



میکر

دلیل

دلیل

میکر

محتوى الدليل



البدء باستخدام لوحة ابتكار ميكرو 4

- 4..... مقدمة حول وحدات التحكم الدقيقة
- 6..... برمجة وحدة التحكم الدقيقة
- 8..... ابتكار ميكرو

أنشطة ميكرو 42

- 42..... مصباح LED المضمن باللوحة
- 44..... شبكة مصابيح LED
- 55..... قراءة الأزرار
- 59..... قراءة درجة الحرارة
- 60..... قراءة شدة الضوء المحيط
- 61..... جهاز الإنذار
- 65..... قراءة مستوى الصوت
- 66..... لوحات المنافذ
- 69..... وحدات NeoPixels
- 74..... مقياس التسارع ثلاثي المحاور

إنشاء برنامجك الأول 28

- 28..... واجهة أردوبلوكلي
- 30..... أقسام العناصر البرمجية
- 37..... إنشاء، وحفظ، وتحميل مشاريعك
- 38..... برمجة لوحة ابتكار ميكرو



ibtikar[®]
EDU TECH SOLUTIONS

دليل ميكرو

ISBN: 978-9948-24-388-0

تأليف: مهند التكروري

حقوق الطبع والنشر محفوظة © لعام 2018 لصالح شركة ابتكار لحلول تكنولوجيا التعليم.

جميع الحقوق محفوظة. لا يجوز نسخ أي جزء من هذا المنشور أو توزيعه بأي شكل أو بأي وسيلة، أو تخزينه في قاعدة بيانات أو نظام استنساخ، دون الحصول على موافقة خطية مسبقة من شركة ابتكار لحلول تكنولوجيا التعليم، بما في ذلك، على سبيل المثال لا الحصر، التخزين أو النقل عبر شبكة الويب، أو البث عن بُعد للتعليم، باستثناء حالات الاقتباسات القصيرة المسردة في المقالات أو المراجعات النقدية. حقوق حصرية لشركة ابتكار لحلول تكنولوجيا التعليم للتصنيع والتصدير. لا يمكن إعادة تصدير هذا الكتاب من الدولة التي تم بيعه فيها من قبل ابتكار لحلول تكنولوجيا التعليم.

إذا كانت النصائح والمعلومات الواردة في هذا الكتاب صحيحة ودقيقة في تاريخ النشر، فلا يتحمل المؤلف ولا الناشر أي مسؤولية قانونية عن أي أخطاء أو سهو وجد في الكتاب. ولا يقدم الناشر أي ضمان، صريحًا أو ضمنيًا، بشأن المواد الواردة في هذا الدليل. ولن تتحمل شركة ابتكار لحلول تكنولوجيا التعليم وتجارها وموزعيها أي مسؤولية عن أي أضرار ناجمة أو مزعومة بشكل مباشر أو غير مباشر في هذا الدليل. تسعى ابتكار لحلول تكنولوجيا التعليم إلى لتوفير المعلومات بشأن العلامات التجارية للشركات والمنتجات المذكورة في هذا الدليل عن طريق الاستخدام المناسب للأصول. ومع ذلك، لا يمكن لشركة ابتكار لحلول تكنولوجيا التعليم ضمان دقة تلك المعلومات.

البدء باستخدام لوحة ابتكار ميكرو

إذا كان النظام يحتوي على وحدة إدخال لاستشعار البيئة المحيطة، ووحدة تحكم لمعالجة الإشارات المستقبلية ووحدة إخراج لإرسال المعلومات أو للتحكم في جهاز الإخراج، فحينئذٍ يمكن تسميته بالنظام المدمج.

أين يمكنك العثور على الأنظمة المدمجة؟

تتحكم الأنظمة المدمجة في العديد من أجهزة الاستخدام اليومي الشائعة. ويمكنك العثور على وحدات التحكم الدقيقة في الغسالات وأفران الميكروويف والسيارات والمساعد وغيرها من الأجهزة الذكية.

فيما يلي مثال على النظام المدمج، وهو وحدة مكيف الهواء التي تستخدمها في منزلك وفي سيارتك. يقوم مكيف الهواء بتبريد الهواء عن طريق تبديد الحرارة. وتعد وحدة التحكم الرئيسية في وحدة مكيف الهواء هي نظام كمبيوتر مدمج يستشعر درجة حرارة الغرفة أو السيارة، ويقارنها بدرجة الحرارة المطلوبة التي تختارها، ثم يتحكم في عملية التبريد لضبط درجة الحرارة.

مقدمة حول وحدات التحكم الدقيقة

التعريف

وحدة التحكم الدقيقة (microcontroller) عبارة عن كمبيوتر صغير بشريحة واحدة. على عكس الكمبيوتر الشخصي الذي يمكنه القيام بمهام مختلفة في آن واحد، يمكن أن تقوم وحدة التحكم الدقيقة بإجراء مهمة واحدة فقط في كل أمر يتم إرساله إليها.

عادةً ما يتم تضمين وحدة التحكم الدقيقة داخل النظام، يمكنك اعتبارها بمثابة عقل النظام.

هل تعلم؟
الكلمة "ميكرو" (micro) تعني صغير جدًا.

أنواع البرمجة

يمكن برمجة وحدات التحكم الدقيقة باستخدام واجهة برمجة مرئية أو نصية. في البرمجة المرئية، يمكنك استخدام عناصر رسومية لإنشاء البرامج. ويمكنك أيضًا سحب عناصر البرنامج وإفلاتها، والنقر فوقها، واستخدام القوائم، والنماذج، ومربعات الحوار وغيرها. ويوجد خلف كل عنصر من برنامجك عشرات أو حتى مئات من أسطر التعليمات البرمجية. ويساعد هذا النوع من البرمجة المبرمجين الجدد على فهم البرمجة بسهولة.

يعتمد النوع الآخر من البرمجة على النص. وفي البرمجة النصية، يجب مراعاة القواعد اللغوية. يمكنك قص البرامج النصية ونسخها ولصقها مما يمنحك المزيد من المرونة مقارنة بسحب عنصر رسومي واحد وإفلاته في كل مرة في الطريقة الأخرى.

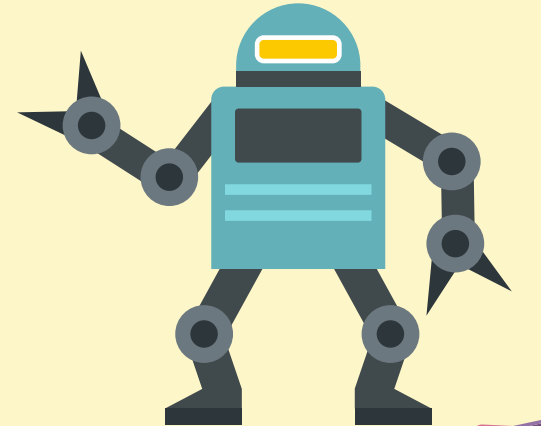


برمجة وحدة التحكم الدقيقة

ما المقصود بالبرنامج؟

البرنامج عبارة عن مجموعة من الأوامر والتعليمات البرمجية التي يمكن إرسالها إلى الكمبيوتر أو وحدة التحكم الدقيقة للقيام بمهمة محددة. وتتميز أجهزة الكمبيوتر باستخدام لغة خاصة تتكون من الرقمين 0 و 1. ويصعب علينا كبشر الكتابة بهذه اللغة. ولهذا السبب، أنشأ المبرمجون لغات عالية المستوى تسمح لنا بكتابة برامج الكمبيوتر بلغة تشبه اللغة التي نفهمها، ثم يتم استخدام محول

برمجي (compiler) لتحويل البرنامج إلى لغة الكمبيوتر، مما يسمح لنا بتعديل البرامج المكتوبة وفهمها بسهولة وسلاسة.





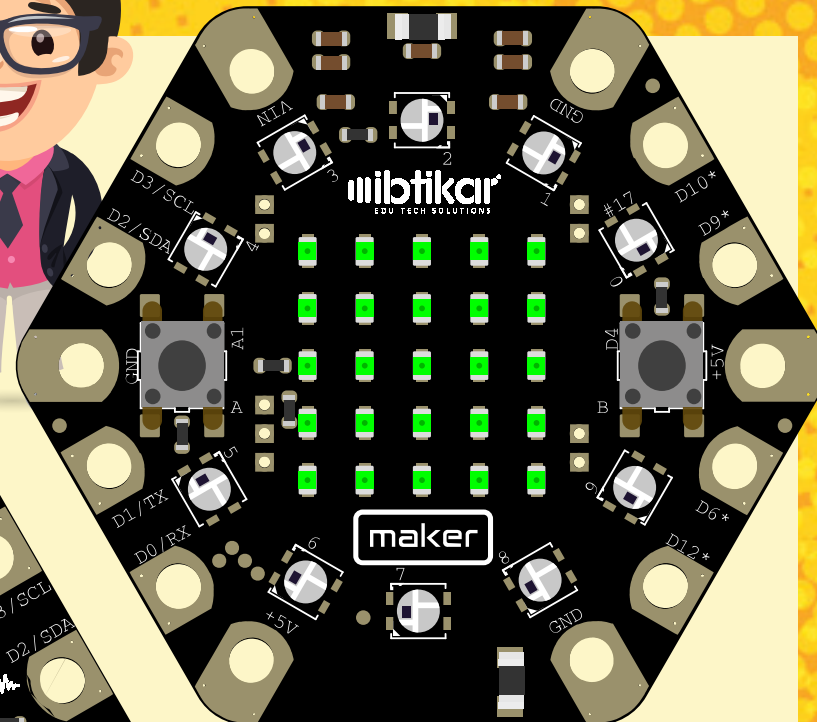
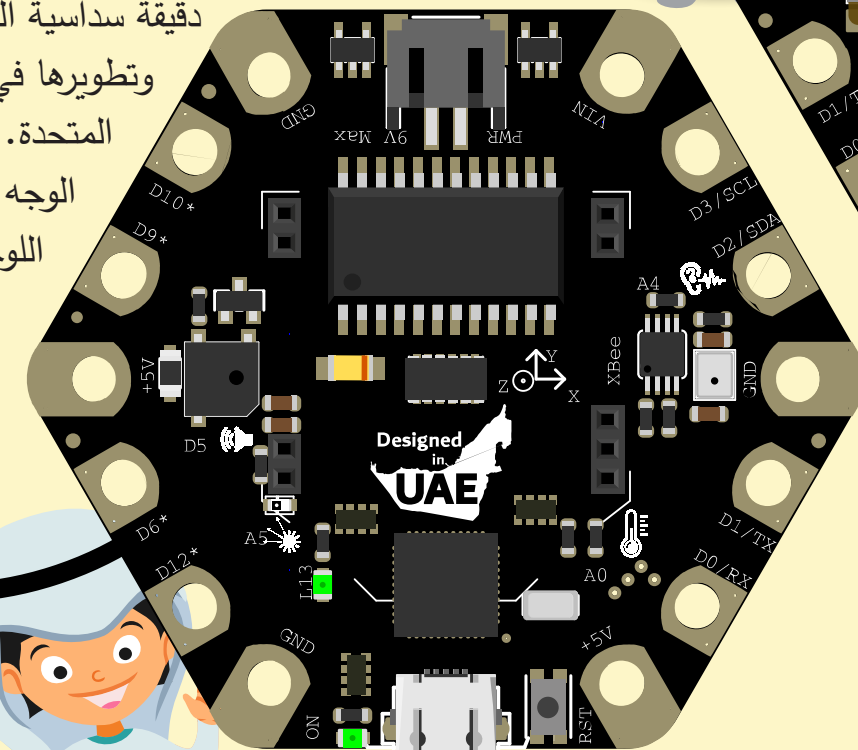
الميزات

تحتوي لوحة ميكرو على الميزات التالية:

- ← 25 مصباح LED على شكل شبكة
- ← 10 مصابيح NeoPixel RGB LED
- ← 2 زر ضغط (على الجانب الأيسر والأيمن)
- ← مستشعر درجة الحرارة
- ← مستشعر الضوء المحيط
- ← مستشعر الصوت
- ← مكبر صوت صغير (جهاز إنذار صوتي مغناطيسي)
- ← مقياس التسارع ثلاثي المحاور
- ← 8 منافذ إدخال/إخراج، حيث تعمل 7 منافذ منها كوحدات إدخال باللمس
- ← مقبس XBEE للسماح بإضافة شبكة Wi-Fi أو البلوتوث
- ← وحدة تحكم متوافقة مع آردوينو

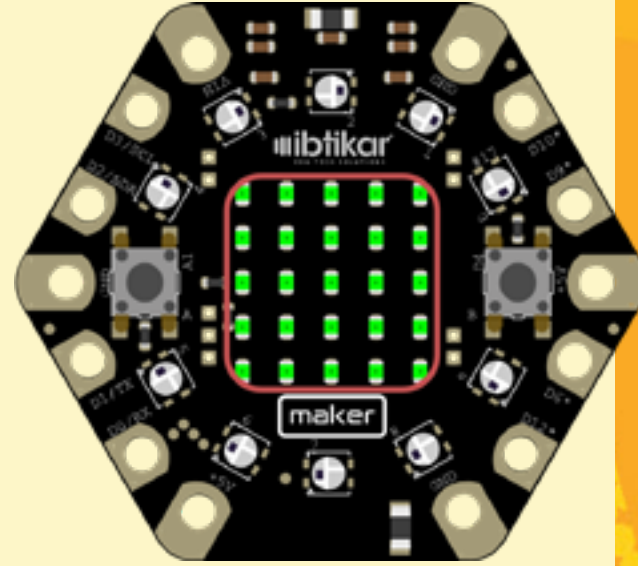
ابتكار ميكرو الجهاز الحجم

إن لوحة ابتكار ميكرو (Ibtikar Maker) هي لوحة تحكم دقيقة سداسية الشكل، تم تصميمها وتطويرها في دولة الإمارات العربية المتحدة. يوضح الشكلان التاليان الوجه الأمامي والخلفي من اللوحة.



1. شبكة مصابيح LED

يوجد 25 مصباحًا يمكنك تشغيلها أو إيقاف تشغيلها. ويمكن التحكم في كل مصباح LED بشكل فردي، مما يسمح لك بإنشاء أنماط متعددة، على سبيل المثال، يمكنك عرض الأحرف أو الأرقام أو النصوص أو الرموز التعبيرية أو أي نمط آخر تفضله.



هل تعلم؟
أن أحرف (LED) هي اختصار للمصطلح صمام ثنائي باعث للضوء (Light Emitting Diode). يختلف مصباح LED عن المصباح الكهربائي التقليدي الذي اخترعه توماس أديسون، حيث لا يوجد به أسلاك خاصة تولد الضوء عند مرور الكهرباء خلالها. وتستخدم مصابيح LED مواد أشباه الموصلات المتقدمة، وهي نفس المادة الموجودة داخل رقائق الكمبيوتر. وتعد هذه المصابيح أفضل من مصابيح الإضاءة التقليدية حيث تدوم لفترة أطول، وتتميز بقوة أكثر ومعدل استهلاك أقل بكثير للطاقة.



الجانب الأمامي والخلفي

قبل البدء باستخدام لوحة ميكرو، من المهم أن تعرف أين يوجد كل عنصر وما الغرض منه. توجد المكونات التالية في الجهة الأمامية من لوحة ميكرو:

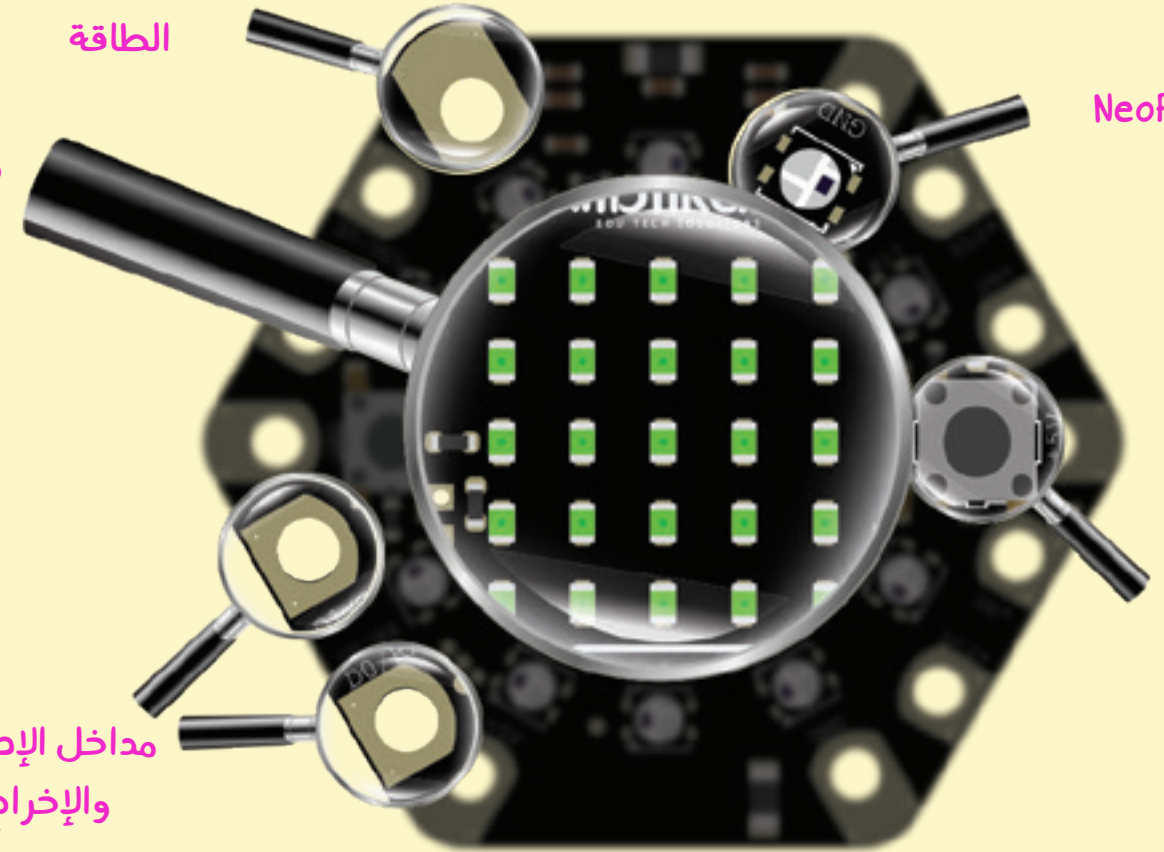
مداخل الطاقة

شبكة مصابيح LED

وحدات NeoPixel

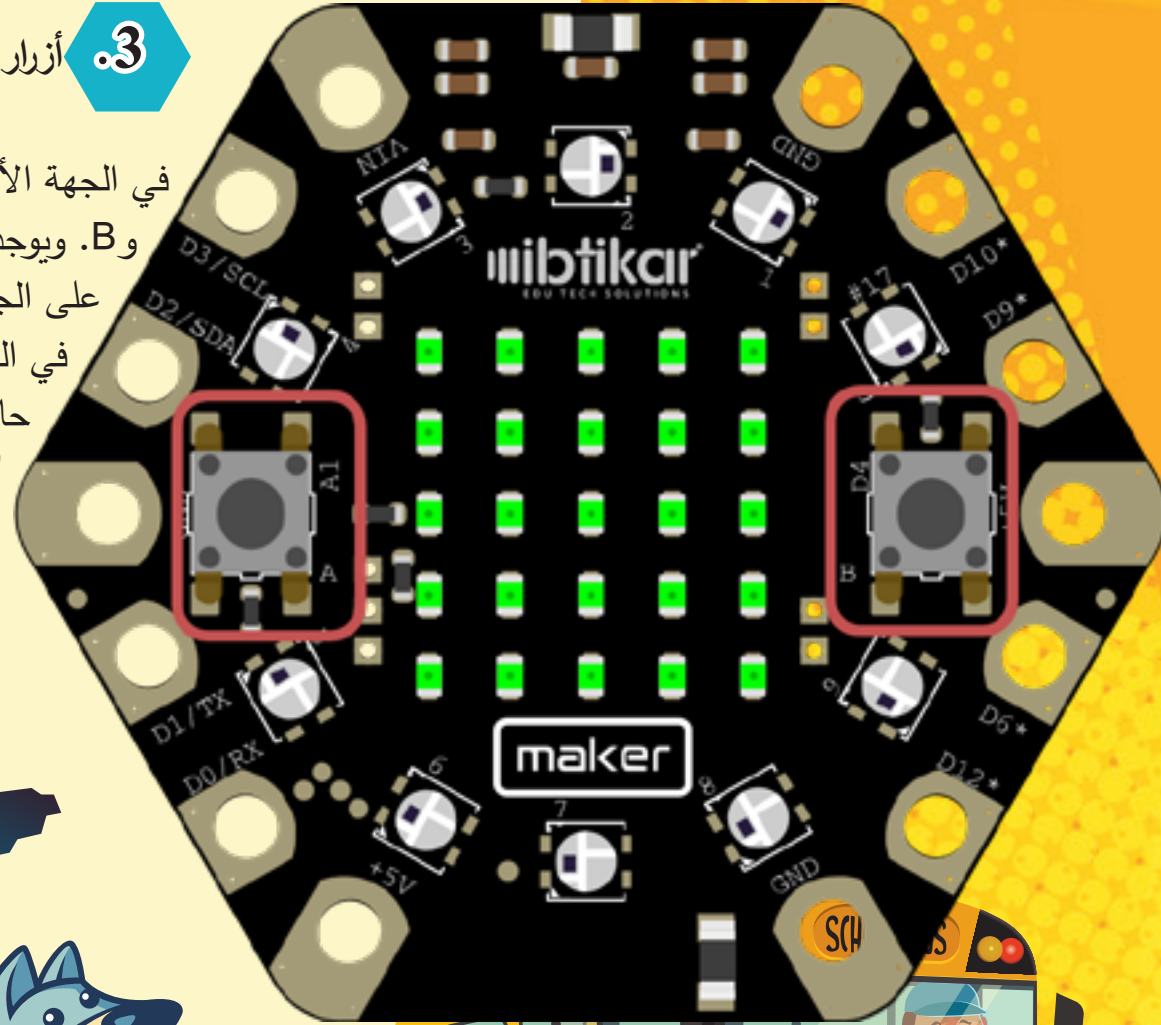
أزرار الضغط

مداخل الإدخال والإخراج



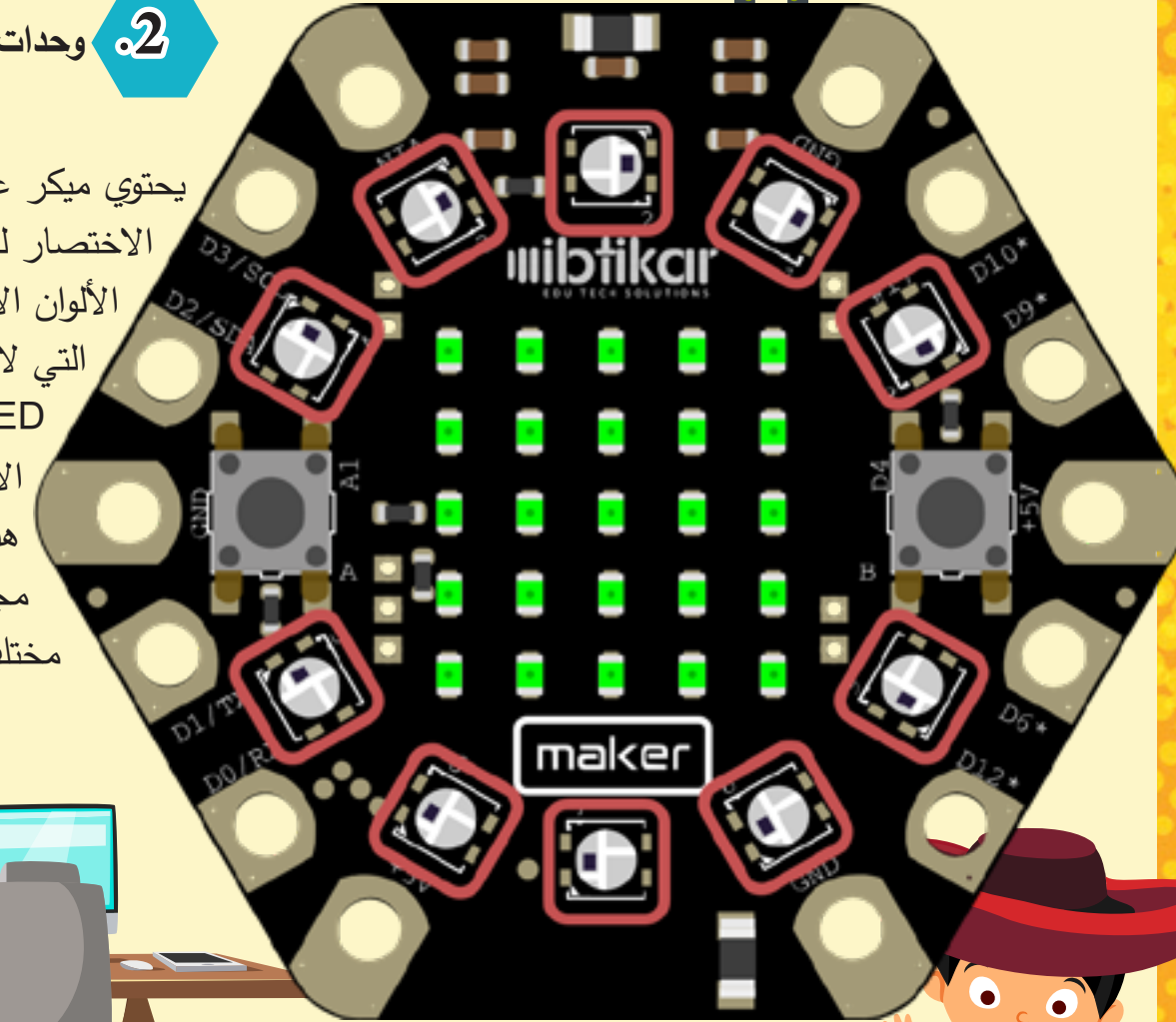
3. أزرار الضغط

في الجهة الأمامية اللوحة، يوجد زران من أزرار الضغط، هما A وB. ويوجد الزر A على جانب اللوحة الأيسر، بينما الزر B على الجانب الأيمن. وهذان الزران هما أدوات إدخال للمستخدم في اللوحة. ويعني هذا بأنه يمكنك برمجة اللوحة لمعرفة حالة الزر إذا تم الضغط عليه أو تحريره. وكلمة "مضغوط" تعني رقم 1 وكلمة "محرر" تعني رقم 0.



2. وحدات NeoPixels

يحتوي ميكرو على 10 مصابيح RGB LED. وتمثل RGB الاختصار لكلمات **أحمر**، **أخضر**، و**أزرق**، والتي تمثل الألوان الأساسية. وعلى عكس المصابيح الخمسة والعشرون التي لا تستطيع تغيير لونها، يمكن برمجة مصابيح RGB LED لعرض أي لون من خلال الجمع بين الألوان الأساسية الثلاثة. تخيل كل مصباح من هذه المصابيح مثل ثلاثة مؤشرات ضوئية صغيرة مجتمعة، ولكل مصباح من هذه المصابيح لون مختلف (**أحمر**، أو **أخضر** أو **أزرق**).





1. مستشعر درجة الحرارة

جهاز استشعار تناظري يقيس درجة حرارة اللوحة.

2. مستشعر الضوء

جهاز استشعار تناظري يقيس شدة الإضاءة المحيطة. يعطي مستشعر الضوء التناظري القيم بين 0 و1023.

3. مكبر الصوت

جرس مغناطيسي يمكن أن يُصدر نغمات صوتية. ويمكنك من خلاله التحكم في التردد الوارد إلى الجرس، وإنشاء نغمات صوتية مختلفة.



في الجهة الخلفية للوحة، توجد المكونات التالية:

مقياس التسارع

مكبر صوت صغير

مستشعر الضوء

مصباح LED

مقبس USB صغير

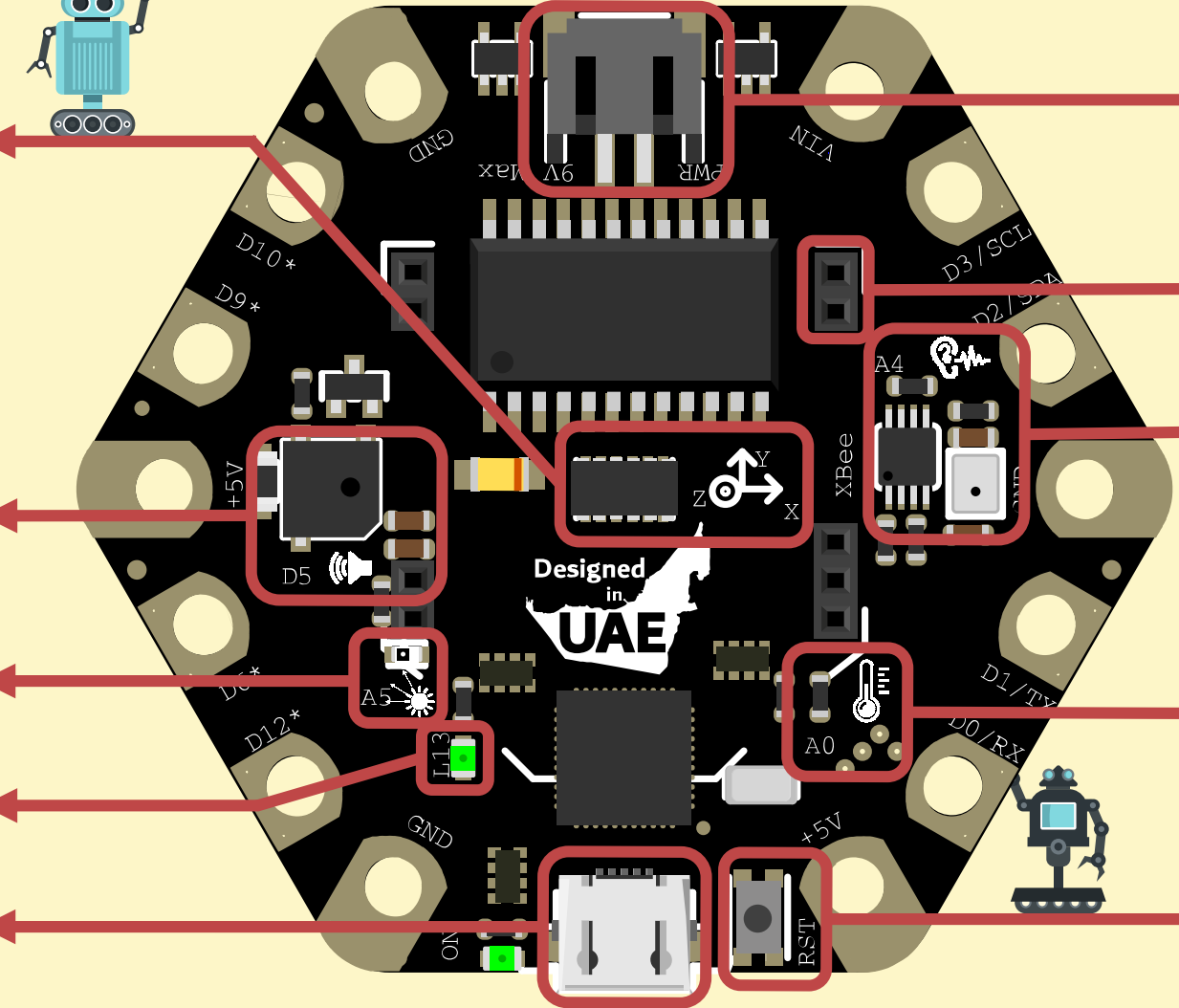
مدخل البطارية

مقبس XBEE

مستشعر الصوت

مستشعر درجة الحرارة

زر إعادة الضبط



4. مستشعر الصوت

جهاز استشعار تناظري يقيس شدة الصوت. يمكنك استخدامه لاكتشاف ما إذا كان هناك صوت/تصفيق بالقرب من اللوحة.

5. مقياس التسارع ثلاثي المحاور

مستشعر مثبت في منتصف اللوحة. ويتم استخدامه لقياس التسارع، وهو المعدل الزمني لتغير السرعة. يمكن لهذا الجهاز قياس التسارع الثابت مثل الجاذبية للكشف عن درجة ميل سطح معين مثل ما يحدث عند إمالة هاتفك المحمول.

يمكن لمقياس التسارع أيضًا اكتشاف التسارع الديناميكي: بدء أو توقف مفاجئ للتسارع.

ملاحظة

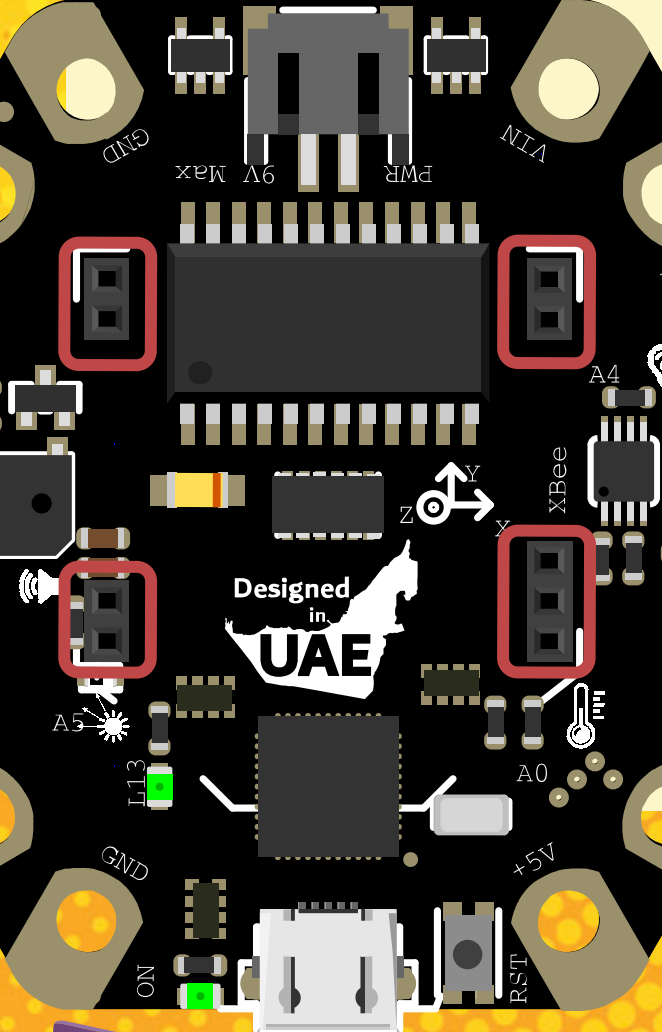
يمكن لمقياس التسارع في ميكرواستشعار المحاور الثلاثة (X و Y و Z). ويتم تحديد الاتجاهات الموجبة والسالبة لكل محور بالنظر إلى الرسم ثلاثي المحاور الصغير الموجود بجوار المستشعر.

6. مقابس XBEE

تسمح هذه المقابس بإضافة شبكة Fi-Wi أو البلوتوث. وهذا يفيد إذا كنت تريد التحكم في لوحة ميكرو لاسلكيًا أو القراءة من خلالها. ويمكنك استخدام هاتفك أو جهازك اللوحي، على سبيل المثال، للاتصال بلوحة ميكرو.

ملاحظة

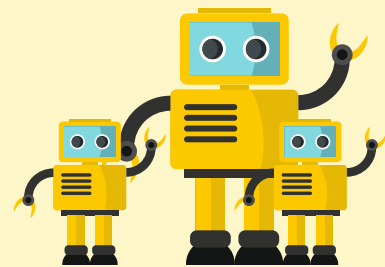
وحدة شبكة Wi-Fi/البلوتوث غير مقدمة مع المجموعة ويتم شراؤها بشكل مستقل.





10. مصباح LED الخاص بالطاقة

يتم تشغيل هذا المصباح عند تشغيل ميكرو.



10



9. مقبس USB صغير

يسمح لك هذا المقبس بتوصيل لوحة الميكرو بالكمبيوتر باستخدام كابل USB، ويمكنك بعد ذلك برمجة اللوحة وإرسال البيانات واستقبالها وإمداد اللوحة بالطاقة.

11. مدخل البطارية

يعمل هذا المدخل كمصدر خارجي للطاقة. ويمكنك استخدامه لتشغيل اللوحة عند عدم توصيل كابل USB. ويمكن أن يكون هذا مفيداً للغاية خصوصاً عند عدم رغبتك في إبقاء مشروعك متصلاً بجهاز الكمبيوتر طوال الوقت.

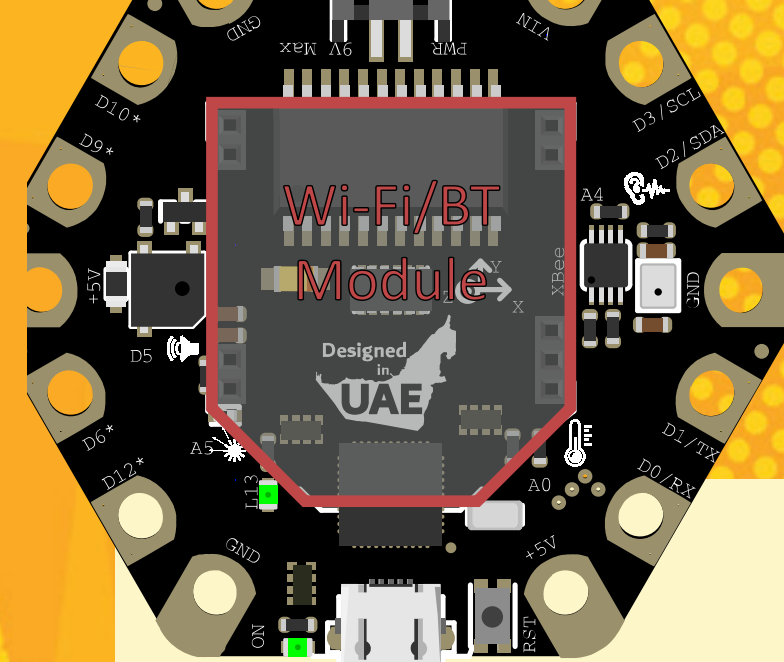
11

8. زر إعادة الضبط

يتم من خلال هذا الزر إعادة تشغيل اللوحة أو إعادة ضبطها.



8



يوجد بجانب كل مقبس بعض الخطوط البيضاء الدقيقة المرسومة على اللوحة. وتساعدك هذه الخطوط على توصيل وحدة شبكة Wi-Fi/البلوتوث في الاتجاه الصحيح، كما هو موضح في الصورة.

7. مصباح LED

يمكن التحكم في هذا المصباح بواسطة المستخدم وهو متصل بالمنفذ رقم 13.



بالإضافة إلى الميزات المذكورة أعلاه، يوجد أيضًا 14 منفذًا على الحافة الخارجية للوحة ميكرو. ويمكن الوصول إلى هذه المنافذ إما من الجهة الأمامية أو الخلفية من اللوحة وتنقسم إلى فئتين.

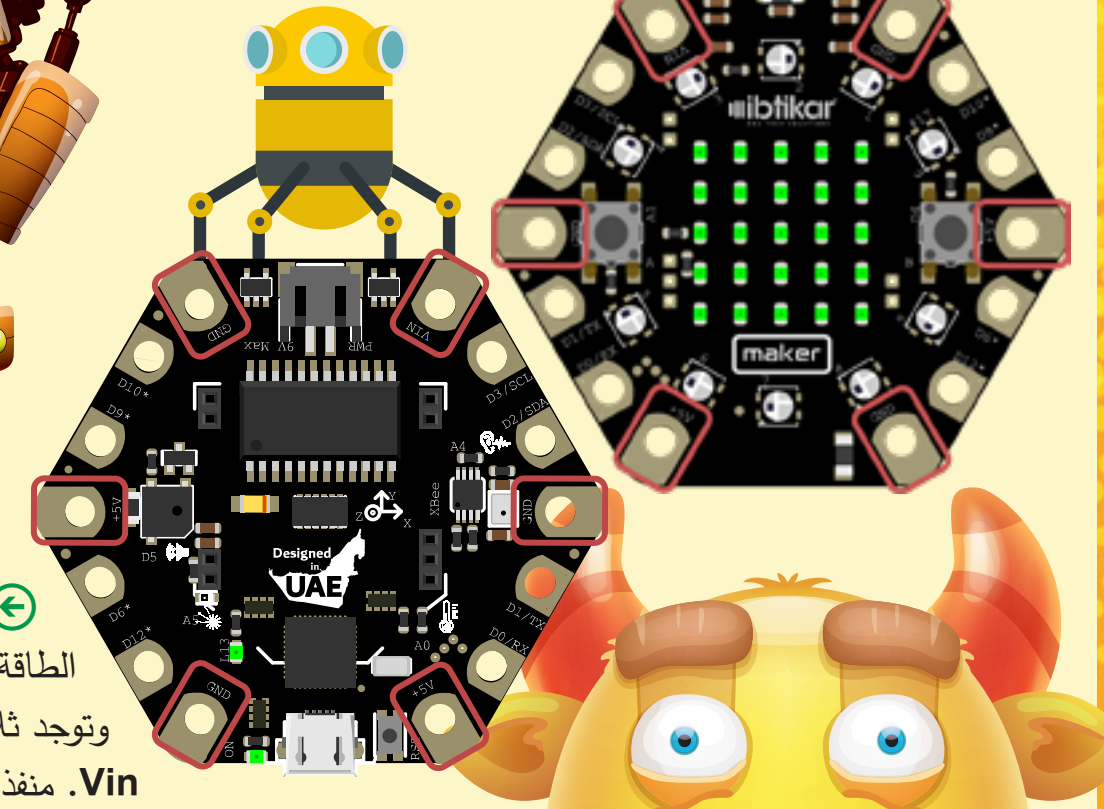
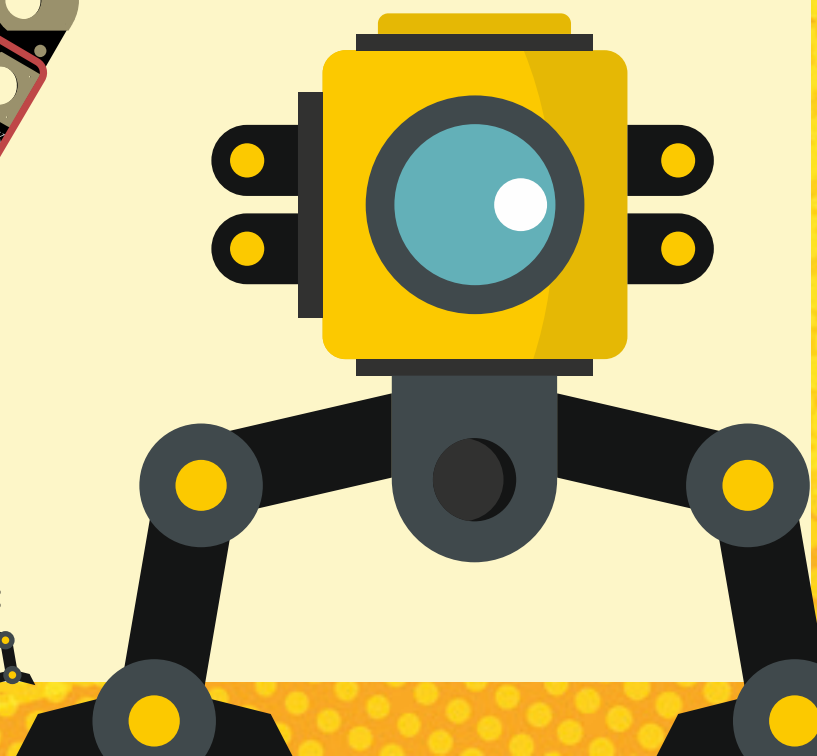
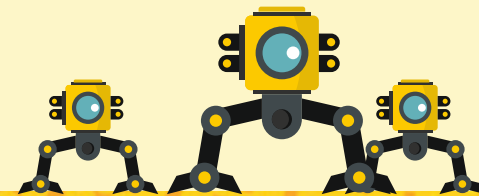
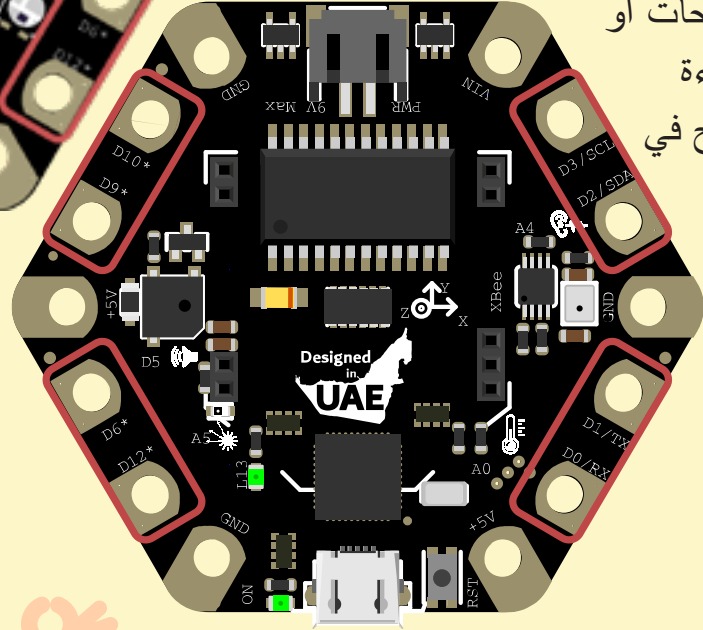
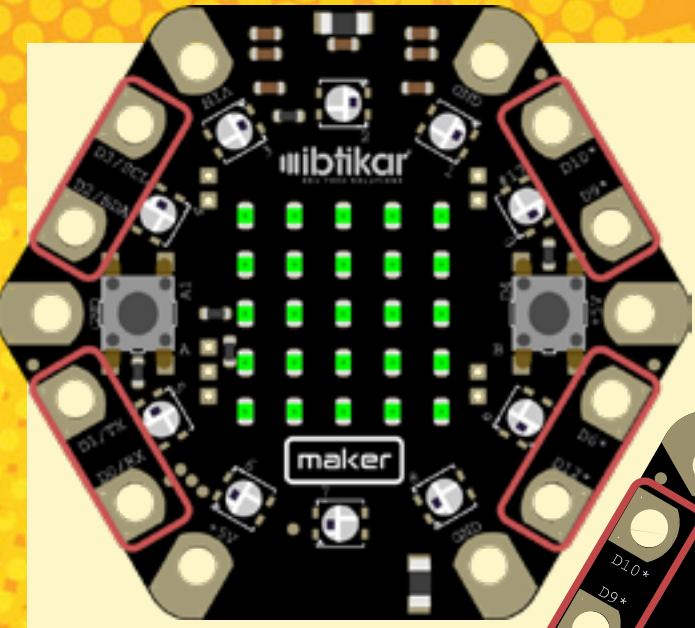
ملاحظة

يجب أن يكون الجهد الكهربائي من 6 فولت إلى 9 فولت. وقد يؤدي تزويد اللوحة بجهد مختلف عن ذلك الموصى به إلى حدوث عطل في اللوحة أو الإضرار بها.

منافذ الطاقة: يمكن لهذه المنافذ تزويد الطاقة إلى وحدات الإدخال أو الإخراج الخارجية. وتوجد ثلاثة منافذ أرضية، واثنين من منافذ 5V ومنفذ Vin. منفذ Vin له نفس الجهد الكهربائي الوارد من مدخل البطارية الخارجي.

منافذ الإدخال والإخراج: يمكن ربط هذه المنافذ بوحدات إدخال وإخراج إضافية مما يسمح لك بزيادة إمكانات لوحة ميكرو. ويمكنك القيام بذلك عن طريق إضافة زر ضغط آخر على سبيل المثال، أو حتى وحدة استشعار مثل مستشعر الحركة أو اللهب، وكلاهما غير موجود في اللوحة.

تسمح لك بعض المنافذ بالاتصال باللوحات أو المكونات الأخرى. وتسمح لك المنافذ الأخرى بقراءة وكتابة الإشارات التناظرية والرقمية كما هو موضح في القسم التالي.



تهيئة المنافذ

تحتوي لوحة ميكر على المنافذ التالية. بعض هذه المنافذ خارجية (أي على الجانبين الخارجيين للوحة) وبعضها داخلي، مثل أزرار الضغط. من المهم معرفة المنفذ الخاص بكل عنصر على اللوحة بحيث يمكنك التحكم الكامل في لوحة الميكر.

المنفذ	الوظيفة
D0/RX	منفذ حر، إدخال/إخراج رقمي واتصال تسلسلي (استقبال)
D1/TX	منفذ حر، إدخال/إخراج رقمي واتصال تسلسلي (إرسال)
D2/SDA	منفذ حر، إدخال/إخراج رقمي واتصال I2C
D3/SCL	منفذ حر، إدخال/إخراج رقمي واتصال I2C، إخراج تناظري
D4	زر الضغط A
D5	مكبر الصوت
*D6	منفذ حر، إدخال/إخراج رقمي، إدخال/إخراج تناظري
D7	مقاطعة التسارع (غير مستخدم حالياً)
D8	اختيار رقاقة مقياس التسارع
*D9	منفذ حر، إدخال/إخراج رقمي، إدخال/إخراج تناظري
*D10	منفذ حر، إدخال/إخراج رقمي، إدخال/إخراج تناظري
*D12	منفذ حر، إدخال/إخراج رقمي، إدخال تناظري
D13	مصباح LED مضمن
D17	وحدات NeoPixels
A0	مستشعر درجة الحرارة
A1	زر الضغط B
A4	مستشعر الصوت
A5	مستشعر الضوء



البرنامج

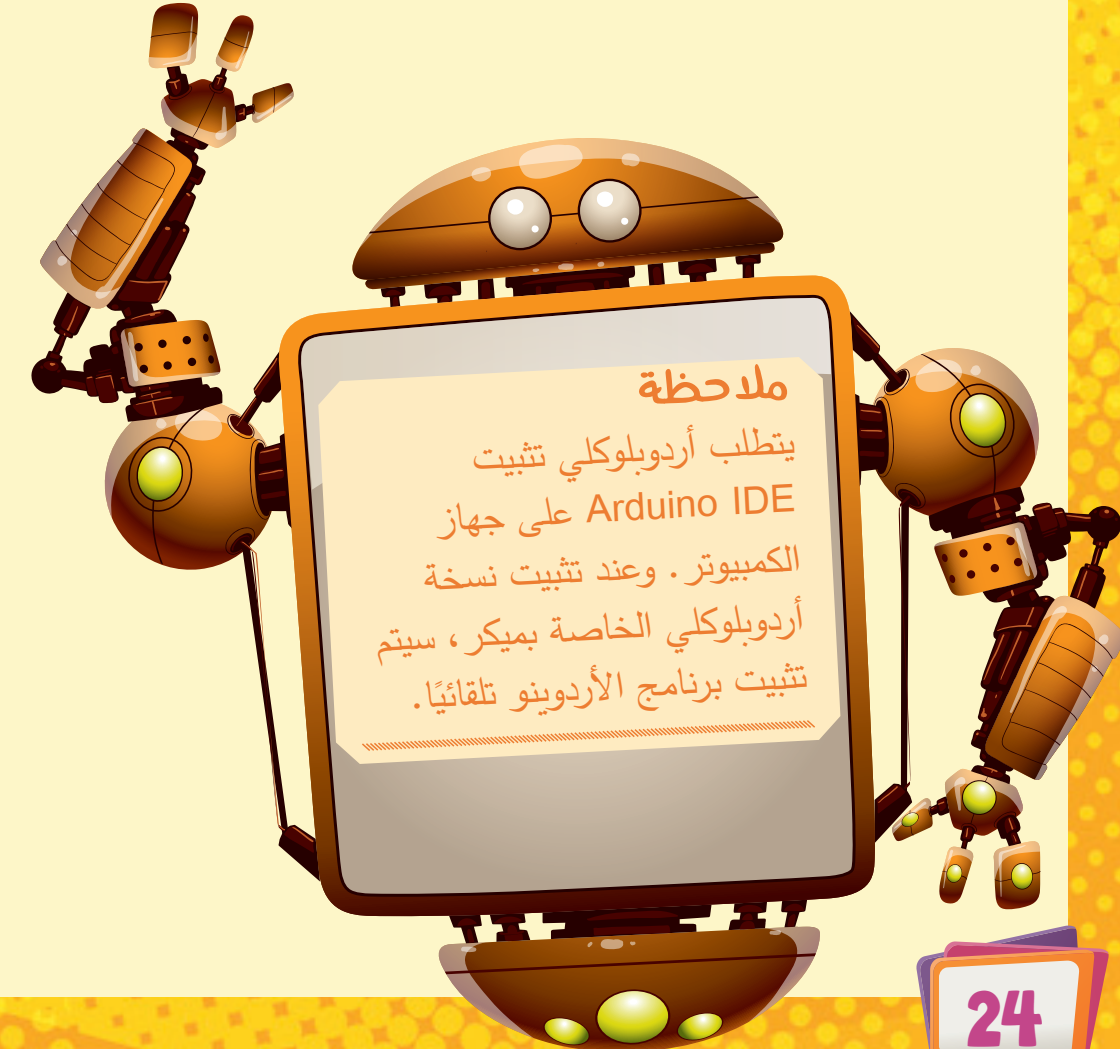
بيئة التطوير المتكاملة لاردينو

تسمح لك بيئة التطوير المتكاملة لاردينو (Arduino IDE) بكتابة وتحميل البرامج إلى لوحة ميكر، باستخدام محرر النصوص. وتحتوي هذه الواجهة مفتوحة المصدر على منطقة رسائل وشريط أدوات يضم أزراراً للتحقق من البرامج وتحميلها بالإضافة للوظائف العامة الأخرى مثل إنشاء الملفات وحفظها. يمكنك قص البرامج ونسخها ولصقها.



أردوبلوكلي

أردوبلوكلي (Ardublockly) هي واجهة برمجة مرئية لاردينو. وتعتمد هذه الواجهة على مكتبة بلوكلي المتوفرة لدى جوجل.



Ardublockly

في البرمجة المرئية، يمكنك استخدام عناصر رسومية لإنشاء البرامج. ويمكنك أيضًا سحب عناصر البرنامج وإفلاتها، والنقر فوقها، واستخدام القوائم، والنماذج، ومربعات الحوار وغيرها. يوجد خلف كل عنصر من هذا العناصر المرئية عشرات أو حتى مئات التعليمات البرمجية. وهو ما يساعد المبرمجين الجدد على فهم البرمجة بسهولة.

لغة برمجة بايثون

بايثون هي لغة برمجة عالية المستوى وتستخدم على نطاق واسع للبرمجة العامة (general-purpose programming)، وتم إصدار هذه اللغة لأول مرة في عام 1991. وتتميز بفلسفة تصميم تركز على قابلية قراءة البرامج (ولا سيما باستخدام المسافة البادئة لتحديد حدود البرامج، بدلاً من الأقواس المعقوفة أو الكلمات الرئيسية) وبناء الجمل (مجموعة من القواعد) التي تسمح للمبرمجين بإظهار المفاهيم في أسطر أقل من الكود، مقارنة بلغات البرمجة الأخرى.

لتثبيت أحدث إصدار، يمكنك زيارة موقع بايثون الرسمي.

يمكن استخدام لغة بايثون للاتصال التسلسلي مع لوحة ميكرو والتي تسمح لك باستخدام جميع إمكانات بايثون المثبتة على الكمبيوتر، وذلك باستخدام مكتبة بايثون التي تسمى PySerial. وعلى عكس برنامج أردوينو وأردوبلوكلي والتي تقوم بتحميل الأكواد على لوحة الميكرو، يمكنك الكتابة إلى ميكرو مباشرة، وتسمح لك PySerial بإرسال واستقبال البيانات بين جهاز الكمبيوتر ولوحة الميكرو.

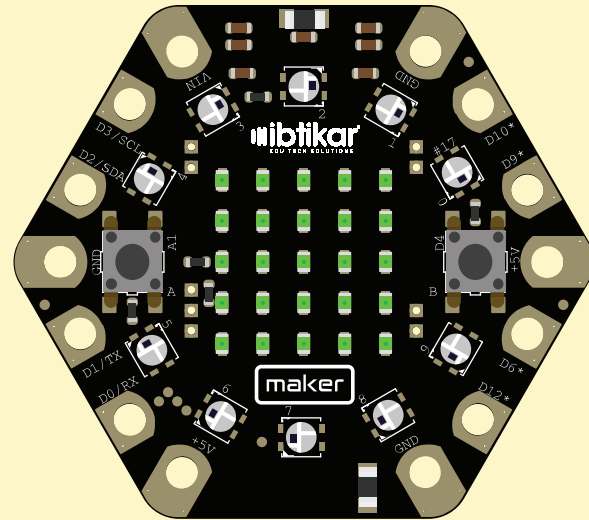
في هذا الدليل سيتم التركيز فقط على بيئة البرمجة أردوبلوكلي.

 python™



ماذا يوجد في الصندوق؟

عند فتح الصندوق، سوف تجد ما يلي:



لوحة ميكر من ابتكار



8 كابلات Crocodile (4 أنثى و 4 ذكر)



كابل USB صغير



حامل بطارية 9 فولت

إمداد لوحة ميكر بالطاقة

موصل USB

يسمح لك كابل USB الصغير بتوصيل جهاز الميكر بجهاز الكمبيوتر الذي يعمل على تشغيل اللوحة ويسمح لك بإرسال البيانات واستلامها. ويكون كابل USB كما في الصورة.



مصدر الطاقة الخارجي

خيار الطاقة الآخر هو استخدام مصدر خارجي بجهد كهربائي يتراوح من 5 إلى 9 فولت. ويتوفر لديك في هذه المجموعة، حامل بطارية خاص ببطارية 9 فولت.

يشتمل حامل البطارية على مفتاح ON/OFF (تشغيل/إيقاف التشغيل) لتسهيل تشغيل لوحة الميكر، دون الحاجة إلى توصيل البطارية وفصلها في كل مرة.





تنقسم هذه الواجهة إلى الأقسام الأربعة التالية:

1. قسم عناصر الكود المرئية: في هذا القسم، تقوم بسحب العناصر التي تريدها من الجانب الأيسر وإفلاتها في المنطقة البيضاء لتشكيل برنامجك. ويمكنك تكبير وتصغير العناصر وتعديل قيمها أو حذفها.

2. قسم كود الاردوينو: يعرض هذا القسم كود اردوينو الذي يتوافق مع العناصر التي قمت بسحبها. يتم إنشاء هذا الكود تلقائيًا بواسطة البرنامج ولا يمكنك تعديله ما لم تقم بإجراء التغيير في قسم العناصر المرئية. ولا يزال بإمكانك نسخ كود الاردوينو إلى واجهة Arduino IDE وتغييره من خلالها.

3. قسم مخرجات Arduino IDE: عند تحميل البرنامج إلى لوحة الميكر، ستظهر حالة عملية التحميل أو أي رسالة خطأ في هذا القسم. ويمكنك إخفاء هذا القسم أو إظهاره من خلال النقر عليه.

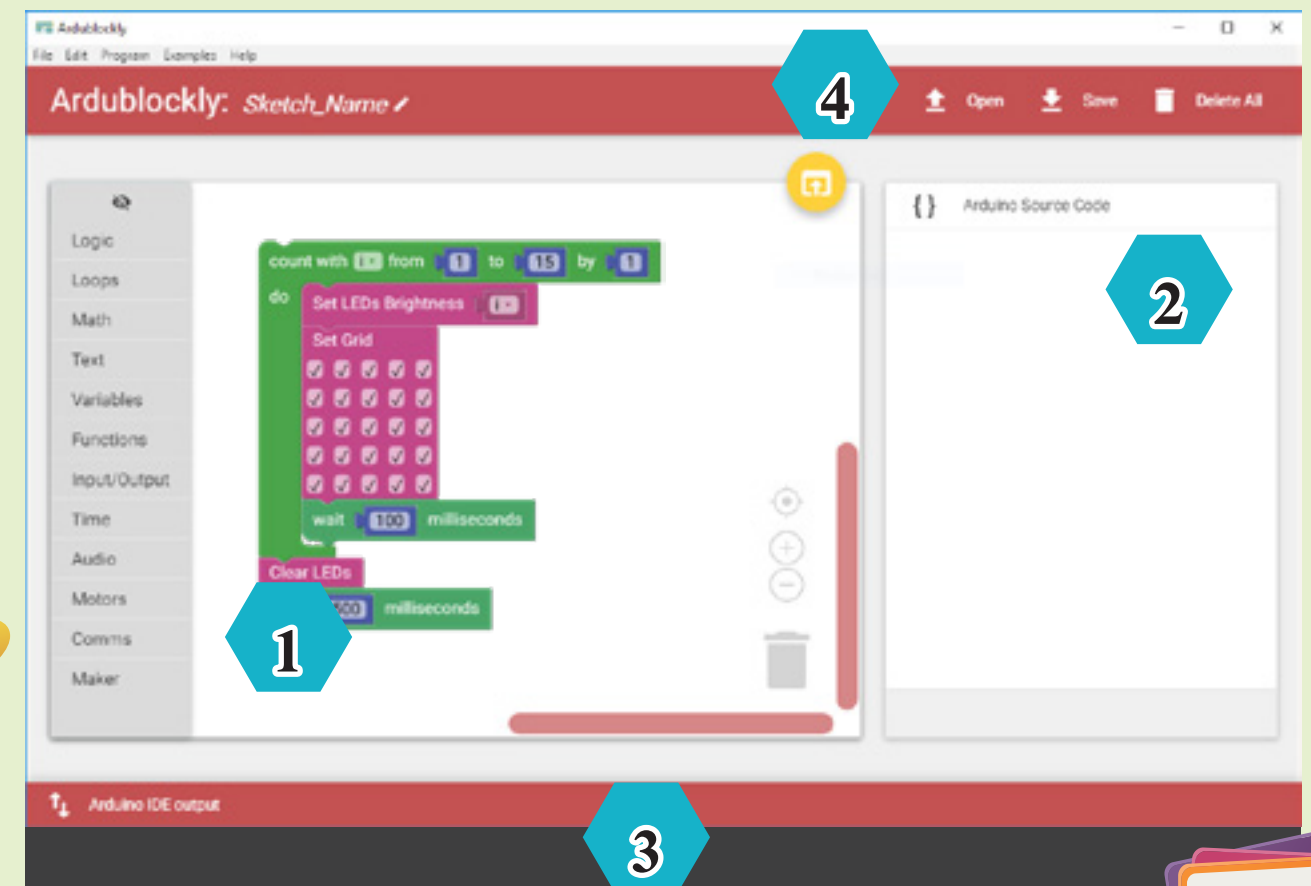
4. قسم الوصول السريع: يحتوي هذا القسم على أزرار الاستخدام الشائعة مثل، فتح، وحفظ وحذف الكل.



إنشاء برنامجك الأول

واجهة أردوبلوكلي

تظهر في الشكل التالي واجهة أردوبلوكلي.

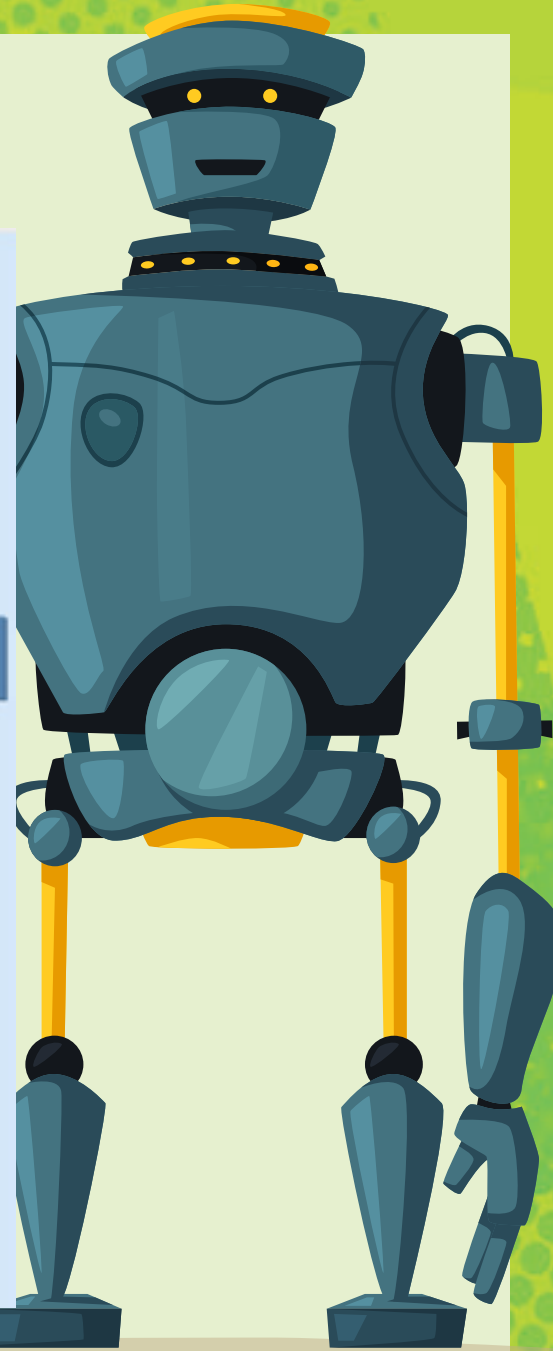
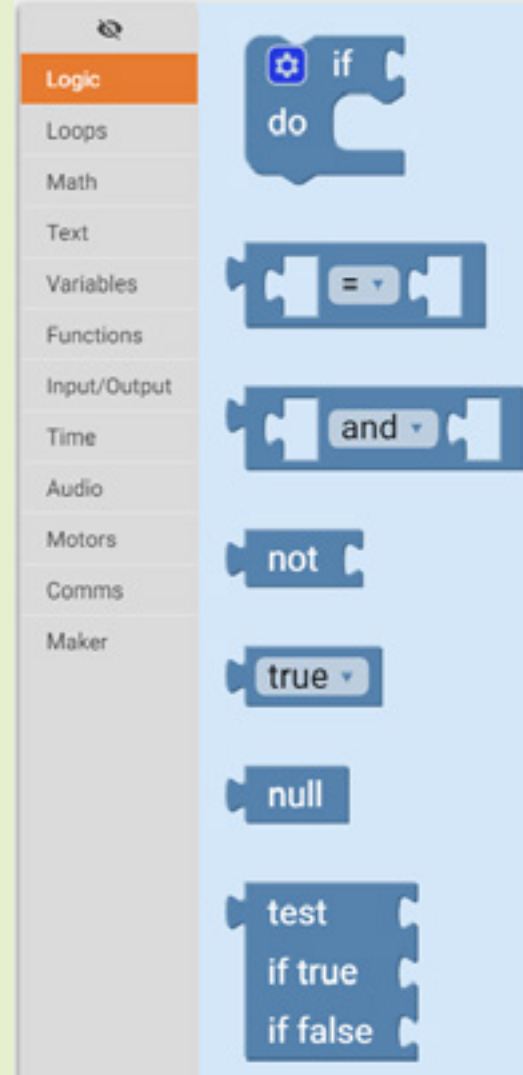


المنطق

تتبع هذه الفئة نظام الجبر المنطقي، وهو فرع من فروع الجبر حيث تحتوي المتغيرات على قيمتين متاحيتين فقط؛ وهما **صحيح (True)** أو **خطأ (False)**.

عوامل المقارنة

تقارن هذه العوامل القيم على جانبيها وتحدد العلاقة بينها. وترجع القيمة إما **صحيح** أو **خطأ** استنادًا لنتيجة مقارنة هذه القيم مع بعضها البعض.

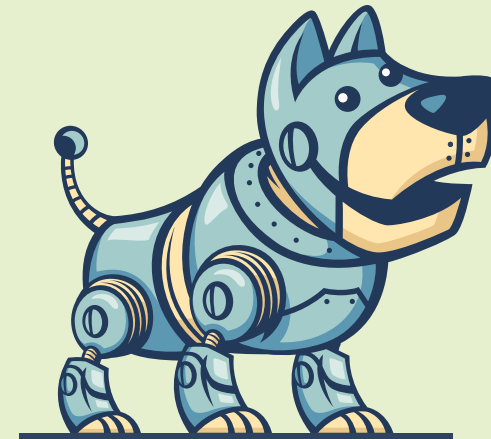


أقسام العناصر البرمجية

يوجد في أروبولوكلي، العديد من الأقسام المختلفة للعناصر المرئية. وكل قسم له غرض خاص. وفيما يلي الأقسام الرئيسية:

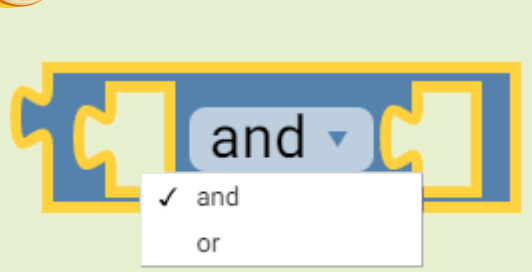
- المنطق
- المتغيرات
- الوقت
- الرياضيات
- التكرار
- النصوص
- الوظائف
- الميكرو

أكثر الأقسام شيوعًا هو قسم **ميكرو والمنطق والتكرار**. هذه الأقسام الثلاثة موضحة في الصفحات التالية، وسيتم توضيح أي قسم آخر عن باقي الأقسام بمجرد الحاجة إلى عنصر منه.



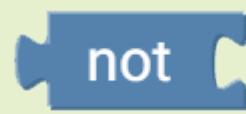


العمليات المنطقية



1. عبارة **and** التي تقوم بإرجاع القيمة **صحيح** عندما يكون كلا القيم المدخلة **صحيحة**.

2. عبارة **or** التي تقوم بإرجاع القيمة **صحيح** عندما يكون كلا المدخلات أو أحدهما **صحيحًا**.

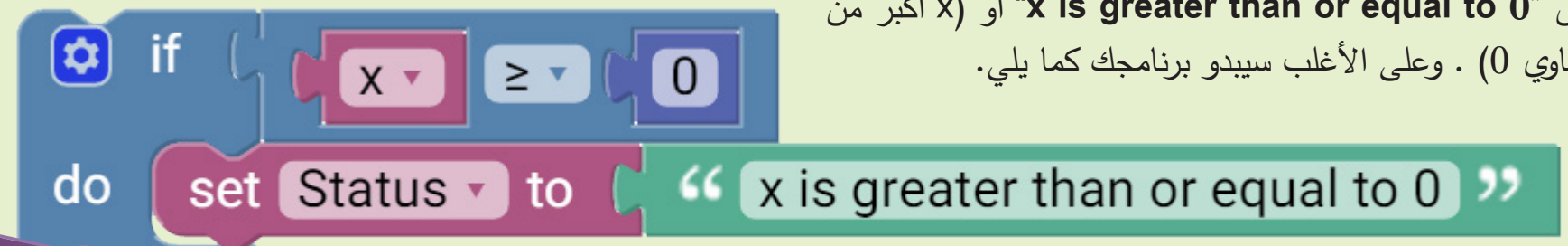


3. عبارة **not** التي تحول مدخلاتها إلى العكس. فإذا كان المدخل **صحيحًا**، فسوف يصبح **خطأ**، والعكس صحيح.

العبارات الشرطية

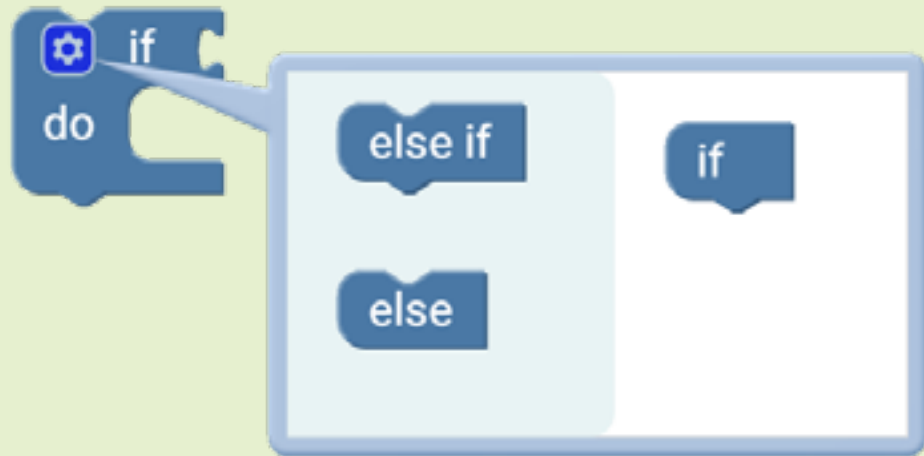
تعمل العبارات الشرطية على تنفيذ إجراءات مختلفة بناءً على الشروط الخاصة بها. وهناك عدة طرق للقيام بذلك، ومن هذه الطرق استخدام الجملة الشرطية **if** والتي يمكنك تكوينها، حسب حاجتك.

أبسط شكل من أشكال الجمل الشرطية هو **if**. وإذا تحقق الشرط، يلزم أن تفعل شيئاً، مثلاً؛ لديك رقم مُخزّن في متغير يسمى **x** وتريد التحقق من قيمة هذا المتغير، فإذا كانت القيمة أكبر من أو تساوي صفر، فأنت تريد تحديث متغير آخر يسمى **Status** (الحالة) بالنص **"x is greater than or equal to 0"** أو (x أكبر من أو يساوي 0). وعلى الأغلب سيبدو برنامجك كما يلي.



المعامل	الوصف
=	إذا كانت قيم المعاملين متساوية، فإن الشرط يصبح صحيحًا.
≠	إذا كانت قيم المعاملين غير متساوية، يصبح الشرط صحيحًا.
<	إذا كانت قيمة المعامل الأيسر أقل من قيمة المعامل الأيمن، يصبح الشرط صحيحًا.
≤	إذا كانت قيمة المعامل الأيسر أقل من أو تساوي قيمة المعامل الأيمن، يصبح الشرط صحيحًا.
>	إذا كانت قيمة المعامل الأيسر أكبر من قيمة المعامل الأيمن، يصبح الشرط صحيحًا.
≥	إذا كانت قيمة المعامل الأيسر أكبر من أو تساوي قيمة المعامل الأيمن، يصبح الشرط صحيحًا.





يمكن أن يتم إنشاء عبارة **if** بأشكالها المختلفة ببساطة عن طريق النقر على أيقونة الترس الصغيرة في العنصر **if**. ويمكنك بعد ذلك السحب والإفلات بقدر ما تريد من العنصر **else if**، وعنصر **else** واحد، ثم إصاقها بالعنصر **if**.



لنفترض أنك تريد عبارة **if** مع عنصرين من **else if**. بعد سحب العناصر المطلوبة من الجانب الأيسر من القائمة وإفلاتها بالجانب الأيمن، سوف ينتهي بك الأمر إلى ما يلي.



الشكل الآخر من حالات **if** الشرطية هو نمط **if - else**. يسمح لك هذا النمط بالتحقق من الحالة، وإذا كانت **صحيحة** أم لا. فإذا تحقق الشرط، **تفعل** شيئاً معيناً وإذا لم يتحقق الشرط، **تقم** بإجراء عمل آخر.

في المثال نفسه، إذا كانت قيمة **x** أكبر من أو تساوي صفر، فأنت تريد تحديث المتغير **Status** (الحالة) بالنص **"x is greater than or equal to 0"**. عدا ذلك، قم بتحديث حالة المتغير بالنص **"x is less than 0"** أو **(x أقل من 0)**.

```

if [x] >= 0
do [set Status to "x is greater than or equal to 0"]
else [set Status to "x is less than 0"]

```



الشكل الثالث من عبارة **if** الشرطية، هو نموذج **if - else if - else**. يسمح لك هذا النمط بالتحقق من شروط متعددة والقيام بإجراءات مختلفة.

في المثال نفسه، إذا كانت قيمة **x** أكبر من صفر، فأنت تريد تحديث حالة المتغير بالنص **"x is greater than 0"**. ولكن إذا كانت قيمة **x** أقل من صفر، فأنت تريد تحديث حالة المتغير بالنص **"x is less than 0"**. عدا ذلك، قم بتحديث حالة المتغير بالنص **"x equals 0"** أو **(x تساوي 0)** نظرًا لأنها الحالة الأخيرة التي قد تواجهها.

```

if [x] > 0
do [set Status to "x is greater than 0"]
else if [x] < 0
do [set Status to "x is less than 0"]
else [set Status to "x equals zero"]

```

التكرار

في بعض الأحيان من المهم تكرار البرنامج دون توقف. إحدى الطرق هي تكرار العناصر التي تحتاج إليها، لكن هذا لا معنى له إذا كان البرنامج طويلاً أو إذا كنت تريد تشغيل البرنامج دون توقف. لحسن الحظ، توجد عناصر برمجية تسمح لك بتكرار برنامجك أو جزء منه لعدد معين من المرات، أو لحين استيفاء شرط معين أو حتى التكرار إلى الأبد.

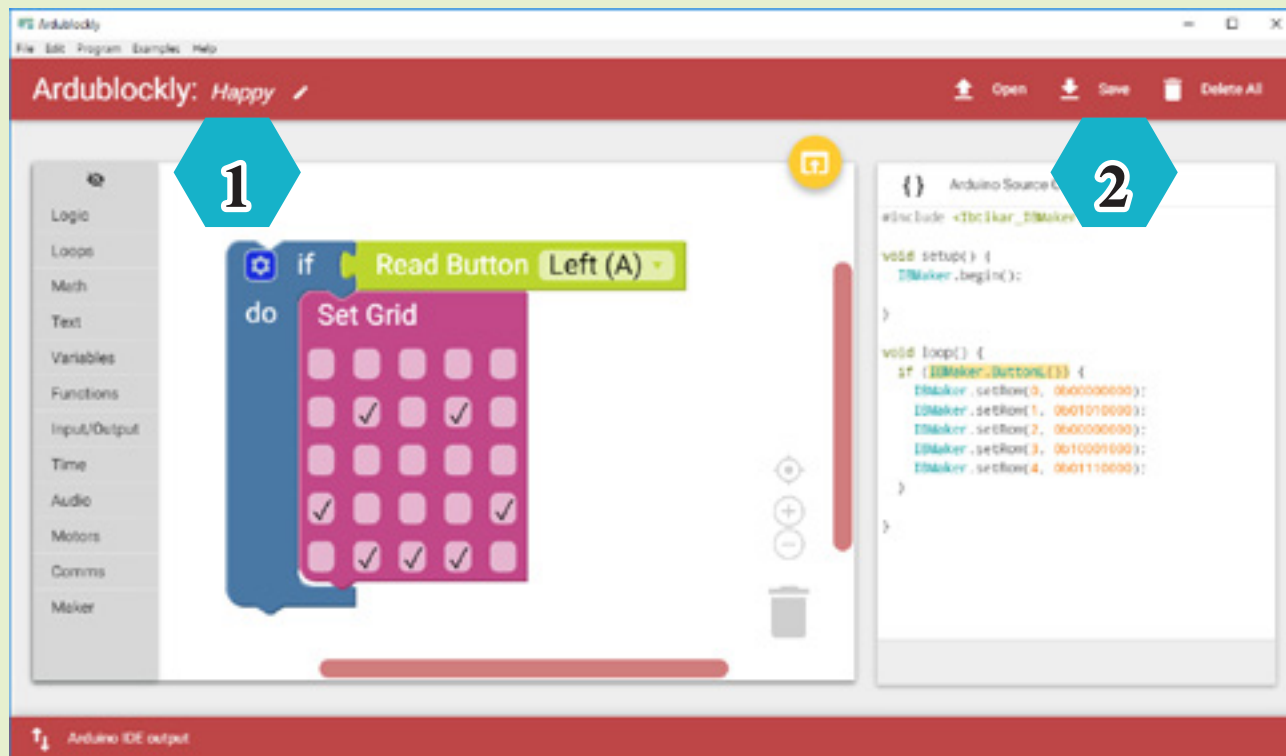


The screenshot shows the 'Loops' category in the Arduino IDE's block palette. It contains several loop blocks: 'repeat 10 times', 'repeat while', 'count with i from 1 to 10 by 1', and 'break out of loop'. The 'repeat 10 times' block is highlighted.



إنشاء، وحفظ، وتحميل مشاريعك

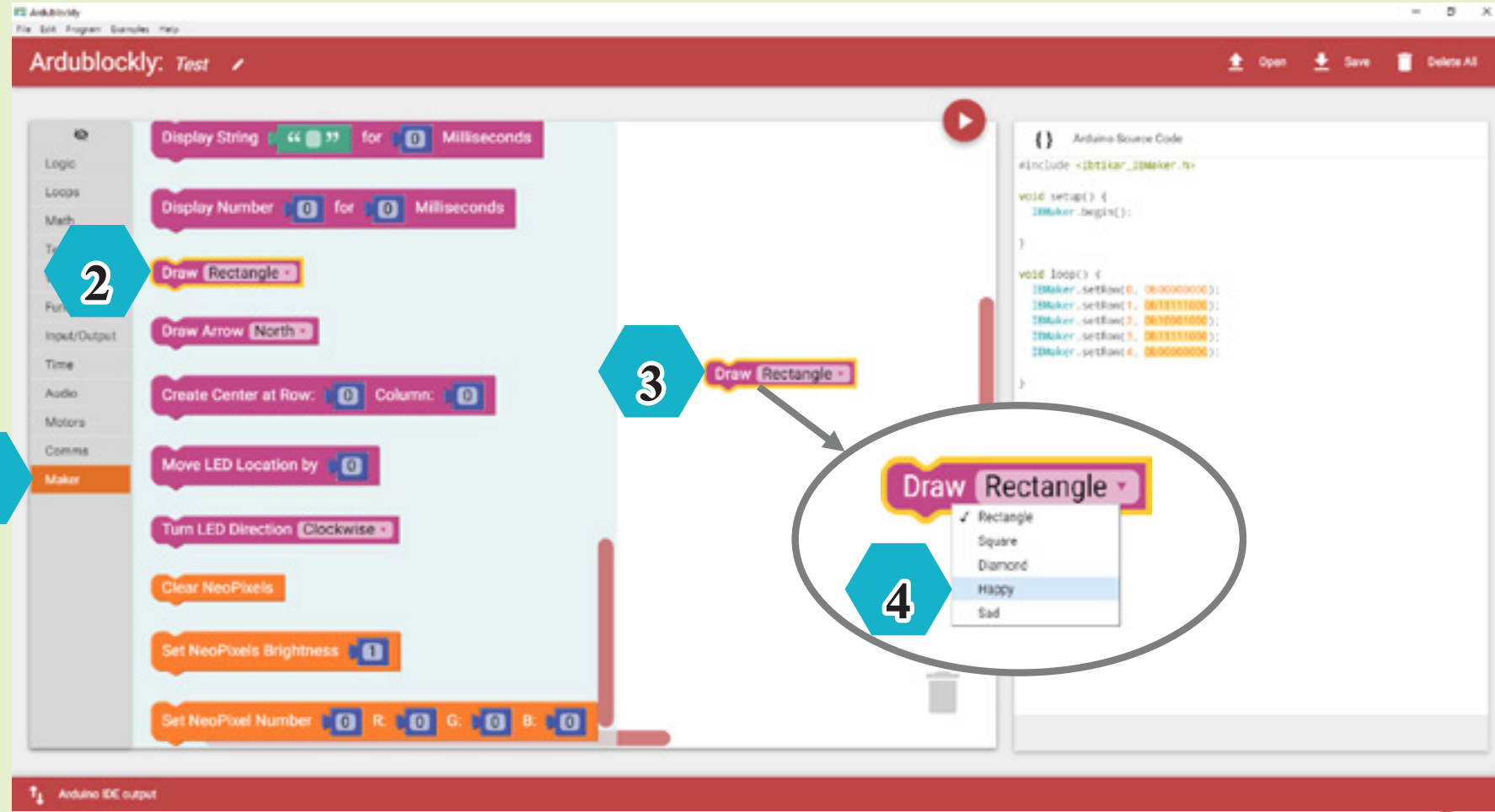
يمكنك فتح وحفظ وحذف جميع العناصر باستخدام قسم الوصول السريع. تعد ميزة حذف الكل مفيدة إذا كان لديك العديد من العناصر في برنامجك وتريد التخلص منها دفعة واحدة. إذا كنت تريد حفظ برنامج، فاكتب أولاً الاسم الذي تريد حفظه به كما هو موضح في الخطوة 1. ومن ثم انقر فوق حفظ كما هو موضح في الخطوة 2 وسوف تظهر نافذة تسمح لك باختيار الموقع الذي تريد حفظ البرنامج به.



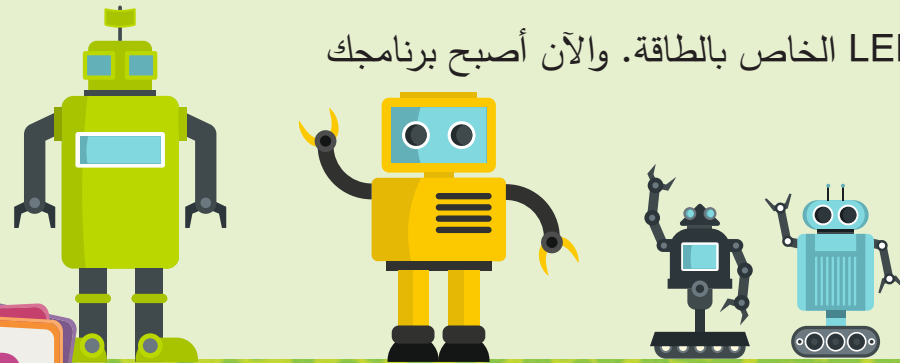
The screenshot shows the Arduino IDE interface. The 'if' block is highlighted with a red circle and the number 1, and the 'Set Grid' block is highlighted with a red circle and the number 2.

إذا قررت لاحقاً فتح برنامج آخر قمت بحفظه من قبل، فكل ما عليك فعله هو النقر على فتح وسوف تظهر لك النافذة. حدد موقع البرنامج الذي تريد فتحه وانقر فوق فتح.





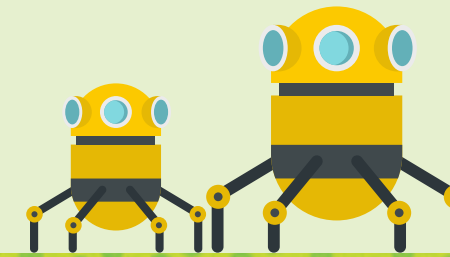
قم بتوصيل جهاز الميكر بالكمبيوتر. يجب أن يضيء مصباح LED الخاص بالطاقة. والآن أصبح برنامجك جاهزاً للتحميل إلى ميكر.

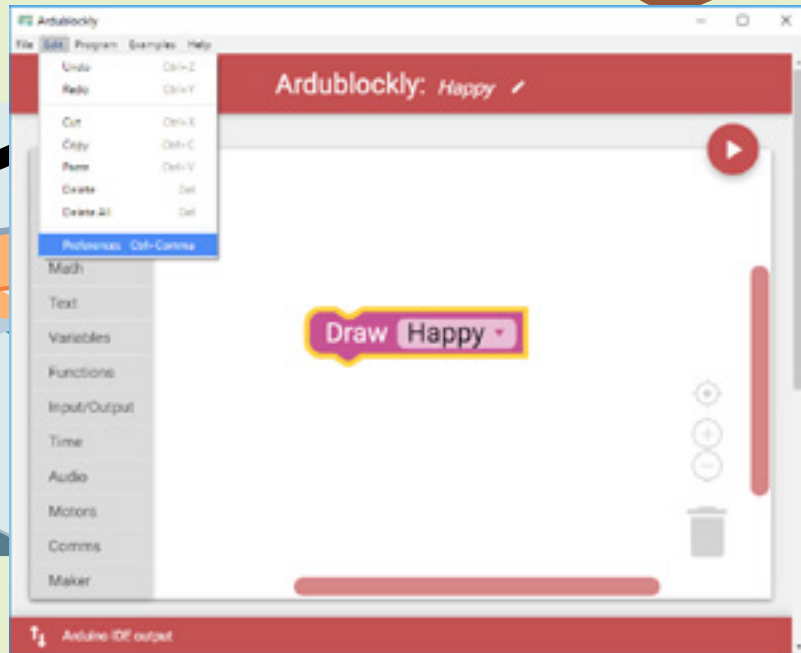


برمجة لوحة ابتكار ميكر

أول برنامج يمكنك تجربته هو عرض وجه سعيد على شبكة مصابيح LED. للقيام بذلك، ستحتاج إلى عنصر Draw. اتبع الخطوات التالية:

1. اذهب إلى قائمة Maker.
2. قم بالتمرير حتى تعثر على العنصر Draw.
3. اسحب عنصر Draw إلى مساحة العمل.
4. قم بتغيير الشكل الذي تريد رسمه من Rectangle إلى Happy.

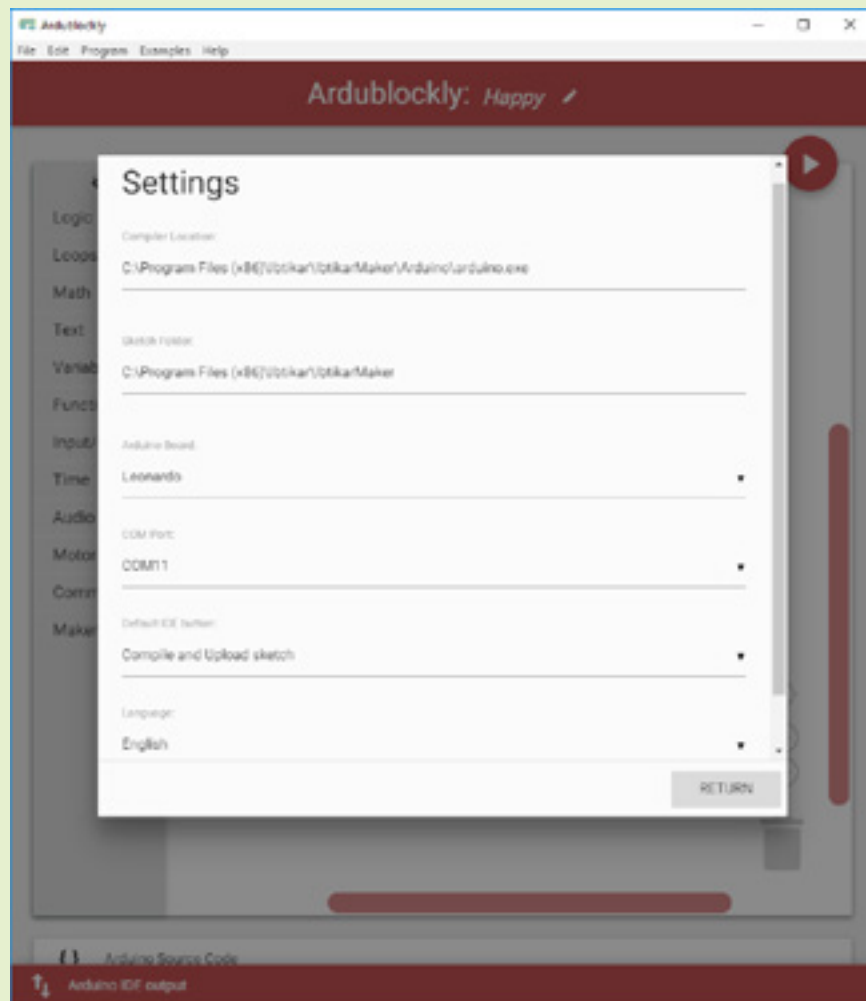




سيفتح هذا نافذة الإعدادات. تحقق من وجود برنامج التحويل البرمجي (Compiler) وبرامج الأردوينو في نفس الدليل كما هو موضح. ويتم اختيار هذه المواقع بشكل افتراضي عند تثبيت أردوبلوكلي. وتأكد من أن لوحة أردوينو المحددة هي ليوناردو (Leonardo) وأنها متصلة بمنفذ COM. قد يكون رقم منفذ COM مختلفًا على جهاز الكمبيوتر الخاص بك. وبعد أن تنتهي من القيام بجميع هذه الإعدادات، سوف تصبح لوحة ميكرو جهازة للبرمجة.

في حالة توصيلك بالفعل، ولكن لا تزال تتلقى نفس الرسالة، اتبع الخطوات التالية.

من قائمة تحرير، انتقل إلى التفضيلات



قم بحفظ البرنامج. يمكنك تسميته "Happy" (سعيد).

في بعض الأحيان، قد تتلقى رسالة من Arduino IDE تفيد بأن المنفذ التسلسلي غير متوفر. يمكن أن يحدث هذا إذا نسيت أن تقوم بتوصيل جهاز الميكرو بجهاز الكمبيوتر.

5. مرر الماوس بالقرب من منطقة تحميل البرامج وسوف يتم توسيع القائمة، وسوف ترى ثلاثة خيارات متاحة، بحيث يمكنك إما تحميل البرنامج مباشرة إلى ميكرو، أو التحقق منه قبل تحميله أو فتح البرنامج باستخدام واجهة Arduino IDE.

حمل البرنامج إلى ميكرو وانتظر حوالي 10 ثوانٍ. لكي تشاهد وميض مصباح LED المثبت على الجهة الخلفية للوحة أثناء تحميل البرنامج. وبعد ذلك يجب أن ترى الوجه السعيد على شبكة LED.



أنشطة ميكرو

نشاط 2: وميض مصباح LED

في هذا النشاط، سوف يومض مصباح LED المضمن في اللوحة كل ثانية. وستحتاج إلى عنصري LED وعنصري انتظار كما هو موضح.

Set Built-in LED to ON

wait 1000 milliseconds

Set Built-in LED to ON

wait 1000 milliseconds

قم بترتيب العناصر كما هو موضح وتحميل البرنامج وسوف يومض مصباح LED كل ثانية.

Set Built-in LED to ON

wait 1000 milliseconds

Set Built-in LED to OFF

wait 1000 milliseconds



مصباح LED المضمن باللوحة

يمكن التحكم في مصباح LED المضمن بالجهة الخلفية للوحة لتشغيله (ON) أو إيقاف تشغيله (OFF). وهذا المصباح لديه عنصر خاص به في قائمة Maker.

نشاط 1: تشغيل مصباح LED

في هذا النشاط، سيكون مصباح LED الموجود على اللوحة مضاءً بشكل دائم. وتعني كلمة "دائم" أن مصباح LED سيظل قيد التشغيل إلى أن تقوم بإلغاء توصيل ميكرو أو فصل البطارية. وستحتاج إلى عنصر برمجي واحد كما هو موضح. قم بتحميل البرنامج وسوف يتم تشغيل مصباح LED.

Set Built-in LED to ON

ON
OFF

Set Built-in LED to ON



← عنصر **Set Row** للتحكم في مصابيح LED في صف معين مرة واحدة.

← عنصر **Set Column** للتحكم في مصابيح LED في عمود معين مرة واحدة.

Draw Rectangle ▾

Draw Arrow North ▾

Set LED ON ▾ on Row: 0 Column: 0

Set Row 0

Set Column 0

Set Grid

شبكة مصابيح LED

يمكن للوحة ميكروالتحكم في مصابيح LED الخمسة والعشرين مرة واحدة أو كل مصباح على حدة. توجد في قائمة **Maker** العديد من العناصر المتاحة. وتسمح لك بعض هذه العناصر بعرض الأرقام أو الحروف أو تمرير النصوص أو حتى رسم الأشكال.

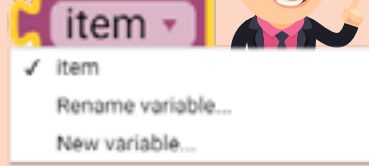
نشاط 3: رسم الأشكال

يمكنك رسم أشكال على ميكروالتحكم باستخدام طرق مختلفة. على سبيل المثال، يمكنك استخدام:

← عنصر **Draw Shape** لرسم مستطيل أو مربع أو شكل ماسي أو يمكنك رسم وجه سعيد أو حزين.

← عنصر **Draw Arrow** لرسم أسهم بالاتجاهات المختلفة. يمكنك رسم الشمال، الشمال الشرقي، الشرق، الجنوب الشرقي، الجنوب، الجنوب الغربي، الغرب أو الشمال الغربي.

← عنصر **Set LED** لتشغيل أو إيقاف تشغيل مصابيح LED بشكل فردي عن طريق تحديد الصف والعمود.



نظرًا لأنك تحتاج إلى التنقل بين الصفوف والأعمدة في الوقت نفسه، فإنك تحتاج إلى متغيرين مختلفين، أحدهما للصفوف والآخر للأعمدة. يمكنك إنشاء متغير جديد بالنقر فوق السهم الصغير في عنصر المتغير، ثم النقر فوق **"New variable"** (متغير جديد).

يمكنك أيضًا النقر على واحد من عناصر **for-loop** ثم النقر فوق **New variable**. وسوف تعمل كلتا الطريقتين على إنشاء متغير يسمى **"j"** تلقائيًا.

قم الآن بتوصيل العناصر كما هو موضح في الشكل. نظرًا لأن شبكة LED هي عبارة عن 5×5 والترقيم يبدأ فيها من 0، وبالتالي يجب أن يتم حساب كل حلقة من 0 إلى 4.

إذا قمت بتبديل الحلقات الداخلية والخارجية، فسوف يكون النمط عموديًا بدلاً من أفقيًا.

```

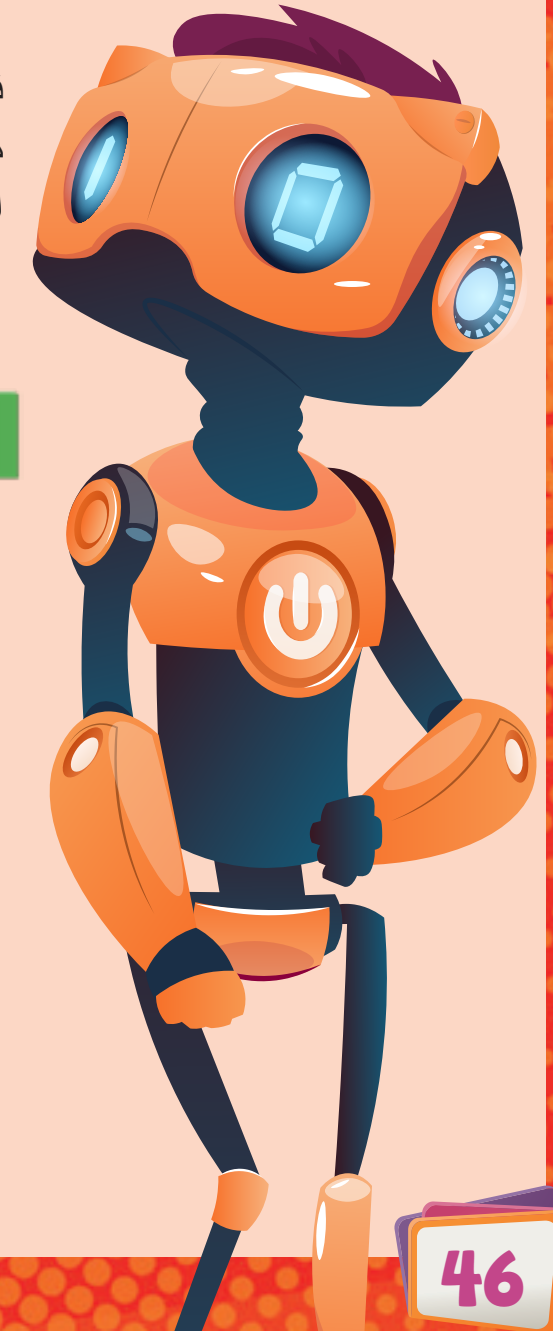
count with i from 0 to 4 by 1
do
  count with j from 0 to 4 by 1
  do
    Set LED ON on Row: i Column: j
    wait 100 milliseconds
    Set LED OFF on Row: i Column: j
    wait 100 milliseconds
  
```



في النشاط التالي، ستقوم بإنشاء برنامج لوميض كل مصباح LED بشكل فردي باستخدام حلقتين من الحلقات التكرارية **for-loops**، إحداها للصفوف والأخرى للأعمدة. وسوف تحتاج إلى العناصر التالية.

```

count with i from 1 to 10 by 1
do
  count with i from 1 to 10 by 1
  do
    Set LED ON on Row: 0 Column: 0
    Set LED ON on Row: 0 Column: 0
    wait 1000 milliseconds item item
    wait 1000 milliseconds item item
  
```



نشاط 5: عرض الحروف

لعرض رمز واحد (سواء كان حرفًا من حروف الأبجدية أو رقمًا أو أي رمز آخر) في ميكرو، فأنت بحاجة إلى العنصر البرمجي التالي.

Display Character ' ' for 0 Milliseconds

يقبل هذا العنصر حرفًا واحدًا لعرضه. ويمكنك في المعامل الثاني تحديد مدة العرض. وهذا يعني أنه سيتم عرض الحرف الذي تريده في الحال وبدون التمرير وسيستمر العرض للمدة التي تحددها.

لمشاهدة الاختلاف، جرب البرنامج التالي والذي بمجرد تحميله إلى ميكرو، سترى رمز علامة الاستفهام لمدة ثانية واحدة، ثم ترى الرقم 4 لثانية أخرى. وسوف يتكرر هذا بشكل دائم.

Display Character ' ? ' for 1000 Milliseconds

Display Character ' 4 ' for 1000 Milliseconds



نشاط 4: عرض الأرقام

لعرض الأرقام على ميكرو، ستحتاج إلى العنصر **Display Number**.

يقبل هذا العنصر عددًا صحيحًا (أرقام بدون كسر). كما يقبل أيضًا المتغيرات، لذا يمكنك عرض الأرقام، على سبيل المثال. وفي المعامل الثاني للعنصر، يمكنك تحديد مدة خطوة التمرير.

سيعرض البرنامج التالي الأرقام من 1 إلى 15 مع مدة تمرير 100 ميلي ثانية. ستحتاج إلى العناصر التالية:

Display Number 0 for 0 Milliseconds

count with i from 1 to 10 by 1
do
item

Display Number 0 for 0 Milliseconds

قم بترتيب العناصر وتغيير المعاملات الخاصة بها كما هو موضح. وانقر فوق عنصر المتغير وقم بتغيير اسمه ليتوافق مع المتغير الموجود في حلقة التكرار، والذي سيكون في حالتك هو المتغير "i".

count with i from 1 to 15 by 1
do
Display Number i for 100 Milliseconds

قم الآن بتحميل البرنامج إلى ميكرو وستشاهد الأرقام من 1 إلى 15 على شبكة LED.



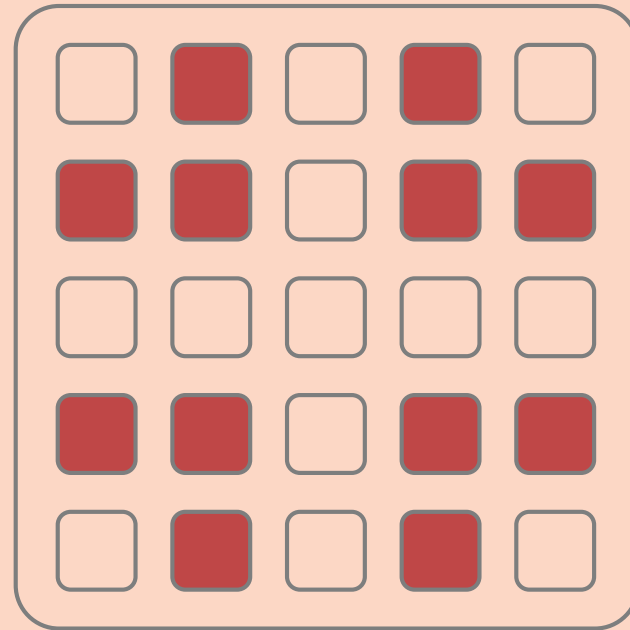
نشاط 7: تحريك ودوران مصباح LED

Create Center at Row: 0 Column: 0

Move LED Location by 0

Turn LED Direction Clockwise

نظرًا لأن مصابيح LED الخمسة والعشرين تشكل شبكة، يمكنك اعتبارها كنظام إحداثيات. وهو ما يعني أنه بإمكانك إنشاء نقطة أصل، وتحريك مصباح LED في اتجاه معين وتدويره إما باتجاه عقارب الساعة أو عكس اتجاه عقارب الساعة. مما يمكنك من إنشاء أنماط معقدة في عدد قليل من العناصر البرمجية بسهولة. في هذا النشاط، ستقوم بإنشاء النمط التالي مع تشغيل مصباح LED واحد في كل مرة.



نشاط 6: تمرير النصوص

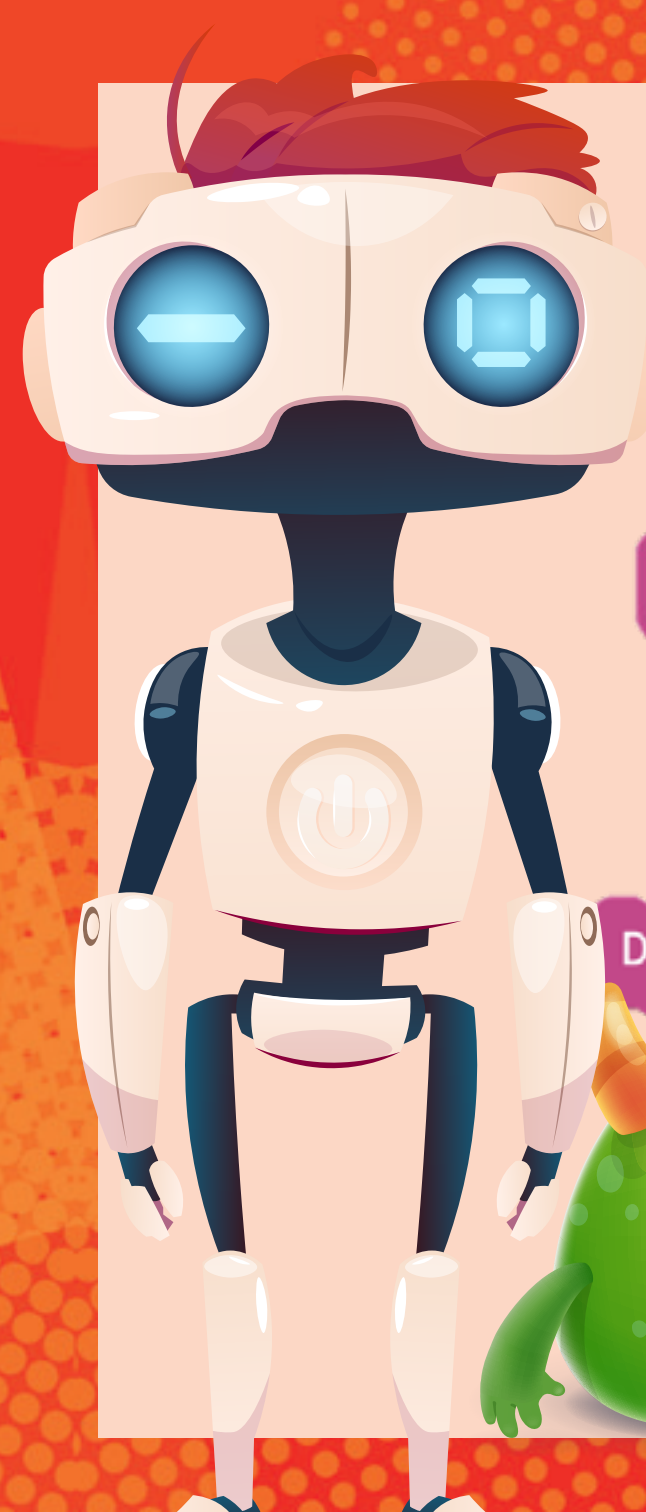
إذا كنت ترغب في تمرير نص، مثل اسمك على سبيل المثال، فسوف تحتاج إلى عنصر برمجي ثالث يسمى **Display String**. لاحظ الفرق بين الاقتباس الفردي في عنصر **Display String** و **Character** والاقتباس المزدوج في عنصر **Display String**.

Display String " " for 0 Milliseconds

في المثال التالي، سنقوم بتمرير "Hello Maker" (مرحبًا ميكرو) لمدة 200 ميلي ثانية لكل خطوة تمرير.

Display String "Hello Maker" for 200 Milliseconds

قم بتحميل البرنامج إلى ميكرو وتحقق من النتيجة. والآن قم بتمرير اسمك.



يجب أن تبدأ نقطة الأصل من النقطة (1,0).
ثم يتحرك مصباح LED خطوتين، ويدور
في اتجاه عقارب الساعة، ثم يتحرك خطوة
واحدة، ويدور عكس اتجاه عقارب الساعة وفي
النهاية يتحرك خطوة واحدة. إذا تم تكرار النمط
نفسه أربع مرات، فسوف تحصل على النمط
المطلوب. بما أن هذه العملية سريعة جداً،
فسوف يلزم إضافة تأخير زمني في كل جزء. قم
بترتيب برنامجك كما هو موضح.

Arduino run first:

Create Center at Row: Column:

Arduino loop forever:

repeat times

do

Move LED Location by

Turn LED Direction

wait milliseconds

Move LED Location by

Turn LED Direction

wait milliseconds

Move LED Location by

Turn LED Direction

wait milliseconds

يمكنك التفكير في هذا كمنط يعيد نفسه 4 مرات، كما هو موضح.

Loop 1

	■		■	
			■	■

Loop 2

				■
			■	■
			■	

Loop 3

■	■			
■				

Loop 4

	■			
■	■			
■				

Arduino run first:

Create Center at Row: Column: X 1

Arduino loop forever:

Move LED Location by X 3

Turn LED Direction X 3

wait milliseconds X 3

repeat times

do

1 X

1 X

سوف تحتاج إلى العناصر التالية.
يلزم توفر عنصر اردوينو ذو اللون
الأرجواني عندما تريد تشغيل بعض
أجزاء التعليمات البرمجية مرة واحدة
وتشغيل بعضها الآخر بشكل دائم.

قراءة الأزرار

يحتوي ميكرو على زرّين يمكنك قراءةتهما. وهذان الزران هما، الزر A على الجانب الأيسر للوحة ميكرو والزر B على الجانب الأيمن للوحة. وفي قائمة **Maker**، هناك نوعان من العناصر البرمجية ويمكنك استخدامها، على النحو التالي؛ واحد للتحقق مما إذا كان الزر مضغوطاً أم لا، بينما يقوم العنصر الآخر بإرجاع عدد المرات التي قمت بالضغط فيها على الزر.

يحتاج العنصران البرمجان إلى اختيار الزر الذي تريد قراءته. بالإضافة لذلك فإن العنصر الثاني يحتاج إلى الحد الأقصى لعدد

Read Button Left (A)

Left (A)
Right (B)

Read Button Left (A) Count 2 Times

النقرات التي تريد الوصول إليه، على سبيل المثال، إذا كانت القيمة 2، فهذا يعني أن العنصر يُرجع القيمة 0 إذا لم يتم النقر فوقه، والقيمة 1 إذا تم النقر فوقه مرة واحدة والقيمة 2 إذا تم النقر فوقه مرتين. إذا نقرت ثلاث مرات، فلن يتم اكتشاف ذلك، وسيكون الحد الأقصى هو 2.

ملاحظة

لتجنب إيذاء عينيك، لا تنظر مباشرة إلى شبكة LED عندما يكون السطوع مرتفعاً.

نشاط 8: سطوع مصابيح LED

العنصران الأخيران اللذان يتحكمان في شبكة LED هما **LEDs Brightness** (سطوع مصابيح LED) و **Clear LEDs** (إطفاء مصابيح LED). يسمح لك العنصر البرمجي الخاص بالسطوع، تغيير درجة سطوع مصابيح LED الخمسة والعشرين مرة واحدة. ويمكنك تغيير السطوع من 1 إلى 15. بينما يقوم العنصر الآخر بإيقاف تشغيل جميع مصابيح LED في وقت واحد. وهذا مفيد إذا قمت بعمل نمط معين، ثم تريد إيقاف تشغيل الشبكة دون الحاجة إلى إيقاف تشغيل كل مصابيح LED بشكل فردي.

في هذا النشاط، سوف تزيد قيمة السطوع من 1 إلى 15 بزيادة خطوة واحدة في كل مرة. داخل حلقة التكرار، ستقوم بإضافة عنصر السطوع، وتشغيل جميع مصابيح LED وتنتظر 100 ملي ثانية. وبمجرد الانتهاء من حلقة التكرار، سوف يتم إطفاء جميع المصابيح في وقت واحد والانتظار 500 ملي ثانية، ويجب أن يبدو برنامجك كما يلي.

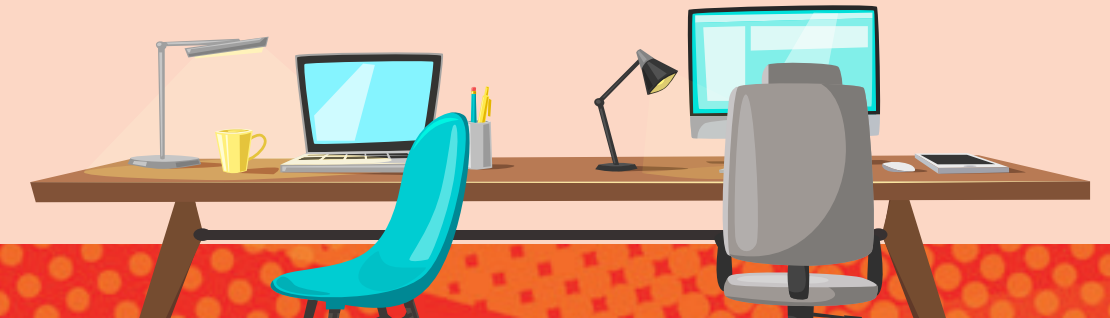


نشاط 9: قراءة كلا الزرين

في هذا النشاط، سوف تقرأ كلا الزرين وتعرض رمزًا على شبكة LED. إذا تم الضغط على الزر A، فسوف يتم عرض رمز "A" على الشبكة. وبالمثل، إذا تم الضغط على الزر B، فسيتم عرض رمز "B" على الشبكة. جرب البرنامج التالي وحمله إلى ميكرو.



```
if Read Button Left (A)
do
  Display Character 'A' for 200 Milliseconds
if Read Button Right (B)
do
  Display Character 'B' for 200 Milliseconds
```




نشاط 10: اكتشاف النقرات المتعددة

في هذا النشاط، ستقرأ عدد المرات التي يتم فيها الضغط على الزر A وتعرض وجهًا سعيدًا إذا وصلت إلى الحد الأقصى لعدد النقرات. وإذا لم تصل إلى الحد الأقصى، فسوف تعرض وجهًا حزيبًا.

نظرًا لأن عنصر **Count Button Read** سيعرض قيمًا مختلفة بناءً على عدد المرات التي تضغط فيها على الزر، فمن الأفضل قراءة العنصر وتخزين القيمة في متغير، ثم بناءً على القيمة المخزنة، يمكنك اتخاذ القرارات. سوف تحتاج إلى العناصر التالية.

جرب البرنامج التالي وحمله إلى ميكرو وحاول أن تصل إلى الحد الأقصى لعدد النقرات.



```
Read Button Left (A) Count 2 Times
set item to
Draw Rectangle Draw Rectangle
item 0
if Read Button Left (A) Count 5 Times
do
  Draw Happy
else
  Draw Sad
```





قراءة درجة الحرارة

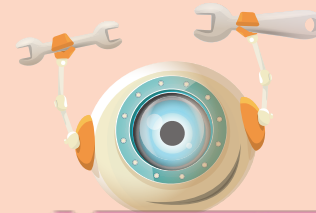
يمكن استخدام جهاز استشعار درجة الحرارة في ميكرو لقياس درجة حرارة اللوحة. يسمح لك عنصر البرمجة في قائمة **Maker** بقراءة درجة الحرارة إما بوحدة درجة مئوية (سلسيوس) أو فهرنهايت.

Read Temperature Celsius (C)

✓ Celsius (C)
Fahrenheit (F)

نشاط 12: عرض درجة الحرارة

في هذا النشاط، سوف تقوم بتمرير درجة الحرارة بوحدة درجة مئوية (سلسيوس) وفهرنهايت على شبكة LED. وللقيام بذلك، ستحتاج إلى العناصر التالية.



Display Number 0 for 0 Milliseconds Read Temperature Celsius (C)

Display Number 0 for 0 Milliseconds Read Temperature Celsius (C)

Display String "0" for 0 Milliseconds

Display String "0" for 0 Milliseconds

Display Number Read Temperature Celsius (C) for 200 Milliseconds

Display String "C" for 200 Milliseconds

Display Number Read Temperature Fahrenheit (F) for 200 Milliseconds

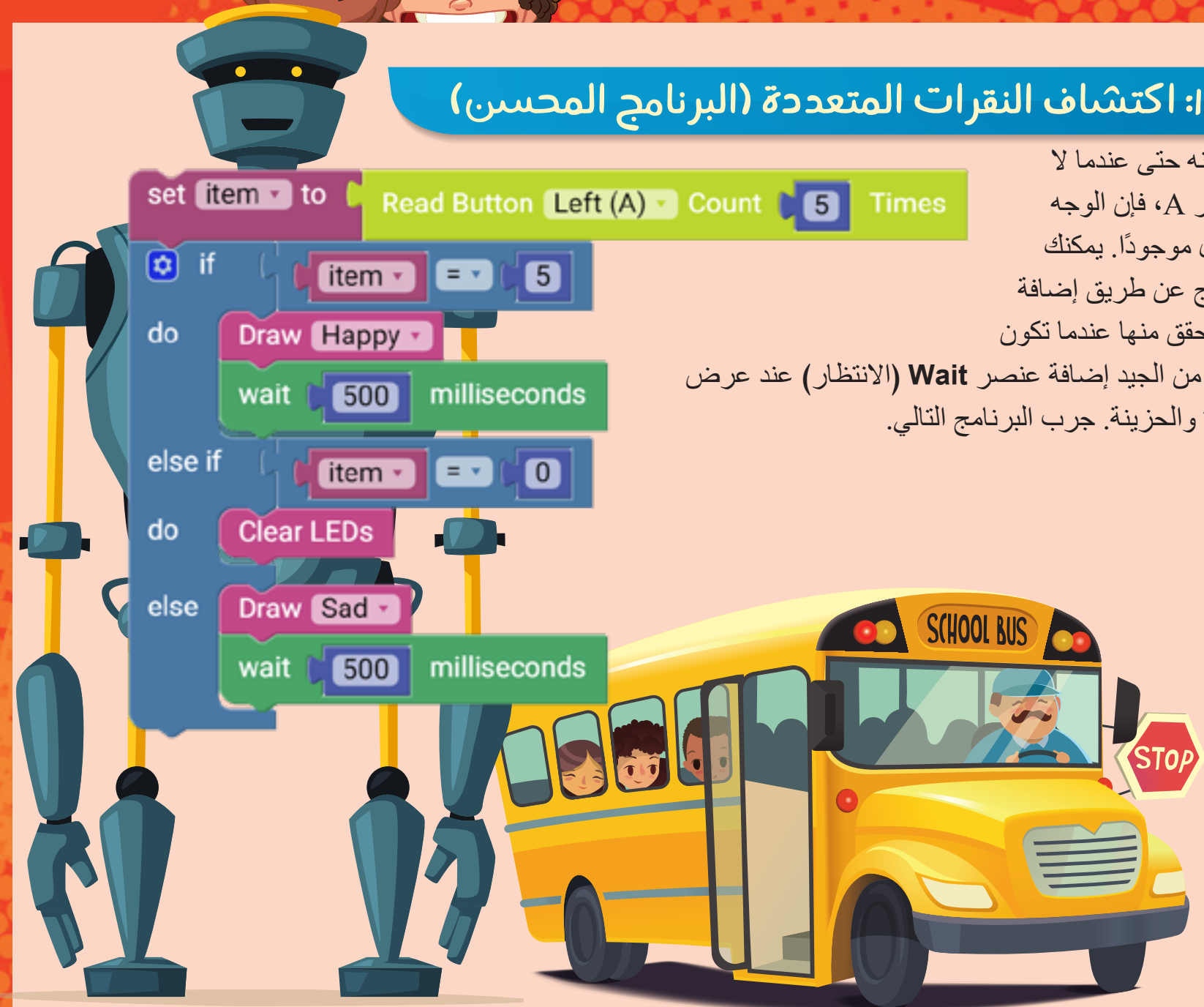
Display String "F" for 200 Milliseconds

جرب البرنامج التالي وحمله إلى ميكرو.



النشاط 11: اكتشاف النقرات المتعددة (البرنامج المحسن)

ربما لاحظت أنه حتى عندما لا تضغط على زر A، فإن الوجه الحزين لا يزال موجودًا. يمكنك تحسين البرنامج عن طريق إضافة حالة أخرى للتحقق منها عندما تكون القراءة صفرًا. من الجيد إضافة عنصر **Wait** (الانتظار) عند عرض الوجوه السعيدة والحزينة. جرب البرنامج التالي.



set item to Read Button Left (A) Count 5 Times

if item = 5

do Draw Happy

wait 500 milliseconds

else if item = 0

do Clear LEDs

else Draw Sad

wait 500 milliseconds

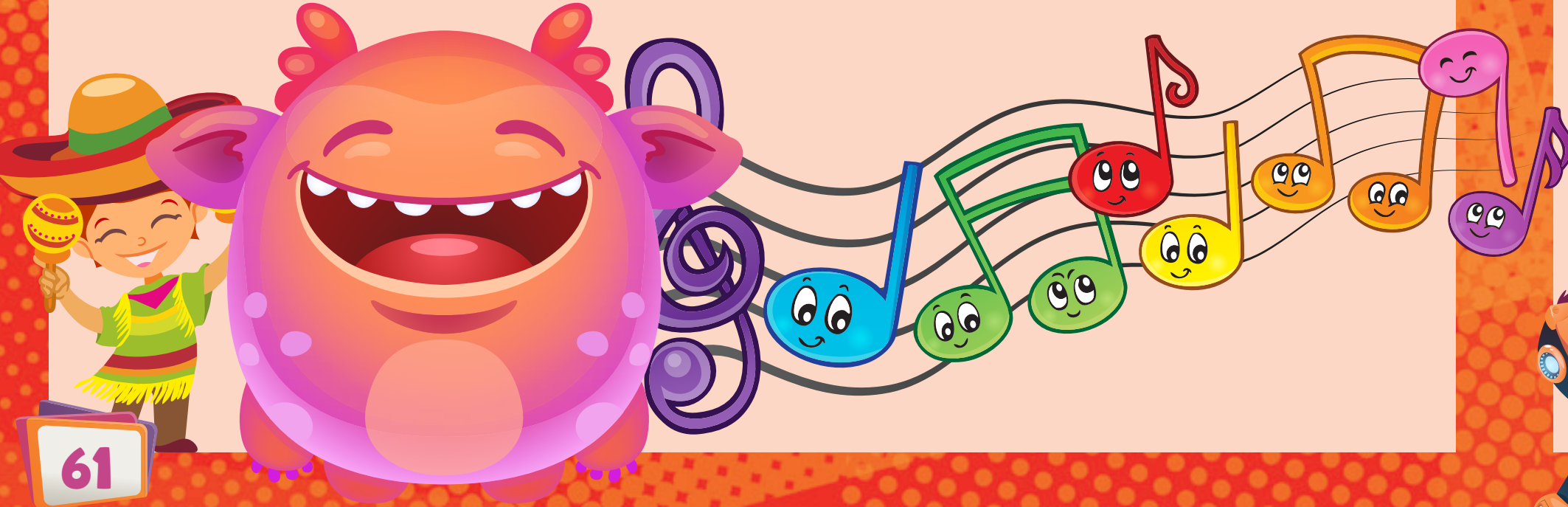
جهاز الإنذار

يعمل جهاز الإنذار المغناطيسي على توليد النغمات عن طريق التحكم في تردد النغمة ومدة استمرارها. ويظهر جهاز الإنذار كعنصر برمجي واحد في قائمة **Maker**. ويعد هذا العنصر في حد ذاته عبارة عن برنامج كامل.

Play Tone: Frequency 100 Duration 250

نشاط 14: إنشاء النغمات المختلفة

باستخدام 3 عناصر من **Tone Play**، قم بإنشاء نغمات بترددات مختلفة ومدة ثابتة. ومن ثم قم بإنشاء نغمات بتردد ثابت ومدة مختلفة. وفي كل حالة، حمّل البرنامج إلى لوحة ميكرو ولاحظ الفرق.



قراءة شدة الضوء المحيط

يقوم مستشعر الإضاءة في ميكرو بقياس شدة الإضاءة المحيطة. ويقوم بإرجاع القيمة من 0 إلى 1023 والتي تمثل مستوى شدة الإضاءة. وكلما كانت القيمة مرتفعة فهذا يعني أن مستوى شدة الإضاءة المُقاس مرتفع (يوجد ضوء). يمكنك العثور على العنصر البرمجي الخاص بهذا المستشعر في قائمة **Maker**.

Read Ambient Light



Display Number 0 for 0 Milliseconds

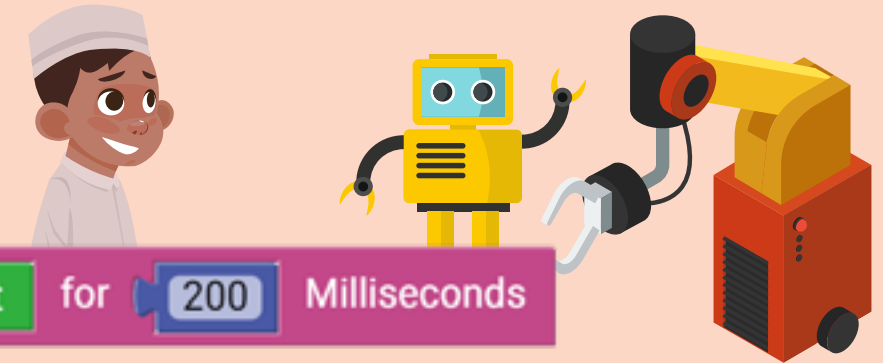
Read Ambient Light

نشاط 13: عرض قيمة الإضاءة المحيطة

في هذا النشاط، ستقوم بتمرير قيمة الإضاءة المحيطة على شبكة LED. ويمكنك أثناء النشاط استخدام مصباح يدوي وتوجيهه نحو المستشعر لرؤية التغير في شدة الإضاءة. وللقيام بذلك، ستحتاج إلى العناصر التالية.

جرب البرنامج التالي وحمله إلى ميكرو.

Display Number Read Ambient Light for 200 Milliseconds



يجب أن تبدو برامجك على النحو التالي.

البرنامج 1:

ترددات مختلفة ومدة ثابتة

```
Arduino run first:  
Play Tone: Frequency 300 Duration 250  
wait 500 milliseconds  
Play Tone: Frequency 400 Duration 250  
wait 500 milliseconds  
Play Tone: Frequency 500 Duration 250  
wait 500 milliseconds  
Arduino loop forever:
```

```
Arduino run first:  
Play Tone: Frequency 300 Duration 250  
wait 500 milliseconds  
Play Tone: Frequency 300 Duration 500  
wait 500 milliseconds  
Play Tone: Frequency 300 Duration 750  
wait 500 milliseconds  
Arduino loop forever:
```

البرنامج 2:

تردد ثابت ومدة مختلفة

هل يمكنك ملاحظة الفرق؟

البرنامج 1:

ترددات مختلفة ومدة ثابتة

```
Play Tone: Frequency 300 Duration 250  
Play Tone: Frequency 400 Duration 250  
Play Tone: Frequency 500 Duration 250
```

البرنامج 2:

تردد ثابت ومدة مختلفة

```
Play Tone: Frequency 300 Duration 250  
Play Tone: Frequency 300 Duration 500  
Play Tone: Frequency 300 Duration 750
```

إذا حاولت تحميل هذه البرامج، ستلاحظ أنه ليس من السهل التمييز بينها. قد يكون من المزعج أيضًا الاستماع إليها لأنها تتكرر بشكل مستمر.

إذا كنت ترغب في تشغيل البرنامج مرة واحدة، فقم بإضافة العنصر البرمجي **Arduino run first**. ويمكن إضافة هذا العنصر مرة واحدة في البرنامج ويعد مفيدًا عندما تريد تهيئة المتغيرات أو تعيين خصائص بعض المكونات مثل السطوح على سبيل المثال. وأيضا قم بإضافة عنصر **Wait** (الانتظار) بعد كل نغمة للحصول على الوقت الكافي للاستماع إليها.

نشاط 15: تغيير النغمات باستخدام حلقات التكرار

يقبل عنصر **Play Tone** المعاملات كالأرقام الصحيحة أو المتغيرات. في هذا النشاط، ستقوم بتغيير التردد من 100 إلى 2000 هرتز بخطوة 50 هرتز باستخدام عنصر **for loop**. سيتم تحديد مدة النغمة بمقدار 250 ميلي ثانية. وسيتم تشغيل جميع هذه العناصر مرة واحدة في الكود، لذلك يلزم استخدام العنصر **Arduino run first**.

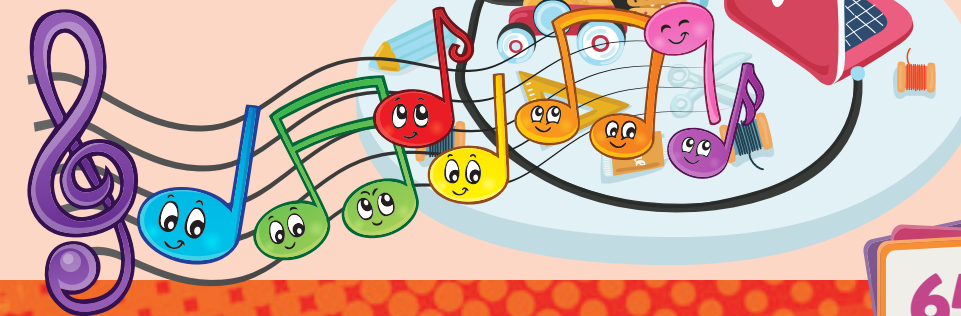
Arduino run first:

count with **i** from 100 to 2000 by 50

do Play Tone: Frequency **i** Duration 250

Arduino loop forever:

حمّل البرنامج إلى ميكرو و استمتع بالموسيقى التي أنشأتها للتو.



قراءة مستوى الصوت

يقوم مستشعر الصوت في ميكرو بقياس مستوى شدة الصوت. يظهر مستشعر الصوت كعنصر واحد في قائمة **Maker**. تعتبر القيم المُقاسة باستخدام ميكرو استثناءً مقارنة ببعض المستشعرات الموجودة بالأسواق، فبدلاً من زيادة القيمة عند اكتشاف مستشعر الصوت للضوضاء، فإنها تقل. ويتراوح النطاق الذي قد تصل إليه إذا صفتت بيديك على سبيل المثال ما بين 0 و 400.

نشاط 16: الاستجابة للصوت

في هذا النشاط، سوف تكتشف التغيير في مستوى الصوت عن طريق تشغيل نغمة. وسوف يعتمد تردد هذه النغمة على مستوى الصوت المُقاس. ويمكنك تمرير قيمة الصوت المُقاس مباشرةً إلى عنصر **Play Tone**.

يمكنك أيضاً القيام بالمزيد، كما يمكنك تغطية نطاق أوسع من الترددات عن طريق تحويل قيم نطاق الصوت (0-400) إلى قيم مدى التردد (0-2000) هرتز. لتجنب سماع ضوضاء عندما يكون التردد منخفضاً، يمكنك وضع عنصر **Play Tone** داخل عنصر **if**، بحيث يمكنك سماع النغمة فقط عندما تتجاوز القيمة 350.

```
set item to Map Value Read Sound From Range [ 400 - 0 ] to Range [ 0 - 2000 ]
if item > 350
do Play Tone: Frequency item Duration 200
wait 10 milliseconds
wait 1 milliseconds
```

حمّل البرنامج إلى ميكرو و صفق بيدك بالقرب من مستشعر الصوت. هل سمعت شيئاً؟

الآن ضع فمك بالقرب منه وانفخ باتجاه جهاز الاستشعار. هل سمعت نغمات موسيقية مختلفة؟

ملاحظة

لتجنب تلف جهاز الإنذار، لا تستخدم قيم تردد عالية (أعلى من 2000 هرتز).



سوف تحتاج إلى العناصر التالية.

- 7 X Read Capacitive D0 X 1
- 1 X set item to item X 7
- 7 X > or X 3
- 4 X Draw Arrow North if do X 4

جرب البرنامج التالي وحمله إلى ميكرو.

```

set item to 200
if Read Capacitive D0 > item
do Draw Arrow South West
if Read Capacitive D2 > item or Read Capacitive D3 > item
do Draw Arrow North West
if Read Capacitive D6 > item or Read Capacitive D12 > item
do Draw Arrow South East
if Read Capacitive D9 > item or Read Capacitive D10 > item
do Draw Arrow North East
  
```

لوحات المنافذ

يمكن ربط المنافذ الثمانية مع وحدات إضافية للإدخال والإخراج مما يسمح لك بتوسيع إمكانات لوحة ميكرو. كما يمكنك استخدام 7 منافذ منها كأجهزة استشعار تعمل باللمس.

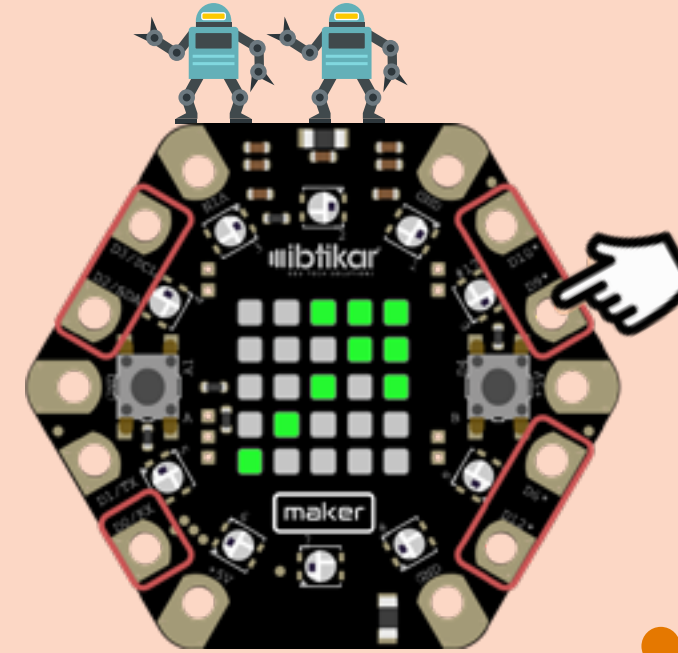
نشاط 17: التحقق مما إذا تم لمس لوحة ميكرو

في هذا النشاط، ستقوم ببرمجة ميكرو لعرض شكل السهم استنادًا إلى المنفذ الذي لمستته. على سبيل المثال، إذا لمست منفذ 9 أو منفذ 10، يجب أن ترى سهمًا باتجاه هذه المنافذ، كما هو موضح.

سوف تختار قيمة عتبة (Threshold)، وليكن 200 وتقوم باتخاذ القرارات حسب تلك القيمة. ونظرًا لأن قيمة العتبة ستكون واحدة لجميع منافذ اللمس، يمكنك استخدام متغير وتعيين قيمة 200 له. وسيؤدي ذلك إلى تسهيل الأمر إذا كنت تريد تغيير القيمة لاحقًا.

Read Capacitive D0

- D0
- D2
- D3
- D6
- D9
- D10
- D12



وحدات NeoPixels

يحتوي ميكرو على 10 مصابيح RGB. وعلى عكس المصابيح الخمسة والعشرين التي لا تستطيع تغيير لونها، يمكن برمجة مصابيح RGB LED لعرض أي لون من خلال الجمع بين الألوان الأساسية الثلاثة. تخيل كل مصباح من هذه المصابيح مثل ثلاثة مؤشرات ضوئية صغيرة مجتمعة، ولكل مصباح من هذه المصابيح لون مختلف (أحمر وأخضر وأزرق).

يمكن التحكم بهذه المصابيح باستخدام العناصر التالية من قائمة Maker.

Set NeoPixel Number 0 R: 0 G: 0 B: 0

Set NeoPixels Brightness 1 Clear NeoPixels

Set NeoPixel Number 0 Color White

لكل مصباح NeoPixel هنالك رقم مكتوب بجواره. ويساعدك ذلك في تحديد المصباح الذي تريد التحكم فيه.

نشاط 18: البيانو

في هذا النشاط، ستقوم ببرمجة ميكرو لتشغيل نغمة حسب لوحة اللمس التي قمت بالضغط عليها. جرب البرنامج التالي وحمله إلى ميكرو.

```
set item to 100
if Read Capacitive D0 > item
do Play Tone: Frequency 400 Duration 250
if Read Capacitive D2 > item
do Play Tone: Frequency 450 Duration 250
if Read Capacitive D3 > item
do Play Tone: Frequency 500 Duration 250
if Read Capacitive D10 > item
do Play Tone: Frequency 550 Duration 250
if Read Capacitive D9 > item
do Play Tone: Frequency 600 Duration 250
if Read Capacitive D6 > item
do Play Tone: Frequency 650 Duration 250
if Read Capacitive D12 > item
do Play Tone: Frequency 700 Duration 250
```

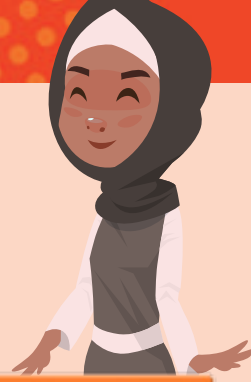
نشاط 20: تتابع التلاشي

في هذا النشاط، ستقوم بتغيير سطوع مصابيح NeoPixels باستخدام حلقة التكرار **for-loop**. يبدأ البرنامج التالي بإيقاف تشغيل جميع مصابيح LED، ثم ينتقل إلى حلقة التكرار الخاصة بالسطوع. ويمكن تغيير السطوع من 0 إلى 255. ويمكنك اختيار أي لون تريده لكل مصباح من مصابيح NeoPixel.

قم بإضافة 10 ملي ثانية تأخير بين كل تحديث سطوع. وبمجرد الانتهاء من الحلقة، سينتظر البرنامج نصف ثانية قبل البدء من البداية مرة أخرى.

جرب البرنامج التالي وحمله إلى ميكرو.

```
Clear NeoPixels
count with i from 0 to 255 by 1
do
  Set NeoPixels Brightness i
  Set NeoPixel Number 0 R: 255 G: 0 B: 0
  Set NeoPixel Number 1 R: 255 G: 127 B: 0
  Set NeoPixel Number 2 R: 255 G: 255 B: 0
  Set NeoPixel Number 3 R: 127 G: 255 B: 0
  Set NeoPixel Number 4 R: 0 G: 255 B: 0
  Set NeoPixel Number 5 R: 0 G: 0 B: 255
  Set NeoPixel Number 6 R: 0 G: 255 B: 255
  Set NeoPixel Number 7 R: 255 G: 0 B: 255
  Set NeoPixel Number 8 R: 75 G: 0 B: 130
  Set NeoPixel Number 9 R: 255 G: 255 B: 255
wait 10 milliseconds
wait 500 milliseconds
```



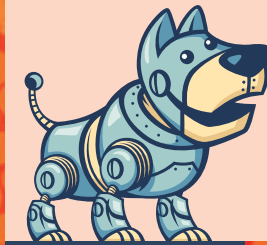
نشاط 19: عجلة الألوان

في هذا النشاط، ستعطي كل مصباح NeoPixel لونًا مختلفًا من القائمة المنسدلة. وبما أن هناك 10 وحدات NeoPixels، فإنك ستحتاج إلى 10 عناصر.

جرب البرنامج التالي وحمله إلى ميكرو.

هل رأيت جميع مصابيح NeoPixels ملونة بالألوان التي اخترتها؟

```
Set NeoPixel Number 0 Color White
Set NeoPixel Number 1 Color Red
Set NeoPixel Number 2 Color Lime
Set NeoPixel Number 3 Color Yellow
Set NeoPixel Number 4 Color Cyan
Set NeoPixel Number 5 Color Magenta
Set NeoPixel Number 6 Color Green
Set NeoPixel Number 7 Color Purple
Set NeoPixel Number 8 Color Teal
Set NeoPixel Number 9 Color Navy
```



نشاط 22: تكرار ألوان ومصابيح RGB LED

ماذا لو كنت تريد تغيير قيم RGB؟

في هذه الحالة، يمكنك إضافة عنصر **for-loop** منفصل لكل قيمة. في البرنامج التالي، ستقوم بتغيير قيمة 10 وحدات NeoPixels مرة واحدة. وستبدأ حلقات RGB الثلاث من 0 إلى 255 وبخطوة مقدارها 50. ففي كل مرة تنتهي فيها حلقة اللون الأزرق، ستغير حلقة اللون الأخضر القيمة **G** بخطوة مقدارها 50. وفي كل مرة تنتهي فيها حلقة اللون الأخضر، سوف تغير حلقة اللون الأحمر قيمة **R** بخطوة مقدارها 50.

جرب البرنامج التالي وحمله إلى ميكر.

```
Clear NeoPixels
do
  count with R from 0 to 255 by 50
do
  count with G from 0 to 255 by 50
do
  count with B from 0 to 255 by 50
do
  count with LED from 0 to 9 by 1
do
  Set NeoPixel Number LED R: R G: G B: B
wait 100 milliseconds
```

يمكنك تقليل مقدار الخطوة الخاصة بكل لون، لكن هذا سيجعل برنامجك يستغرق وقتاً أطول لإنهاء هذه الحلقات.

نشاط 21: تكرار مصابيح RGB LED

في بعض الأحيان، يكون من السهل تكرار رقم مصباح NeoPixel بدلاً من إضافة 10 عناصر برمجية. وسوف يساعد هذا في جعل برنامجك أصغر وأسهل في الفهم.

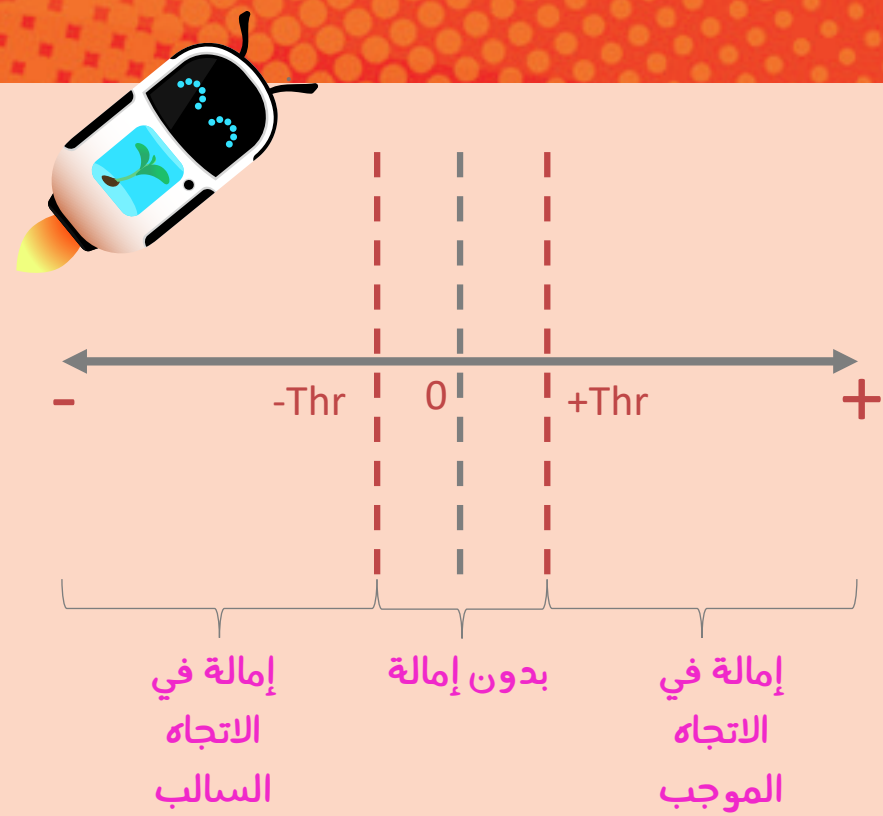
في هذا النشاط، ستقوم بتعيين 10 وحدات NeoPixels للون معين باستخدام عنصر **NeoPixel** وعنصر **for-loop**.

جرب البرنامج التالي وحمله إلى ميكر.

Clear NeoPixels

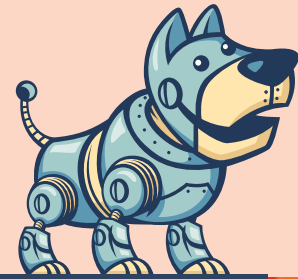
```
count with LED from 0 to 9 by 1
do
  Set NeoPixel Number LED R: 189 G: 71 B: 71
wait 250 milliseconds
```

قد تكون قيمة مقياس التسارع عبارة عن رقم موجب أو رقم سالب أو صفر. عندما تكون قيمة التسارع في المحور السيني صفراً، على سبيل المثال، فهذا يعني أن ميكرو مستوٍ ولا يميل في هذا الاتجاه. في بعض الأحيان، قد تحصل على قيمة موجبة أو سالبة صغيرة حتى إذا لم يكن ميكرو مائلاً. لتجنب هذا، ستحتاج إلى قياس هذه الإزاحة والتعويض عنها. وتكمن إحدى الطرق البسيطة والفعالة في إيجاد الإزاحة بإضافة أو طرح رقم من القيمة المُقاسة، والتحقق مما إذا كان هذا يحل المشكلة أم لا. يمكنك الاستمرار في معايرة القيمة المُقاسة حتى تلغي تأثير الإزاحة.



معايرة هذا المستشعر خارج نطاق هذا الدليل. لمزيد من التفاصيل حول كيفية معايرة المستشعر، يمكنك زيارة بوابة ميكرو (Maker portal) عبر الرابط التالي (maker.ibtikar.io). في الأمثلة التالية، تم إيجاد قيم الإزاحة باستخدام الطريقة الموضحة على بوابة ميكرو.

يوجد موضوع آخر مهم هو حساسية المستشعر. ماذا لو رغبت في أن تقوم اللوحة بالكشف عن الميل بعد قيمة معينة؟ وهذا يعني أنه عند إمالة اللوحة، يمكنك إنشاء قيمة العتبة (Threshold)، وبالتالي إذا تجاوزتها، فأنت متأكد من أن ميكرو مائل كما هو موضح.



مقياس التسارع ثلاثي المحاور

هذا المستشعر مثبت في منتصف اللوحة. ويتم استخدامه لقياس التسارع، وهو المعدل الزمني لتغير السرعة. ويمكن لهذا المستشعر قياس التسارع الثابت مثل الجاذبية للكشف عن درجة ميل سطح معين مثل ما يحدث عند إمالة هاتفك المحمول.

يوجد عنصران لبرمجة مقياس التسارع في قائمة **Maker**. يستخدم الأول لتمكين أو تعطيل المستشعر. ويسمح لك العنصر الثاني بقراءة قيمة التسارع في أحد المحاور الثلاثة (X أو Y أو Z).

Enable Acceleration Sensor FALSE ▾

Read Acceleration of Axis X ▾

✓ X
Y
Z

ملحوظة

إذا كنت تستخدم مستشعر التسارع، فيجب ألا تلمس لوحات اللمس 2 و 3 لأنهما يشتركان في نفس خطوط البيانات. إذا لمستهما عن طريق الخطأ، فسوف يتوقف ميكرو عن العمل وستحتاج إلى إعادة تعيينه باستخدام زر إعادة التشغيل على الجانب الخلفي. ولا يزال بإمكانك استخدام اللوحات الأخرى التي تعمل باللمس.

نشاط 24: الكشف عن الإمالة باتجاه عدة محاور (4 اتجاهات)

في هذا النشاط، ستقوم بإنشاء برنامج للكشف عن الإمالة في المحورين X و Y. سوف يظهر الاتجاه على شكل سهم (الشمال أو الشرق أو الجنوب أو الغرب) باستخدام شبكة LED. مثل النشاط السابق، سوف تقوم بإنشاء قيمة العتبة، والتي إذا تجاوزتها، فأنت على يقين من إمالة ميكرو. وسوف تكون هذه القيمة واحدة لكلا الاتجاهين.

بعد عدة محاولات تجريبية، تم العثور على الإزاحة في محور X بمقدار 34 تقريباً وفي المحور Y بمقدار 85 تقريباً. وقد تختلف هذه القيم قليلاً على اللوحة الخاصة بك. جرب البرنامج التالي وقم بتحميله إلى ميكرو.

```

Arduino run first:
Enable Acceleration Sensor TRUE
set threshold to 25 as Decimal

Arduino loop forever:
set x to Read Acceleration of Axis X 33.5
set y to Read Acceleration of Axis Y 85
if x > threshold
do Draw Arrow East
wait 100 milliseconds
else if x < -1 * threshold
do Draw Arrow West
wait 100 milliseconds
else Clear LEDs
if y > threshold
do Draw Arrow South
wait 100 milliseconds
else if y < -1 * threshold
do Draw Arrow North
wait 100 milliseconds
else Clear LEDs
    
```



نشاط 23: الكشف عن الإمالة باتجاه محور فردي

في النشاط التالي، ستقوم بإنشاء برنامج للكشف عن الإمالة في اتجاه المحور X. وسوف يظهر الاتجاه على شكل سهم (شرق أو غرب) باستخدام شبكة LED.

في هذا النشاط، ستحتاج أولاً إلى تمكين مستشعر التسارع، ثم تحديد المتغير الخاص بقيمة العتبة وتعيينه إلى قيمة صغيرة، على سبيل المثال 25. بما أن مقياس التسارع يُرجع قيم عشرية بكسور، فيجب أن يكون نوع المتغير يقبل هذا النوع من القيم أيضاً.

في الحلقة الرئيسية، اقرأ المستشعر في الاتجاه المطلوب. إذا كانت هناك إزاحة، فسوف تحتاج إلى تعويضها. بعد عدة محاولات تجريبية، تم العثور على الإزاحة في محور X بمقدار 34 تقريباً. وقد تختلف هذه القيمة قليلاً على اللوحة الخاصة بك.

بمجرد قراءة القيمة وتعويض الإزاحة إن وجدت، فأنت تحتاج إلى مقارنة القيمة المُقاسة بقيمة العتبة. إذا كانت القيمة أكبر من مقدار قيمة العتبة بالاتجاه الموجب، فقم برسم السهم الشرقي. وإذا كانت أصغر من قيمة العتبة بالاتجاه السالب، ارسم السهم الغربي. بخلاف ذلك، قم بإطفاء شبكة LED للإشارة إلى عدم وجود إمالة. جرب البرنامج التالي وقم بتحميله إلى ميكرو.

```

Arduino run first:
Enable Acceleration Sensor TRUE
set threshold to 25 as Decimal

Arduino loop forever:
set x to Read Acceleration of Axis X 33.5
if x > threshold
do Draw Arrow East
else if x < -1 * threshold
do Draw Arrow West
else Clear LEDs
wait 100 milliseconds
    
```

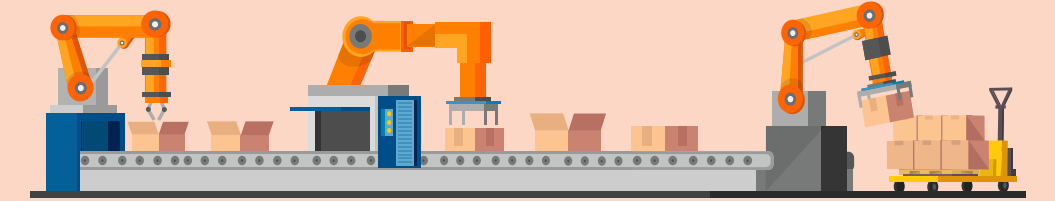

سوف يبدو البرنامج كما يلي.

جرب البرنامج السابق وقم بتحميله إلى ميكرو.

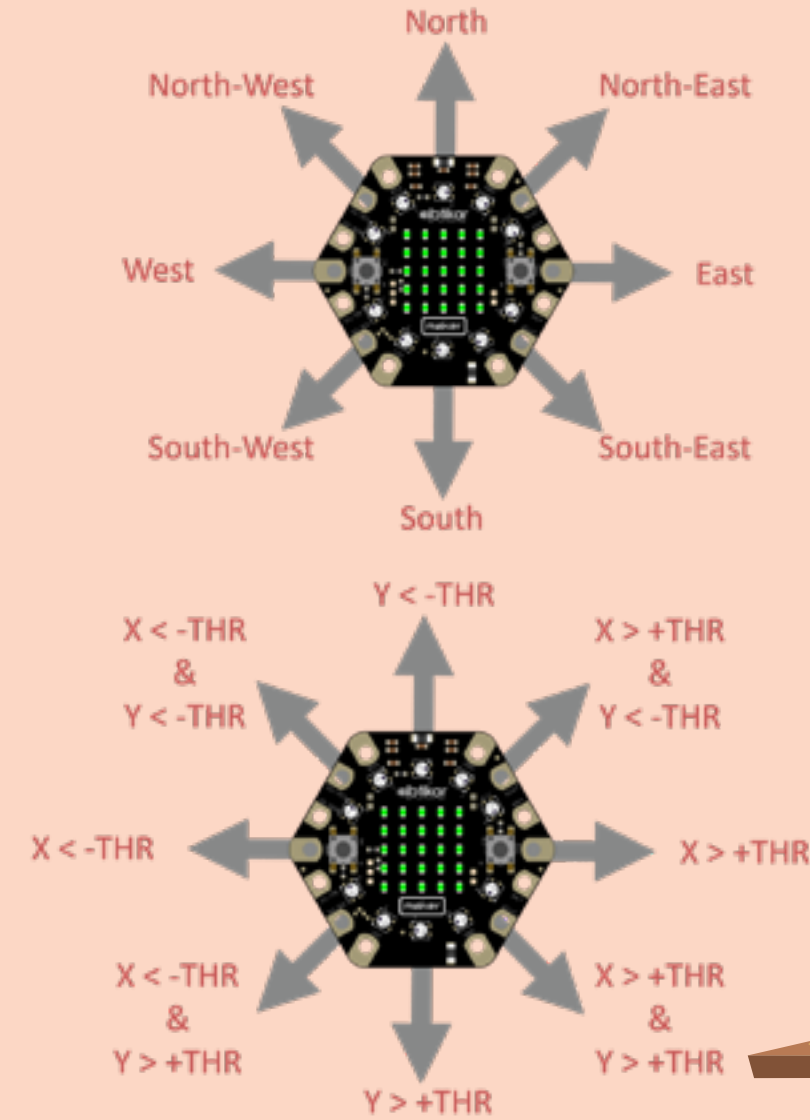


نشاط 25: الكشف عن الإمالة باتجاه عدة محاور (8 اتجاهات)

في هذا النشاط، ستقوم بإنشاء برنامج للكشف عن الإمالة في المحورين X و Y. سوف يظهر الاتجاه على شكل سهم (شمال، شمال شرق، شرق، جنوب شرق، جنوب، جنوب غرب، غرب، أو شمال غرب) باستخدام شبكة LED.



مثل النشاط السابق، سوف تقوم بإنشاء قيمة العتبة؛ وسوف تكون هذه القيمة واحدة لكلا الاتجاهين. قيم الإزاحة هي أيضا واحدة لكلا المحورين كما في النشاط السابق.





TM
maker



ibtikar
EDU TECH SOLUTIONS