من المخرجات من منظم الحرارة (محرك التيار المستمر). أنشئ الدائرة وقم ببرمجتها باستخدام لغة C++ لتمثيل التنظيم التلقائي لدرجة الحرارة.

وزارة التعليم
Ministry of Education
2023 - 1445

ماذا تعلّمت

- تحديد المزايا والمخاطر لأنظمة الأمان المبنية على إنترنت الأشياء.
- تعيين أمثلة حول أجهزة إنترنت الأشياء المستخدمة في أنظمة الحماية الذكية.
- استخدام الأوامر الأساسية في لغة C++.
- برمجة جهاز تحكم الأردوينو الدقيق باستخدام لغة C++.
- إنشاء دائرة إلكترونية في تينكركاد وبرمجتها باستخدام لغة C++.

المصطلحات الرئيسة

تكرار	loop()
لغة البرمجة الكائنية	Object Oriented Programming Language
كائن	Object
إعداد	setup()
الحماية الذكية	Smart Security

لغة C++	C++
فئة	Class
لغة برمجة عالية المستوى	High Level Programming Language
لوحة مفاتيح	Keypad
شاشة LCD	LCD display

7. الرسائل في إنترنت الأشياء

سيتعرف الطالب في هذه الوحدة على التطبيقات الخاصة بالمدن الذكية، وعلى أساسيات بروتوكول نقل القياس عن بعد في قائمة انتظار الرسائل (Message Queuing Telemetry Transport - MQTT)، كما سينشئ تطبيق إنترنت الأشياء باستخدام متحكم الأردوينو وبروتوكول (MQTT)، وفي الختام سيقوم بإجراء عمليات لتحليل البيانات على التطبيق المدمج.

أهداف التعلُّم

بنهاية هذه الوحدة سيكون الطالب قادرًا على أن:

- يتعرّف على طبقات هيكلية المدن الذكية.
- يحدّد أمثلة على المدن الذكية.
- يصف وظيفة بروتوكول (MQTT).
- يصنّف جودة الخدمة (QoS) لبروتوكول (MQTT).
- يستخدم البرمجة النصية في بايثون لنشر الرسائل إلى عميل MQTT X .
- إنشاء ملف بيانات جسون (JSON) لتخزين التقارير.
- استخدام مُفكرة جوبيتر(Jupyter) لإجراء عمليات تحليل البيانات في ملف بيانات (JSON).

الأدوات

- بيئة واجهة الأردوينو (Arduino IDE)
- جيت برينز باي تشارم (JetBrains PyCharm)
- دوائر أوتوديسك تينكركاد (Autodesk Tinkercad Circuits)
- عميل MQTT X

وزارة التعليم
Ministry of Education
2023 - 1445

رابط الدرس الرقمي

www.ien.edu.sa

الدرس الأول

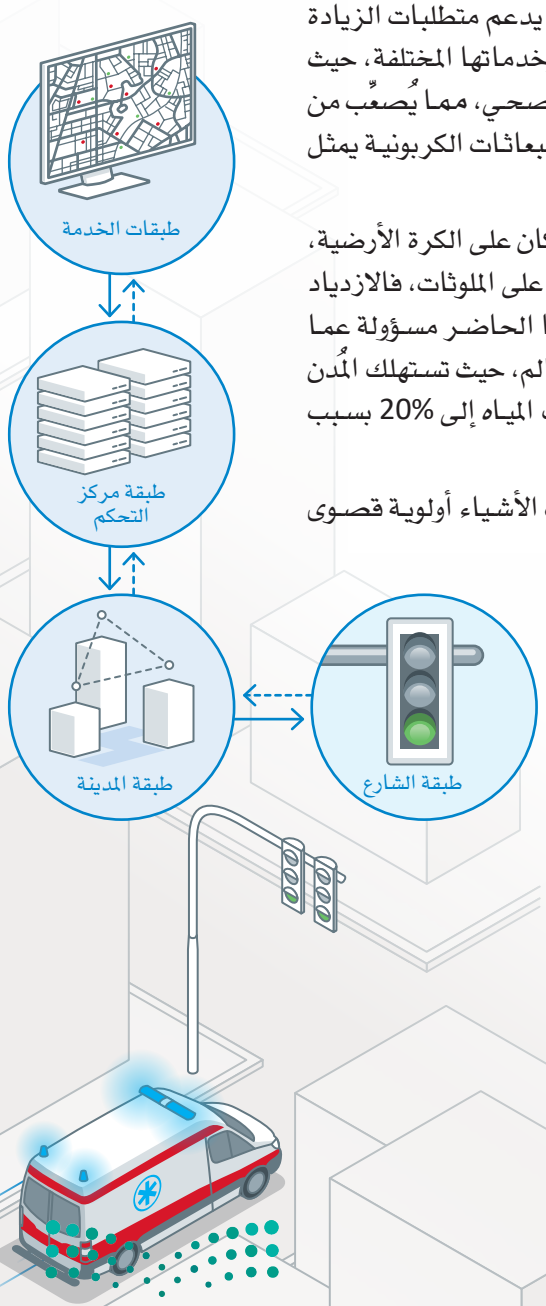
المدن الذكية وبروتوكول MQTT

المدن الذكية Smart Cities

بدأت غالبية المدن كمراكز حَضرية متواضعة، وذلك دون وجود تخطيط مُسبق يدعم متطلبات الزيادة السكانية المتسارعة. يُؤثر التوسع العُمراني المطّرد للمدن على بنيتها التّحتية وخدماتها المختلفة، حيث يتم تجاوز الطاقة الاستيعابية القصوى للطرق والجسور وأنظمة الصرف الصحي، مما يُصعِّب من طبيعة الحياة فيها، ويجعل توفير الأساسيات مثل الماء والكهرباء وتقليل الانبعاثات الكربونية يمثل تحديًا مباشرًا في هذه المدن.

تزداد الانبعاثات الحرارية واستهلاك الطاقة مع الزيادة الكبيرة في عدد السكان على الكرة الأرضية، كما أن تركُّز السكان في مناطق معينة يَحدُّ من قدرة النظام البيئي على التغلب على الملوثات، فالازدياد في الانبعاثات والنفايات يسهم في تسارع التغير المناخي. وتعدّ المُدن في وقتنا الحاضر مسؤولة عما يقرب من 80%–60 من انبعاثات الطاقة وغازات الاحتباس الحراري في العالم، حيث تستهلك المُدن 60% من مجموع المياه الصالحة للشرب، بينما تصل نسبة الفاقد من تلك المياه إلى 20% بسبب التسريبات في شبكات المياه.

يُعدُّ تحسين الموارد ومعالجة النفايات والانبعاثات باستخدام تقنيات إنترنت الأشياء أولوية قصوى لجميع السلطات المسؤولة عن المدن في جميع أنحاء العالم.



شكل 7.1: مُخطط إنترنت الأشياء في المدينة الذكية

هيكلية المدينة الذكية باستخدام إنترنت الأشياء
A Smart City IoT Architecture

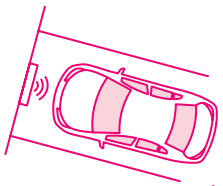
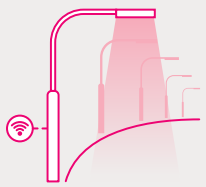
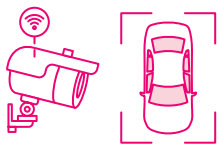
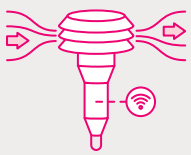
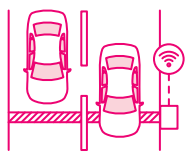
يتمثل التحدي الرئيس لحلول إنترنت الأشياء الذكية في ربط أنظمة معقدة متعددة في تقنية موحدة، وتوجد العديد من مُخططات المدن الذكية المقترحة، ومن أبرزها المُخطط المبني على تقسيم شبكة إنترنت الأشياء للمدن الذكية إلى أربع طبقات، وتشمل كل من طبقة **الشارع** (Street)، وطبقة **المدينة** (City)، وطبقة **مركز البيانات** (Data Center)، وطبقة **الخدمات** (Services).

تنتقل البيانات من الأجهزة الموجودة في طبقة الشارع إلى طبقة شبكة المدينة، حيث يتم دمجها وتوحيدها وتخزينها، وتقوم طبقة مركز البيانات بتغذية المعلومات في طبقة الخدمات التي تشتمل على تطبيقات مزود الخدمة للمدينة.

طبقة الشارع Street Layer

تتكون طبقة الشارع من أجهزة ومُستشعرات تجمع البيانات وتعمل وفقًا لمتطلبات النظام المتكامل، وذلك وفقًا لمكونات الشبكات اللازمة لجمع هذه البيانات واختزالها، وتُستخدم في طبقة الشارع مجموعة من الأجهزة ذات الاستخدامات المختلفة في المدن الذكية، كما يظهر في الجدول أدناه:

جدول 7.1: أجهزة ومستشعرات طبقة الشارع

النوع	الوصف
 مُستشعر مغناطيسي (Magnetic sensor)	يكتشف المستشعر المغناطيسي عملية رَكْنِ المركبات، من خلال مراقبة التغيرات في المجال المغناطيسي له عند اقتراب جسم معدني ثقيل مثل سيارة أو شاحنة.
 مُستشعرات الإضاءة (Lighting controller)	يمكن لمُستشعر الإضاءة التحكم في الإنارة بناءً على المتغيرات البيئية وعلى الوقت.
 كاميرات المراقبة (Video cameras)	يمكن لكاميرات المراقبة وتقنيات تحليل الصور التعرف على السيارات والوجوه وحالة المرور في مختلف التطبيقات الخاصة بالمرور والحماية.
 مُستشعر جودة الهواء (Air quality sensor)	يمكن لمُستشعر جودة الهواء اكتشاف وتحديد كميات الغازات والجسيمات في الهواء، لمعرفة كمية التلوث بدقة في موقع محدد.
 العدادات (Counters)	تستطيع العدادات المثبتة في الشوارع تسجيل عدد المركبات المُتحركة في الشارع أو المتوقفة في منطقة وقوف عامة، وذلك من أجل توفير التحليلات المرورية وحالة المرور للسائقين.

طبقة المدينة City Layer

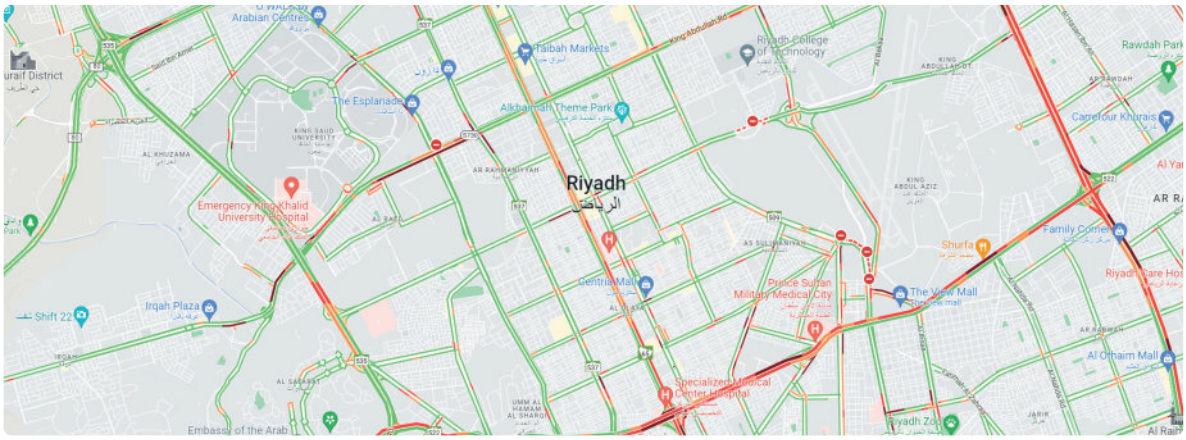
يُمكن النظر إلى طبقة المدينة على أنها طبقة النقل المباشرة بين أجهزة إنترنت الأشياء الطرفية (Edge devices)، ومراكز البيانات أو الإنترنت. ويجب نشر موجهات الشبكة ومحولاتها في هذه الطبقة في مستوى أعلى من طبقة الشارع لدعم نقل البيانات الضخمة، كما يجب أن تنقل طبقة المدينة البيانات من خلال أنواع عديدة من البروتوكولات لتطبيقات إنترنت الأشياء المختلفة. قد تتسم بعض هذه التطبيقات بالحساسية للتأخر الزمني أو لفقدان الحِزم، حيث يمكن أن يؤدي فقدان حِزمة معينة إلى تشغيل تنبيه أو إنشاء تقرير غير صحيح. لذلك، يجب أن تكون طبقة المدينة مَرنة لضمان وصول حِزمة البيانات المُرسلة من مُستشعر أو بوابة إلى وجهتها دائمًا.

طبقة مركز البيانات Data Center Layer

تُرسل البيانات التي تجمعُها المستشعرات إلى مركز البيانات للمعالجة والتخزين، وبناءً على عمليات معالجة البيانات هذه، تُحدّد المعلومات والأنماط المهمة، ثم إنشاء الأفكار ودعم القرارات. على سبيل المثال، يمكن لمركز البيانات إعطاء تصور لحركة المرور على مستوى المدينة، ومساعدة السلطات في تحديد أهمية إضافة مركبات نقل جماعي إضافية أو تقليلها. ويُمكن استخدام نفس بيانات حركة المرور لإدارة مدة عمل إشارات المرور في المدينة ومزامنتها تلقائيًا للتحكم بالازدحام المروري. وتُعدُّ الخدمات السحابية وتخزين البيانات سحابيًا ضرورة لتطوير أي حل شامل لإنترنت الأشياء، كما يمكن تخزين هذه البيانات في مراكز البيانات التابعة لسلطات المدينة أو للشركات الخاصة، وذلك حسب التشريعات المحلية.

طبقة الخدمات Services Layer

تكْمُن الأهمية الفعلية لإنترنت الأشياء في الخدمات التي تقدمها للسلطات والمواطنين، ويجب عرض البيانات التي تُعالج وفقًا للمتطلبات الخاصة بكل مُستهلك للبيانات، ووفق متطلبات تجربة المستخدم وحالات الاستخدام المختلفة. يمكن مثلًا إعادة توجيه الحافلات وأنظمة النقل العام الأخرى لتجنب مواقع الازدحام المروري المتكررة، كما يمكن تسيير المزيد من قطارات الأنفاق بصورة تلقائية وذلك استجابةً لزيادة الازدحام المروري، وتوقع قرارات الركاب باختيار وسائل النقل العام للتنقل بدلًا من السيارات الخاصة في الأيام التي تشهد حركة مرور مُكثفة.



شكل 7.2: التحديث الفوري للحركة المرورية

مثال

تُخطط وزارة الشؤون البلدية والقروية والإسكان لتنفيذ أكثر من 50 مشروعًا لمدن ذكية متصلة بإنترنت الأشياء بحلول عام 2030. تشمل هذه المشاريع الإدارة الذكية لحركة المرور ومواقف السيارات وأنظمة الحفاظ على البيئة، وكذلك إدارة التخلص من النفايات، والإسكان الذكي، وأنظمة إدارة الأراضي، وذلك لتحقيق الهدف الرئيس وهو تحسين نوعية حياة المواطنين وتحقيق الاستدامة المالية وجودة الخدمة.

تطبيقات المدينة الذكية Smart City Applications

الإنارة الذكية للشوارع Connected Street Lighting

شكل 7.3: الإنارة الذكية للشوارع

تُعدُّ إنارة الشوارع إحدى أكثر المرافق الحضرية تكلفة، لكونها تمثل ما يصل إلى40% من إجمالي تكلفة الطاقة، وتبحث المدن عن طرق لخفض تكاليف الإنارة مع تحسين كفاءة التشغيل وخفض النفقات الأولية. يمكن أن يؤدي تثبيت نظام الإنارة الذكية للشوارع إلى توفير كبير في الطاقة مما يُتيح المجال لتقديم خدمات جديدة. ويُعدُّ الانتقال إلى استخدام تقنيات **الدايودات المشعة للضوء** (LEDs) في طليعة الوسائل المُستخدمة للتحول من الإنارة التقليدية إلى الإنارة الذكية للشوارع، وتتميز الدايودات المشعة للضوء بالاستهلاك المُنخفض للطاقة، مما يجعلها مناسبة بشكل مثالي لتطبيقات الحلول الذكية. ويُمكن أيضًا تعديل ألوان الدايودات المشعة للضوء وشدتها وفقًا للحاجة والظروف المحيطة.

التحكم الذكي في الحركة المرورية Smart Traffic Control

شكل 7.4: التحكم الذكي في الحركة المرورية

يُعدُّ الازدحام المروري من أكثر المشاكل شيوعًا في المدن الحديثة، حيث يسهم بشكل كبير في التلوث البيئي وفقدان الإنتاجية، ويشمل الحل الذكي لضبط وتنظيم الحركة المرورية في المدن توفير المعلومات حول عدد السكان، وحركة التنقل، وأعداد المركبات على الطريق، حيث يتم إرسال تلك البيانات إلى المسؤولين عن تخطيط وتنظيم حركة المرور لاتخاذ الإجراءات اللازمة. ومن الممكن تفعيل التطبيقات المرورية من خلال البيانات الواردة من مستشعرات إنترنت الأشياء وذلك لتخفيف الازدحام والتحكم في الحركة المرورية، ويستطيع مخططو المُدن من خلال تحليل البيانات التي تم جمعها خلال فترات زمنية معينة إنشاء استراتيجيات أكثر فاعلية لتقليل الازدحام المروري. يتسبب الازدحام المروري في ارتفاع حوادث المرور، والتي بدورها تزيد من الازدحام المروري حتى تلك البسيطة منها، ويتمثل أحد الحلول الشائعة في التحكم في تدفق حركة السيارات بناءً على كثافة الحركة المرورية. يمكن للتطبيق الذي يكتشف كثافة الحركة المرورية الفورية تنظيم مدة دورة إشارة المرور لتقييد أو إزالة تأثير الازدحام المروري بالتحكم في عدد المركبات المنضمة إلى حركة المرور على الطرق الرئيسة.

البيئة المتصلة Connected Environment

شكل 7.5: محطة جودة هواء ذكية

تُراقب غالبية المدن الكبيرة جودة الهواء، ولكن الكثير من محطات مراقبة جودة الهواء تستخدم معدات مُراقبة قديمة ومُكلفة لجمع هذه البيانات، وعادةً ما تجمع هذه المحطات قراءات دقيقة جدًا، ولكنها تتسم بمحدودية المدى الذي تجمع منه البيانات، وبالتالي يُحتمل ألا يتم تغطية كامل المدينة بالنقاط الكافية، وتؤدي محدودية البيانات التي تُجمع إلى عدم القدرة على تحديد أنماط جودة الهواء بشكل صحيح. إن تكلفة محطات مراقبة جودة الهواء وحجمها يجعلان من الصعب توفير العدد الكافي من هذه المحطات لتوفير معلومات موثوقة على مستوى محلي وتتبع انتقال التلوث في أرجاء المدن على مدى فترة زمنية معينة.

تنبيهات الأمان الذكية Smart Safety Alerts

شكل 7.6: وحدة على جانب الطريق (RSU)

توجد على جانب الطريق وحدة **اتصالات مخصصة للاتصالات قصيرة المدى** (Dedicated Short-Range Communications – DSRC) تعمل كبوابة بين **وحدة التواصل داخل المركبة** (On-Board Unit – OBU) والبنية التحتية للاتصالات، كما تعمل **وحدة الاتصال على جانب الطريق** (Roadside Unit – RSU) بمثابة جهاز اتصال لاسلكي على جانب الطريق وتُوفِّر الاتصال ودعم المعلومات للمركبات المارة بما فيها تحذيرات السلامة والمعلومات المرورية.

تتطلب المدينة الذكية دائمًا معرفة فورية وشاملة بجودة الهواء، ولجمع هذه البيانات تتطلب المدن الذكية ما يلي:

- أنظمة بيانات مفتوحة تتلقى قياسات جودة الهواء من محطات المراقبة الموجودة.
- مستشعرات إنترنت الأشياء منخفضة التكلفة وذات مستوىً من الدقة مماثلًا لذلك الذي يمكن الحصول عليه من محطات جودة الهواء.
- إمكانية تمثيل للبيانات البيئية متوافر للسلطات وللمواطنين، وتخزين سجلات بيانات جودة الهواء السابقة لتتبع الانبعاثات زمنيًا وتحديد اتجاهاتها.

مثال

يهدف مشروع ذا لاين (The Line) في مدينة نيوم الكبرى في المملكة العربية السعودية إلى دمج أحدث تقنيات المدن الذكية المتطورة لتصبح البيئة الحضرية الأكثر تقدمًا من الناحية التقنية.

ستعتمد نيوم بشكل كبير على حلول إنترنت الأشياء للمدن الذكية للوصول إلى هدفها المتمثل في أن تصبح مدينة خالية من الانبعاثات، دون سيارات أو ازدحام مروري.

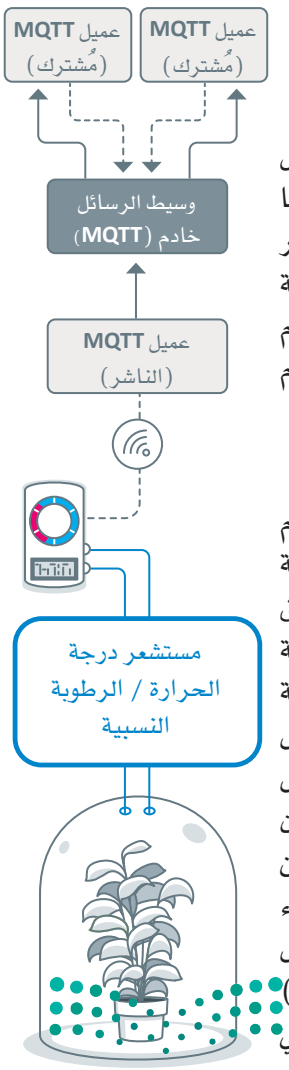
بروتوكول نقل القياس عن بُعد في قائمة انتظار الرسائل

Message Queuing Telemetry Transport - MQTT

مقدمة إلى بروتوكول (MQTT)

Introduction to MQTT

طرح المهندسون من شركتي IBM و Arcom في نهاية التسعينيات من القرن الماضي فكرة تطوير بروتوكول غير معقد وموثوق وفعّال، وكذلك منخفض التكلفة، وذلك لمراقبة الأعداد الكبيرة من المستشعرات، وإدارتها والتعامل مع بياناتها من موقع خادم مركزي، وتحديدًا للاستخدام في قطاعي النفط والغاز، نتج عنه تطوير بروتوكول **نقل القياس عن بُعد في قائمة انتظار الرسائل** (MQTT)، والذي تم توحيده الآن من قِبَل مؤسسة المعايير الدولية المنظمة (OASIS). يُستخدم بروتوكول (MQTT) على نطاق أوسع من استخدام بروتوكول (HTTP) في تطبيقات إنترنت الأشياء، وذلك بشكل أساسي لسهولة إنشاء هياكل مُعقدة باستخدام الأجهزة التي تُرسل وتستقبل حزم البيانات.



شكل 7.7 وظيفة MQTT

أساسيات MQTT Basics

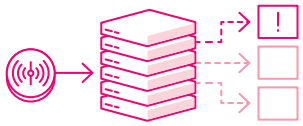
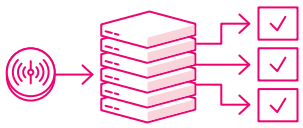
يمكن لعميل (MQTT) أن يكون **ناشرًا** (Publisher) لإرسال البيانات إلى خادم (MQTT) يعمل كخادم رسائل، ويسمى أيضًا **بوسيط الرسائل** (Message Broker). يتلقى خادم (MQTT) الاتصال من شبكة الناشرين ورسائل التطبيق، ويدير هذا الخادم أيضًا عمليات الاشتراك وإلغاء الاشتراك ويقدم بيانات التطبيق لعملاء (MQTT) الذين يعملون **كمُشتركين** (Subscribers). يُمكن للعملاء الاشتراك لاستلام كافة البيانات أو جزء منها من مجموعة معلومات الناشر (MQTT). ويُطلق على عميل (MQTT) في هذه الحالة اسم **المشترك** (Subscriber). يؤدي تضمين وسيط الرسائل في (MQTT) إلى الفصل بين عملية نقل البيانات بين الناشرين والمشتركين، فالناشرون والمشتركون يجهلون بعضهم بعضًا، ويضمن وسيط رسائل (MQTT) إمكانية تأخير المعلومات وتخزينها في حالة فشل الشبكة، وهو ما يُعدُّ ميزة لعملية الفصل بين الناشرين والمشتركين، ولهذا السبب، لا يُطلب من الناشرين والمشتركين الاتصال بالإنترنت في آنٍ واحد. وتتكون جلسة (MQTT) لكل عميل وخادم من أربعة مراحل وهي: إنشاء الجلسة، والمصادقة، وتبادل البيانات، وإنهاء الجلسة، وكل عميل يتصل بخادم لديه مُعرف عميل فريد يحدد جلسة (MQTT) بين الطرفين، ويعامل الخادم كل عميل على حدة عند إرسال رسالة تطبيق إلى العديد من العملاء. من عيوب بروتوكول (MQTT) أنه أبطأ في الإرسال من بروتوكول (HTTP)، كما أنه يجب تنفيذ اكتشاف الموارد وخدمات النسخ الاحتياطي من قِبل المستخدم، ويؤخذ على هذا البروتوكول أيضًا قصوره الأمني في عملية التشفير، وكذلك صعوبة توسيع نطاقه مع زيادة عدد الأجهزة والوسطاء.

جودة الخدمة Quality of Service - QoS

يوفر بروتوكول (MQTT) ثلاث درجات من **جودة الخدمة** (QoS)، وتُطبّق جودة الخدمة لبروتوكول (MQTT) أثناء تبادل رسائل التطبيق مع الناشرين أو المشتركين، كما يتعلق بروتوكول التسليم بشكل أساسي بتسليم رسائل التطبيق من مُرسل واحد إلى مُستلم آخر.

يعرض الجدول الآتي مستويات جودة الخدمة الثلاثة لبروتوكول (MQTT):

جدول 7.2: جودة مستويات الخدمة

المستوى	الوصف
 **مستوى جودة الخدمة 0 (مرة واحدة على الأكثر)** • لا يمكنه التعامل مع الفشل. • لا يتكرر أبدًا.	هذه خدمة بيانات غير مؤكدة، وتعرف باسم أفضل جهد أو (مرة واحدة على الأكثر). يُسلم الناشر رسالة واحدة إلى الخادم الذي ينقلها إلى كل مشترك، ولا يستقبل المُستلم أي إجابة، ولا يحاول المرسل إرسال البيانات مرة أخرى. يتلقى المستلم الرسالة إما مرة واحدة أو لا يتلقاها على الإطلاق.
 مستوى جودة الخدمة 1 (مرة واحدة على الأقل) • يستطيع التغلب على فقدان الاتصال. • يمكن أن يتكرر.	يضمن مستوى جودة الخدمة هذا إرسال الرسائل مرة واحدة على الأقل بين الناشر والخادم، ثم بين الخادم والمشتركين. يضمن هذا المستوى التسليم مرة واحدة على الأقل.
 **مستوى جودة الخدمة 2 (مرة واحدة بالضبط)** • يستطيع التغلب على فقدان الاتصال. • لا يمكن أن يتكرر.	يُعدُّ هذا أعلى مستوى لجودة الخدمة، ويُستخدم في الحالات التي لا تسمح بفقدان الرسالة أو تكرارها. يحتوي مستوى جودة الخدمة هذا على تكلفة إضافية نظرًا لأن كل حزمة تتضمن متغيرًا اختياريًا يحتوي على تعريف الحزمة، ويوفر هذا المستوى "خدمة مضمونة" تسمى التسليم "مرة واحدة بالضبط"، ولا يهم عدد مرات إعادة المحاولة طالما تم إرسال الرسالة مرة واحدة بدقة.

مثال

يُمكن أن تتعرض كائنات إنترنت الأشياء في المدن الذكية للمخاطر بسبب هيكليتها المركزية، حيث لا تتناسب أساليب الحماية التقليدية مع بيئة إنترنت الأشياء المتطورة. ستُطوّر في المملكة العربية السعودية تقنيات سلسلة الكتل (Blockchain) لإنترنت الأشياء في المدن الكبرى لتقليل النقاط المركزية لحالة فشل الشبكة التي تعتمد على الهيكلية الموزعة. ستعتمد شبكة مشروع نيوم العملاق على تقنيات سلسلة الكُتل لإنترنت الأشياء لتوفير بُنية تحتية آمنة للشبكة ولمواطنيها.

تمرينات

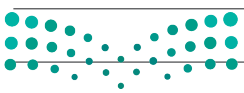
1

حدِّد الجملة الصحيحة والجملة الخاطئة فيما يلي:	صحيحة	خاطئة
1. تُطوّر تقنيات المدن الذكية لتحسين الحركة المرورية فقط.	○	○
2. يجب أن تكون موجهات الشبكة في طبقة المدينة مرنة لمواجهة حالات فُقدان البيانات المُحتملة في الحزم.	○	○
3. تُرسل البيانات مُباشرة من طبقة الشارع إلى طبقة مركز البيانات.	○	○
4. يمكن تخزين البيانات الموجودة في طبقة مركز البيانات على الخوادم الخاصة للشركات.	○	○
5. تحتوي طبقة الخدمات على التطبيقات التي يستخدمها سكان المدينة.	○	○
6. ينحصر استخدام أنظمة إنارة الشوارع الذكية على الدايودات المشعة للضوء (LEDs).	○	○
7. لا يُمكن استخدام البيانات التاريخية التي جُمعت على مدى فترات معينة في الماضي لتوقُّع الحركة المرورية المستقبلية.	○	○
8. يمكن استخدام حلول بيئية مبنيّة على إنترنت الأشياء للحد من الانبعاثات الضارة داخل المدن.	○	○
9. أُنشئ بروتوكول (MQTT) لربط العديد من المُستشعرات من خلال نقطة خدمة واحدة.	○	○
10. في الاتصال الأساسي ببروتوكول (MQTT)، يُدرك الناشر والمشترك وجود الطرف الآخر.	○	○

2 ما الدافع الأساسي وراء تطوير المدن الذكية؟ دوّن أفكارك أدناه.

3 أنشئ مُخططًا يوضح كيفية تدفق البيانات في هيكلية إنترنت الأشياء في المدينة الذكية.

4 اعرض أمثلة حول استخدام المستشعرات في طبقة شوارع المدينة الذكية.



5 صِف كيف يُمكن استخدام الأنظمة المتطابقة في طبقة مركز البيانات في تطبيقات متعددة. دوِّن أفكارك أدناه.

6 اعرض مثالين على تطبيقات المدن الذكية وصِفهما بإيجاز. دوِّن أفكارك أدناه.

7 صِف باختصار آلية عمل بروتوكول (MQTT).

8 صَنِّف درجات جودة الخدمة الثلاث لبروتوكول (MQTT).

9 أنشئ مخططًا لمثال على ثلاثة أجهزة تتصل بواسطة بروتوكول (MQTT)، أحدها كناشر، والجهازان الآخران كمشتركين.

رابط الدرس الرقمي

www.ien.edu.sa

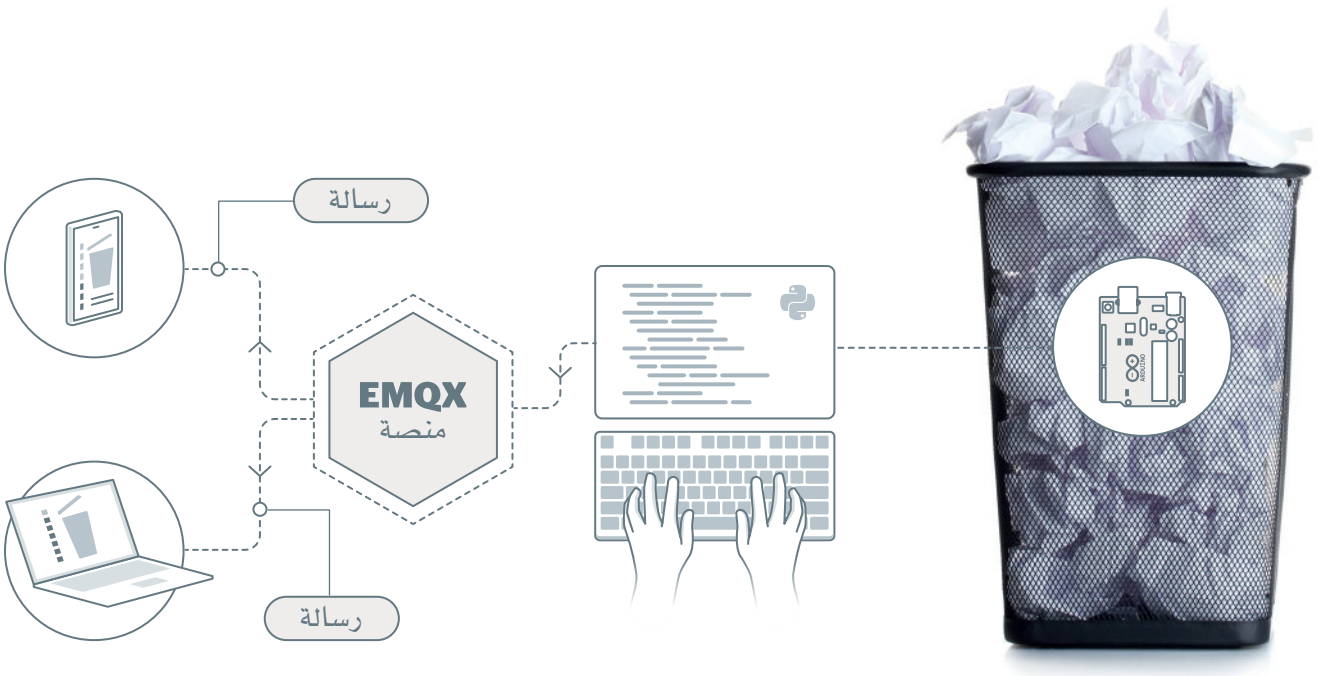
الدرس الثاني

تصميم وبرمجة جهاز ذكي لإنترنت الأشياء

إدارة النفايات الذكية

Smart Waste Management

نظرًا للاكتظاظ السكاني قد لا تُجمع ولا تُعالج كميات كبيرة جدًا من النفايات والمُخلفات بكفاءة، مما يتسبب في زيادة كمية النفايات في عدة أماكن، وتحدث هذه المشكلة بسبب تجاوز سعة حاويات القمامة دون إزالتها في الوقت المناسب. لكن باستخدام حاويات النفايات الذكية، يمكن أن تُرسل رسائل تنبيهية لإعلام مركبات جمع النفايات بهذه الحاويات. كذلك من خلال عمليات تحليل البيانات الملاءمة يُمكن استنباط الأفكار حول كيفية تعبئة حاويات النفايات لتحسين العملية برمتها بشكل أكثر كفاءة.



شكل 7.8: مشروع إدارة النفايات الذكية بالأردوينو وبروتوكول MQTT

سنُنشئ في هذا الدرس نموذجًا أوليًا لحاوية قمامة ذكية تحسب متوسط عدد المرات المطلوبة لتصل إلى سعتها الكاملة. ستُرسل رسالة إلى وسيط (MQTT) كل مرة تُستخدم فيها الحاوية، وعندما تمتلئ الحاوية، تُرسل رسالة أخرى إلى مُتحكم النظام الذي يُنتج التقارير عن الحاوية. ستستخدم في هذا المشروع مُتحكم أردوينو يُمثل حاوية ذكية، وستقوم ببرمجته باستخدام بروتوكول (Firmata) وبايثون، كما ستستخدم منصة (MQTT) لتوزيع الرسائل.

EMQX

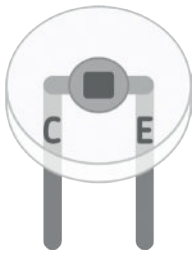
(EMQX) هو وسيط (MQTT) مفتوح المصدر عالي الأداء مع مُحرك لمعالجة الرسائل بصورة فورية. يُستخدم لدعم تدفق الأحداث بواسطة عدد كبير من أجهزة إنترنت الأشياء.

مكونات وأدوات المشروع Components & Tools for Project

الترانزستور الضوئي Phototransistor

هو مُكون كهربائي يعمل عند تعرضه للضوء، حيث تتدفق كمية متناسبة من التيار العكسي عند سقوط الضوء على المستشعر، وتُستخدم أجهزة الترانزستورات الضوئية على نطاق واسع لاكتشاف وتحويل نبضات الإضاءة إلى إشارات كهربائية.

ترانزستور ضوئي



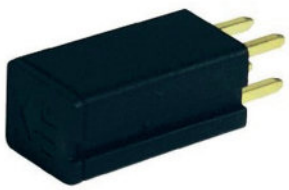
الرمز في المحاكي

رمز تخطيطي

شكل 7.9: مُستشعر ضوئي

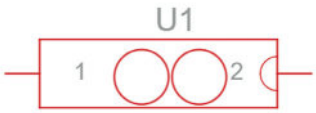
مُستشعر الإمالة Tilt Sensor

يُستخدم مُستشعر الإمالة لقياس درجة الميل على عدة محاور، وتقوم مستشعرات الإمالة بتقييم وضع الميل بالنسبة للجاذبية وتُستخدم في تطبيقات مختلفة، حيث تجعل اكتشاف الميل أو الاتّجاه أمرًا ميسورًا.



مُستشعر الإمالة

الرمز في المحاكي



رمز تخطيطي

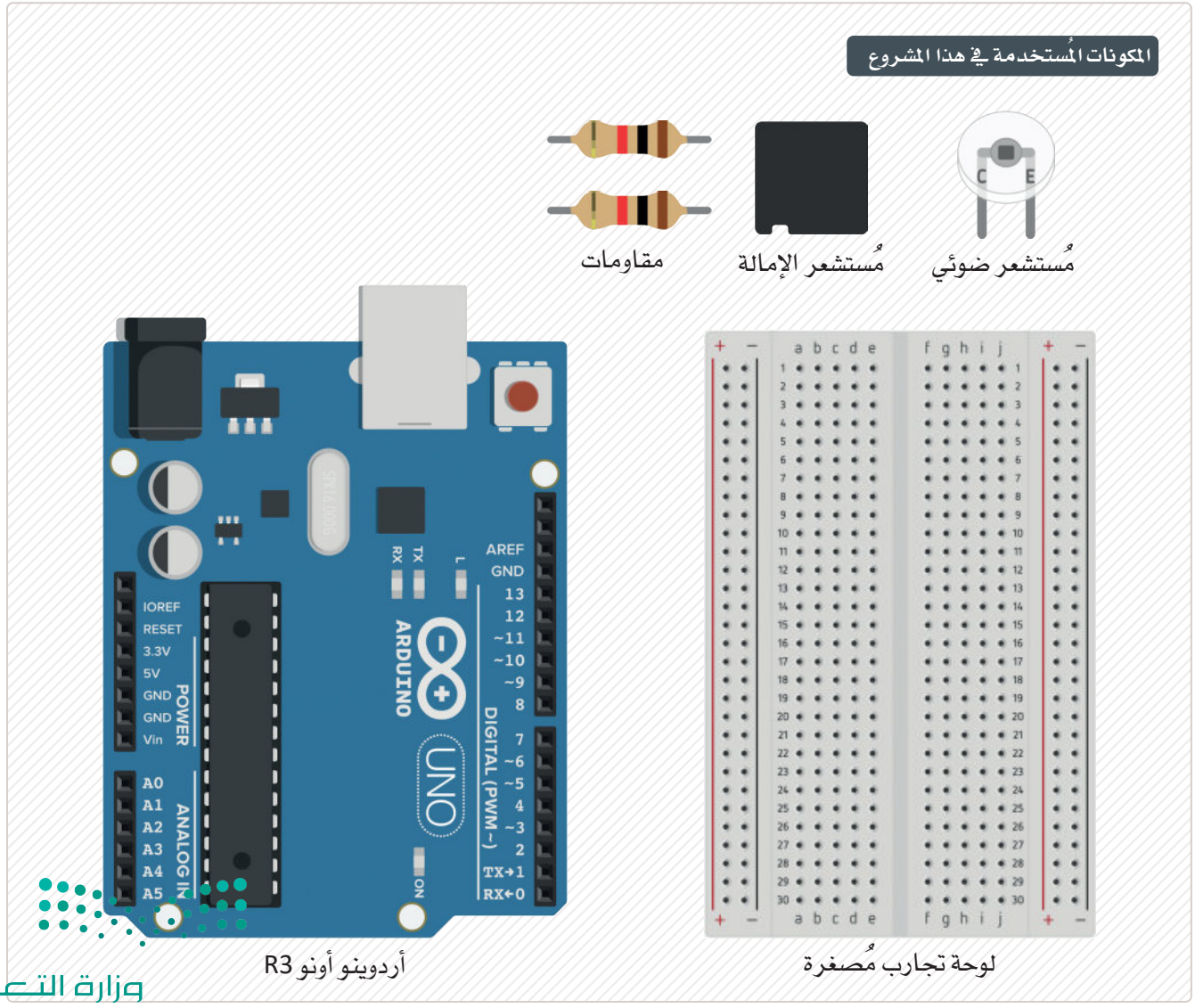
شكل 7.10: مستشعر الإمالة

النموذج الأولي باستخدام الأردوينو Arduino Prototype

سيراقب مُتحكم الأردوينو حالة الحاوية، ويجمع بيانات الأحداث المختلفة ويرسلها من خلال بروتوكول (Firmata). سيُستخدم مستشعر الإمالة لتسجيل استخدام الحاوية في كل مرة، ومحاكاة حركة غطاء الحاوية، وسيعمل الترانزستور الضوئي كمستشعر عند الوصول إلى حدٍ مُعين مما يعني أن الحاوية مليئة بالنفايات.

ستحتاج إلى المكونات الآتية:

- **لوحة أردوينو أونو R3** (Arduino Uno R3).
- **لوحة تجارب مُصغرة** (Breadboard Small).
- **مُستشعر ضوئي** (Phototransistor).
- **مُستشعر إمالة** (Tilt Sensor).
- **مقاومتان 1 كيلو أوم** (Two resistors 1kΩ).



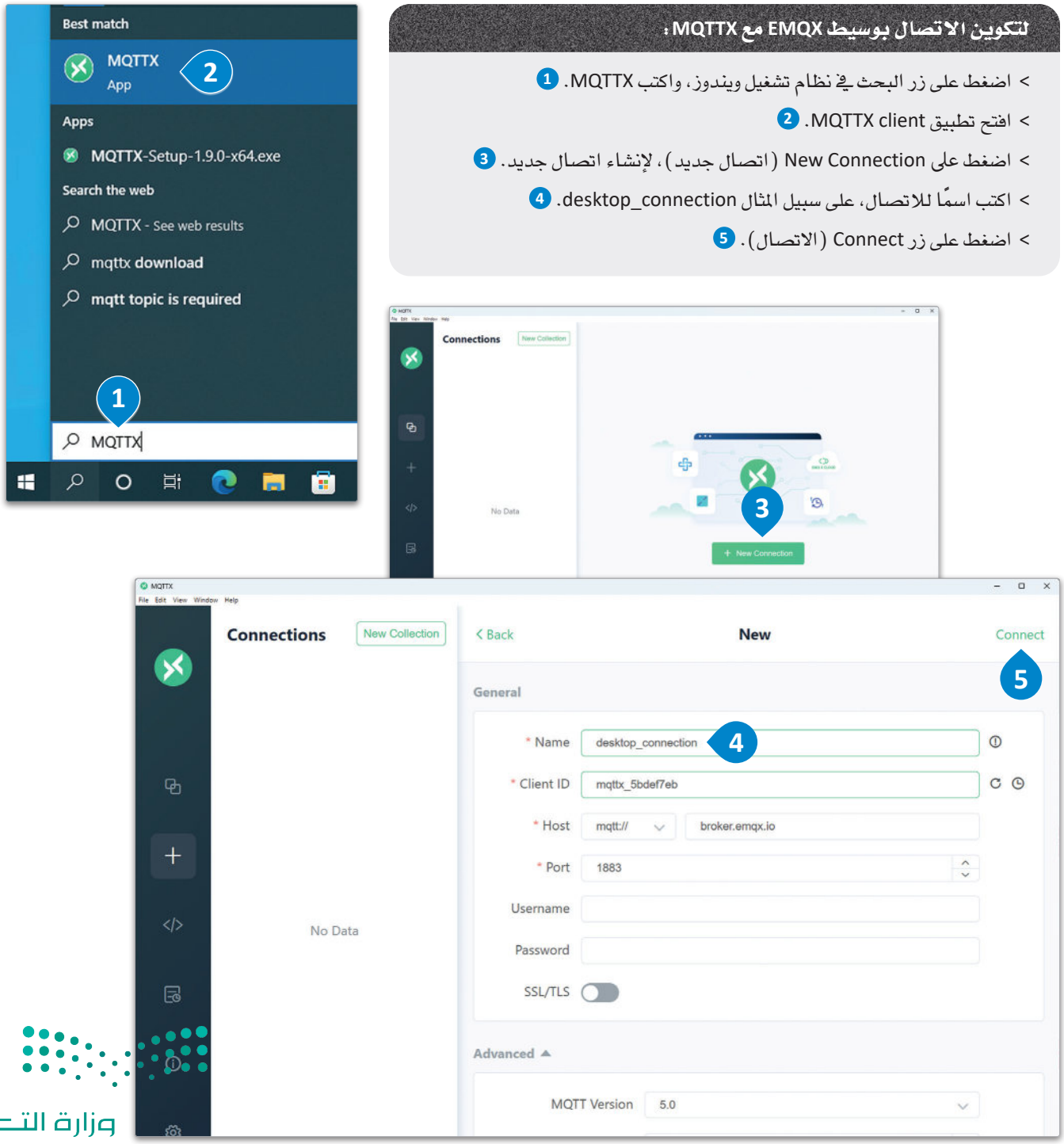
شكل 7.11: مكونات مشروع النفايات الذكية

الاتصال بوسيط EMQX العام Connecting to the EMQX Public Broker

ستحتاج أولًا إلى تثبيت التطبيق المكتبي MQTTX client، ثم اختبار الاتصال مع وسيط (EMQX) العام. لتثبيت تطبيق MQTTX Agent، قم بزيارة موقع الويب: https://mqttx.app/ وقم بتنزيل أحدث إصدار. قم بتشغيل المُثبِّت لإكمال عملية التثبيت. ستقوم الآن بفتح البرنامج العميل، وإنشاء اتصال جديد مع وسيط (EMQX).

لتكوين الاتصال بوسيط EMQX مع MQTTX:

- اضغط على زر البحث في نظام تشغيل ويندوز، واكتب MQTTX. 1
- افتح تطبيق MQTTX client. 2
- اضغط على New Connection (اتصال جديد)، لإنشاء اتصال جديد. 3
- اكتب اسمًا للاتصال، على سبيل المثال desktop_connection. 4
- اضغط على زر Connect (الاتصال). 5



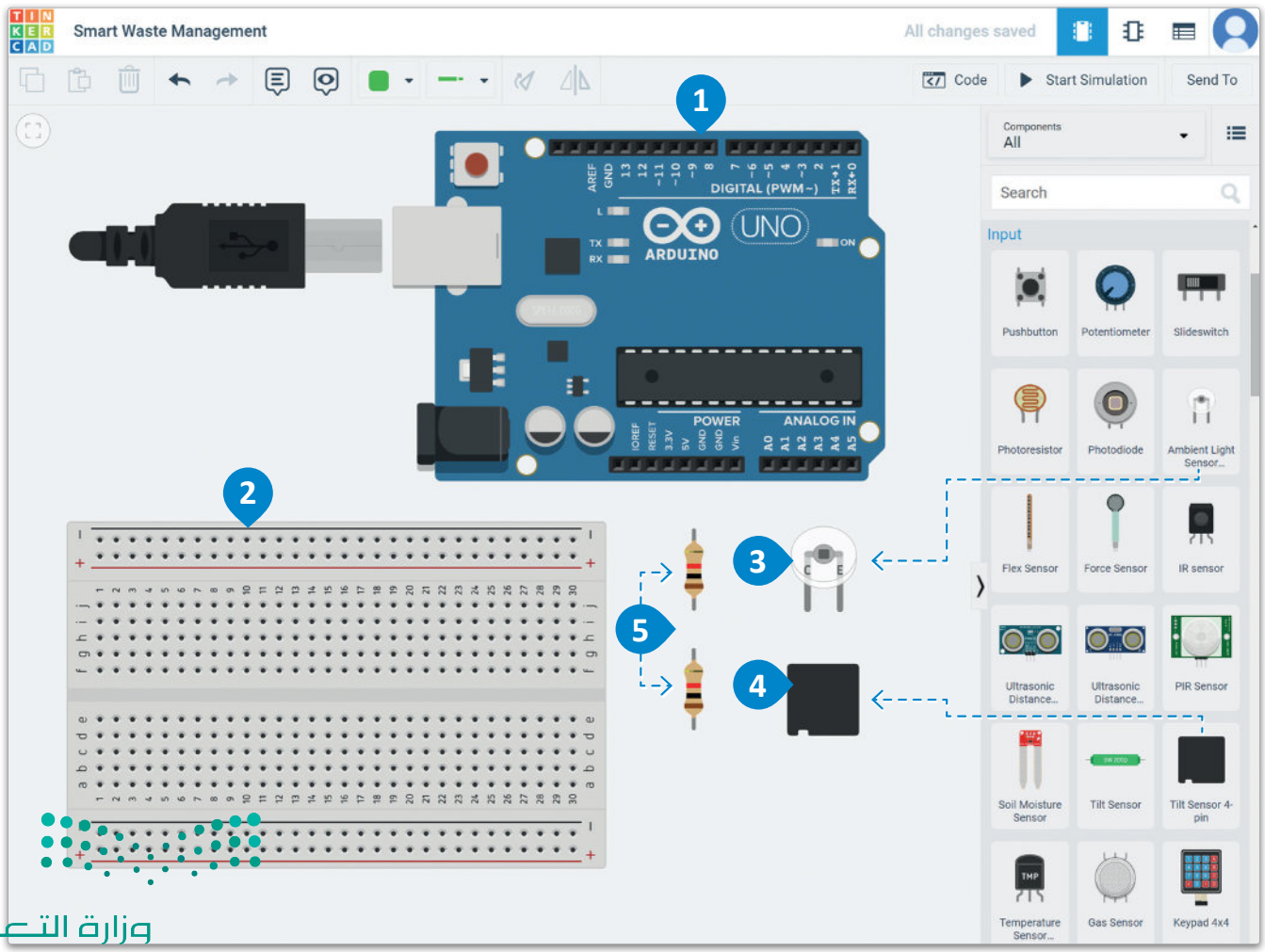
شكل 7.12: إعداد الاتصال بوسيط EMQX مع MQTTX

دائرة الأردوينو Arduino Circuit

ستبدأ في إنشاء دائرة الأردوينو عن طريق إضافة المكونات المطلوبة داخل مساحة عمل دوائر تنكركاد.

لتثبيت المكونات:

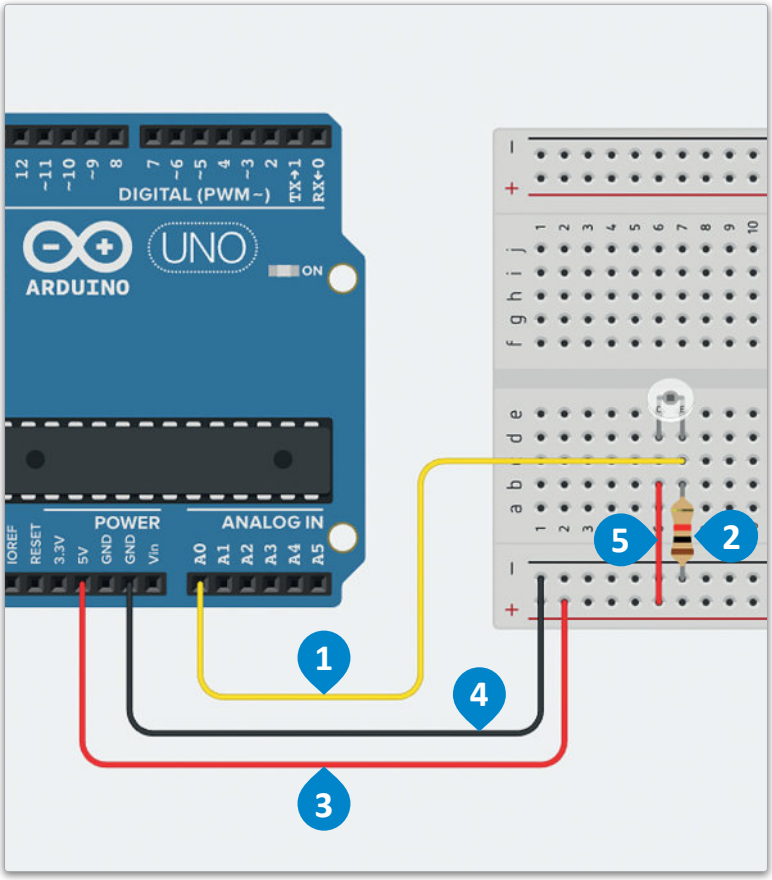
- ابحث عن Arduino Uno R3 (لوحة أردوينو أونوR3) من مكتبة Components (المُكونات)، واسحبها وأفلتها في مساحة العمل. 1
- ابحث عن Breadboard Small (لوحة توصيل الدوائر الصغيرة) من مكتبة Components (المُكونات)، واسحبها وأفلتها في مساحة العمل. 2
- ابحث عن Phototransistor (مُستشعر الإضاءة [الترانزستور الضوئي])، من مكتبة Components (المُكونات) واسحبه وأفلته في مساحة العمل. 3
- ابحث عن Tilt Sensor 4-pin (مُستشعر الإمالة بأربعة أطراف) من مكتبة Components (المُكونات)، واسحبه وأفلته في مساحة العمل. 4
- ابحث عن Resistor (المقاومة) من مكتبة Components (المكونات) واسحب اثنتين منها وأفلتهما في مساحة العمل. 5



شكل 7.13: تثبيت مكونات الدائرة

لتوصيل الترانزستور الضوئي:

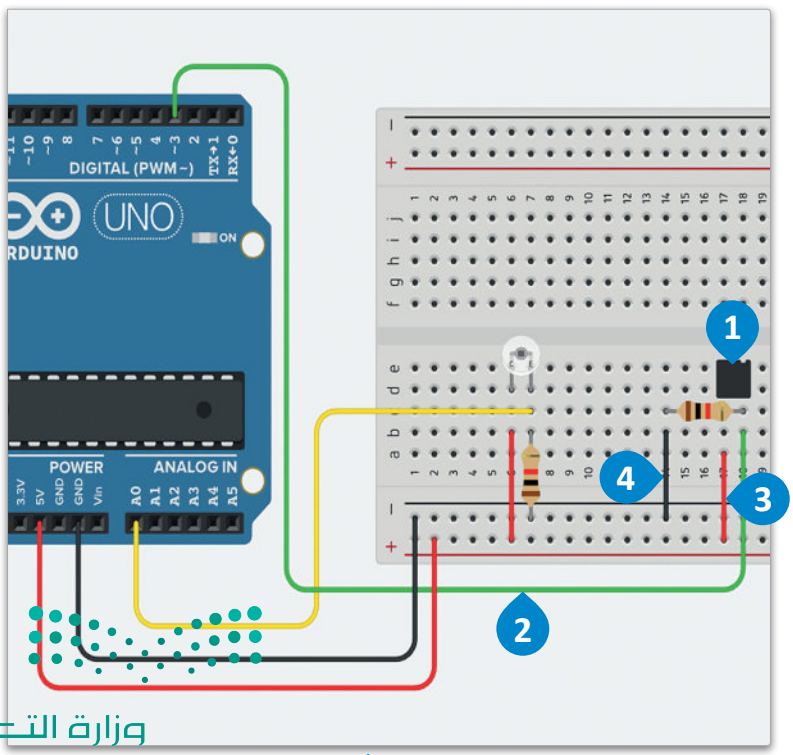
- قم بتوصيل طرف Emitter (الباعث) الخاص بالترانزستور الضوئي بالطرف التناظري A0 في الأردوينو، وغيّر لون السلك إلى yellow (الأصفر). 1
- قم بتوصيل الطرف الثاني للمقاومة الواحدة مع الصف نفسه الذي تم توصيل باعث الترانزستور الضوئي به، ثم وصّل الطرف الأول من المقاومة بالعمود السالب من لوحة الدوائر الصغيرة. 2
- قم بتوصيل طرف 5v (5 فولت) للوحة الأردوينو أونو R3 بالعمود الموجب من لوحة التجارب، وغيّر لون السلك إلى Red (الأحمر). 3
- قم بتوصيل GND (الطرف الأرضي) للوحة الأردوينو أونو بالعمود السالب للوحة التجارب، وغيّر لون السلك إلى Black (الأسود). 4
- قم بتوصيل طرف المُجمع الخاص بالترانزستور الضوئي بالعمود الموجب في لوحة التوصيل المُصغرة. 5



شكل 7.14: توصيل الترانزستور الضوئي

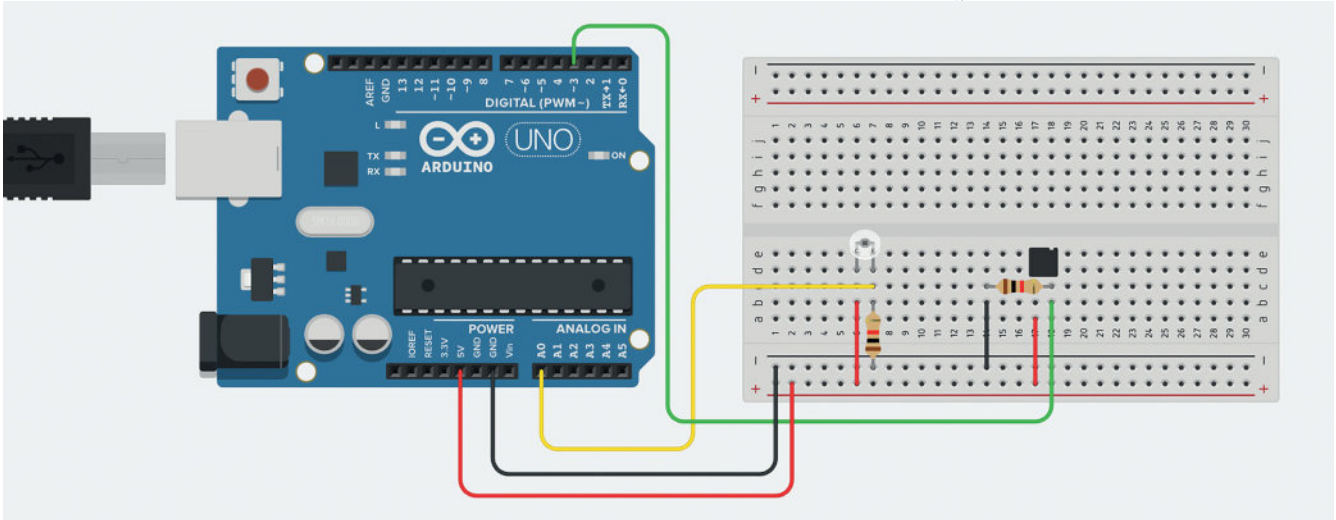
لتوصيل مستشعر الإمالة:

- قم بتوصيل الطرف الثاني للمقاومة الأخرى بالطرف الثاني لمُستشعر الإمالة. 1
- قم بتوصيل الطرف الثاني من مستشعر الإمالة بالطرف الرقمي 3 للوحة الأردوينو، وغيّر لون السلك إلى Green (الأخضر). 2
- قم بتوصيل الطرف الرابع من مستشعر الإمالة بالعمود الموجب من لوحة الدوائر الصغيرة وغيّر لون السلك إلى Red (الأحمر). 3
- قم بتوصيل الطرف الأول من المقاومة بالعمود السالب من لوحة الدوائر الصغيرة وغيّر لون السلك إلى Black (الأسود). 4



شكل 7.15: توصيل مُستشعر الإمالة

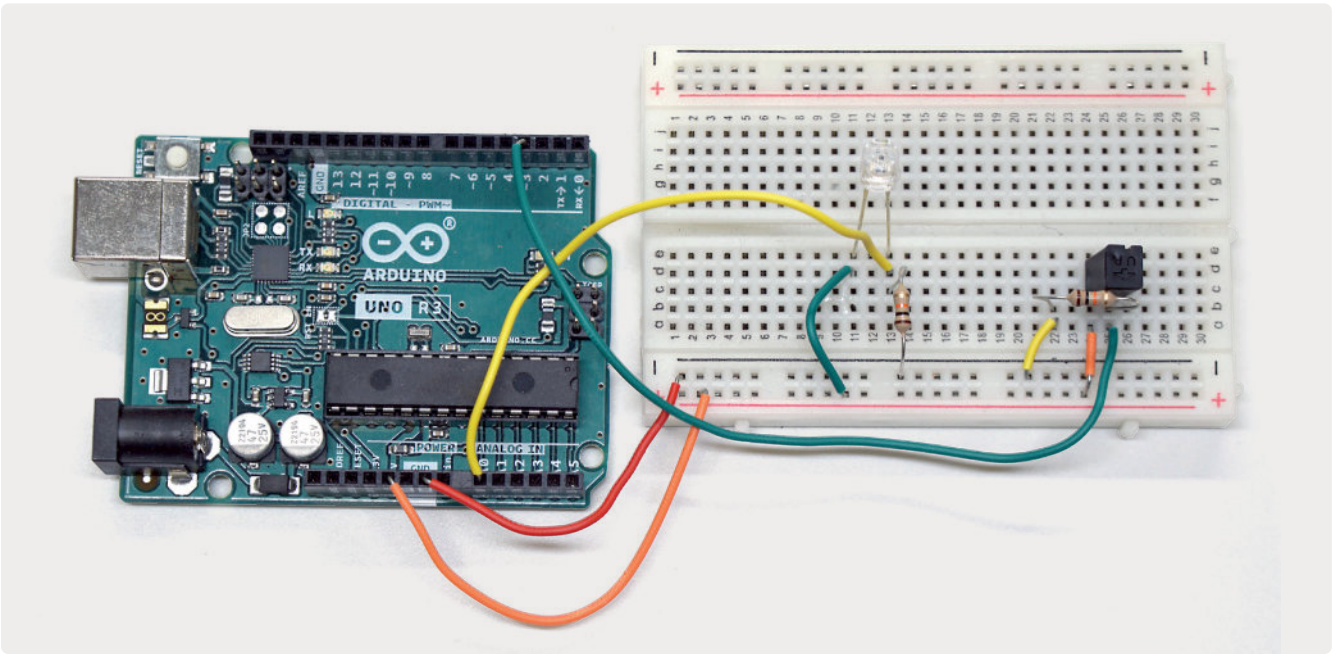
الدائرة بشكلها النهائي Complete Circuit



شكل 7.16: الدائرة بصورتها النهائية في تنكركاد

صورة الدائرة الفعلية Physical Circuit

تمثِّل هذه الصورة الشكل الذي ستبدو عليه الدائرة.



شكل 7.17: صورة للدائرة

تتصل المكونات بالأطراف الآتية:

شكل 7.18: توصيل الأطراف بالمكونات

برمجة الأردوينو Programming the Arduino

ستبدأ بتحميل بروتوكول StandardFirmata من خلال بيئة عمل Arduino IDE لإعداد قناة اتصال بين الأردوينو والبرنامج الذي ستقوم بكتابته بلغة البرمجة بايثون Python.

افتح PyCharm وقم بتثبيت حِزمة paho-mqtt من خلال نظام إدارة الحِزم (pip).

في PyCharm، افتح الواجهة الطرفية (Terminal) في مُجلد عملك، واكتب الأمر الآتي:

```
pip install paho-mqtt
```

قم بإنشاء ملف بايثون جديد باسم mqtt_arduino.py، ثم في بداية مقطعك البرمجي، ثم قم باستيراد الحزم الآتية:

- **datetime**: إنشاء طوابع زمنية للرسائل التي نرسلها.
- **time**: التحكم في سير البرنامج.
- **json**: العمل مع كائنات JSON.
- **pyfirmata**: التواصل مع لوحة الأردوينو من خلال بروتوكول Firmata.
- **paho.mqtt.client**: إنشاء عملاء للتواصل مع وسطاء MQTT.

```
from datetime import datetime
import time
import json
import pyfirmata
import paho.mqtt.client as mq
```

أنشئ المتغيرات الآتية التي ستُستخدم لعميل MQTT الذي ستنشئه. سيكون اسم العميل CLIENT_ID. أما MQTT_BROKER فهو عنوان الوسيط العام الذي توفره EMQX الذي ستتصل به. ويشير TOPIC إلى اسم الموضوع الذي سيشترك فيه العميل. يشير PORT إلى منفذ الخادم الافتراضي للاتصال بالوسيط. وختامًا فإن FLAG_CONNECTED سيستخدم كمتغير إشارة في دالة معالج الأحداث لاحقًا.

```
# Variables to setup MQTT client
CLIENT_ID = "PUBLISHER_01"      # ID of the client
MQTT_BROKER = "broker.emqx.io"  # Address of the broker
TOPIC = "waste/drops"           # Topic to subscribe to
PORT = 1883                     # Default server port
FLAG_CONNECTED = False          # Connection flag
```

جدول 3.3: متغيرات الاتصال بوسيط MQTT

المُتغير	الوصف
CLIENT___ID	اسم عميل MQTT.
MQTT___BROKER	عنوان وسيط MQTT المستهدف.
TOPIC	الموضوع الذي سيشترك فيه العميل.
PORT	منفذ الخادم المراد الاتصال به.
FLAG___CONNECTED	مُتغير إشارة للتحقق من اتصال الخادم.

أضف الأسطر الآتية، والتي مهمتها تهيئة الاتصال بالأردوينو باستخدام بروتوكول (Firmata) وكذلك تكوين الأطراف الخاصة بمستشعر الإضاءة ومُستشعر الإمالة المُستخدمين للحصول على البيانات.

```
board = pyfirmata.Arduino('COM4')    # Specify communication port
it = pyfirmata.util.Iterator(board)  # Select the board to connect
it.start()                           # Connect to board

# Selecting the sensor pins
light_sensor_pin = board.get_pin('a:0:i')
tilt_sensor_pin = board.get_pin('d:3:i')
```

أنشئ المتغيرات الآتية بالأسماء الآتية: can_full وهو وَسْمٌ يُحدد ما إذا كانت حاوية القمامة قد مُلئت أم لا، و garbage_drops وهو عدَّاد لتتبع عدد مرات الاستخدام لتعبئة الحاوية بالكامل.

```
can_full = False    # Flag to indicate whether the can is full
garbage_drops = 0   # Counter for the garbage drops
```

قم بإنشاء الدالة الآتية التي تعيد تعيين مُتغيري can_full و garbage_drops في كل مرة تكون فيها الحاوية ممتلئة، وترسل رسالة إلى العميل حول هذا الموضوع.

```
def reset_can():
    global garbage_drops  # Access the garbage_drops variable
    global can_full       # Access the can_full variable
    garbage_drops = 0     # Reset the counter to 0
    can_full = False      # Clear the can
```

قم بإنشاء الدالة الآتية لإرسال رسالة للعميل تفيد بأنه تم استخدام الحاوية. ستقوم أولًا بإنشاء متغير باسم timestamp لتسجيل الوقت، وإنشاء كائن قاموس Dictionary بالخصائص timestamp و garbage_drops و can_full. ستقوم بتحويل هذا القاموس إلى كائن JSON، ثم نشره إلى موضوع المُشترك "waste/drops" من خلال العميل.

```
def publish_message():
    global garbage_drops  # Access garbage_drops variable
    global can_full       # Access can_full variable
    # Create a custom format for the timestamp
    timestamp = str(datetime.now().strftime("%H:%M:%S"))
    msg_dictionary = {    # Creating the JSON object
        "timestamp": timestamp,
        "garbage_drops": garbage_drops,
        "can_full": can_full
    }
    msg = json.dumps(msg_dictionary)  # Convert dictionary to JSON
    try:
        result = client.publish(TOPIC, msg)  # Publish message
    except:
        print("There was an error while publishing the message")
    time.sleep(2)
```

إنشاء تنسيق يتناسب مع بيانات الوقت.

إنشاء كائن قاموس JSON لإرسال البيانات.

قم بإنشاء دالة معالج الأحداث الآتية، والتي ستطبع رسالة تأكيد إلى الواجهة الطرفية Terminal حول نجاح الاتصال بالعميل أو فشله. وسيطات الدالة هي وسيطات افتراضية تُستخدم لربط هذه الدالة بمعالج الأحداث المناسب الذي توفره مكتبة paho.mqtt.client.

```
def on_connect(client, userdata, flags, rc):
    global FLAG_CONNECTED  # Access the FLAG_CONNECTED variable

    if rc == 0:            # If rc is 0 the client connected successfully
        FLAG_CONNECTED = True
        print("Connected to MQTT Broker!")
    else:
        print("Failed to connect to MQTT Broker!")
```

هذا المتغير يُرسل من خلال مكتبة paho ليظهر حالة الاتصال.

ستقوم في الجزء الرئيس من البرنامج بتهيئة عميل MQTT، وربط معالج الأحداث on_connect بالدالة المذكورة أعلاه، ثم الاتصال بوسيط MQTT المُحدد، والاشتراك في الموضوع المحدد.

```
client = mq.Client(CLIENT_ID)     # Initialize an MQTT client
client.on_connect = on_connect    # Bind the on_connect event handler
client.connect(MQTT_BROKER, PORT) # Connect to the specified MQTT broker
client.subscribe(TOPIC, 0)        # Subscribe to the specified topic
```

قم بإنشاء التكرار الرئيس للبرنامج. إذا كانت قيمة light_value أقل من 0.200، فستُعدّ الحاوية ممتلئة.

```
while True:
    # Get sensor values
    light_value = light_sensor_pin.read()
    tilt_value = tilt_sensor_pin.read()

    if (light_value is not None) and (tilt_value is not None):
        print("Light levels : " + str(light_value))
        print("Tilt levels : " + str(tilt_value))
        print("Garbage drops : " + str(garbage_drops))

        # If there is a tilt, add 1 to the counter
        if (tilt_value == True):
            garbage_drops += 1
            # If there is a tilt and the can is full
            # publish a message and reset the can
            if (light_value <= 0.200):
                can_full = True
                publish_message()
                reset_can()
            publish_message()
        time.sleep(1)
```

سيُبدأ في نشر البيانات عندما تمتلئ حاوية القمامة التي يُكشف عنها في ظروف الإضاءة المنخفضة.

البرنامج بشكله النهائي Complete Code

```
from datetime import datetime
import time
import json
import pyfirmata
import paho.mqtt.client as mq

# Variables to setup MQTT client
CLIENT_ID = "PUBLISHER_01"      # ID of the client
MQTT_BROKER = "broker.emqx.io"  # Address of the broker
TOPIC = "waste/drops"           # Topic to subscribe to
PORT = 1883                     # Default server port
FLAG_CONNECTED = False          # Connection flag

board = pyfirmata.Arduino('COM4')   # Specify communication port
it = pyfirmata.util.Iterator(board) # Select the board to connect
it.start()                          # Connect to board

# Selecting the sensor pins
light_sensor_pin = board.get_pin('a:0:i')
tilt_sensor_pin = board.get_pin('d:3:i')

can_full = False    # Flag to indicate whether the can is full
garbage_drops = 0   # Counter for the garbage drops

def reset_can():
    global garbage_drops  # Access garbage_drops variable
    global can_full       # Access can_full variable
    garbage_drops = 0     # Reset the counter to 0
    can_full = False      # Clear the can
```

```
def publish_message():
    global garbage_drops   # Access garbage_drops variable
    global can_full        # Access can_full variable

    # Create a custom format for the timestamp
    timestamp = str(datetime.now().strftime("%Y-%m-%d %H:%M:%S"))

    # Creating the dictionary object
    msg_dictionary = {
        "timestamp": timestamp,
        "garbage_drops": garbage_drops,
        "can_full": can_full
    }
    msg = json.dumps(msg_dictionary)  # Convert dictionary to JSON

    try:
        result = client.publish(TOPIC, msg)  # Publish message
    except:
        print("There was an error while publishing the message")

    time.sleep(2)
    print("Message sent to the MQTT broker")

def on_connect(client, userdata, flags, rc):
    global FLAG_CONNECTED  # Access the FLAG_CONNECTED variable

    if rc == 0:            # If rc is 0 the client connected successfully
        FLAG_CONNECTED = True
        print("Connected to MQTT Broker!")
    else:
        print("Failed to connect to MQTT Broker!")
```

```
client = mq.Client(CLIENT_ID)     # Initialize an MQTT client
client.on_connect = on_connect    # Bind the on_connect event handler
client.connect(MQTT_BROKER, PORT) # Connect to the specified MQTT broker
client.subscribe(TOPIC, 0)        # Subscribe to the specified topic

while True:
    # Get sensor values
    light_value = light_sensor_pin.read()
    tilt_value = tilt_sensor_pin.read()

    if (light_value is not None) and (tilt_value is not None):
        print("Light levels : " + str(light_value))
        print("Tilt levels : " + str(tilt_value))
        print("Garbage drops : " + str(garbage_drops))

        # If there is a tilt, add 1 to the counter
        if (tilt_value == True):
            garbage_drops += 1

            # If there is a tilt and the can is filled
            # publish a message and reset the can
            if (light_value <= 0.200):
                can_full = True
                publish_message()
                reset_can()

            publish_message()

        time.sleep(1)
```

اختبار الوسيط Testing the Broker

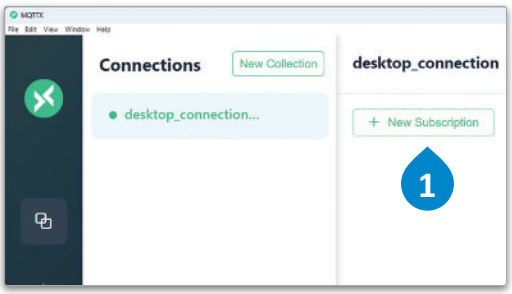
تُعدُّ EMQX وسيط MQTT عام لاختبار وتطوير تطبيقات MQTT . تساعد EMQX في تطوير النماذج الأولية لتطبيقات إنترنت الأشياء دون الحاجة إلى وضع البنى التحتية وتطوير الوسيط. ستستخدم عميل MQTTX client للتحقق من نشر الرسائل، ثم ستُنشئ مقطعًا برمجيًا آخر في بايثون يستقبل الرسائل المنشورة، ويُنشئ تقاريرَ عن الحاوية، ويحُلل البيانات الموجودة في تلك التقارير. بعد تحميل مخطط StandardFirmata إلى الأردوينو، نفِّذ تعليمات بايثون وحرِّك لوحة التجارب لتفعيل مُستشعر الإمالة. في كل مرة يُفعّل بها المُستشعر، ستُزاد عداد النفايات والذي يشير إلى عدد المرات التي تم بها فتح الحاوية افتراضيًا لوضع القمامة. وعند تفعيل مُستشعر الإمالة مع تغطية مُستشعر الإضاءة، سينشر البرنامج رسالة مفادها أن الحاوية مُمتلئة، ويعيد ضبط عداد النفايات. ستقوم في الدرس التالي بتحليل البيانات بناءً على الرسائل المنشورة. لاختبار نشر رسائلك بصورة صحيحة، ستستخدم العميل المكتبي MQTTX Agent قبل تنفيذ مقاطع بايثون البرمجية، ستستخدم عميل MQTTX للاشتراك في موضوع "waste/drops". سينتظر العميل الآن استلام الرسائل التي تُنشر من خلال برنامج بايثون وتُوزّع من خلال وسيط EMQX العام.

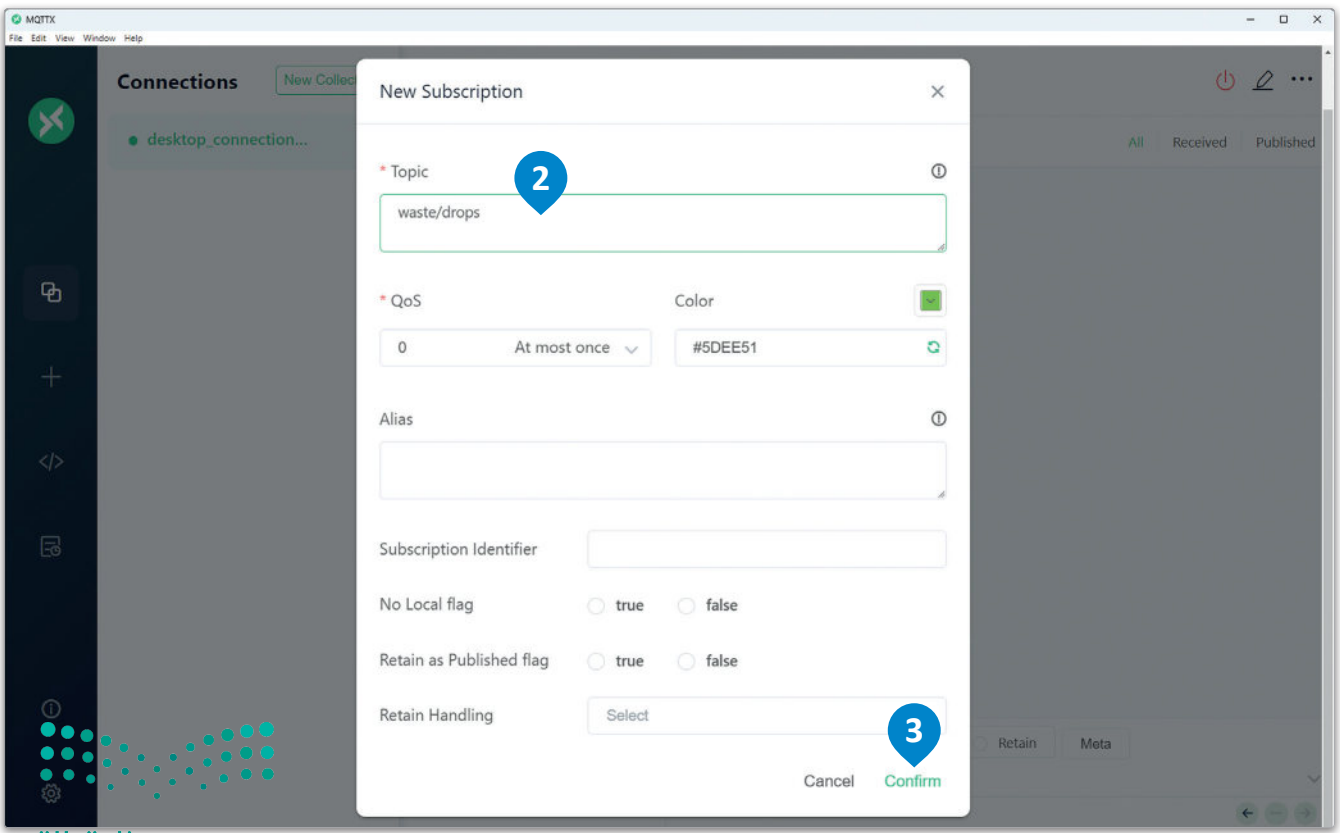
لاستخدام MQTTX للاشتراك في موضوع المحدد:

> في قائمة Connections (اتصالات) في تبويبة desktop_connection (سطح المكتب)، اضغط على زر New Subscription (اشتراك جديد). 1

> في مربع نص Topic (الموضوع)، اكتب "waste/drops". 2

> اضغط على زر Confirm (تأكيد). 3



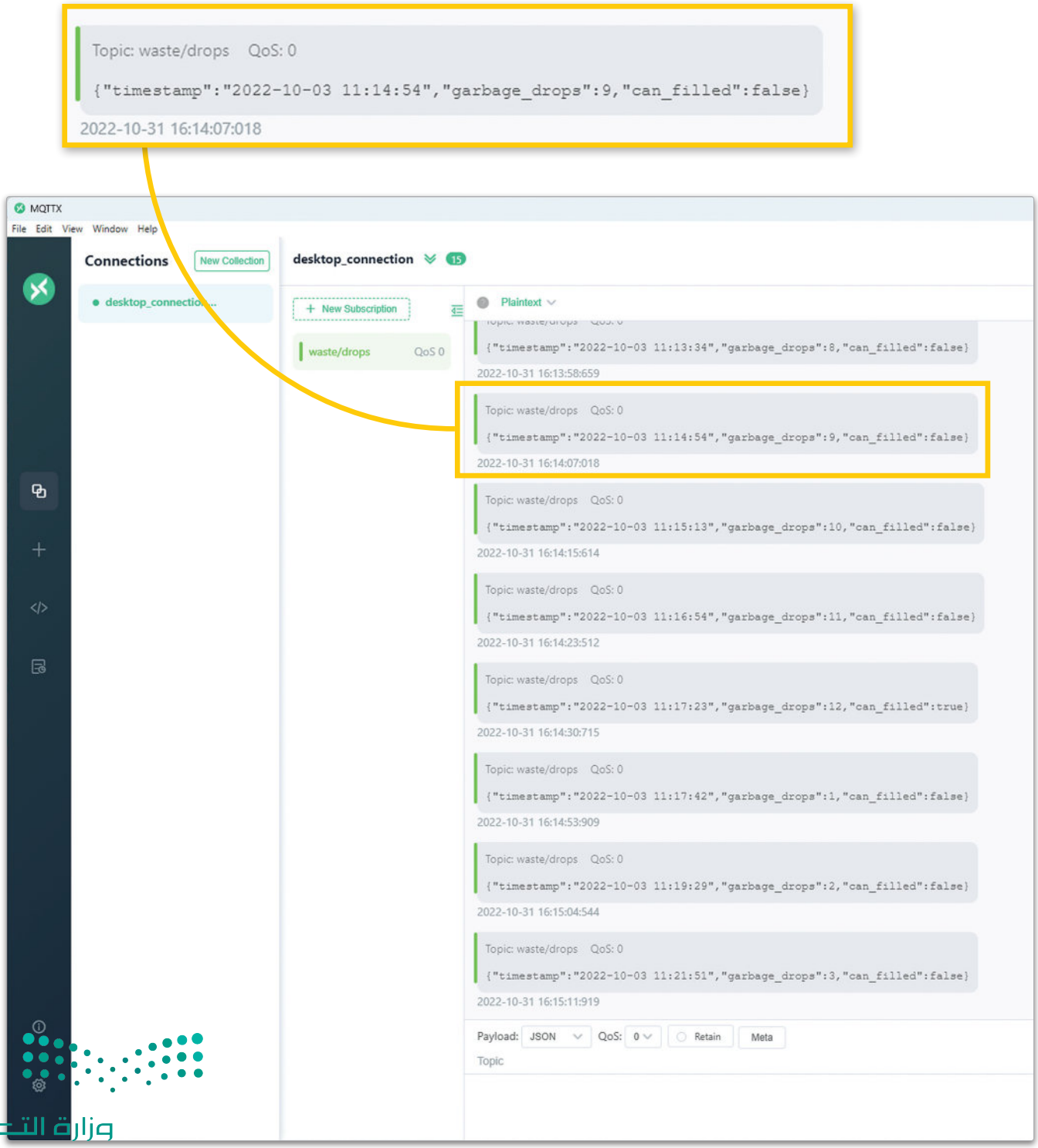


شكل 7.19: استخدام MQTTX للاشتراك في موضوع محدد

عرض الرسائل من خلال عميل MQTTX

Viewing Messages through the MQTTX Client

بعد تنفيذ المقاطع البرمجية في بايثون، وبدء نشر الرسائل، سيتم استقبال الرسائل من خلال العميل المكتبي MQTTX كما يظهر أدناه.



شكل 7.20: عرض الرسائل من خلال عميل MQTTX

تمرينات

1 أنشئ مُخططًا لشبكة MQTT مع لوحة أردوينو واحدة تعمل كناشر، واثنتان تعملان كمُستقبلات.

2 قدِّمْ وصفًا للترانزستور الضوئي ومكونات مستشعر الإمالة وحالات استخدامهما.

3 قُمْ بتحليل ماهية وسيط EMQX العام ومدى مساهمته في تطوير تطبيقات النماذج الأولية لإنترنت الأشياء.

4 قُمْ بتحديث معالج الحدث لحدث on_connect الذي يطبع معلومات التكوين إلى وحدة التحكم المُقدمة إلى عميل MQTT.

5 قُمْ بتحديث الدالة publish_message() لنشر رسالة زمنية timestamp تعرض التاريخ والوقت بأكملهما وخاصية جديدة تحتوي على مُعرِّف العميل.

6 أنشئ مقطعًا برمجيًا بلغة بايثون يتيح للمستخدم كتابة الموضوع الذي يريد الاشتراك به، والرسالة التي يريد إرسالها ثم نشرها من خلال وسيط EMQX العام.

اختبر برنامجك باستخدام العميل المكتبي MQTT X.

رابط الدرس الرقمي

www.ien.edu.sa

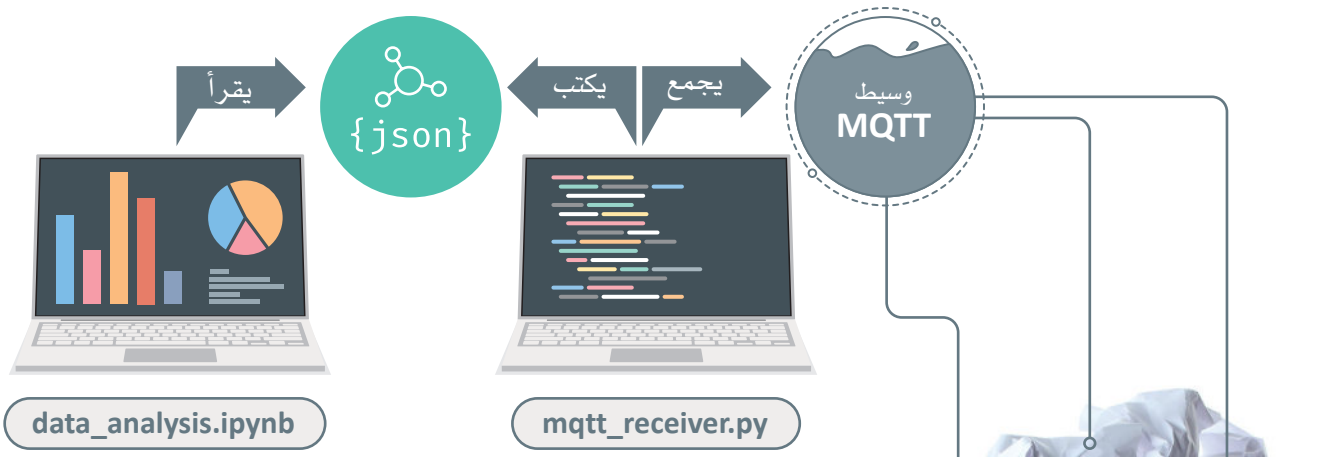
الدرس الثالث

إنشاء حل ذكي لإدارة النفايات

إدارة النفايات الذكية وتحليل البيانات

Smart Waste Management and Data Analysis

لقد قُمت في الدرس السابق بإنشاء نموذج أولي ذكي لإدارة حاوية نفايات باستخدام متحكم أردوينو دقيق يراقب بيئته المحيطة، ثم يُنتج البيانات من المُستشعرات، وينشرها كرسائل إلى موضوع في MQTT. يجب أن يتم جمع البيانات ثم معالجتها لتكوين تصورات مختلفة ووضع الخطط بناءً على هذه البيانات.



شكل 7.21: تحليل بيانات مشروع النفايات الذكية

ستقوم في هذا الدرس بإنشاء مقطع برمجي في بايثون يشترك في نفس الموضوع الخاص بوسيط MQTT الذي نُشرت عليه الرسائل. سيتم ستُجمع هذه الرسائل، وفي كل مرة تمتلئ فيها سلة النفايات، سيُنشأ تقرير ويُخزّن في ملف آخر. سيحتوي هذا الملف فقط على البيانات من التقارير التي أُنشئت، ثم ستُنشئ بعد ذلك مفكرة جوبيتر تقوم بفتح الملف وبتحليل البيانات وإنشاء رؤى وتصورات حولها.

سيُسمّى هذان الملفان باسم mqtt_receiver.py وdata_analysis.ipynb على التوالي، سيكتب البرنامج الأول البيانات المحفوظة في ملف JSON، وسيقرأ البرنامج الثاني البيانات من ملف JSON ويقوم بتحليلها.

إنشاء ملف بيانات جسون Creating the JSON Data File

ستقوم بإنشاء ملف بيانات JSON بواسطة مصفوفة فارغة، وسيقوم البرنامج mqtt_receiver.py بإلحاق كل تقرير أُنشئ ككائن JSON بالمصفوفة، ثم سيفتح ملف data_analysis.py ملف JSON هذا، وسيقرأ محتويات مصفوفة JSON السابقة ويقوم بعمليات تحليل البيانات.

افتح PyCharm ثم أنشئ ملفًا جديدًا في مجلدك الخاص باسم data.json. ثم أنشئ كائنًا لمصفوفة فارغة داخل هذا الملف كما هو موضح أدناه. سيُلحق mqtt_receiver.py كائنات JSON الخاصة بالتقارير المُنشأة بالمصفوفة المعروضة أدناه. احفظ الملف data.json ثم أغلقه.

```
[]
```

قم بإنشاء ملف بايثون جديد باسم mqtt_receiver.py، وفي بداية المقطع البرمجي، قم باستيراد الحزم الآتية:

- **datetime**: تُنشئ طابعًا زمنيًا للرسائل المُرسلة.
- **json**: للتعامل مع كائنات JSON.
- **paho.mqtt.client**: لإنشاء عملاء للتواصل مع وسطاء MQTT.
- **os**: للتعامل مع الملفات الموجودة على حاسبك الخاص.

```
from datetime import datetime
import json
import paho.mqtt.client as mq
from os import path
```

قم بإنشاء المتغيرات الآتية data_file و data_file_objects والتي ستتفاعل مع ملف بيانات JSON.

```
data_file = "your_file_path"  # Absolute path to the JSON data file
data_file_objects = []  # This contains the objects from the JSON data file
```

تأكد من إدخال امتداد ملف البيانات الصحيح.

قم بإنشاء المتغيرات الآتية التي ستُستخدم لعميل MQTT الذي ستُنشئه باسم CLIENT_ID. بينما يشير MQTT_BROKER إلى عنوان الوسيط العام الذي توفره EMQX الذي سيتم الاتصال به، و TOPIC إلى اسم الموضوع الذي سيشترك فيه العميل. و PORT إلى مَنفذ الخادم الافتراضي للاتصال بالوسيط، وFLAG_CONNECTED الذي سيستخدم كمتغير إشارة في دالة معالج الأحداث لاحقًا.

```
# Variables to setup MQTT client
CLIENT_ID = "RECEIVER_01"       # ID of the client
MQTT_BROKER = "broker.emqx.io" # Address of the broker
TOPIC = "waste/drops"           # Topic to subscribe to
PORT = 1883                     # Default server port
FLAG_CONNECTED = False          # Connection flag
```

قم بإنشاء المتغيرات الآتية messages_stack و reports والتي ستُستخدم لتخزين المعلومات من الرسائل المنشورة.

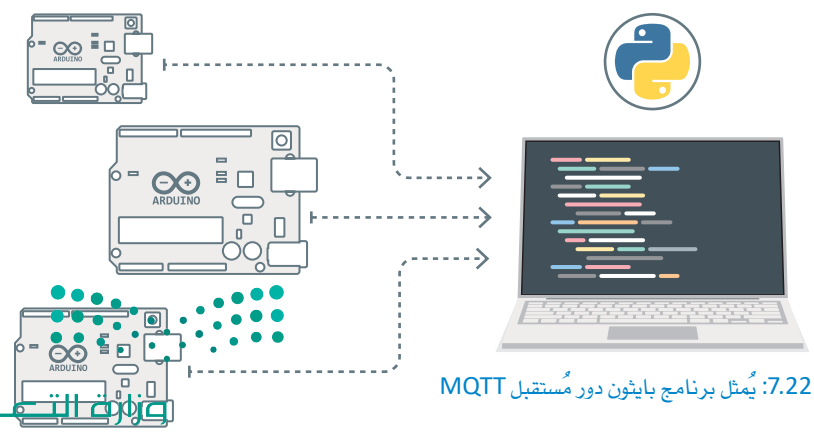
```
messages_stack = []  # The array with the messages per can filling
reports = []  # The array with all the generated report objects
```

قم بإنشاء دالة معالج الأحداث الآتية والتي تطبع رسالة تأكيد إلى الواجهة الطرفية Terminal حول نجاح الاتصال بالعميل من عدمه. وسيطات الدالة هي وسيطات افتراضية يجب استخدامها لربط هذه الدالة بمعالج الأحداث المناسب الذي توفره المكتبة paho.mqtt.client.

```
def on_connect(client, userdata, flags, rc):
    global FLAG_CONNECTED  # Access the FLAG_CONNECTED variable

    if rc == 0:            # If rc is 0 the client connected successfully
        FLAG_CONNECTED = True
        print("Connected to MQTT Broker!")
    else:
        print("Failed to connect to MQTT Broker!")
```

```
def on_message(client, userdata, msg):
    global messages_stack  # Access the messages_stack variable

    # Decode the message payload
    payload = str(msg.payload.decode())

    # Convert the payload to a JSON object and append it
    # to the messages stack
    payload_object = json.loads(payload)
    messages_stack.append(payload_object)

    # When you receive a message, print it to the terminal
    print("||---- MESSAGE RECEIVED ----||\n")
    print("Payload: " + str(payload_object))

    # If the payload object has the can_filled flag set to True
    # generate a report for the filled can
    if payload_object["can_filled"] == True:
        generate_report()
```

إنشاء بيانات التقرير فقط عند امتلاء حاوية القمامة.



شكل 7.22: يُمثل برنامج بايثون دور مُستقبل MQTT

يمكن لبرنامج بايثون هذا أن يعمل كمُستقبل MQTT لجمع رسائل من عدة ناشري أردوينو في نفس الوقت. يمكن توسعة هذا البرنامج ليتمكن من معالجة بيانات JSON مع إضافة المزيد من الحقول التي تحتوي على معلومات حول الناشرين أيضًا.

قم بإنشاء دالة generate_report الآتية والتي ستنشئ كائن JSON للتقرير وتلحقه بملف البيانات في كل مرة تُستلم فيها رسالة تشير إلى امتلاء حاوية النفايات.

```
def generate_report():
    global data_file_objects  # Access data_file_objects variable
    global messages_stack     # Access messages_stack variable
    global reports            # Access reports variable

    # Getting the first and last objects from the messages_stack
    first_msg = messages_stack[0]
    last_msg = messages_stack[len(messages_stack) - 1]

    # Converting the string attributes of the messages objects to datetime
    time_filled_timestamp = last_msg["timestamp"]
    first_timestamp = datetime.strptime(first_msg["timestamp"],"%H:%M:%S")
    last_timestamp = datetime.strptime(last_msg["timestamp"], "%H:%M:%S")

    garbage_drops = last_msg["garbage_drops"]

    # Calculating the time_to_fill by comparing the timestamps
    # of the first and last fillings
    time_delta = last_timestamp - first_timestamp
    time_to_fill = time_delta.total_seconds() / 60
    report_id = len(reports)  # This will be used for object indexing
    # The JSON object that will be appended to the JSON data file
    report = {
        "id": report_id,
        "timestamp": time_filled_timestamp,
        "garbage_drops": garbage_drops,
        "time_to_fill": time_to_fill
    }
```

ستُستخدم الرسالتان الأولى والأخيرة لحساب فرق الوقت.

ستُستخدم عملية الطابع الزمني لحساب فرق الوقت.

```
    # Append the new report to the objects of the data file
    # and write the data_file_objects array to the data file
    data_file_objects.append(report)
    with open(data_file, 'w') as file:
        json.dump(data_file_objects, file, indent=4, separators=(',',': '))

    # Append the report object to the reports array and to the JSON data file
    # and clear the messages stack
    reports.append(report)
    messages_stack = []
```

تحديث محتويات مصفوفة البيانات وملف البيانات.

ستتحقق في الجزء الرئيس من البرنامج مما إذا كانت البيانات موجودة وتقوم بفتحها، ثم ستقوم بتهيئة عميل MQTT، وتربط معالجات الأحداث on_connect و on_message بالدوال المذكورة أعلاه، ثم تقوم بالاتصال بوسيط MQTT، والاشتراك في الموضوع المحدد والاستماع إلى الرسائل الواردة.

```
#  Check if the data file exists
if path.isfile(data_file) is False:
    raise Exception("Data file not found")
# Read the contents of the JSON data file
with open(data_file) as fp:
    data_file_objects = json.load(fp)

client = mq.Client(CLIENT_ID)      # Initialize an MQTT client
client.on_connect = on_connect     # Bind the on_connect event handler
client.on_message = on_message     # Bind the on_message event handler
client.connect(MQTT_BROKER, PORT)  # Connect on the specified MQTT broker
client.subscribe(TOPIC, 0)         # Subscribe to the specified topic
client.loop_forever()              # Listen continuously for messages
```

التأكد من وجود ملف البيانات لتجنب الأخطاء.

البرنامج بشكله النهائي Complete Code

```
from datetime import datetime
import json
import paho.mqtt.client as mq
from os import path

data_file = "your_file_path"  # Absolute path to the JSON data file
data_file_objects = []  # This contains the objects from the JSON data file

# Variables to setup MQTT client
CLIENT_ID = "RECEIVER_01"      # ID of the client
MQTT_BROKER = "broker.emqx.io" # Address of the broker
TOPIC = "waste/drops"          # Topic to subscribe to
PORT = 1883                    # Default server port
FLAG_CONNECTED = False         # Connection flag

messages_stack = []  # The array with the messages per can filling
reports = []  # The array with all the generated report objects

def on_connect(client, userdata, flags, rc):
    global FLAG_CONNECTED  # Access the FLAG_CONNECTED variable

    if rc == 0:            # If rc is 0 the client connected successfully
        FLAG_CONNECTED = True
        print("Connected to MQTT Broker!")
    else:
        print("Failed to connect to MQTT Broker!")

def on_message(client, userdata, msg):
    global messages_stack  # Access the messages_stack variable
```

```
    # Decode the message payload
    payload = str(msg.payload.decode())

    # Convert the payload to a JSON object and append it
    # to the messages stack
    payload_object = json.loads(payload)
    messages_stack.append(payload_object)

    # When you receive a message, print it to the terminal
    print("||---- MESSAGE RECEIVED ----||\n")
    print("Payload: " + str(payload_object))

    # If the payload object has the can_filled flag set to True
    # generate a report for the filled can
    if payload_object["can_filled"] == True:
        generate_report()
def generate_report():
    global data_file_objects  # Access data_file_objects variable
    global messages_stack     # Access messages_stack variable
    global reports            # Access reports variable

    # Getting the first and last objects from the messages_stack
    first_msg = messages_stack[0]
    last_msg = messages_stack[len(messages_stack) - 1]

    # Converting the string datetimes to datetime objects
    first_timestamp = datetime.strptime(first_msg["timestamp"], "%H:%M:%S")
    last_timestamp = datetime.strptime(last_msg["timestamp"], "%H:%M:%S")
    garbage_drops = last_msg["garbage_drops"]
    # Calculating the time_to_fill by comparing the timestamps
    # of the first and last fillings
    time_delta = last_timestamp - first_timestamp
```

```
    time_to_fill = time_delta.total_seconds() / 60
    report_id = len(reports)  # This will be used for object indexing

    # The JSON object that will be appended to the JSON data file
    report = {
        "id": report_id,
        "garbage_drops": garbage_drops,
        "time_to_fill": time_to_fill
    }
    # Append the new report to the objects of the data file
    # and write the data_file_objects array to the data file
    data_file_objects.append(report)
    with open(data_file, 'w') as file:
        json.dump(data_file_objects, file, indent=4, separators=(',',': '))

    # Append the report object to the reports array and to the JSON data file
    # and clear the messages stack
    reports.append(report)
    messages_stack = []
#  Check if the data file exists
if path.isfile(data_file) is False:
    raise Exception("Data file not found")
# Read the contents of the JSON data file
with open(data_file) as fp:
    data_file_objects = json.load(fp)

client = mq.Client(CLIENT_ID)      # Initialize an MQTT client
client.on_connect = on_connect     # Bind the on_connect event handler
client.on_message = on_message     # Bind the on_message event handler
client.connect(MQTT_BROKER, PORT)  # Connect on the specified MQTT broker
client.subscribe(TOPIC, 0)         # Subscribe to the specified topic
client.loop_forever()              # Listen continuously for messages
```

تحليل البيانات في جوبيتر Data Analysis in Jupyter Notebook

ستستخدم الآن مُفكرة جوبيتر لإجراء عمليات تحليل البيانات على ملف بيانات JSON. ونظرًا لأن جمع البيانات اللازمة وتحليلها يستغرق وقتًا طويلًا ، فقد تم توفير مجموعة بيانات JSON جاهزة لكي تستخدمها. تحاكي مجموعة البيانات هذه ترك نموذج الأردوينو الأولي قيد التشغيل لفترة طويلة من الوقت.

ملف JSON متاح للتنزيل من هنا:

http://binary-academy.com/dnld/KSA/IOT2/U3_L3_DATA.json

ستقوم أولًا باستيراد المكتبات المطلوبة وقراءة بيانات JSON من الملف.

```
import os
import pandas as pd                    # library used for data manipulation
import matplotlib.pyplot as plt        # library used for plotting data

# The data that will be used, extracted from the JSON dataset
data = pd.read_json('U3_L3_DATA.json','records',convert_dates=['timestamp'])
```

ستقوم بعد ذلك بوصف مجموعة البيانات لاستخراج الخصائص الإحصائية.

```
data.describe().round(0)
```

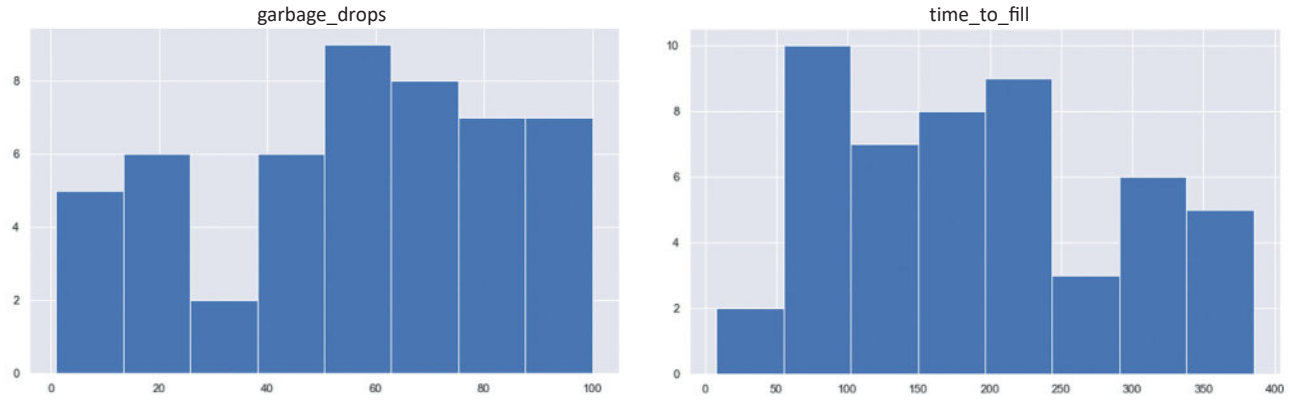
هذا هو ملف البيانات الذي تم تنزيله والذي يجب وضعه في نفس مجلد المشروع.

	id	garbage_drops	time_to_fill
count	50.0	50.0	50.0
mean	24.0	54.0	152.0
std	15.0	30.0	100.0
min	0.0	2.0	5.0
25%	12.0	30.0	60.0
50%	24.0	55.0	147.0
75%	37.0	78.0	235.0
max	49.0	100.0	376.0

شكل 7.23: وصف البيانات

سنقوم بإنشاء مُخططين بيانيين مُجمّعين حسب خصائص garbage_drops و time_to_fill.

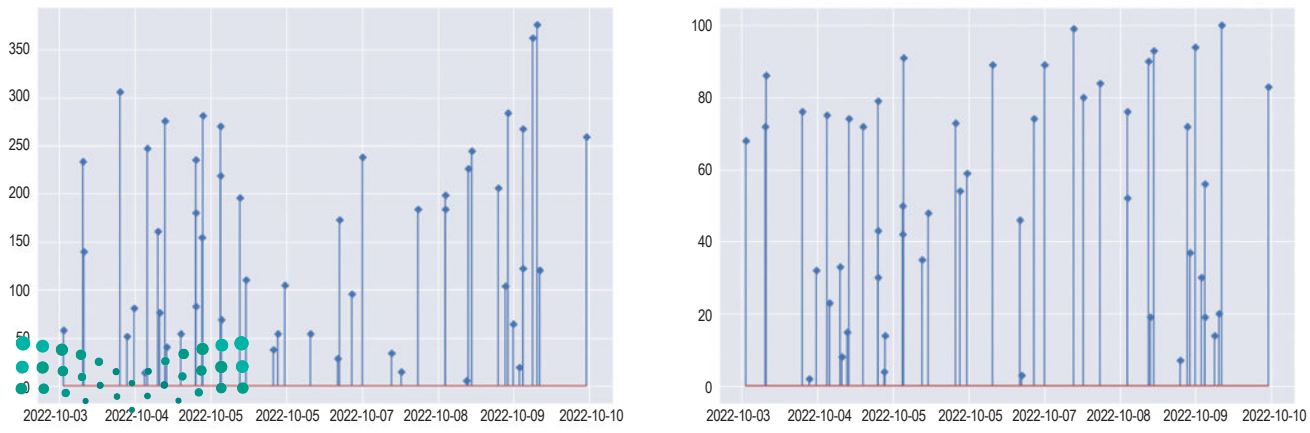
```
# Create histograms for the data using 8 groupings
hist = data.hist(['garbage_drops'],figsize=(10,6),bins=8)
hist = data.hist(['time_to_fill'],figsize=(10,6),bins=8)
```



شكل 7.24: المُخططات البيانية

ستقوم بعد ذلك بإنشاء مُخططين من نوع stem plots لعرض garbage_drops و time_to_fill في كل فترة زمنية.

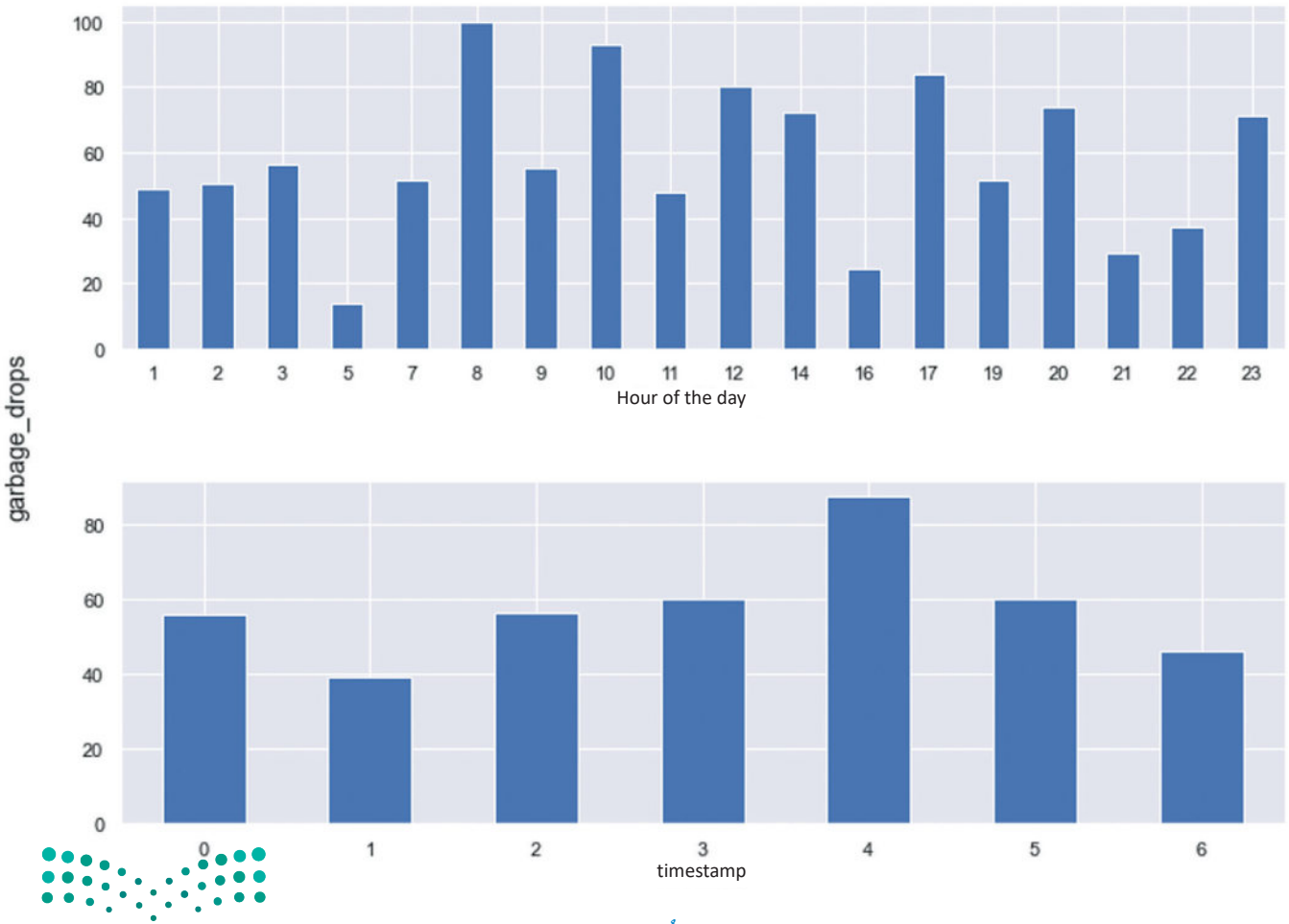
```
# Create stem plots for the data with diamond-shaped ('D') markers
plt.stem(data['timestamp'], data['time_to_fill'], markerfmt='D');
plt.stem(data['timestamp'], data['garbage_drops'], markerfmt='D');
```



شكل 7.25: مخططات stem plots البيانية

ستنشئ في الختام مُخططين يجمعان عدد garbage_drops لكل ساعة من اليوم، ولكل ساعة خلال الأسبوع.

```
# Create 2 plots, one that groups mean garbage amount by hour and one by day
fig, (ax1,ax2) = plt.subplots(2,figsize=(12, 8))
fig.supylabel('garbage_drops')
data.groupby(data["timestamp"].dt.hour)["garbage_drops"].mean().plot(kind='bar',
rot=0, xlabel='Hour of the day', ax=ax1);
# Monday = 0, Sunday = 6
data.groupby(data["timestamp"].dt.day_of_week)["garbage_drops"].mean().
plot(kind='bar', rot=0, ax=ax2);
```



شكل 7.26: مُخططات تجميعية

تمرينات

1 قم بإنشاء مُخطط يوضِّح العلاقة بين ملفي البايثون وملف JSON المحتفظ بالبيانات.

2 قُمْ بإنشاء ملف بلغة بايثون يتصل بثلاثة موضوعات ويكتب معالج أحداث **on_connect** لطباعة معلومات التكوين والموضوعات التي اشترك فيها العميل إلى الواجهة الطرفية Terminal.

3 قُمْ بتحديث الكائن on_message لطباعة المعلومات إلى الواجهة الطرفية حول العميل الذي نشر البيانات، وكذلك الموضوع الذي استُلمت البيانات منه.

4 قُمْ بإنشاء ملف JSON جديد يحتوي على جميع القيم من قائمة الرسائل، واستخدم دالة create_report() لإلحاق قيم messages_stack بملف JSON الجديد.

5 في مُفكرة جوبيتر، قم بإنشاء مُخطط مبعثر (Scatter Diagram) جديد لنفس البيانات التي قمت بمعالجتها في الدرس.

6 أضف مقطعًا برمجيًا آخر بلغة بايثون يستقبل الرسائل التي نشرتها من البرنامج في التمرين السادس من الدرس الثاني. عندما تستقبل رسالة، اطبع المعلومات المتعلقة بالناشر والمُستقبل والموضوع المشترك على الواجهة الطرفية Terminal.

المشروع

تُستخدم الاتصالات مع بروتوكول (MQTT) على نطاق واسع في المشاريع المختلفة على أرض الواقع. يُمكن تطبيق نفس الهيكليات الخاصة بالاتصال من خلال الناشرين والوسطاء والمستلمين على مجالات أخرى مختلفة. ستقوم بإنشاء حل متكامل لحديقة ذكية متصلة تُراقب بواسطة بروتوكول (MQTT)، ويُمكن بعد ذلك تعميم مثل هذه الهيكليات على تطبيقات صناعية لحدائق ذكية أكبر حجمًا.

1. قم بإنشاء دائرة جديدة باستخدام لوحة أردوينو ومستشعر درجة الحرارة ومستشعر رطوبة التربة وترانزستور ضوئي.

2. قم بإنشاء برنامج بايثون آخر بمثابة مُستقبل للبيانات التي جُمعت بواسطة الأردوينو. سيُطلب من المستخدم اختيار الموضوع الذي سيتلقى حوله البيانات، ثم يقوم بإنشاء عميل للاشتراك في هذا الموضوع. ستُخزن الرسائل ويُعرض تنبيه إذا وجد ارتفاعًا في القيم المحدثة.

3. قم بإنشاء برنامج بايثون يشترك في موضوع بكافة القراءات ويحفظها في ملف JSON. سيُطلب من المستخدم اختيار ما إذا كان يريد الاستماع إلى الوسيط وجمع البيانات، أو إنشاء تمثيل للقراءات المُخزنة بالفعل.

4. قم بتشغيل مقاطع بايثون البرمجية الثلاثة في آنٍ واحد، واضبط بيئة الأردوينو لتحديث قراءات البيانات ومراقبة النتائج.

ماذا تعلّمت

- تحليل الطبقات الهيكلية للمدن الذكية.
- نشر الرسائل باستخدام بروتوكول MQTT.
- إنشاء برنامج بايثون لنشر الرسائل إلى عميل MQTTX Client.
- تخزين التقارير في ملف بيانات JSON.
- إجراء عمليات تحليل البيانات على ملف بيانات JSON باستخدام مُفكرة جوبيتر.

المصطلحات الرئيسة

Publisher	ناشر
Quality of Service	جودة الخدمة
Receiver	مُستقبل
Services Layer	طبقة الخدمات
Street Layer	طبقة الشارع
Subscriber	مُشترك
Tilt Sensor	مُستشعر الإمالة
Tracker	مُتعقب

City Layer	طبقة المدينة
Client	عميل
Data Center Layer	طبقة مركز البيانات
Message Broker	وسيط الرسالة
MQTT Server	خادم MQTT
Phototransistor	الترانزستور الضوئي
Prototype	نموذج أولي

8. محاكاة شبكة مُستشعرات إنترنت الأشياء اللاسلكية

سيتعرف الطالب في هذه الوحدة على تقنيات إنترنت الأشياء المستخدمة في الصناعات الذكية. ستستخدم بيئة كاب كاربون (CupCarbon) لإنشاء شبكات من المُستشعرات ومحاكاتها. وفي الختام ستُنشئ نموذجًا أوليًا لنظامٍ للإنذار ومراقبة الحرائق، ونموذجًا آخر أوليًا خاصًا بالصناعة الذكية والأتمتة.

أهداف التعلُّم

بنهاية هذه الوحدة سيكون الطالب قادرًا على أن:

- يتعرف على تقنيات إنترنت الأشياء المُستخدمة في الأغراض الصناعية.
- يتعرف على استخدام الأتمتة الصناعية وأنظمة التحكم في المصانع المُتصلة بإنترنت الأشياء.
- ينشئ ويمثّل شبكات إنترنت الأشياء باستخدام كاب كاربون (CupCarbon).
- يتعرف على محاكاة شبكة إنترنت الأشياء باستخدام كاب كاربون (CupCarbon).
- ينشئ مقاطع برمجية بلغة بايثون لبرمجة عُقد الشبكة.
- ينشئ نموذجًا أوّليًا لمراقبة الحرائق والإنذار بإنترنت الأشياء.
- ينشئ نموذجًا أوّليًا للصناعة الذكية والأتمتة بإنترنت الأشياء.

الأدوات

- كاب كاربون (CupCarbon)

وزارة التعليم
Ministry of Education
2023 - 1445

رابط الدرس الرقمي

www.ien.edu.sa

الدرس الأول

مقدمة إلى كاب كاربون

الصناعة الذكية Smart Industry

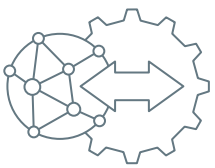
أصبح لتقنيات إنترنت الأشياء تأثير كبير على جميع مجالات الحياة، بما فيها المجالات الصناعية، وذلك من أجل تقليل التكلفة وتحسين الكفاءة. أدى تطور النماذج الصناعية وازدياد المنافسة إلى تحول التركيز إلى الابتكار وتحسين نماذج الأعمال. قامت الشركات على مدى عقود طويلة بمحاولات لخفض التكلفة الإجمالية لمنتجاتها من خلال خفض تكلفة عمليات التصنيع وسلاسل التوريد، ولكنها أدركت أن المحاولات المستمرة لخفض التكلفة يؤثر بشكلٍ سلبي على خدمة العملاء وجودة الإنتاج. أدت بعض تقنيات إنترنت الأشياء إلى إحداث تغييرات ذات أثر كبير في عمليات التصنيع ومن ذلك:

التصنيع القائم على البيانات Data-Driven Manufacturing

تعمل البيانات الضخمة على تغيير عالم الصناعة، فقد أصبح بإمكان المُصنِّعين الوصول إلى جميع البيانات التي تُنشأ وتُجمع بواسطة الآلات وذلك بهدف المراقبة الفورية للجودة، وتحسين **كفاءة الآلات الكُلية** (Overall Equipment Effectiveness – OEE)، وتقليل وقت الإنتاج. يُعدّ **مؤشر كفاءة الآلات الكلية** (OEE) بمثابة المعيار العالمي لحساب الكفاءة الفعلية لعمليات الإنتاج الصناعية. يبحث المصنِّعون عن طرق لاستخدام البيانات الضخمة للاستجابةِ لتحولات السوق والتغيرات في حاجات المستهلكين، وذلك من خلال إدخال تقنيات وأدوات صناعية جديدة.

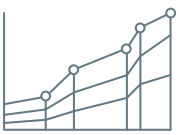
تقارب تقنية التشغيل (OT) وتقنية المعلومات (IT)

Operational Technology (OT) and Information Technology (IT) Convergence



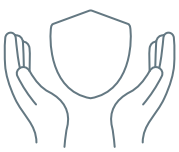
في سياق **إنترنت الأشياء** (Internet of Things – IoT) تضم التقنية التشغيلية في بيئة التصنيع وحدات **تحكم منطقية قابلة للبرمجة** (Programmable Logic Controllers – PLCs)، وأجهزة الحاسب، وغيرها من التقنيات التي تشبه لحد ما تقنية المعلومات، ولكنها تخضع للأعمال التجارية خارج نطاق إدارات تقنية المعلومات. تتيح الشبكات المبنية على بروتوكول IP تكاملًا أعمق بين الآلات وعمليات التصنيع، وتزيل الفجوة بين شبكات الصناعة والأعمال التجارية. يبحث المُصنِّعون عن طرق لدمج عملياتهم في إطار بُنية تحتية موحدة للشبكات تتجاوز طرق التخزين التقليدية.

تقنية أفضل وتكلفة أقل Improved Technology with Lower Costs



أصبحت إمكانية الاتصال والمراقبة وتحسين الأجهزة قابلة للتطوير والأتمتة، وقائمة على بيئات تشغيلية متطورة نتيجة ظهور تقنيات جديدة. في ظل هذا التقدم التكنولوجي الكبير، يمكن اعتبار الآلات جزءًا من نظام شبكة متصل متكامل بدلًا عن كونها نظامًا مستقلًا بذاته عن باقي عملية التصنيع، كما أدى التقارب في الحوسبة والشبكات والحماية إلى تقليل تكلفة توصيل الأجهزة في النظام المتكامل.

تعزيز الكفاءة والسلامة Enhanced Efficiency and Safety



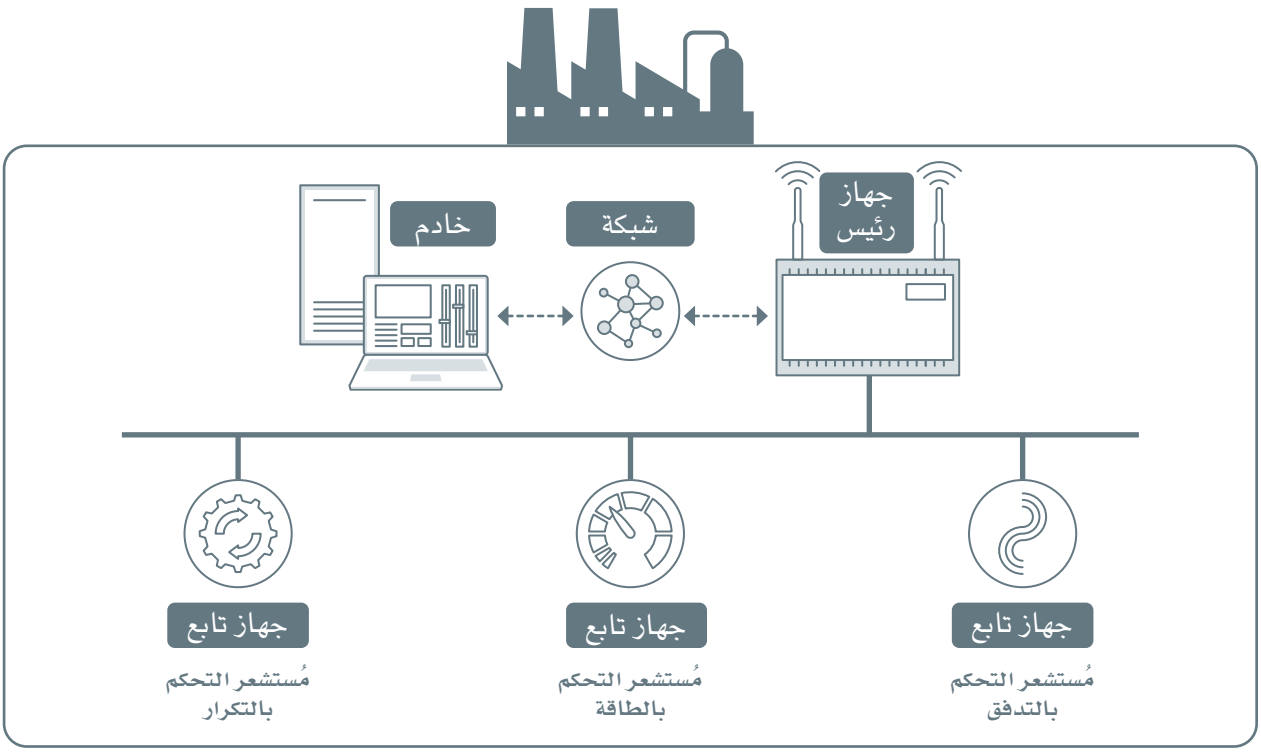
تسعى المصانع، لاسيما في قطاعات الأغذية والمشروبات، إلى الوصول إلى التحكم الآلي والأتمتة والتصنيع دون التدخل البشري لمهمات التصنيع المختلفة. يُمكن توظيف إنترنت الأشياء إضافةً إلى استخدام الروبوتات ومعالجة الصور لتمكين المصانع الحديثة من تحسين الكفاءة والسلامة.

هيكلية المصنع المُتصل An Architecture for the Connected Factory

بدأت الشركات بالدمج بين **الأتمتة الصناعية وأنظمة التحكم** (Industrial Automation and Control Systems – IACS) مع تطبيقات تقنية المعلومات وأدوات التحليلات لتوفير إمكانات تحكم وتحليلات تشغيلية مفيدة للأعمال. يُستخدم هذا الدمج للتحكم في العمليات الأساسية، أو مراقبة استخدام تدابير السلامة عند حدوث أي طارئ. يهدف دمج **الأتمتة الصناعية وأنظمة التحكم** (IACS) إلى تحقيق الجودة والكفاءة في الإنتاج مع الحفاظ على مستوى عالٍ من التكامل والموثوقية.

بروتوكول التحكم في الإرسال وبروتوكول مودبس
Modbus / Transmission Control Protocol - TCP

يُستخدم بروتوكول **مودبس** (Modbus) في القطاع الصناعي بشكل شائع لإدارة الأجهزة الرئيسة والفرعية. تم تحويل **مودبس** (Modbus) إلى بروتوكولات الاتصالات المستخدمة على نطاق واسع مثل Ethernet وTCP/IP، وعلى غرار تقنيات التحكم الأخرى في الأتمتة، يُستخدم مودبس كبروتوكول معياري مفتوح مُوثق ومُثبت في جميع أنحاء العالم. تُعدُّ آلية بروتوكول مودبس في إدارة الأجهزة الرئيسة والتابعة مناسبة تمامًا لطبيعة بروتوكول **التحكم في الإرسال** (TCP) الخاص بالاتصالات، ولكن بشكلٍ أقل تنوعًا.



شكل 8.1: بروتوكول الشبكة Modbus

تحديات المصنع المتصل Connected Factory Challenges

أصبح القطاع الصناعي أحد أبرز أهداف القرصنة الإلكترونية ومهاجمي الإنترنت. تسبب التقارب الحاصل بين الشبكات في المصانع والأعمال التجارية بظهور ثغرات أمنية لعمليات التصنيع، والتي كانت تجري تقليديًا بمعزل عن العمليات الأخرى.

يُعدّ الفصل بين شبكة المصنع الأساسية وشبكة تقنية المعلومات أبسط حل للتغلب على هذه الهجمات في الكثير من الأحيان. رغم أن هذا الحل يُعدّ فعّالًا وعمليًا، إلا أنه سيمنع التواصل مع عمليات الطبقة العليا وسيحدُّ من القدرات الممكنة لتحسين الأعمال المُدعَّمة بإنترنت الأشياء، وقد تظهر المزيد من المخاطر المُحتملة من أجهزة الحاسب المحمولة والأجهزة الحاسوبية الأخرى المتاحة في المصانع للعاملين الذين يتمتعون بوصول غير مُقيَّد للأجهزة.

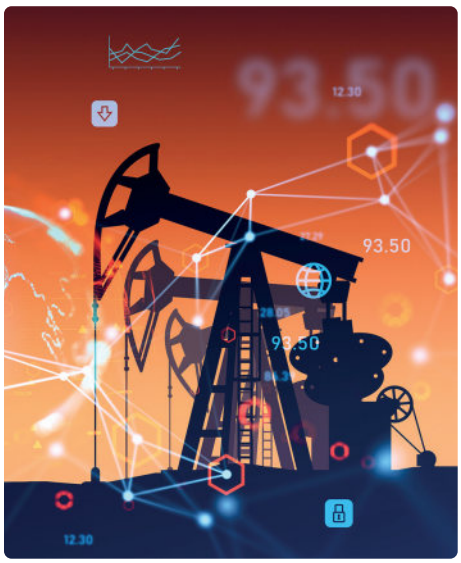
وزارة التعليم
Ministry of Education
2023 - 1445

الحوسبة الطرفية في المصنع المتصل Edge Computing in the Connected Factory

يمكن للآلات الموجودة في المصنع إنتاج كميات هائلة من البيانات، وبالتالي تبرز مشكلة تخزين تلك البيانات، وقد عالجت العديد من المصانع هذه المشكلة من خلال نشر الحواسيب لتخزين هذه البيانات. أدى جمع البيانات من أجهزة الحاسب الموجودة في المصنع إلى ظهور مشكلات عديدة تتعلق بالصيانة والأمان، فكما هو معروف، يتطلب كل حاسب تصحيحات وترقيات لنظام تشغيله، كما تزداد أعطال الأجهزة بشكل ملحوظ في المصانع، إذ إنّ معظم تلك الأجهزة لا تُصمّم لتحمُّل الظروف المختلفة فيها، وتشكِّل هذه المشكلات عائقًا أمام عمليات التصنيع في جمع البيانات ومعالجتها والاستجابة لها بكفاءة. يُمثّل هذا النهج عائقًا كبيرًا أمام تطوير الأفكار والفوائد التجارية المُحتملة التي قد توفرها تحليلات البيانات الصناعية. تساعد التطورات الجديدة في إضافة القدرات المحوسبة في الشبكات الطرفية على حل هذه المشكلات. بدأ المُصنِّعون بإدراك ميزات توصيل الآلات بخدمات الحوسبة المتطورة مع أجهزة الحوسبة الطرفية المُدمجة بالآلة القريبة من الحافة، والتي تتضمن قدرات التبديل والتوجيه والأمن معًا بشكلٍ دائم.

صناعة النفط والغاز Oil and Gas Industry

يُعدّ كل من النفط والغاز من أهم الموارد التي يستخدمها المجتمع الحديث، وذلك بدءًا بالبُنى التحتية للمواصلات، إلى تصنيع المواد البلاستيكية. يعتمد كل عنصر من عناصر الحياة الحديثة تقريبًا على توفُّر السلع التي تعتمد على هذه الصناعة. تهتم شركات النفط والغاز بشكل أساسي بخفض التكاليف، وزيادة الكفاءة والسرعة، وزيادة عوائد الاستثمارات. يُعدُّ التحكم في تكاليف الإنتاج وتعزيز الصحة والسلامة العامة وخاصة في الصناعات الخطرة من بين **مؤشرات الأداء الرئيسة** (Key Performance Indicators – KPLs) الأكثر أهمية في القطاع الصناعي.



شكل 8.2: مراقبة إنتاج النفط

على غرار القطاعات الأخرى، تستخدم شركات النفط والغاز إنترنت الأشياء للعديد من الأغراض بما فيها ما يلي:

- مراقبة حالة المُعدات الصناعية أو سلوكها للرؤية والتحكم.
- تحقيق أقصى قدر من الكفاءة للعمليات والموارد.
- تحسين عملية اتخاذ قرارات الأعمال التجارية.

تحديات الصناعة الرئيسة كمُحركات للتحول إلى الرقمنة Industry Key Challenges As Digitization Drivers

إن **إنترنت الأشياء** (IoT) والرقمنة – وهي عملية الاستفادة من التقدم الكبير في تقنية المعلومات لتطوير حلول وتقنيات جديدة للأعمال وإجراءاتها – تُمهد الطريق لتحقيق مكاسب تحسين الكفاءة التي كانت مستحيلة سابقًا وكذلك نماذج الأعمال الجديدة.

- النمذجة والتحليلات المتقدمة.
- البيانات الضخمة.
- تقارب **تقنية المعلومات** (IT) / **التقنية التشغيلية** (OT).
- الآلات الذكية.
- التنقل والتخزين السحابي.
- إدارة أداء الأصول.

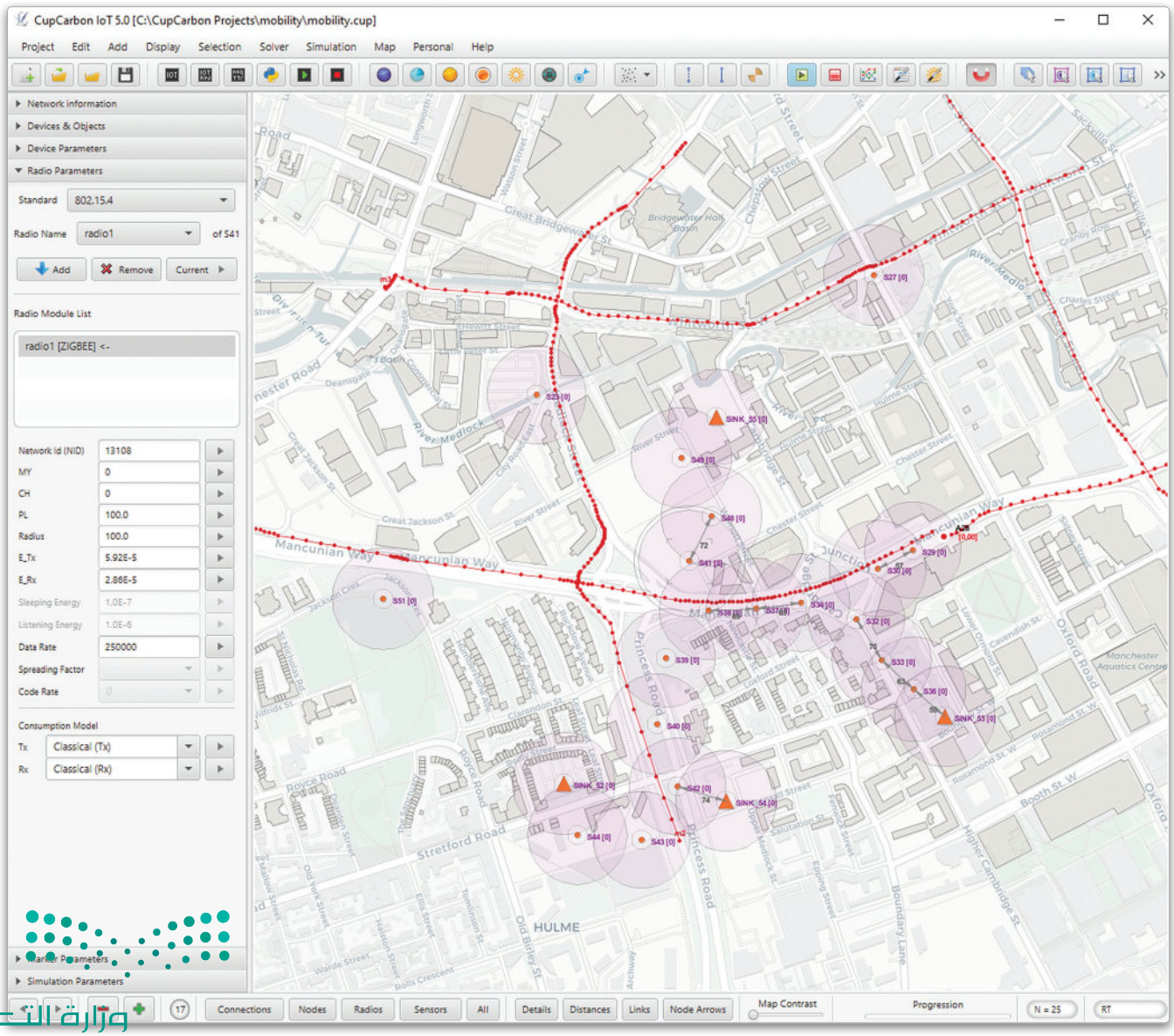
مثال

يُعدّ معمل الغاز في العثمانية أحد أكبر المصانع لمعالجة الغاز في المملكة العربية السعودية. تعمل حلول إنترنت الأشياء والذكاء الاصطناعي على تحسين الإنتاجية وموثوقية هذه المنشأة. فتُستخدم الطائرات دون طيار، والكائنات الذكية لمراقبة مُعدات مصفاة خطوط أنابيب الغاز وتستخدم طرق تحليل البيانات لتحسين الاستخدام. يوجد الآلاف من مُستشعرات إنترنت الأشياء تراقب حقل خريص النفطي.

ما برنامج كاب كاربون؟ What CupCarbon is

برنامج **كاب كاربون** (CupCarbon) هو مدينة ذكية افتراضية، وبيئة محاكاة لشبكة مُستشعرات لاسلكية بإنترنت الأشياء. يُمكن استخدامه لتصميم وإنشاء وتمثيل شبكات إنترنت الأشياء المُكونة من عُقد وأجهزة وأحداث وأمور أخرى. يوفِّر هذا المحاكي عددًا كبيرًا من الأدوات لتكوين الشبكات التي أُنشئت لاختبارها وتحسينها. كما تم به تضمين البروتوكولات الشائعة التي عُرضت سابقًا مثل **زيجبي** (Zigbee) و**واي فاي** (Wi-Fi) و**لورا** (LoRa)، بالإضافة إلى استخدام **خريطة الشارع المفتوح** (OpenStreetMap) كواجهة للتمكُّن من إجراء عمليات المحاكاة من أي مكان.

بعد إنشائك للشبكة المطلوبة بنجاح، يُمكنك استخدام برنامج كاب كاربون لبرمجة وحدة تحكم **أردوينو** (Auruino) يُمكن تجريبها على أجهزة حقيقية. تتواصل العُقد مع بعضها باستخدام البرمجة النصية. يستخدم المحاكي لغته البرمجية الخاصة المعروفة باسم **سينسكريبت** (SenScript)، كما أنه يدعم لغة بايثون. ستقوم في هذه الوحدة ببرمجة العُقد باستخدام لغة بايثون.



شكل 8.3: مشروع في CupCarbon

لتنزيل كاب كاربون (CupCarbon) وتشغيله:

> افتح متصفحك وقم بتنزيل الملف من الرابط:
1 http://binary-academy.com/dnld/KSA/IOT2/BinaryCupCarbon.zip

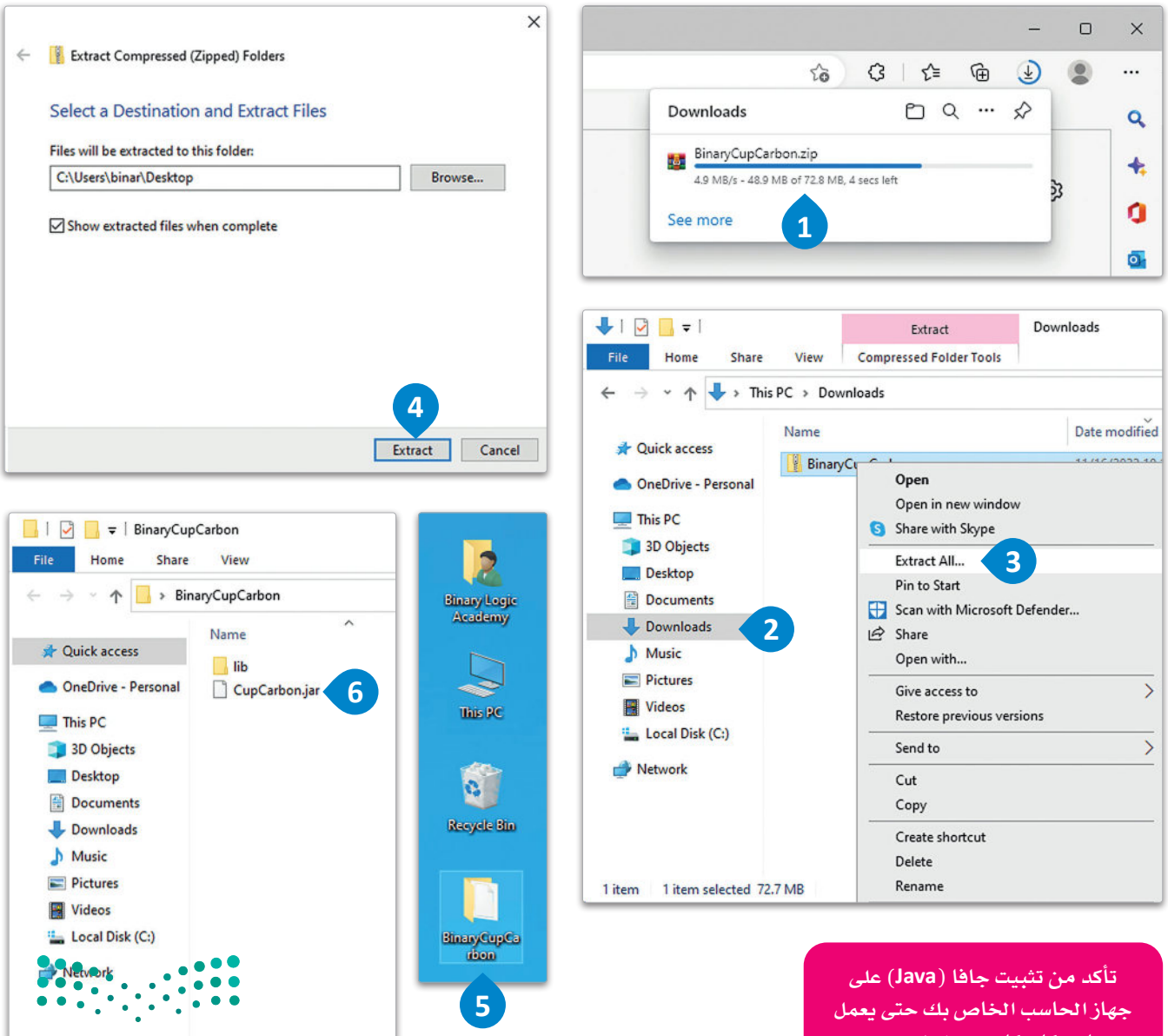
> افتح مستكشف الملفات، وابحث عن الملف الذي تم تنزيله في مُجلد Downloads (التنزيلات). 2

> اضغط بزر الفأرة الأيمن على الملف واختر Extract All (استخراج الكل). 3

> اختر سطح المكتب الخاص بك كوجهة للاستخراج، واضغط على Extract (استخراج). 4

> ابحث عن المجلد المُستخرج في سطح المكتب وافتحه. 5

> اضغط ضغطًا مزدوجًا على CupCarbon.jar لبدء برنامج CupCarbon (كاب كاربون). 6



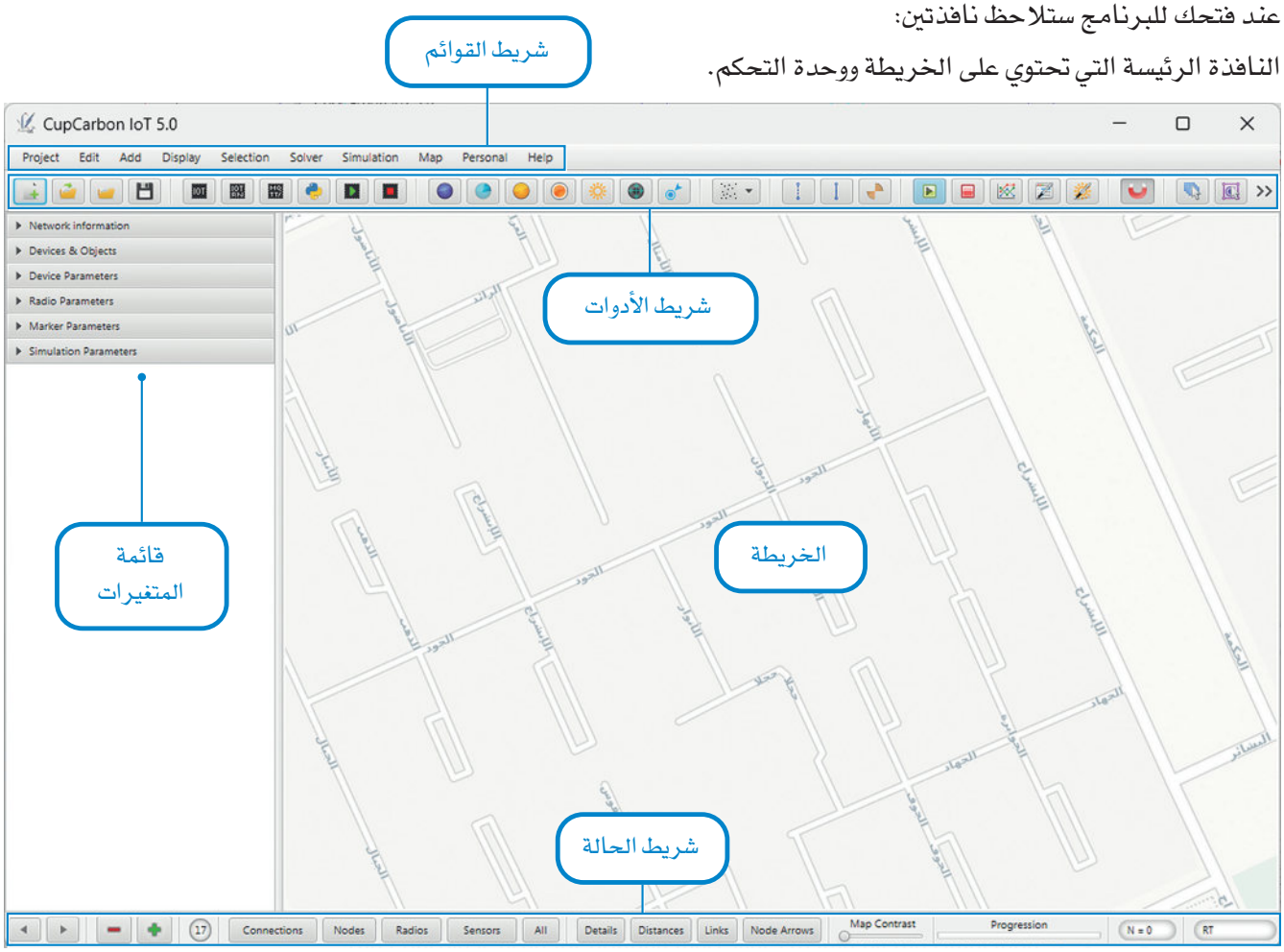
شكل 8.4: خطوات تنزيل كاب كاربون

تأكد من تثبيت جافا (Java) على جهاز الحاسب الخاص بك حتى يعمل برنامج كاب كاربون بشكل صحيح.

نافذة برنامج كاب كاربون The CupCarbon Windows

عند فتحك للبرنامج ستلاحظ نافذتين:

النافذة الرئيسة التي تحتوي على الخريطة ووحدة التحكم.



شكل 8.5: النافذة الرئيسة لبرنامج كاب كاربون

تُستخدم وحدة التحكم لطباعة الرسائل التي أُنشئت بواسطة المحاكاة، ولإخراج رسائل الأخطاء لمساعدة المستخدم على تصحيح الخطأ في البرامج النصية.

شكل 8.6:وحدة تحكم كاب كاربون

بدء الاستخدام Getting Started

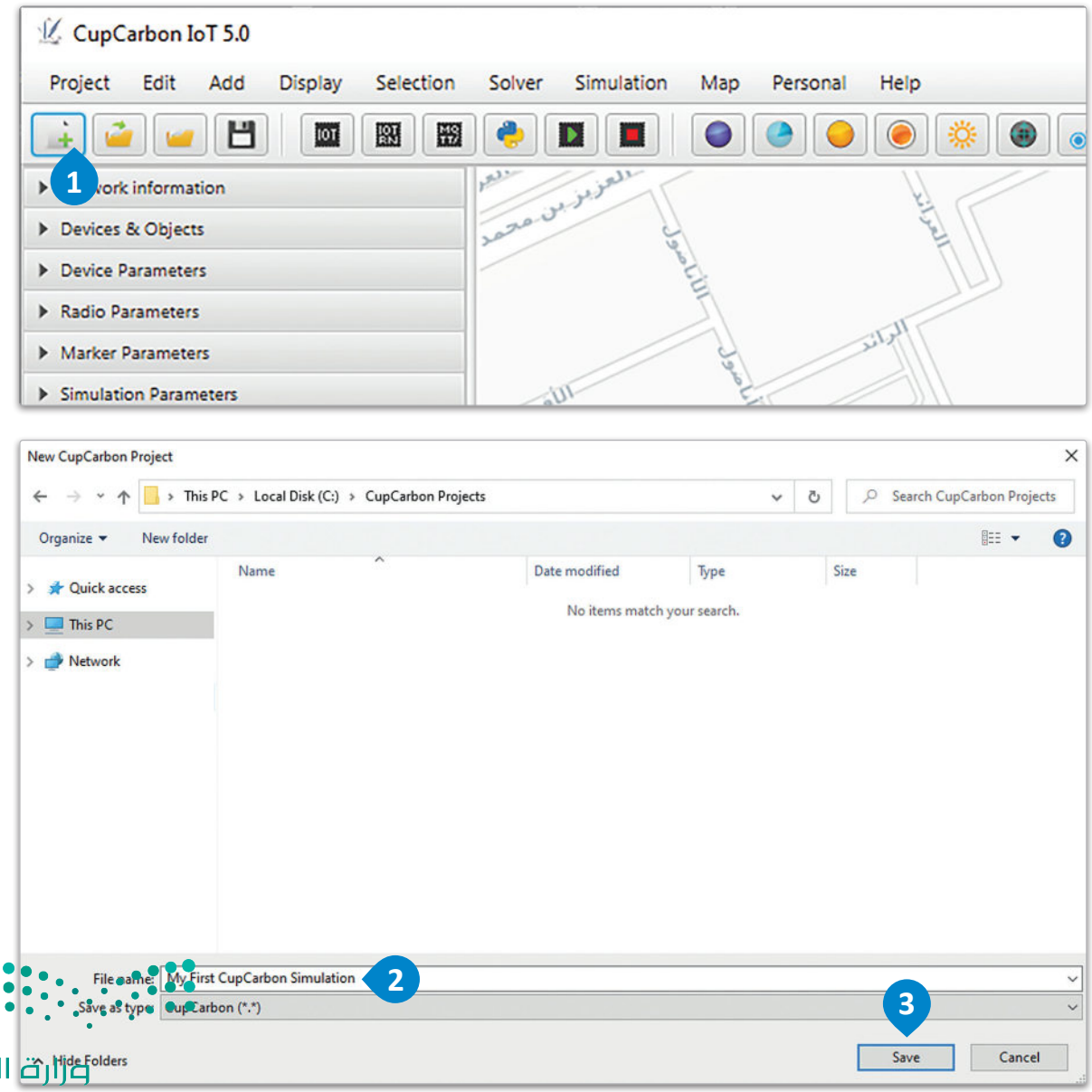
ستُنشئ في هذا الدرس محاكاة بسيطة لعُقدة إنترنت أشياء تطبع رسائل من أجل أن تعتاد على استخدام بيئة كاب كاربون.

في البداية ستنشئ مشروعًا جديدًا:

لإنشاء مشروع جديد:

> اضغط على New Project (مشروع جديد) من شريط الأدوات. 1

> اختر الموقع الذي تريده لحفظ المشروع، ثم اكتب "My First CupCarbon Simulation" في حقل File name (اسم الملف)، 2 ثم اضغط Save (حفظ). 3



شكل 8.7: إنشاء مشروع جديد

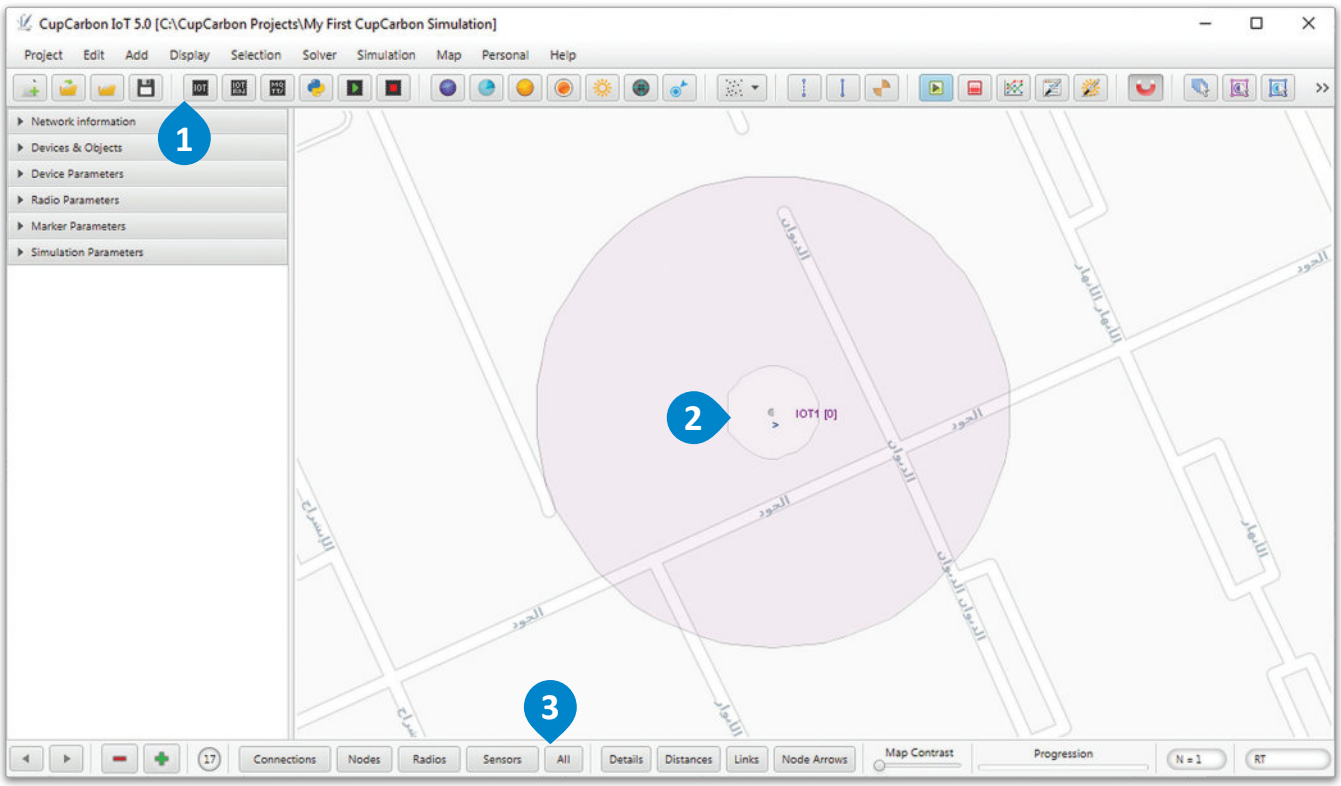
إضافة عُقدة Placing a Node

يمكنك في شريط الأدوات العثور على الكائنات المختلفة التي ستستخدمها في مشاريعك، والتي ستُنتج إما إشارات وتتواصل مع بعضها، أو ستنفذ إجراءات معينة. من هذه الكائنات الكائن IoT Node (عُقدة إنترنت أشياء) والذي يُمكن وضعه على الخريطة، ويمكن إعطاؤه مقطعًا برمجيًّا لتشغيله.

العُقد هي اللبِنات الأساسية لتكوين كاب كاربون. يُعرض في العُقدة المُعرِّف الخاص بها مع دائرتين حولها، دائرة داخلية تشير إلى نصف قطر المُستشعر المُستخدم للكشف عن المُستشعرات، ودائرة خارجية تكشف عن الأجهزة اللاسلكية مثل العُقد الأخرى.

لإضافة عُقدة:

- اضغط على IoT Node (عُقدة إنترنت أشياء) من شريط الأدوات. 1
- اضغط على الخريطة لإضافة العُقدة. 2
- اضغط على All (الكل) من شريط State (الحالة). 3
- اضغط على Esc.



شكل 8.8: إنشاء مشروع جديد

إنشاء المقطع البرمجي Creating a Script

ستقوم الآن بإنشاء مقطع برمجي بسيط يطبع رسالتين ذاتيتين بالتناوب على العُقدة. المقطع البرمجي المُستخدم هو كما يلي:

```
import time
while node.loop():
    node.print("hello")
    time.sleep(1)
    node.print("world")
    time.sleep(1)
```

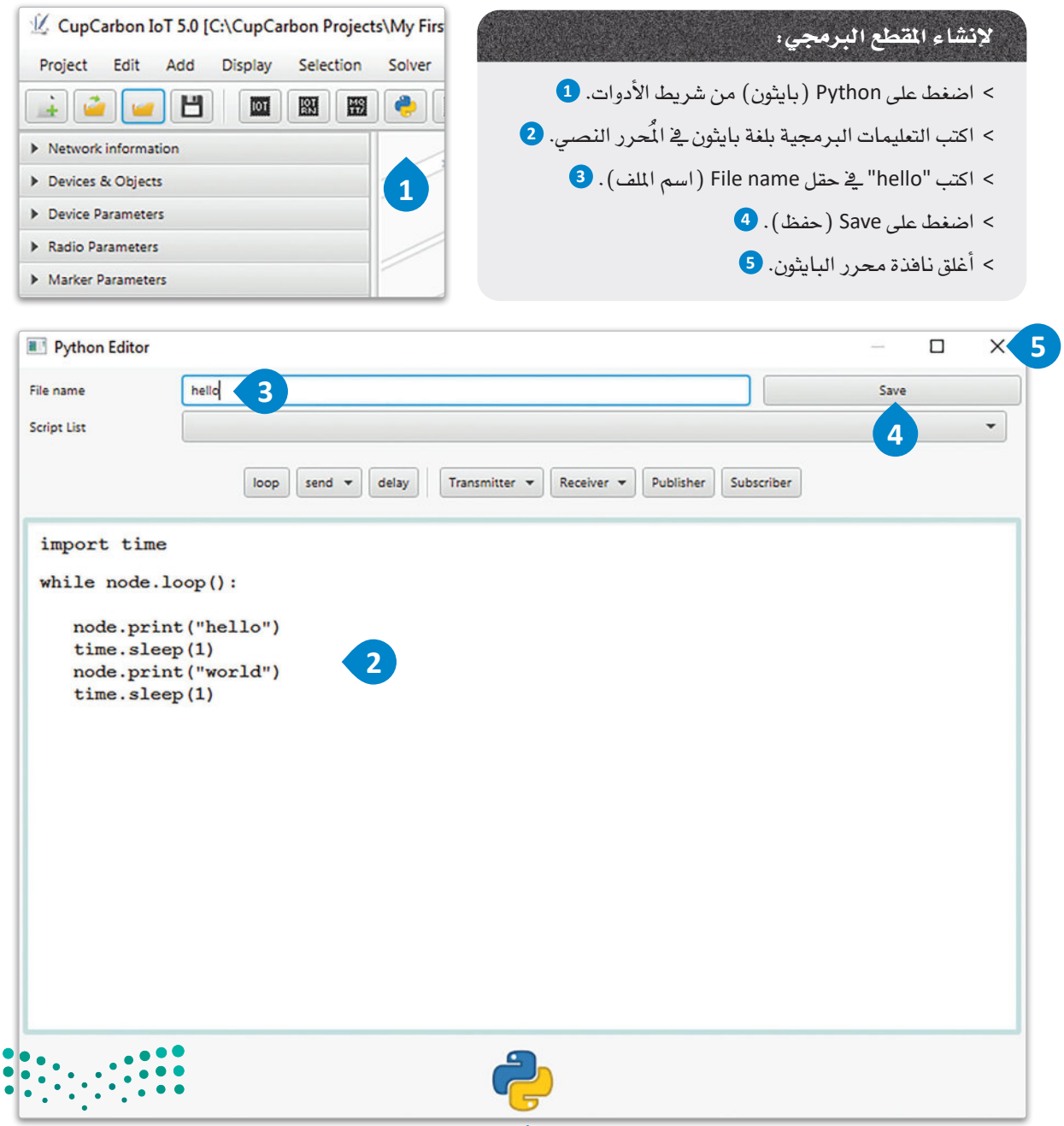
تأكد من استخدام المسافة البادئة المناسبة داخل التكرار (Loop) حتى يعمل المقطع البرمجي بشكل صحيح.

يجب عليك أولًا تضمين مكتبة time في بايثون، ستُستخدم دالة sleep المُدمجة لتأخير عملية الطباعة. يجب تضمين التعليمات البرمجية للعُقدة داخل التكرار while node.loop( ). يمكن للعُقدة الطباعة الذاتية باستخدام node.print( )، ويمكنها "السكون" -أي ألا تفعل شيئًا- باستخدام time.sleep( ). تأخذ دالة print( ) كمُعامل الرسالة المراد طباعتها على شكل نص، على سبيل المثال node.print ("hello world")، وتأخذ دالة sleep كمُعامل عدد موجب يشير لعدد الثواني التي تريدها للعُقدة ليتم التأخير الزمني، على سبيل المثال مع time.sleep( ) سيُنفّذ سكون للعُقدة لمدة 3 ثوان.

ستطبع العُقدة في برنامجك كلمة "hello"، ثم ستنتظر لمدة ثانية واحدة وتطبع "world"، وتنتظر مرةً أخرى لمدة ثانية واحدة، ثم تبدأ مرة أخرى من البداية بلا توقف ما لم يتم إنهاء المحاكاة.

لإنشاء المقطع البرمجي:

> اضغط على Python (بايثون) من شريط الأدوات. 1
> اكتب التعليمات البرمجية بلغة بايثون في المُحرر النصي. 2
> اكتب "hello" في حقل File name (اسم الملف). 3
> اضغط على Save (حفظ). 4
> أغلق نافذة محرر البايثون. 5



شكل 8.9: مُحرر البايثون

لإدراج المقطع البرمجي وتشغيل المحاكاة:

> اضغط على العُقدة. 1

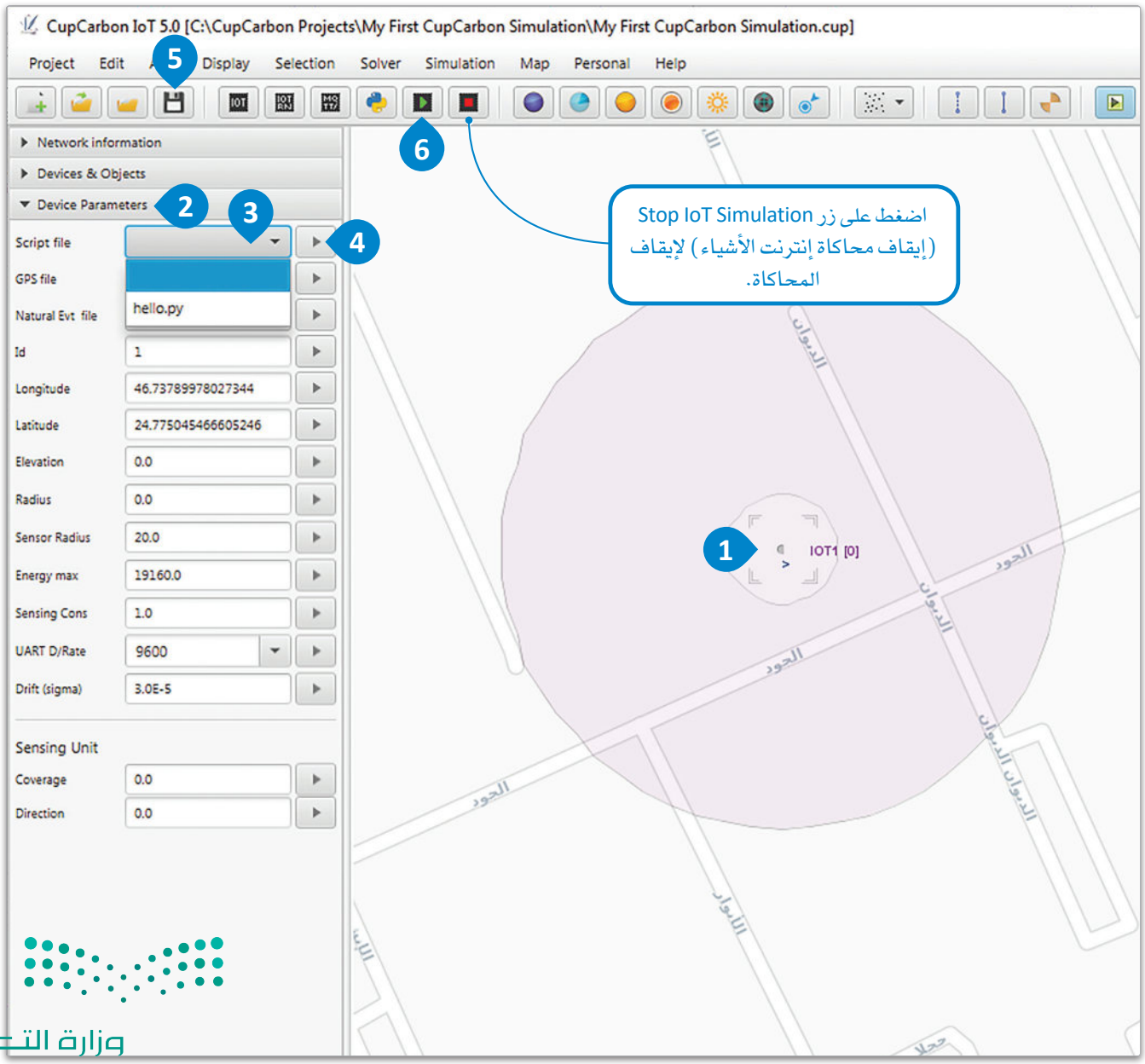
> اضغط على علامة تبويب Device Parameter (مُعامل الجهاز) في قائمة Parameter (مُعامل). 2

> اضغط على صندوق Script file (ملف البرنامج). 3

> من القائمة المنسدلة، اختر المقطع البرمجي hello.py، واضغط على الزر الموجود على اليمين لإدراج المقطع البرمجي في العُقدة. 4

> اضغط على Save project (حفظ المشروع) من شريط الأدوات. 5

> من شريط الأدوات، اضغط على Run IoT Simulation (تشغيل محاكاة إنترنت الأشياء). 6



شكل 8.10: إضافة المقطع البرمجي وتشغيل المحاكاة

كما هو متوقع، فإن العقدة ستتناوب في طباعة النصين "hello" و"world" لمدة ثانية واحدة لكل منهما.

شكل 8.11: حالات المحاكاة

تمرينات

1

حدد الجملة الصحيحة والجملة الخاطئة فيما يلي:	صحيحة	خاطئة
1. لا يُمكن استخدام مراقبة البيانات لزيادة كفاءة تحسين المعدات بشكل عام.	○	○
2. يُمكن لأقسام التقنية التشغيلية (OT) وتقنية المعلومات (IT) الدمج بين جميع قطاعات التصنيع في نطاق شبكي واحد.	○	○
3. يسهم توصيل أجهزة المصنع بشبكة واحدة في تقليل التكاليف.	○	○
4. يمكن للعمليات الأوتوماتيكية التي لا تعمل باللمس في مصنع الأطعمة والمشروبات تحسين جودة المُنتج النهائي.	○	○
5. لا يُمكن أن تتعرض الحواسيب الداخلية في المصانع إلى مخاطر أمنية.	○	○
6. قد تفقد أجهزة المصنع غير الموصولة بالشبكة الطرفية بيانات قيمة في حالة تعطلها.	○	○
7. يُمكن لأنظمة إنترنت الأشياء في صناعات النفط والغاز الحد من تعرض العمال للخطر.	○	○
8. يُمكن في برنامج كاب كاربون (CupCarbon) محاكاة بروتوكول زيجبي (ZigBee) الخاص بالأشياء الذكية.	○	○
9. يُمكن برمجة عُقد كاب كاربون بواسطة لغة بايثون فقط.	○	○
10. يُمكن في برنامج كاب كاربون إنتاج مخططات لوحدات التحكم الدقيقة مثل الأردوينو.	○	○

2 صنِّف تقنيات إنترنت الأشياء الرئيسة التي ستُغيِّر عمليات التصنيع التقليدية.

3 قَدِّم تحليلًا لكيفية تعرض المصانع المتصلة بأنظمة إنترنت الأشياء للهجمات الإلكترونية.

4 صِفْ مدى مساهمة الحوسبة الطرفية في المصانع المتصلة في تحسين كفاءتها وقدرتها الإنتاجية.

5 اذكر كيفية استخدام حلول إنترنت الأشياء الصناعية الذكية لتحسين العمليات في صناعة النفط والغاز.

6 أنشئ عُقدة في برنامج كاب كاربون واكتب برمجتها بحيث يكون هناك تكرار يجعل العُقدة تطبع الرسالة "A" لمدة ثانية واحدة، و"B" لمدة ثانيتين، و"C" لمدة 3 ثوانٍ. قم ببرمجة العُقدة وتشغيل المحاكاة.

7 أنشئ عُقدتين في برنامج كاب كاربون واكتب برمجتهما بواسطة بايثون. سيعرض كل مقطع برمجي رسالة "blink" متكررة. سترسل كل عُقدة في كل مرة رسالة عندما تكون الأخرى غير نشطة.

قم ببرمجة العُقد وتشغيل المحاكاة.

رابط الدرس الرقمي

www.ien.edu.sa

الدرس الثاني

الاتصال في شبكة إنترنت الأشياء

الاتصال بين الأجهزة Communication between Devices

تتكون شبكة إنترنت الأشياء من العديد من الأجهزة التي تُرسل وتستقبل البيانات بين بعضها. تتمايز هذه الأجهزة في إمكانياتها المختلفة مثل النطاق، وعرض النطاق الترددي للبيانات، واستهلاكها للطاقة، وبالتالي تقوم بتشغيل مجموعات مختلفة من الأوامر لتوفير العديد من الوظائف. ستُنشئ في المشروع الآتي شبكة مثل هذه، وستستكشف اللبنات البرمجية الأساسية المكونة لها.

مراقبة الحريق والتحذيرات Fire Surveillance and Notification

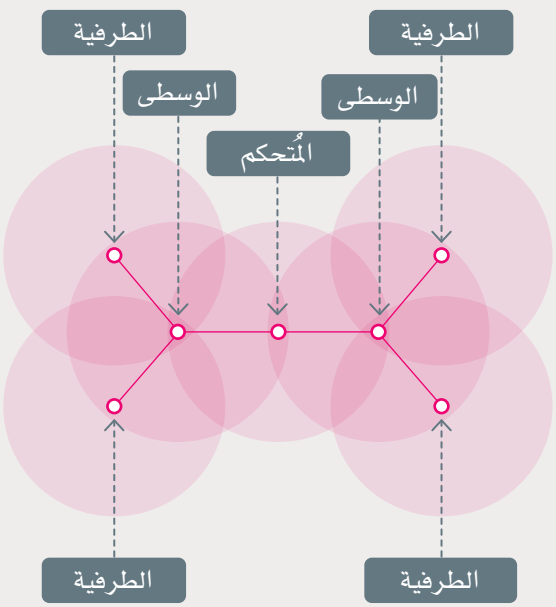
ستُنشئ في هذا الدرس مشروعًا يحاكي نظام مراقبة الحرائق في المصانع.

سيتم في هذه المحاكاة إنشاء حرائق عشوائية في مصنع، وسيبلغ النظام وحدة التحكم الرئيسة بالمصنع عن القطاع الذي يوجد فيه الحريق. سيُنفّذ ذلك باستخدام مجموعة متنوعة من العُقد ذات الوظائف المختلفة التي ستتواصل مع بعضها لتمرير الرسالة بدءًا من العُقد **الطرفية** (Edges) مرورًا بالعُقد **الوسطى** (Proxies) لتصل أخيرًا إلى العُقدة الداخلية: **المُتحكم الرئيس** (main controller).

العُقد ووظائفها
The Nodes and their Functions

ستستخدم ثلاثة أنواع من العُقد في مشروعك لتمثيل قطاعات المصنع المختلفة:

- قد يحدث حريق في **الأطراف** (Edges)، وسيتم تحديده بواسطة مولد أرقام عشوائي. في مثل هذه الحالات، ستطبع العُقدة رسالة ذاتية وترسل رسالة مختلفة تحتوي على رقم القطاع إلى عُقدة **الوسطى** (Proxy) المجاورة لها، ثم ستسكُن لمدة زمنية محددة.
- سيقرأ الوكلاء أي رسائل قد تلقوها ويعيدونها إلى وحدة **المُتحكم الرئيسة** (main controller). كما قد يقومون بطباعة رسالة ذاتية لإبلاغ المستخدم بما يقومون به.
- تقوم وحدة التحكم الرئيسة أيضًا بقراءة أي رسائل قد تكون تلقتها، وتطبع الرسائل التي أُنشئت بواسطة العُقد الطرفية التي تخبر المستخدم بالحريق.



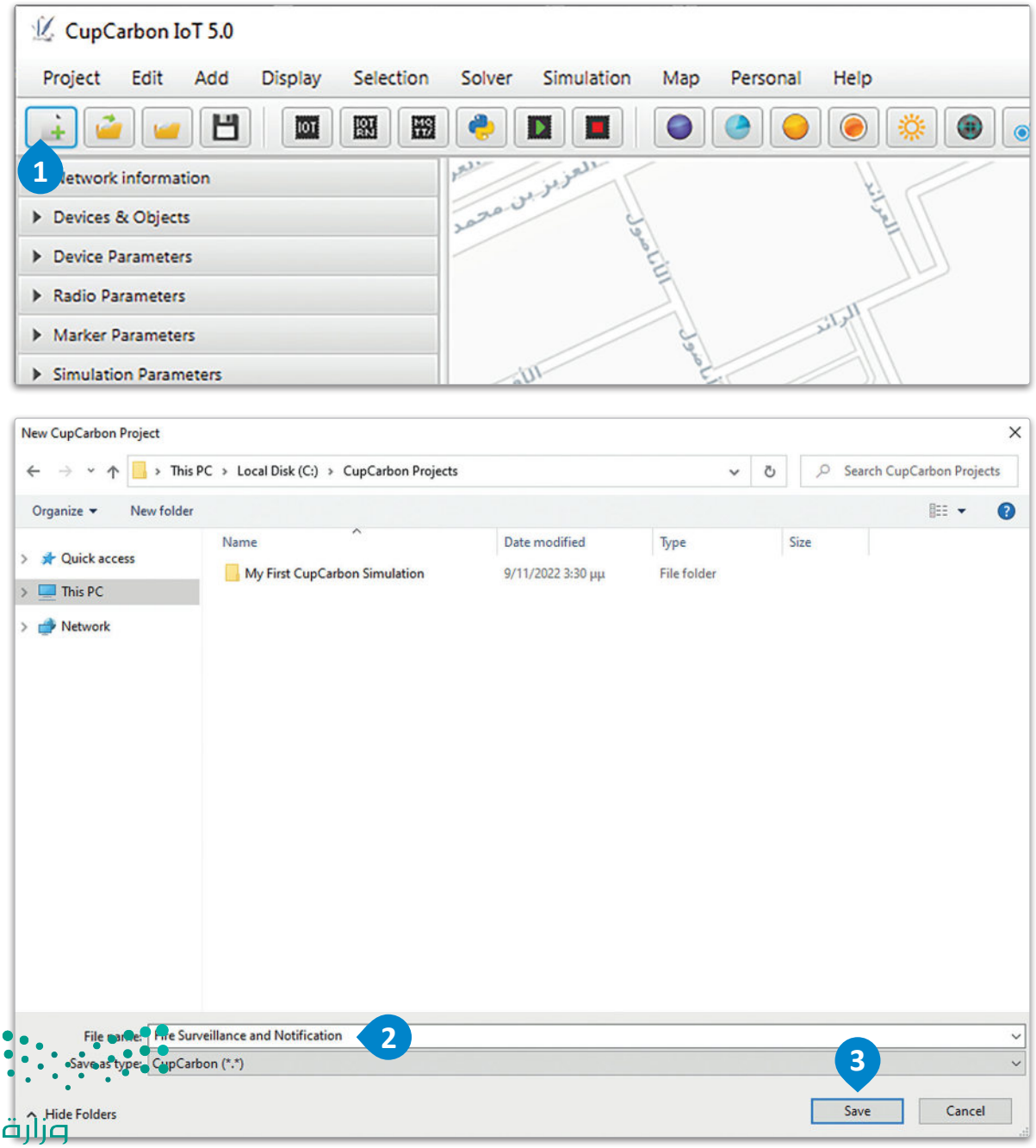
شكل 8.12: العُقد ووظائفها

لنبدأ بإنشاء مشروع جديد:

لإنشاء مشروع جديد:

> اضغط على New Project (مشروع جديد) من شريط الأدوات. 1

> اختر الموقع الذي تريد تخزين المشروع فيه، ثم اكتب "Fire Surveillance and Notification" في حقل File name (اسم الملف) 2 واضغط على Save (حفظ). 3



شكل 8.13: إنشاء مشروع جديد

ابدأ بإنشاء شبكة العُقدة بإضافة وحدة المُتحكم الرئيسة والوسطى:

لإدراج وحدة المُتحكم وعُقد الوسطى:

> اضغط على IoT Node (عُقدة إنترنت أشياء) من شريط Toolbar (الأدوات). 1

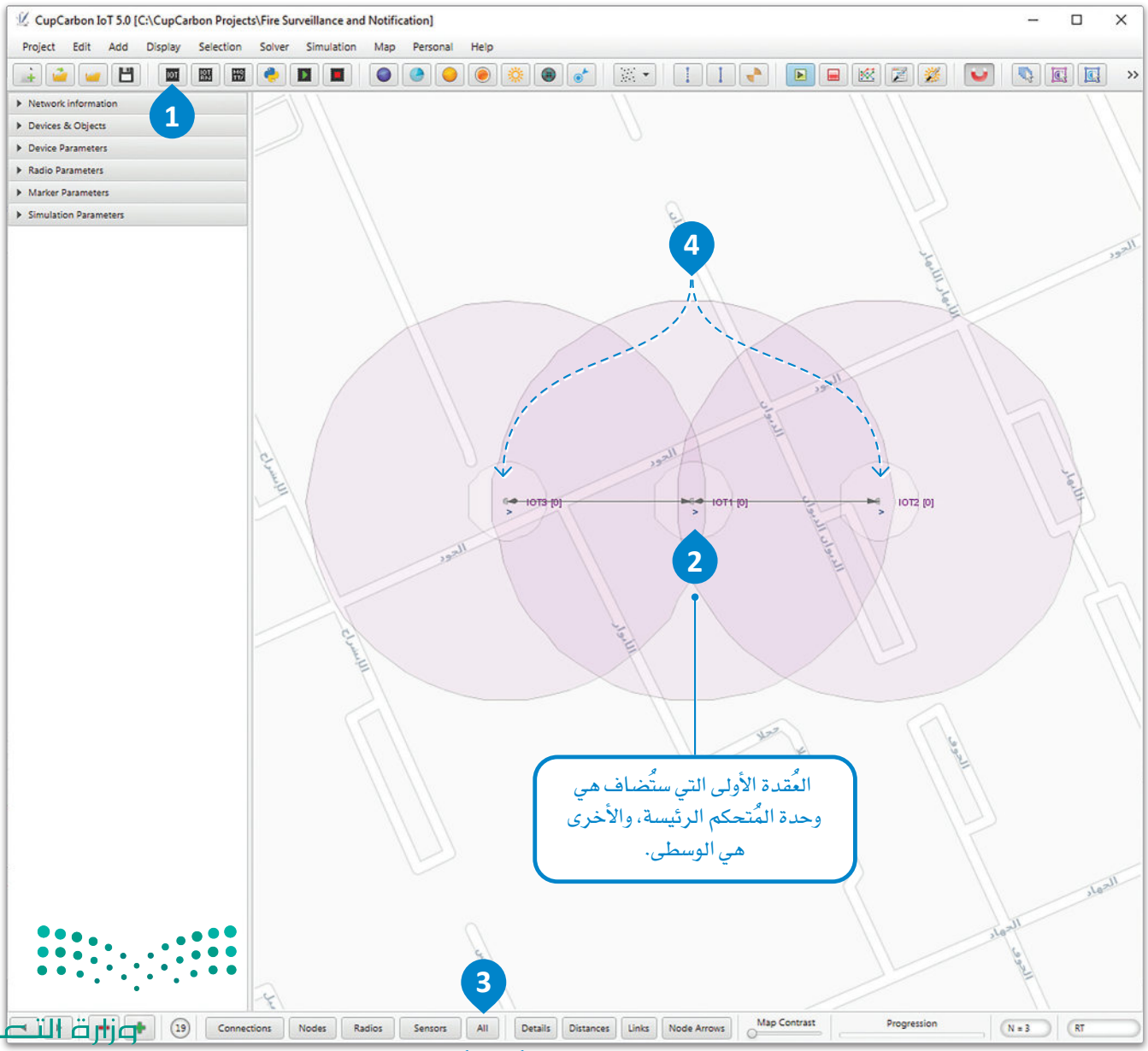
> اضغط على الخريطة لإضافة العُقدة. 2

> اضغط على All (الكل) من شريط State (الحالة). 3

> ضع عُقدتين أخريين على يسار ويمين العُقدة الأولى، وداخل دائرتها الخارجية. 4

> اضغط على Esc.

إذا لم يتم وضع العُقد داخل نصف قطر وحدة المُتحكم، فلن تتمكن من الاتصال. وللتغلب على ذلك اسحبها وأفلتها بالقرب من وحدة المُتحكم حتى يظهر سهم ثنائي الاتجاه يربط بين العُقدتين.



شكل 8.14: إضافة وحدة المُتحكم وعُقد الوسطى

تابع بإضافة العُقد الطرفية:

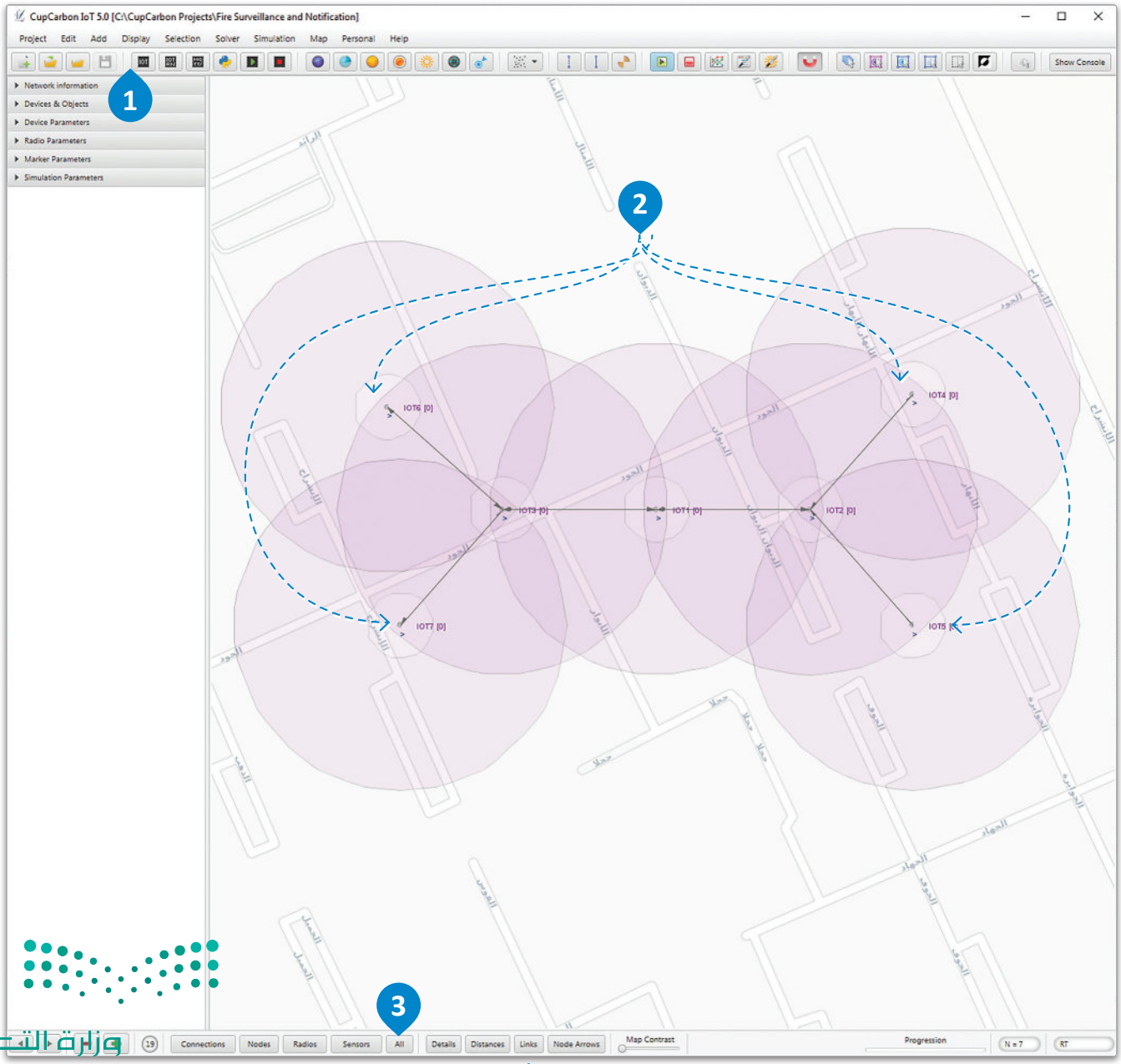
لإدراج العُقد الطرفية :

> اضغط على IoT Node (عُقدة إنترنت أشياء) من شريط Toolbar (الأدوات). 1

> ضع عُقدتين على كل proxy node (عُقدة الوسطى)، وذلك داخل دائرتها الخارجية، ولكن خارج نطاق أي عُقدة أخرى. 2

> اضغط على All (الكل) من شريط الحالة. 3

> اضغط على Esc في لوحة المفاتيح.



شكل 8.15: إضافة العُقد الطرفية

إنشاء المقاطع البرمجية Creating the Scripts

ستتعرف الآن على المقاطع البرمجية التي ستقوم بتشغيل العُقد. لنبدأ ببرمجة العُقد الطرفية.

في البداية، أضف المكتبات اللازمة.

```
import time
import random
```

تأخذ دالة توليد الأرقام العشرية ()randint عددين صحيحين كوسيطين، وتعيد عددًا صحيحًا عشوائيًا داخل نطاق هذين العددين. على سبيل المثال، وفي الحالة السابقة ستُنشئ (1،6)randint عددًا صحيحًا بشكل عشوائي بقيمة بين 1 إلى 6. سيُستخدم الرقم ليمثل القطاع الذي قد يندلع فيه الحريق في كل فترة زمنية. وسيتم تخزين العدد الصحيح في متغير fire.

```
while node.loop():
  fire = random.randint(1, 6)
```

إذا افترضنا أن دالة ()randint ستُرجع الرقم 1، فستكون النيران قد اندلعت افتراضيًا في هذا القطاع. سيتحقق البرنامج مما إذا كانت قيمة المتغير fire تساوي 1، وإذا كان الأمر كذلك، فسيتم تشغيل سلسلة من الأوامر بما فيها طباعة الرسالة "FIRE!" (حريق) على العُقدة نفسها. وإرسال رسالة تحتوي على مُعرف القطاع الخاص بها إلى العُقد الوسطى (proxy node) المجاورة لها.

إن مُعرِّف القطاع هو نفسه رقم مُعرّف العُقدة، وهو عدد صحيح فريد. إذا كان مُعرّف القطاع 5، فستكون الرسالة المُرسلة "FIRE IN SECTOR 5" (حريق في قطاع 5). يمكن إرجاع مُعرف العُقدة وبالتالي القطاع بواسطة الدالة ()id. يتم إرجاع المُعرف كرقم، لذلك يجب "تحويله" إلى نوع نص قبل أن يُربط بالرسالة المتبقية.

```
if fire == 1:
  node.print("FIRE!")
  message = "FIRE IN SECTOR " + str(node.id())
```

يمكن للعُقد إرسال البيانات لبعضها باستخدام دالة ()send. تستخدم الدالة وسيطًا واحدًا فقط، وهو نص الرسالة الذي تقوم ببثه إلى جميع العُقد داخل نطاقها.

```
node.send(message)
```

إذا أنتجت دالة توليد الأرقام العشوائية أي عددٍ صحيح آخر (في حالتنا أي رقم من 2 إلى 6)، فلا يوجد حريق في القطاع، ويتعين على العُقدة ببساطة طباعة نص ذاتي فارغ لمسح أي نص مطبوع سابقًا.

```
  else:
    node.print("")
```

في الختام، ستسكن العُقدة لفترة زمنية عشوائية، وذلك لمحاكاة عشوائية الأحداث في الحياة الواقعية. سيتحقق ذلك باستخدام الدالة ( )uniform التي تعمل مثل دالة ( )randint، ولكنها تُنتج أعدادًا حقيقية وليس فقط أعدادًا صحيحة. ستتراوح فترة السكون في مشروعك بين 1–4 ثوانٍ.

```
  time.sleep(random.uniform(1, 4))
```

المقطع البرمجي النهائي (edge.py)

Complete Code (edge.py)

```
import time
import random

while node.loop():

  fire = random.randint(1, 6)

  if fire == 1:
    node.print("FIRE!")
    message = "FIRE IN SECTOR " + str(node.id())
    node.send(message)
  else:
    node.print("")

  time.sleep(random.uniform(1, 4))
```

التالي هو المقطع البرمجي الخاص بالعُقد الوسطى.

عند استقبال العُقدة للبيانات، تُخزّن في **المخزن المؤقت** (buffer) الخاص بها حتى قراءتها، ولذلك يجب التحقق من حجم المخزن المؤقت في البداية حيث يجب أن تكون قيمته أكبر من صفر (غير فارغ). يُمكن إرجاع حجم المخزن المؤقت بواسطة الدالة bufferSize( ).

```
import time

while node.loop():

  if node.bufferSize() > 0:
```

يُمكنك بعد ذلك قراءة البيانات المُستقبَلة باستخدام الدالة read( ). بعد قراءة الرسالة، تُخزّن في رسالة المُتغير. تقوم العُقدة أيضًا بطباعة رسالة "FORWARDING..." لتوضح أنها تعيد توجيه الرسالة إلى وحدة التحكم الرئيسة.

```
    message = node.read()
    node.print("FORWARDING…")
```

كما هو الحال في البرنامج السابق، ستُرسل الرسالة المُخزنة في المتغير إلى وحدة المُتحكم الرئيسة باستخدام الدالة send( ). وفي هذه المرة وبصرف النظر عن الرسالة، سيستلزم الأمر وسيطة إضافية وهي مُعرف العُقدة المُستقبلة (Node ID). نظرًا لكون الرسالة خاصة بعُقدة واحدة وبمُعرِّف مُحدد، فلا يلزم بث الرسالة، بل يمكن بدلًا من ذلك أن تكون أحادية الإرسال (أي تُرسل إلى عُقدة واحدة فقط). في الحالة السابقة، أُضيفت وحدة المُتحكم الرئيسة أولًا، وبالتالي يكون لها مُعرِّف مساويًا 1.

```
    node.send(message, 1)
```

بعد ذلك، سوف تسكن العُقدة لمدة ثانية واحدة لمنح المُستخدم وقتًا كافيًا لقراءة الرسالة التوضيحية المطبوعة على العُقدة، ثم ستقوم العُقدة بمسح الرسالة وذلك بطباعة نص فارغ.

```
    time.sleep(1)
    node.print("")
```

ينتهي المقطع البرمجي بسكون العُقدة لفترة زمنية صغيرة جدًا (جزء من مائة من الثانية)، مما يمنحها القدرة على الاستجابة في حالة تلقيها الكثير من البيانات.

```
    time.sleep(0.01)
```

المقطع البرمجي النهائي (proxy.py)
Complete code (proxy.py)

```
import time
while node.loop():

  if node.bufferSize() > 0:
    message = node.read()
    node.print("FORWARDING…")
    node.send(message, 1)
    time.sleep(1)
    node.print("")

  time.sleep(0.01)
```

مُعامل الرقم في دالة ( )send هو رقم مُعرف عُقدة وحدة المُتحكم.

يتشابه المقطع البرمجي لوحدة المُتحكم نوعًا ما مع برامج العُقد الوسطى، فهو يفحص المخزن المؤقت، ويقرأ الرسالة المُستلمة، ولكن الرسالة الذاتية المطبوعة هي نفس الرسالة التي أُنشئت في الأصل بواسطة عُقدة الطرفية (edge node).

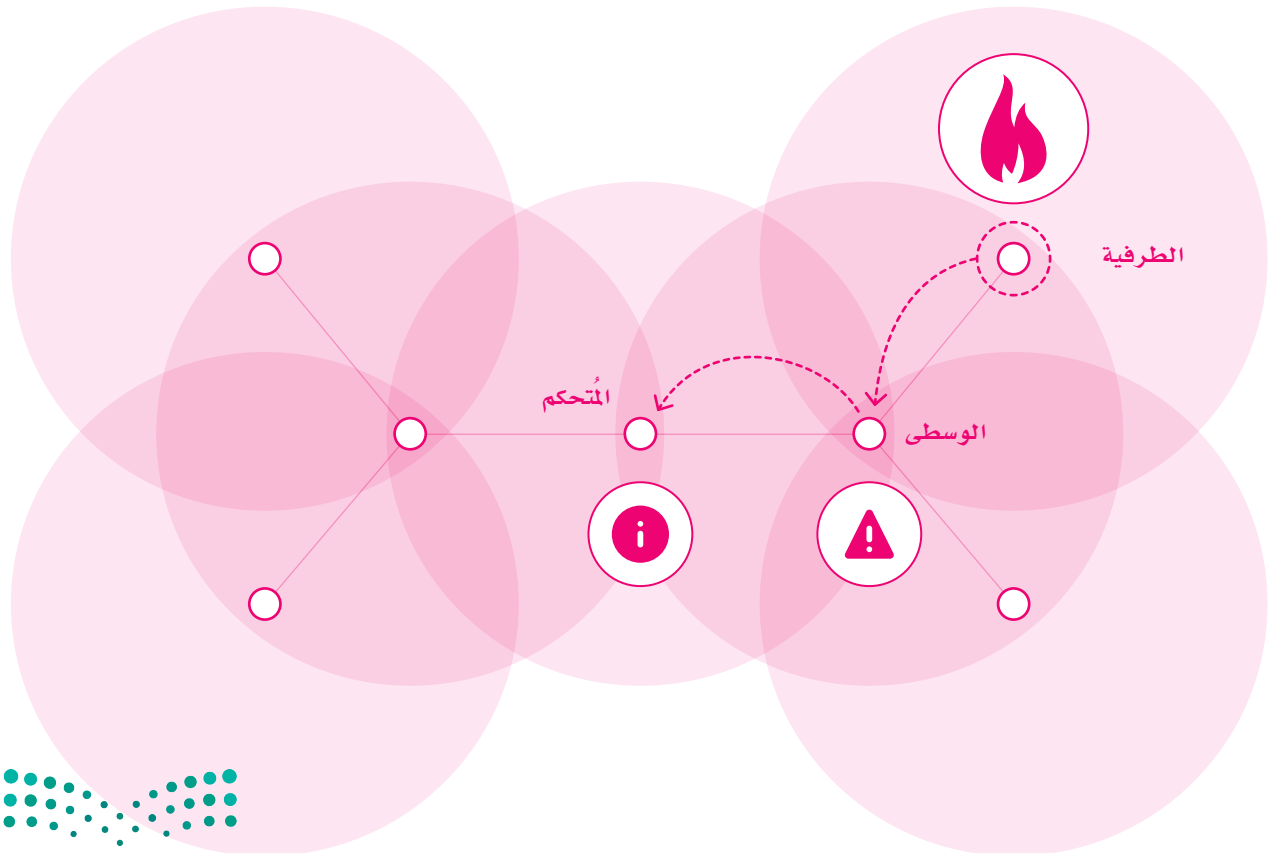
```
import time
while node.loop():

  if node.bufferSize() > 0:
    message = node.read()
    node.print(message)
```

بعد ذلك وكما حدث في عُقد الوسطى، ستسكُن وحدة المُتحكم، ولكن لمدة ثانيتين هذه المرة، ثم تطبع نصًا فارغًا.
وفي الختام ستسكُن لفترة قصيرة بنفس الطريقة التي حدثت مع العُقد الوسطى.

```
    time.sleep(2)
    node.print("")

  time.sleep(0.01)
```

المقطع البرمجي النهائي (controller.py)
Complete code (controller.py)

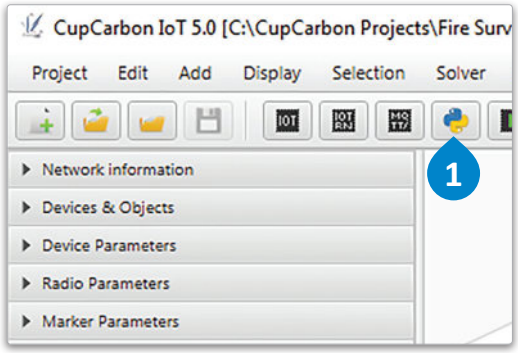
```
import time
while node.loop():

  if node.bufferSize() > 0:
    message = node.read()
    node.print(message)
    time.sleep(2)
    node.print("")

  time.sleep(0.01)
```



شكل 8.16: الشبكة بشكلها النهائي

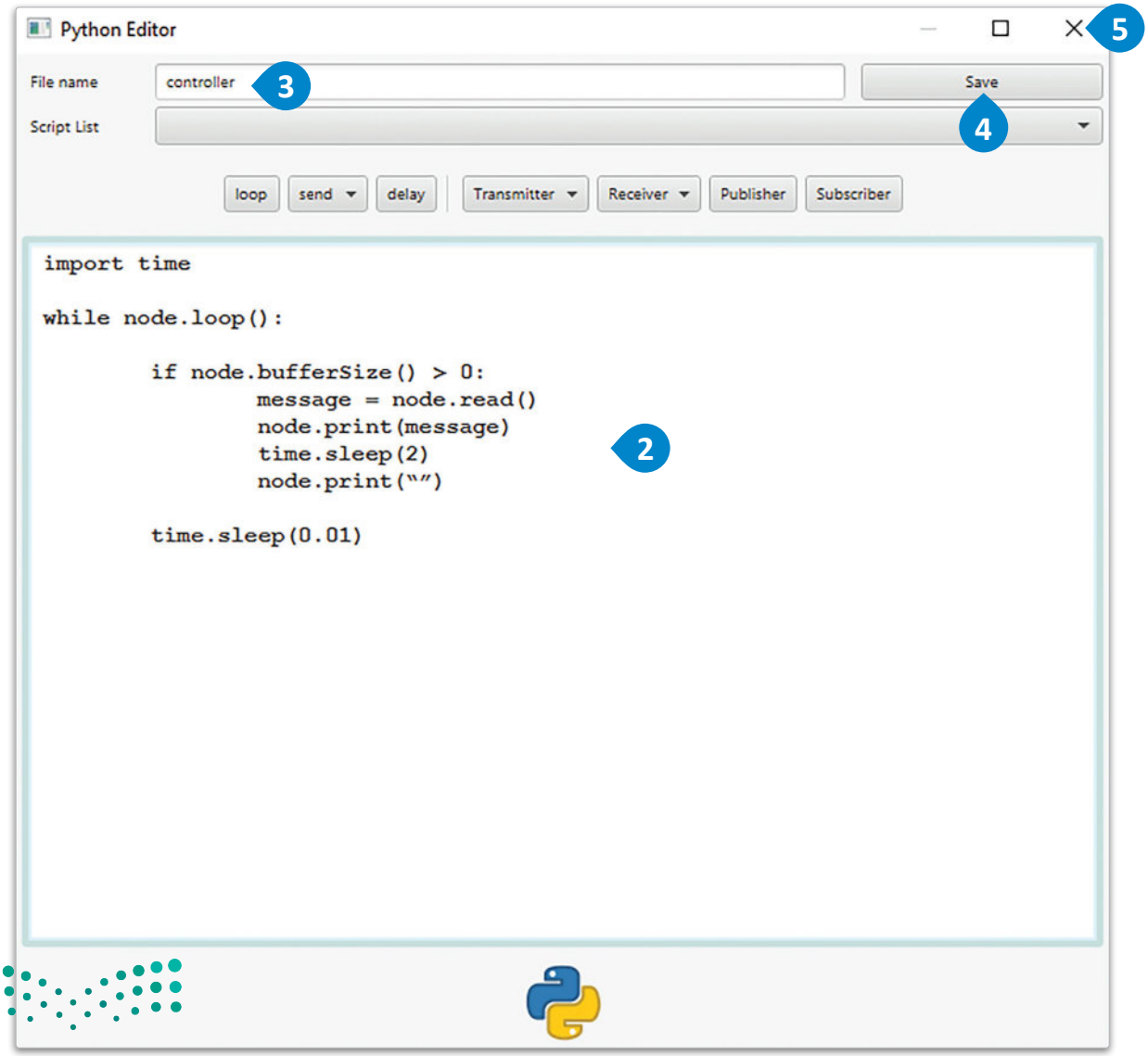
الآن وبعد أن تعرَّفت على وظيفة المقاطع البرمجية، تابع عملك وقم بإنشائها.

لإنشاء البرنامج وتطبيقه على عُقدة وحدة المُتحكم:



لإنشاء المقطع البرمجي:

- اضغط على بايثون في شريط الأدوات. 1
- اكتب أوامر بايثون البرمجية في الحقل الفارغ. 2
- اكتب controller (وحدة المُتحكم) في حقل File name (اسم الملف). 3
- اضغط على Save (حفظ). 4
- أغلق نافذة محرر بايثون النصي. 5



شكل 8.17: إنشاء المقطع البرمجي

لإدراج البرنامج:

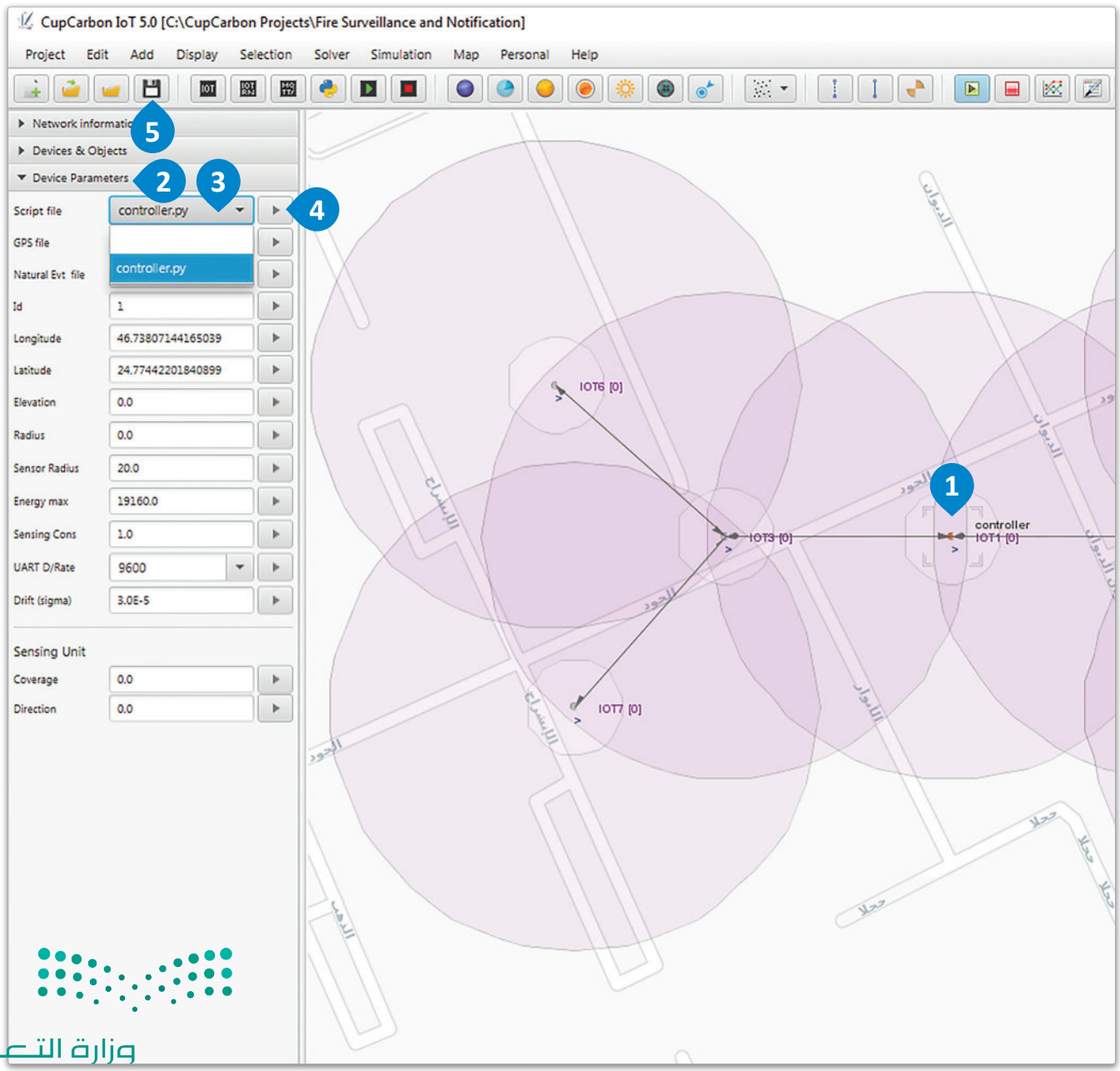
> اضغط على العُقدة. 1

> اضغط على علامة تبويب Device Parameters (مُعاملات الجهاز) في قائمة Parameters (المُعاملات). 2

> اضغط على صندوق Script file (ملف المقطع البرمجي). 3

> من القائمة المنسدلة، اختر ملف controller.py واضغط على الزر الموجود على اليمين لإدراج المقطع البرمجي في العُقدة. 4

> اضغط على Save Project (حفظ المشروع) من Toolbar (شريط الأدوات). 5



شكل 8.18: إضافة المقطع البرمجي

أنشئ المقاطع البرمجية الأخرى بنفس الطريقة، وانسخ أوامرها وطبقّها على العُقد المقابلة لها، بحيث تحتوي جميع العُقد على المقطع البرمجي. عند الانتهاء، اضغط على Run IoT Simulation (تشغيل محاكاة إنترنت الأشياء) من شريط الأدوات.

لاحظ أنه نظرًا لاستخدامك مُولدات أرقام عشوائية، فقد تشتعل حرائق في بعض القطاعات الموجودة على الأطراف أكثر من غيرها والتي قد لا تشتعل فيها حرائق على الإطلاق.

استخدم أسماءً نصية مُعبرة وواضحة للمقاطع البرمجية مثل edge.py و proxy.py.

شكل 8.19: حالات المحاكاة

تمرينات

1. وسِّع مشروعك لدعم عُقدة طرفية (Edge) تضاف لكل عُقدة وسطى (Proxy)، بحيث يكون لكل عُقدة وسطى ثلاثة عُقد طرفية. لا تنْسَ إضافة المقاطع البرمجية داخل العُقد الجديدة.

2. وسِّع مشروعك لدعم عُقدة وسطى إضافية، وأضف عقدتين طرفيتين جديدتين إلى الوسطى، بحيث يكون لدى وحدة المُتحكم الرئيسة ثلاث عُقد وسطى، ولكل عُقدة وسطى عُقدتين طرفيتين. لا تنْسَ إضافة المقاطع البرمجية داخل العُقد الجديدة.

3. حدِّد أي قسم من التعليمات البرمجية يُقرر تكرار حدوث الحرائق. عدِّل مشروعك في برنامج كاب كاربون (CupCarbon) لزيادة احتمال حدوث الحرائق أكثر من السابق.

4. قد يؤدي أي تأخير زمني (latency) في شبكة المصنع إلى تأخير الاتصال بين العُقد. قم بتعديل برنامجك الخاص بعُقد الوسطى لجعل العُقد في وضع السكون لفترة أطول. هل لاحظت وجود أي تأخير أو فقدان لأي رسائل؟ دوِّن ملاحظاتك أدناه.

5. وسِّع مشروعك ليدعم احتمال حدوث تسرب المياه وحدوث الفيضان. عدِّل برنامجك للقطاعات المُعرضة للحرائق، بحيث يعني إرجاع القيمة من دالة توليد الأرقام العشوائية ( ) randint القيمة 2 حدوث تسرب للمياه أو فيضان في هذا القطاع. على العُقدة القيام بطباعة الرسالة المناسبة وإرسالها.

رابط الدرس الرقمي

www.ien.edu.sa

الدرس الثالث

إنترنت الأشياء والأجهزة المحمولة المؤتمتة

الصناعة الذكية والأتمتة Smart Industry and Automation

تُعدُّ الأتمتة ميزة مهمة للتقنية الحديثة، وكذلك فهي عاملٌ مُساهمٌ بشكلٍ رئيس في الثورة الصناعية الرابعة. تُعزّز الصناعة الذكية من خلال تقنيات الأتمتة التي تزيد من الإنتاجية، مما يُتيح تحقيق المزيد من الأرباح. ستُنشئ في المشروع الآتي محاكاة لنظام يفحص منطقة تخزين المصنع للحاويات التي تحتوي على مواد قابلة للتلف إذا تُركت دون تبريد طوال الليل، وذلك باستخدام مركبة تفتيش آلية.

ستتخذ مركبة التفتيش الآلية مسارًا مُحددًا سابقًا في منطقة تخزين المصنع، وستُوضع علامات على الحاويات وفقًا لمحتوياتها من مواد قابلة للتّلف، ومواد طويلة الأمد لا تحتاج إلى التبريد. تحتوي كل حاوية على **رقاقة إنترنت الأشياء** (IoT Tag) ترسل رسالة باستخدام موجاتها اللاسلكية لتُبلغ المركبة الآلية بمحتوياتها. توجد أيضًا بعض محطات الشحن في كافة أنحاء منطقة التخزين لشحن بطارية المركبة التي ستنخفض أثناء حركة المركبة.

شكل 8.20: مركبة صناعية آلية

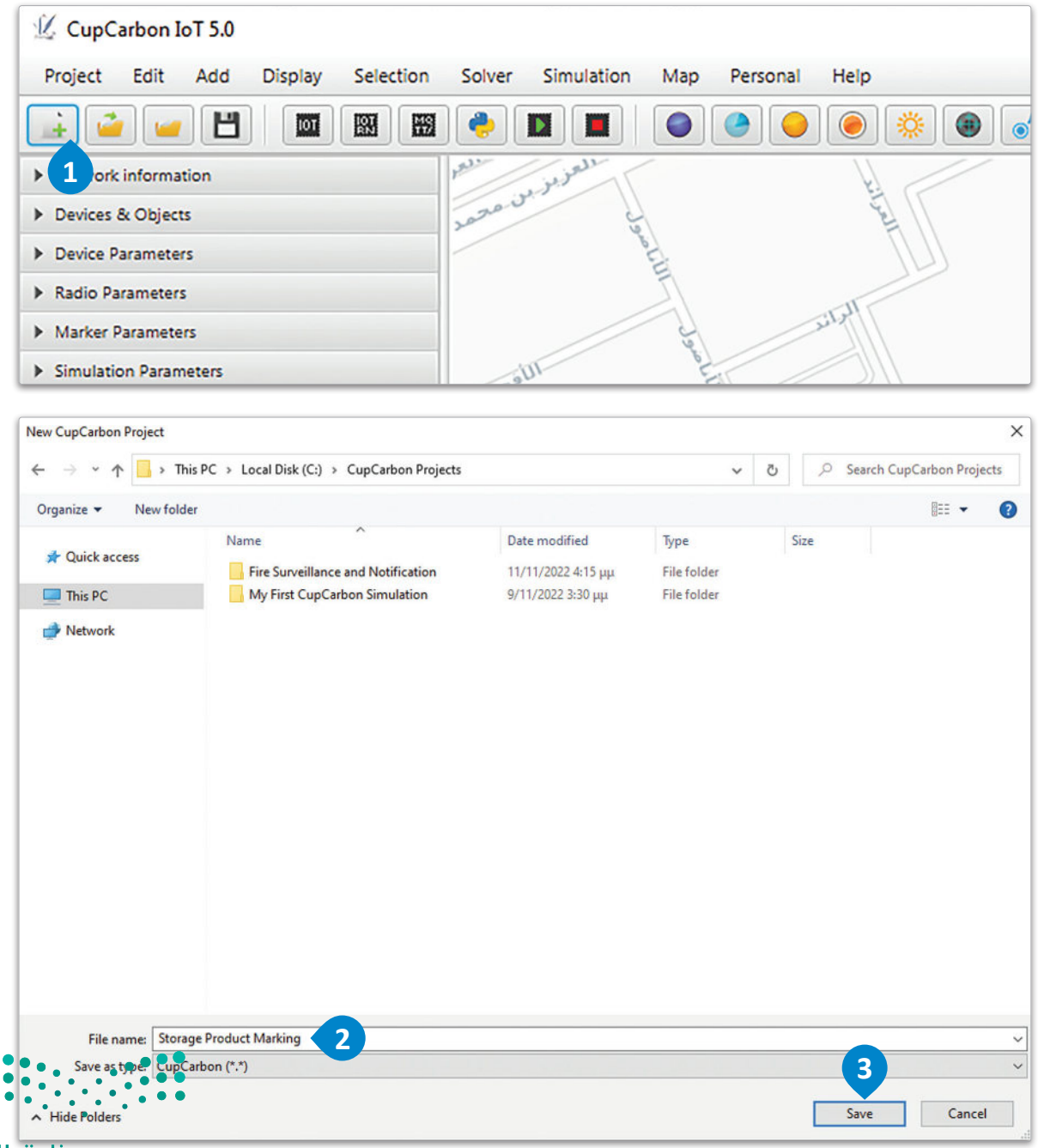
وزارة التعليم
Ministry of Education
2023 - 1445

لنبدأ بإنشاء مشروع جديد:

لإنشاء مشروع جديد:

> اضغط على New Project (مشروع جديد) من Toolbar (شريط الأدوات). 1

> اختر الموقع الذي تريده لحفظ المشروع، اكتب "Storage Product Marking" في حقل File name (اسم الملف)، 2 واضغط على Save (حفظ). 3



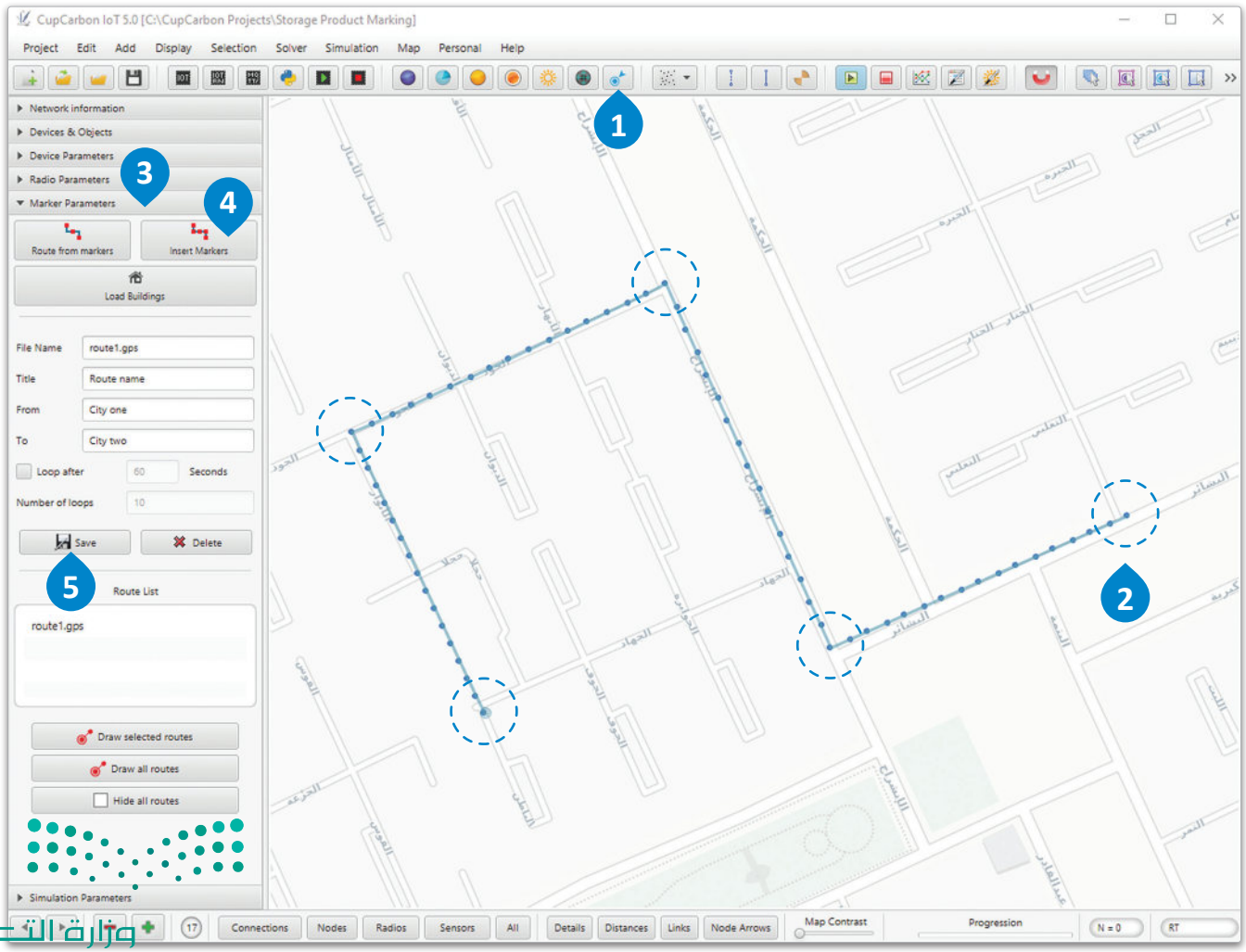
شكل 8.21: إنشاء مشروع جديد

إنشاء مسار مُحدد سابقًا Creating a Predetermined Route

في البداية، ستنشئ المسار الذي ستسير عليه مركبة التفتيش. أولًا، ستضع بعض العلامات على الخريطة لتحديد طبيعة المسار، ثم ستقوم بإضافة بعض العلامات الأخرى لتحديد المسار بشكل دقيق.

لإنشاء المسار:

- اضغط زر Marker (علامة) من Toolbar (شريط الأدوات). 1
- اضغط على الخريطة 5 مرات كما يظهر في الصورة أدناه، مما يُنشئ خطوطًا على طول المسارات. 2
- اضغط على Esc. في لوحة المفاتيح.
- اضغط على علامة تبويب Marker Parameters (معاملات العلامات) في قائمة Parameters (المعاملات). 3
- أضف المزيد من العلامات إلى المسار بالضغط على كل علامة من العلامات الأربعة المضافة، والضغط على Insert Markers (إدراج العلامات) 4 مرات. 4
- اضغط على Save (حفظ). 5



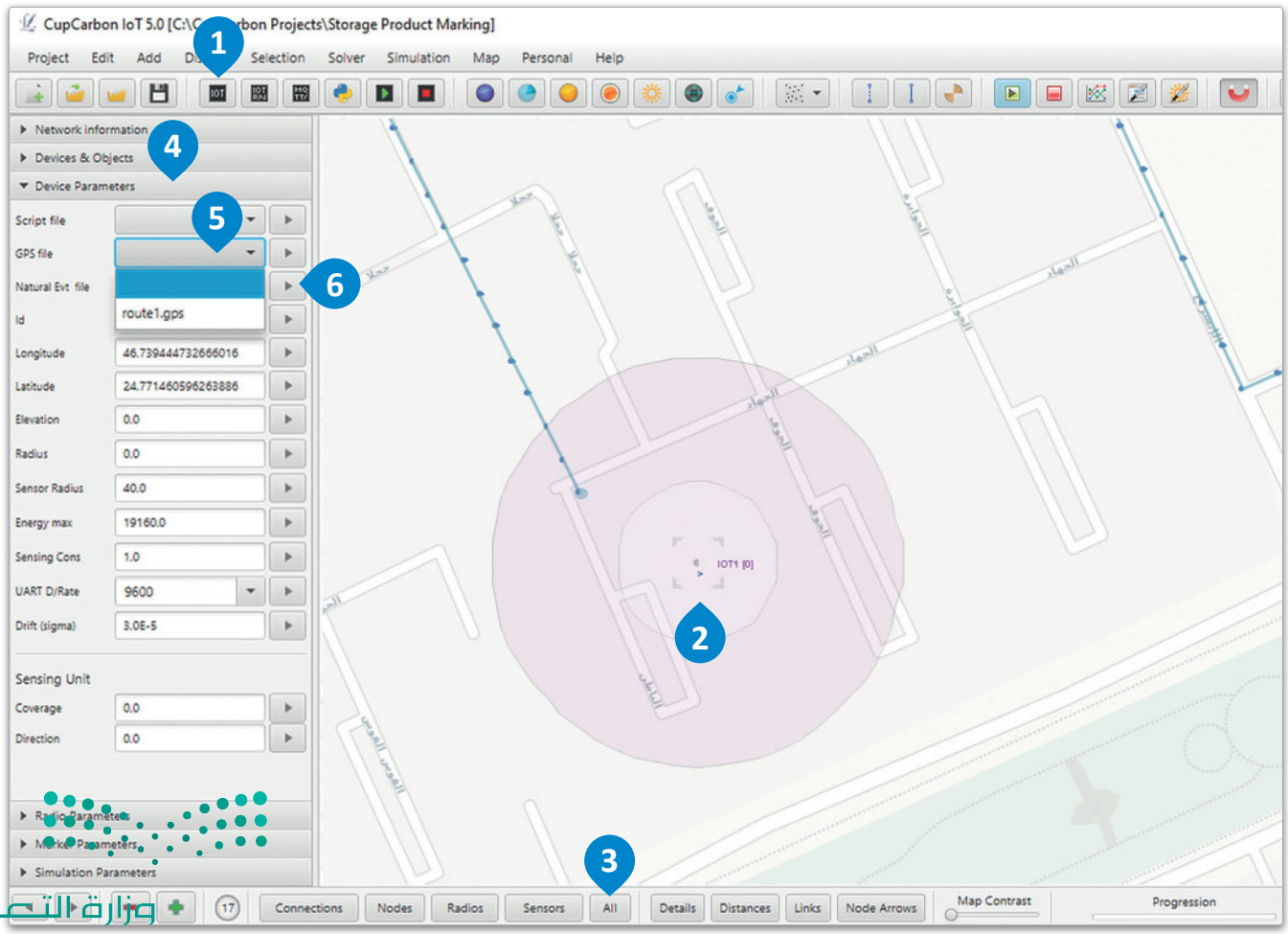
شكل 8.22: إنشاء مسار

إضافة عُقدة مركبة التفتيش Adding the Inspector Vehicle Node

ستتم محاكاة مركبة التفتيش بواسطة عُقدة تتحرك على طول المسار الذي أنشأته سابقًا. ستقوم أيضًا بزيادة نطاق الاستشعار للعُقدة (الدائرة الداخلية) بحيث يمكنها الوصول إلى نطاق محطة الشحن.

لإضافة عُقدة مركبة التفتيش:

- اضغط على IoT Node (عُقدة إنترنت الأشياء) من شريط الأدوات. 1
- اضغط على الخريطة لإضافة العُقدة. 2
- اضغط على All (الكل) من State bar (شريط الحالة). 3
- اضغط على Esc في لوحة المفاتيح.
- اضغط على العُقدة واضغط على الزرين Shift ⇧ + 0 معًا أربع مرات لزيادة نطاق الاستشعار.
- اضغط على علامة التبويب Device Parameters (معاملات الجهاز) في قائمة parameter (معاملات). 4
- اضغط على الصندوق الموجود على يمين ملف GPS. 5
- من القائمة المنسدلة، حدد برنامج route1.gps واضغط على الزر الموجود على اليمين لإدراج المسار في العُقدة. 6



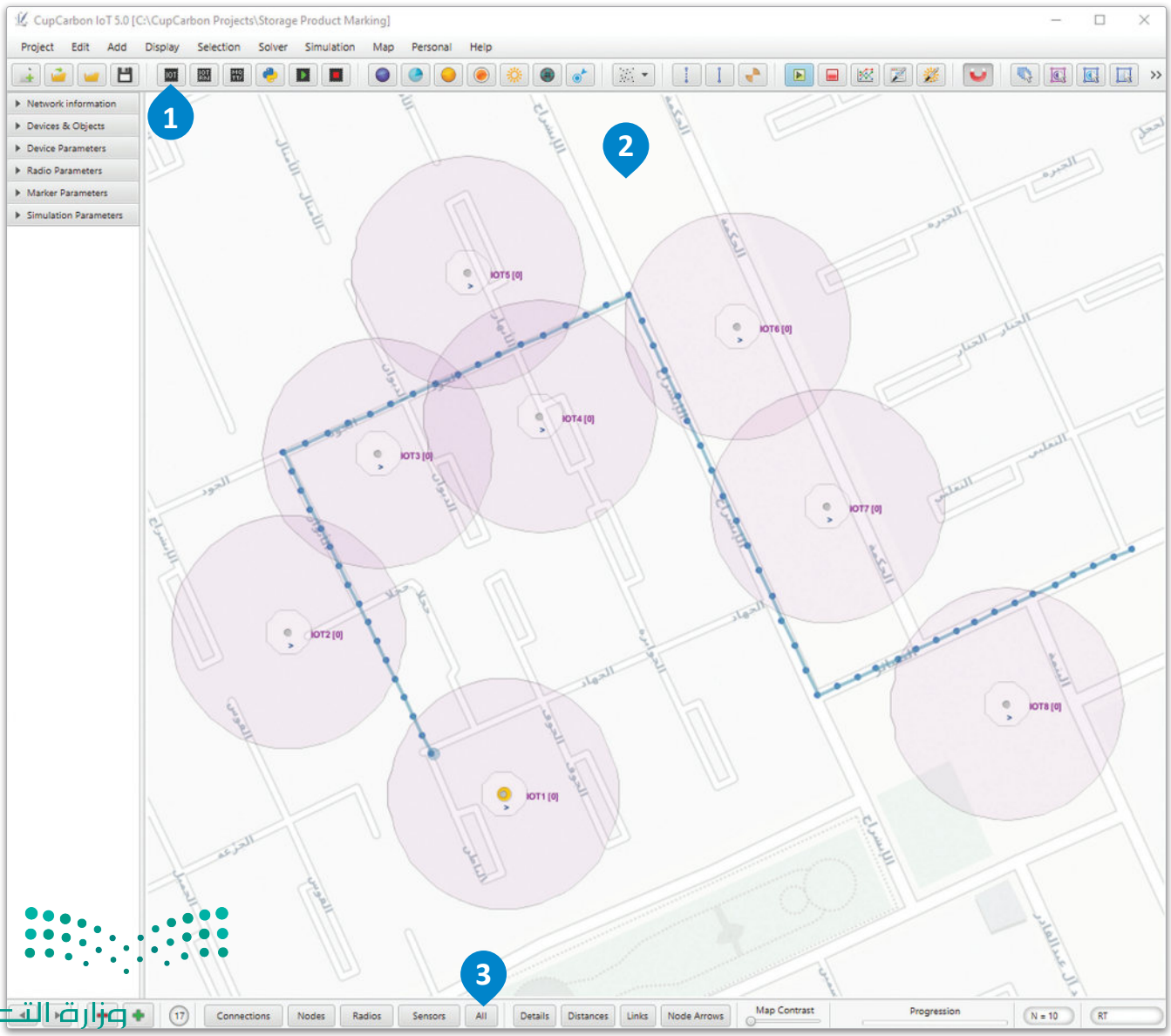
شكل 8.23: إنشاء مسار

إضافة عُقد الحاوية Adding Container Nodes

حان الوقت الآن لإضافة العُقد التي تُمثل الحاويات.

لإضافة عُقد الحاوية :

- اضغط على IoT Node (عُقدة إنترنت الأشياء) من شريط الأدوات. 1
- اضغط على الخريطة وأضف 7 عُقد بالقرب من المسار، بحيث يتضمن نصف قطر كل منها علامة واحدة من المسار على الأقل. 2
- اضغط على All (الكل) من شريط الحالة. 3
- اضغط على زر Esc في لوحة المفاتيح.



شكل 8.24: إنشاء عُقد الحاوية

إضافة محطات الشحن Adding Charging Stations

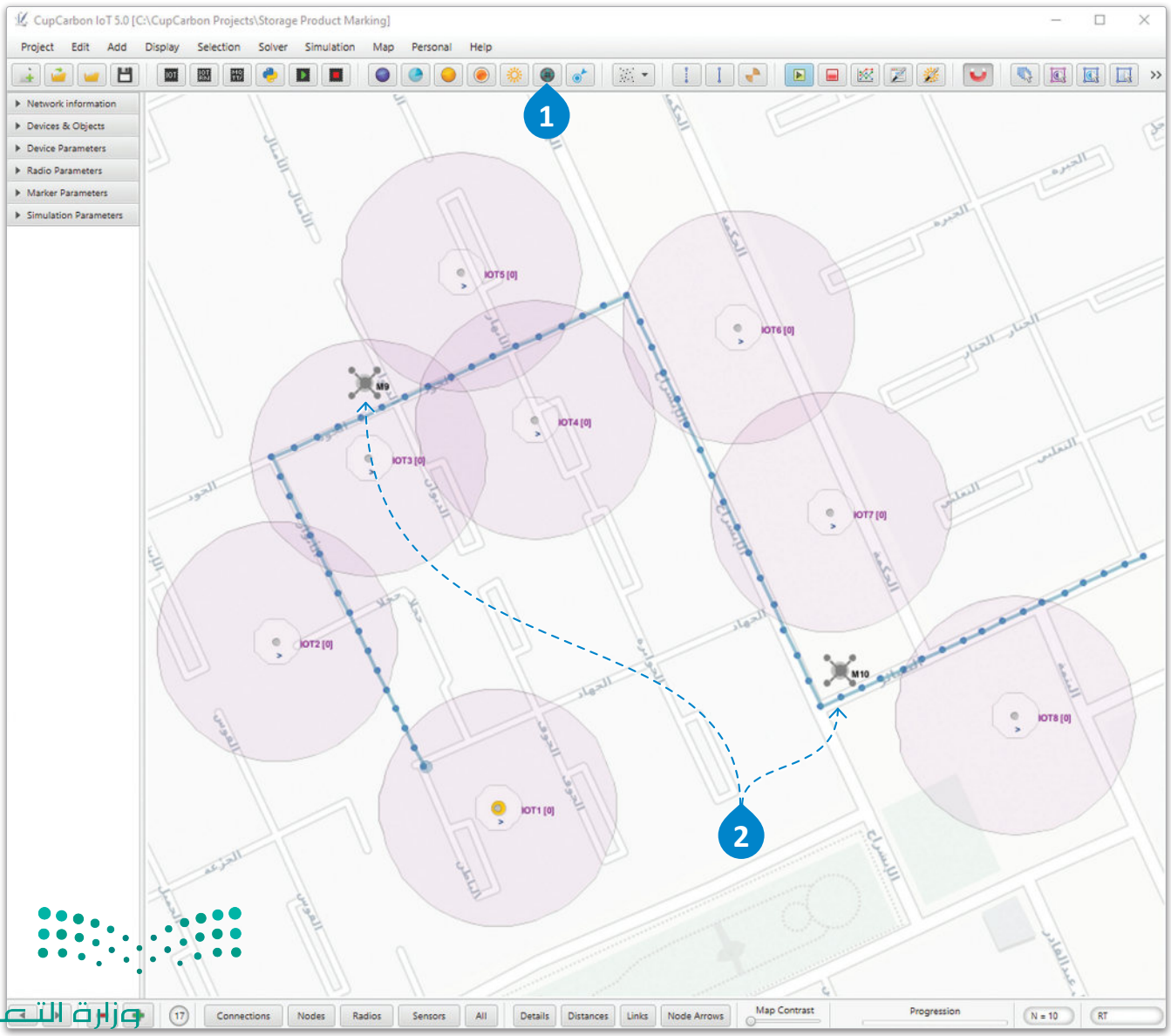
تستهلك مركبة التفتيش الطاقة خلال حركتها في المخزن، مما يتطلب إعادة شحنها. ستقوم بإضافة بعض محطات الشحن على المسار لشحن المركبة أثناء مرورها بقربها، ولهذا الغرض سيتم استخدام نطاق الاستشعار الداخلي.

لإضافة نقاط لمحطات الشَحن:

> اضغط على Mobile (الهاتف المحمول) من Toolbar (شريط الأدوات). 1

> اضغط على map (الخريطة) وأضف عُقدتين على طول المسار بحيث يُمكنهما استشعار المركبة. 2

> اضغط على زر Esc في لوحة المفاتيح.



شكل 8.25: إنشاء عُقد محطة الشحن

إنشاء المقاطع البرمجية Creating the Scripts

ستلقي الآن نظرة على كافة المقاطع البرمجية التي ستستخدمها بِدءًا من برمجة الحاويات. مع العلم بأن برمجة النوعين المختلفين من الحاويات ستكون متطابقة.

ابدأ بإضافة المكتبة اللازمة، وطباعة نص فارغ على العُقدة لإزالة أي نصوص مطبوعة من عمليات التنفيذ السابقة.

```
import time


node.print("")
```

ستقوم حاويات المواد القابلة للتلف ببث رسالة تتضمن محتوياتها ومُعرفها ليُمكن استخدامها من قِبل مركبة التفتيش لتصنيف كل حاوية. المُعرِّف (ID) عدد صحيح، ويجب تحويل نوعه إلى مُتغير نصي قبل ربطه بالنص. تُوضع مسافة بين معلومات المحتويات والمُعرِّف بحيث يمكن إرسال نصٍ واحدٍ فقط في آنٍ واحد باستخدام دالة ()send للإرسال، ثم يتعين عليك إرسال جزئيتين من المعلومات يُفصل بينهما بواسطة المسافة.

```
while node.loop():


    node.send("CONSUMABLES " + str(node.id()))
```

بعد تحليل الحاوية للنص المُستقبل، ستُرسل إما "1" والتي تعني أنه يجب اختيارها، أو ستُرسل "2" والتي تعني أنه يجب ألا يتم ذلك. بدورها، ستطبع الحاوية النص الذاتي "PICK" (التقط) أو "DO NOT PICK" (لا تلتقط)، ثم ستسكُن لمدة ثانية واحدة.

```
    message = node.read()
    if message == "1":
        node.print("PICK")
    elif message == "2":
        node.print("DO NOT PICK")


    time.sleep(1)
```

المقطع البرمجي النهائي (consumples.py)
Complete Code (consumables.py)

```
import time

node.print("")
while node.loop():

  node.send("CONSUMABLES " + str(node.id()))
  message = node.read()
  if message == "1":
    node.print("PICK")
  elif message == "2":
    node.print("DO NOT PICK")

  time.sleep(1)
```

وجه الاختلاف بين البرمجة الخاصة بالمواد القابلة للتلف والمواد طويلة الأمد هو النص المُرسل، ففي المقطع البرمجي الخاص بالمواد طويلة الأمد سيتغير النص من "CONSUMABLES" (قابلة للتلف) إلى "NONCONSUMABLES" (طويلة الأمد).

المقطع البرمجي النهائي (nonconsumables.py)
Complete Code (nonconsumables.py)

```
import time

node.print("")
while node.loop():

  node.send("NONCONSUMABLES " + str(node.id()))
  message = node.read()
  if message == "1":
    node.print("PICK")
  elif message == "2":
    node.print("DO NOT PICK")

  time.sleep(1)
```

هنا المقطع البرمجي الخاص بمركبة التفتيش. في البداية ستُهيّأ البطارية بضبط أقصى طاقة لها لتعادل 100 وحدة طاقة باستخدام الدالة battery.setEMax( )، ثم ضبط مستواها الحالي إلى الحد الأقصى مع دالة battery.init( ).

```
import time


node.battery.setEMax(100.0)
node.battery.init()
```

ستستهلك المركبة بمرور الزمن قدرًا مُعينًا من الطاقة. ولمحاكاة ذلك، استخدم الدالة battery.consume(1.0) لتنفيذ استهلاك وحدة طاقة لكل فترة زمنية محددة.

```
while node.loop():


  node.battery.consume(1.0)
```

لاكتشاف ما إذا كانت أي محطة شحن موجودة في نطاق المركبة، استخدم الدالة isSensorDetecting( )، وعند اكتشاف مُحطة، استخدم battery.init( ) لشحنها إلى الحد الأقصى.

```
  if node.isSensorDetecting():
    node.battery.init()
```

يتعين على المركبة التحقق الآن من جميع الرسائل التي استقبلتها، ثم الرد على مُرسليها (الحاويات). سيتم في البداية تخزين مُتغير القراءة المحلي في recvMsg، ثم باستخدام دالة split( ) سيُفصل النص إلى جزأين وفقًا للمساحة المستخدمة سابقًا، على شكل مصفوفة باسم splitMsg. وهذا يعني أنه في الخلية الأولى من المصفوفة splitMsg [0] سيُحتفظ بمحتويات الحاوية، بينما تحتفظ الخلية الثانية splitMsg [1] بمُعرِّف الحاوية.

```
  for n in range(node.bufferSize()):
    recvMsg = node.read()
    splitMsg = recvMsg.split()
```

إذا كان نص المحتوى "CONSUMABLES"، فسترسل النص "1" بواسطة دالة send( ) إلى حاوية المرسل باستخدام مُعرفها، أما إذا كان نص المحتوى "NONCONSUMABLES" فسيُرسل النص "2". وفي الختام ستسكن لمدة 200 مللي ثانية؛ لأنها تحتاج إلى تحقيق استجابة أكثر من عُقد الحاوية بصفتها تتواصل مع المزيد من العُقد.

```
  if splitMsg[0] == "CONSUMABLES":
    node.send("1", int(splitMsg[1]))
  elif splitMsg[0] == "NONCONSUMABLES":
    node.send("2", int(splitMsg[1]))

  time.sleep(0.2)
```

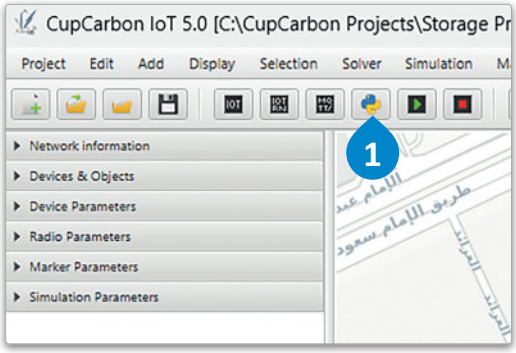
المقطع البرمجي النهائي (inspector.py)

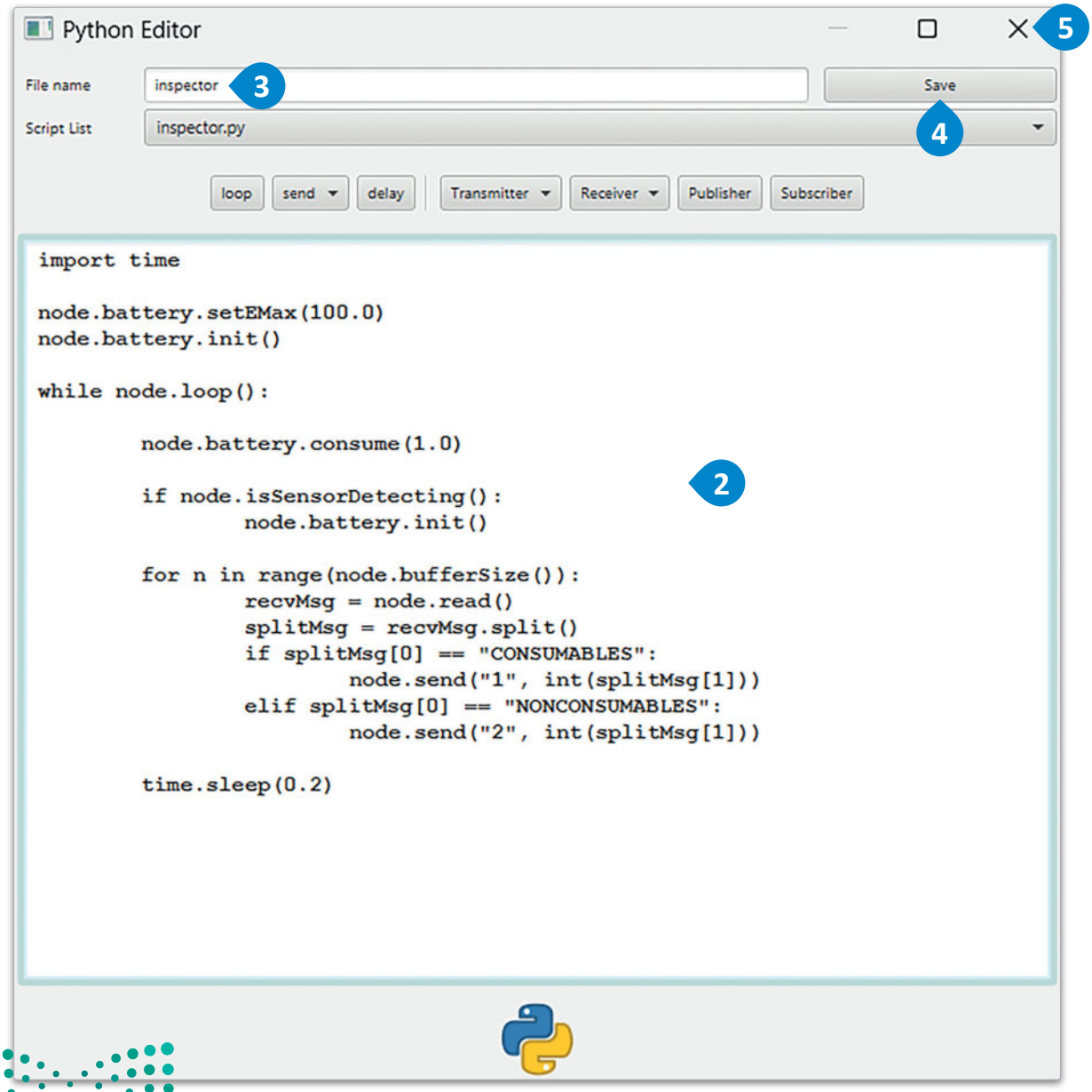
Complete Code (inspector.py)

```
import time

node.battery.setEMax(100.0)
node.battery.init()

while node.loop():

  node.battery.consume(1.0)

  if node.isSensorDetecting():
    node.battery.init()

  for n in range(node.bufferSize()):
    recvMsg = node.read()
    splitMsg = recvMsg.split()
    if splitMsg[0] == "CONSUMABLES":
      node.send("1", int(splitMsg[1]))
    elif splitMsg[0] == "NONCONSUMABLES":
      node.send("2", int(splitMsg[1]))

  time.sleep(0.2)
```

لإنشاء المقطع البرمجي:

> اضغط على Python (بايثون) من شريط الأدوات. 1
> اكتب التعليمات البرمجية في الحقل النصي. 2
> في حقل File name (اسم الملف)، اكتب inspector. 3
> اضغط على Save (حفظ). 4
> أغلق نافذة محرر بايثون النصي. 5






```
import time

node.battery.setEMax(100.0)
node.battery.init()

while node.loop():

        node.battery.consume(1.0)

        if node.isSensorDetecting():
                node.battery.init()

        for n in range(node.bufferSize()):
                recvMsg = node.read()
                splitMsg = recvMsg.split()
                if splitMsg[0] == "CONSUMABLES":
                        node.send("1", int(splitMsg[1]))
                elif splitMsg[0] == "NONCONSUMABLES":
                        node.send("2", int(splitMsg[1]))

        time.sleep(0.2)
```

شكل 8.26: إنشاء المقطع البرمجي

لإدراج المقطع البرمجي:

> اضغط على عُقدة مركبة التفتيش. 1

> اضغط على علامة تبويب Device Parameters (مُعاملات الجهاز) في قائمة Parameters (المُعاملات). 2

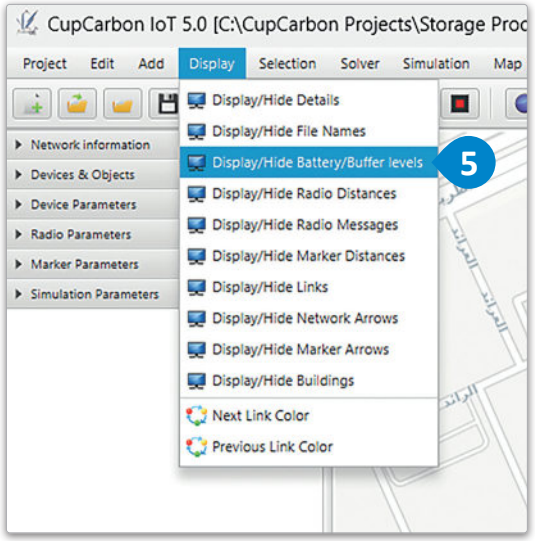
> اضغط صندوق Script file (ملف المقطع البرمجي). 3

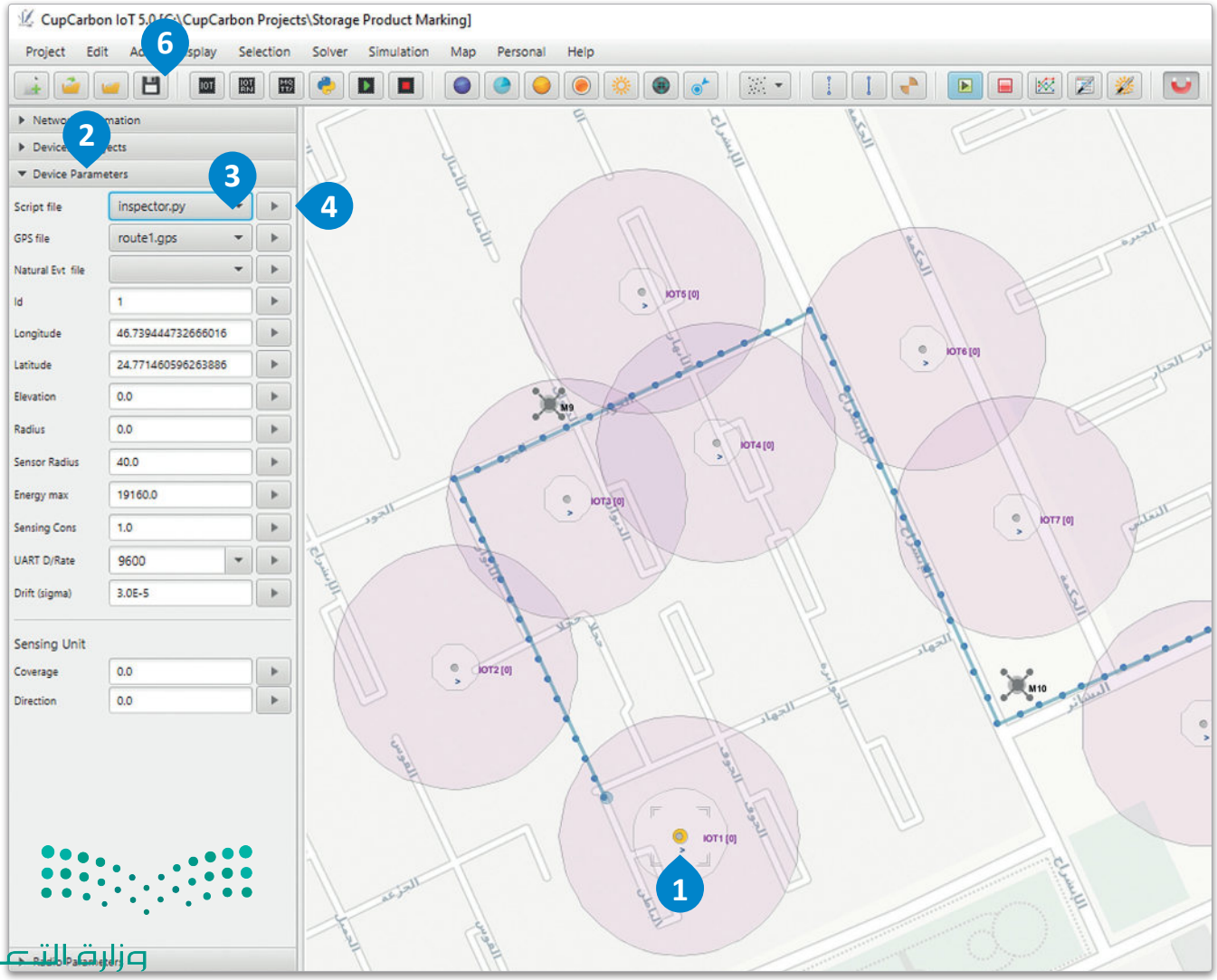
> من القائمة المنسدلة، اختر Inspector.py واضغط على الزر الموجود على اليمين لإدراج المقطع البرمجي في العُقدة. 4

> اضغط على Display > Display/Hide Battery/Buffer levels

> عرض> عرض / إخفاء مستويات البطارية / المخزن المؤقت، من شريط Menu (القوائم). 5

> اضغط على Save Project (حفظ المشروع) من Toolbar (شريط الأدوات). 6

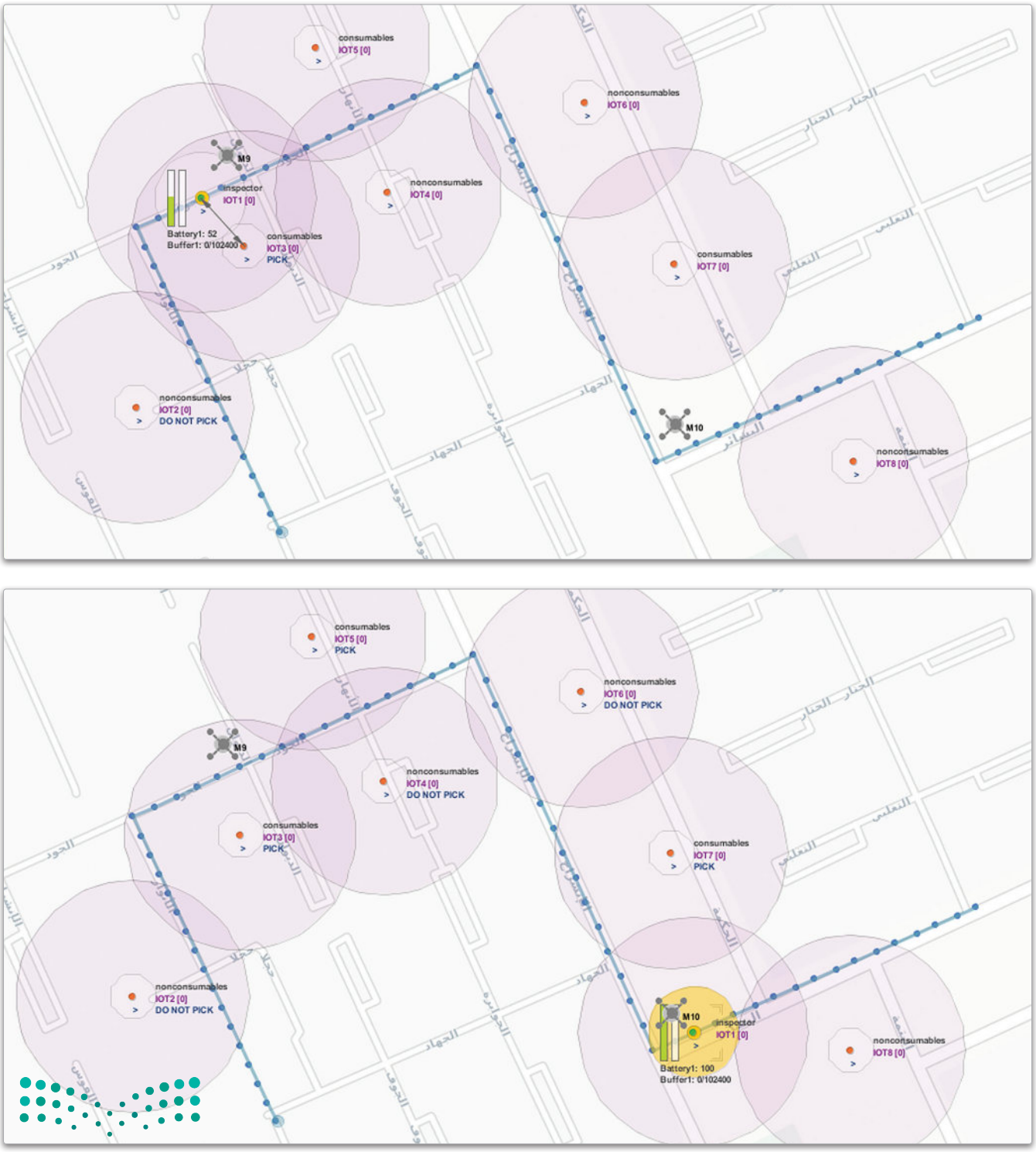




شكل 8.27: إدراج المقطع البرمجي

أنشئ المقاطع البرمجية consumables.py و nonconsumables.py بنفس الطريقة، وطبّق المقطع البرمجي الأول على بعض عُقد الحاويات، والثاني على بقيتها، بحيث تحتوي جميع عُقد الحاوية على أحد هذين المقطعين.

عند الانتهاء، يمكنك الضغط على زر Run IoT Simulation (تشغيل محاكاة إنترنت الأشياء) من شريط الأدوات لبدء المحاكاة.



شكل 8.28: إدراج المقطع البرمجي

تمرينات

1. وسِّع مشروعك بإضافة المزيد من العُقد وإنشاء مسار بالمزيد من العلامات.

   لا تنْسَ إضافة المقاطع البرمجية في العُقد الجديدة.

2. حدِّد ما إذا كان مشروعك يستخدم أقل عدد ممكن من مُحطات الشحن. حاول إزالة محطة، ونقل الأخريات لاختبار فرضيتك.

3. عدِّل البرمجة الخاصة بمركبة التفتيش لكي تستهلك المزيد من الطاقة، ولكي تُستنزف بطاريتها بشكل أسرع. دوِّن نتائجك أدناه.

4. وسِّع مشروعك عن طريق إنشاء نوعٍ ثالثٍ من عُقد الحاوية وهو حاوية فارغة ستُرسل النص Empty (فارغة)، ولن تُحدّد بواسطة مركبة التفتيش.

5. قد يكون لبطء اتصال شبكة المصنع آثار خطيرة على وظائف النظام. عدِّل البرمجة الخاصة بعُقدة مركبة التفتيش لجعل العُقدة تسكُن لمدة أطول. هل حدث تأخير أو فقدان لأي رسائل؟ دوِّن ملاحظاتك أدناه.

المشروع

يُعدّ الاتصال مهمًا جدًا داخل المصنع للتنسيق بين الإدارات والقطاعات المختلفة. عندما يُطبّق الاتصال المتوازي ببراعة، يمكن زيادة الكفاءة والإنتاجية بشكل كبير.

1 ستقوم بمحاكاة نظام توصيل داخل المصنع يتكون من مركبة توصيل تتحرك على مسار مُحدد سابقًا بحيث تُوصل الأجزاء والمواد إلى قطاعات مختلفة. أنشئ شبكة تتكون من وحدة المُتحكم الرئيسة مع 3 عُقد وسطى و3 عُقد طرفية لكل عُقدة وسطى .

2 سوف تمر مركبة التوصيل على كل عُقدة طرفية وتوصل أي أجزاء أو مواد. اكتب مقطعًا برمجيًا للعقد الطرفية لطلب الأجزاء أو المواد عن طريق إرسال نص بهذا الطلب إلى المركبة، واكتب مقطعًا برمجيًا لتزويد المركبة بما طُلب بإعطاء تأكيد.

3 وسِّع مشروعك بحيث تقوم العُقد الطرفية بعد تلقي طلبها بإعادة توجيه الرسالة إلى العُقد الوسطى المقابلة لها بحيث يمكن متابعة الإنتاج. وفي المقابل، ستقوم العُقد الوسطى بإعادة توجيه الرسالة إلى وحدة المُتحكم الرئيسة، والتي ستطبع رسالة إعلامية تفيد بأن طلب القطاع قد تم تلبيته.

4 وسِّع مشروعك ليشمل بطارية في المركبة تُستهلك طاقتها بالكامل في كل تحركٍ لها. أضف عدة محطات شحن عبر الطريق. هل تستخدم أقل عدد مُمكن من محطات الشحن؟

ماذا تعلّمت

- التعرُّف على تقنيات إنترنت الأشياء في الصناعة.
- استخدام برنامج كاب كاربون (CupCarbon) لمحاكاة الشبكات.
- إنشاء مقاطع برمجية بلغة بايثون لبرمجة عُقد الشبكة.
- استخدام بيئة محاكاة كاب كاربون لإنشاء مشاريع إنترنت الأشياء.

المصطلحات الرئيسة

مؤشرات الأداء الرئيسة	Key Performance Indicators
بروتوكول الاتصال مودبس	Modbus
تقنية التشغيل	Operational Technology
الصناعة الذكية	Smart Industry

المصانع المتصلة	Connected Factory
التصنيع المعتمد على البيانات	Data-Driven Manufacturing
الرقمنة	Digitization
حوسبة طرفية	Edge Computing
أنظمة الأتمتة والتحكم الصناعية	Industrial Automation and Control Systems

ملاحظات