دليل لوحة AFF IoT

الإصدار 1.0.2

دليل لوحة AFF IoT

دليلك إلى لوحة AFF IoT

مرحبا بك في *الدليل إلى لوحة AFF IoT*

يحتوي هذا الكتيب على المعلومات اللازمة لاستكشاف اللوحة الإلكترونية AFF IoT.
عند الانتهاء من هذا الكتيب ، سوف تعرف كيفية البدء في إنشاء مشاريعك وتجاربك الخاصة.

هيا نبدأ!

جدول المحتوى

1 استكشاف لوحة AFF IoT
الصفحة 5

2 نظرة عامة على نظام الاردوينو
الصفحة 9

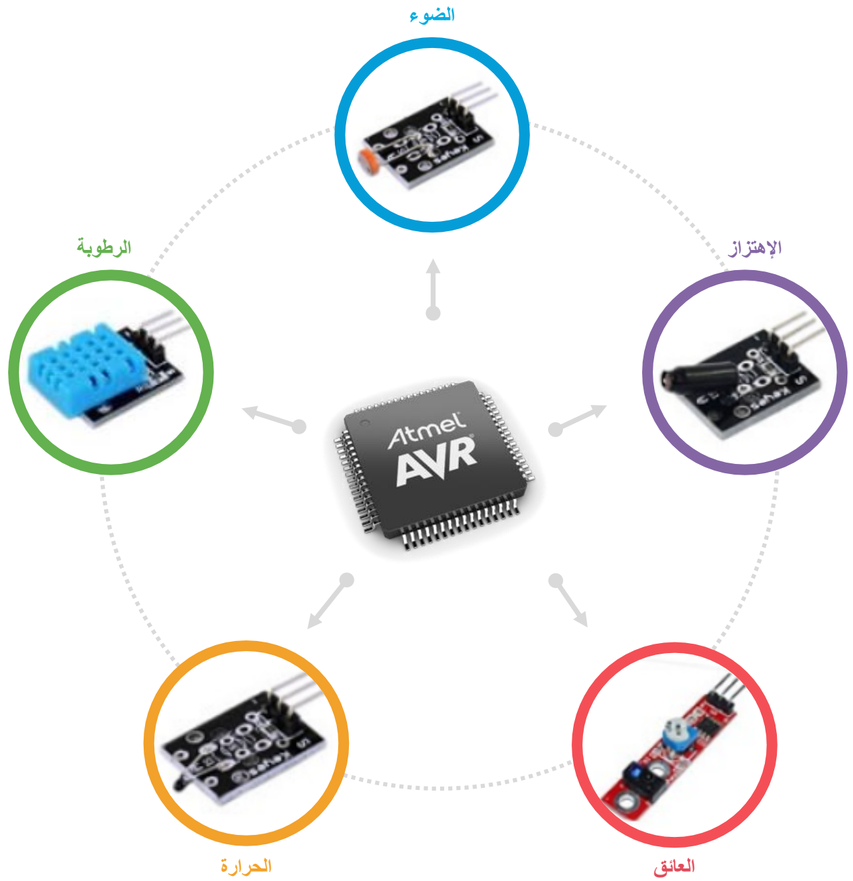
3 تعلم أساسيات البرمجة
الصفحة 13

4 التسجيل في تطبيق AFF
الصفحة 17

5 إنجاز 12 مشروعا
الصفحة 25

مفاهيم أساسية

وجه التشابه بين الدماغ البشري و المتحكم المصغر



المخ هو عضو يعمل كمركز للجهاز العصبي. يجمع البيانات من البيئة باستخدام أعضاء الحواس؛ يعالج البيانات التي تم جمعها وفقًا للاحتياجات الحالية جنبًا إلى جنب مع الظروف السابقة المماثلة ، ثم يرسل أوامر لأداء إجراءات وفقًا لذلك (المشي والتحدث وما إلى ذلك)

المتحكم المصغر هو جهاز كمبيوتر مدمج في دارة واحدة، مكرس لأداء وتنفيذ بعض المهام. فهو يحتوي على ذاكرة، وعلى أجهزة جانبية قابلة للبرمجة لاستقبال أو إخراج المعلومات بالإضافة إلى الدماغ الإلكتروني. فهو يجمع البيانات من البيئة باستخدام أجهزة الاستشعار(الحساسات). يعالج البيانات التي تم جمعها وفقا لاحتياجات المبرمج، ثم يقوم بتنفيذ الإجراءات بناء على ذلك (التحكم في المحرك، تشغيل الضوء، إلخ.)

ماهي لوحة AFF IoT ؟

- يتكون إنترنت الأشياء (IoT) من أشياء متصلة بالإنترنت، يتم التحكم فيها ومراقبتها في أي وقت وفي أي مكان. تقنيا، يتكون من دمج لأجهزة الاستشعار وأجهزة متصلة بالإنترنت عبر الشبكات الثابتة أو اللاسلكية.
- تعد أجهزة إنترنت الأشياء جزء من المفهوم الأكبر: المنزل الذكي. ويشمل الإضاءة، التدفئة، تكييف الهواء، الوسائط و أنظمة الأمن. ويعتبر توفير الطاقة عن طريق ضمان إيقاف تشغيل الأضواء والإلكترونيات تلقائيًا عند عدم استخدامها, من بين الفوائد طويلة المدى لتوظيف أنترنيت الأشياء.
- تعد لوحة AFF IoT عبارة عن متحكم دقيق متوافقة مع لوحة الأردوينو و تتضمن وحدة Wi-Fi مدمجة.

لوحة IoT AFF

المكونات

1 دخل الطاقة (الموصل الأسطواني)

يوفر الطاقة للوحة، ويعمل مع المحول الكهربائي أو البطارية ذات جهد 9 فولت أو 12 فولت. يتراوح نطاق جهد المدخل المقبول من 7 فولت إلى 15 فولت

2 منفذ USB

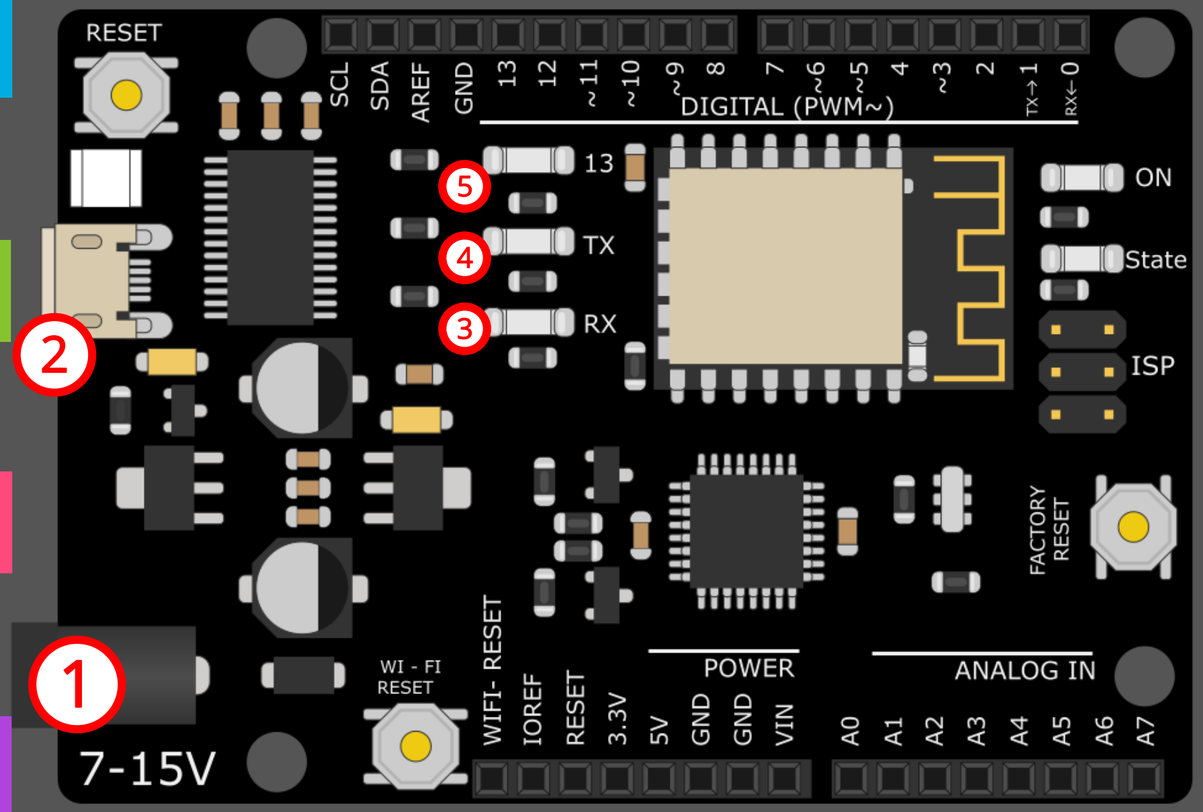
يوفر الطاقة ويربط اللوحة بالكمبيوتر عبر USB.

3-4 الصمام الثنائي المضيء (الاستلام: RX / الإرسال: TX)

تضيء هذه المصابيح عندما يستقبل \ يرسل المتحكم المصغر البيانات من \ إلى الكمبيوتر.

5 الصمام الثنائي المضيء 13 (صمام ثنائي مضيئ مدمج)

يرتبط هذا الصمام بالبرنامج لإظهار ما إذا كان البرنامج يعمل بشكل صحيح أم لا. و هو موصول عبر مقاومة بالمأخذ الرقمي 13.



لوحة IoT AFF

المكونات

6 مآخذ (الأرضي, الرقمي, الإستقبال, الإرسال)

تستخدم كمآخذ مختلفة للمدخلات والمخرجات الرقمية، والاتصالات التسلسلية.

7 الصمام الثنائي المضيء (الأحمر و الأزرق)

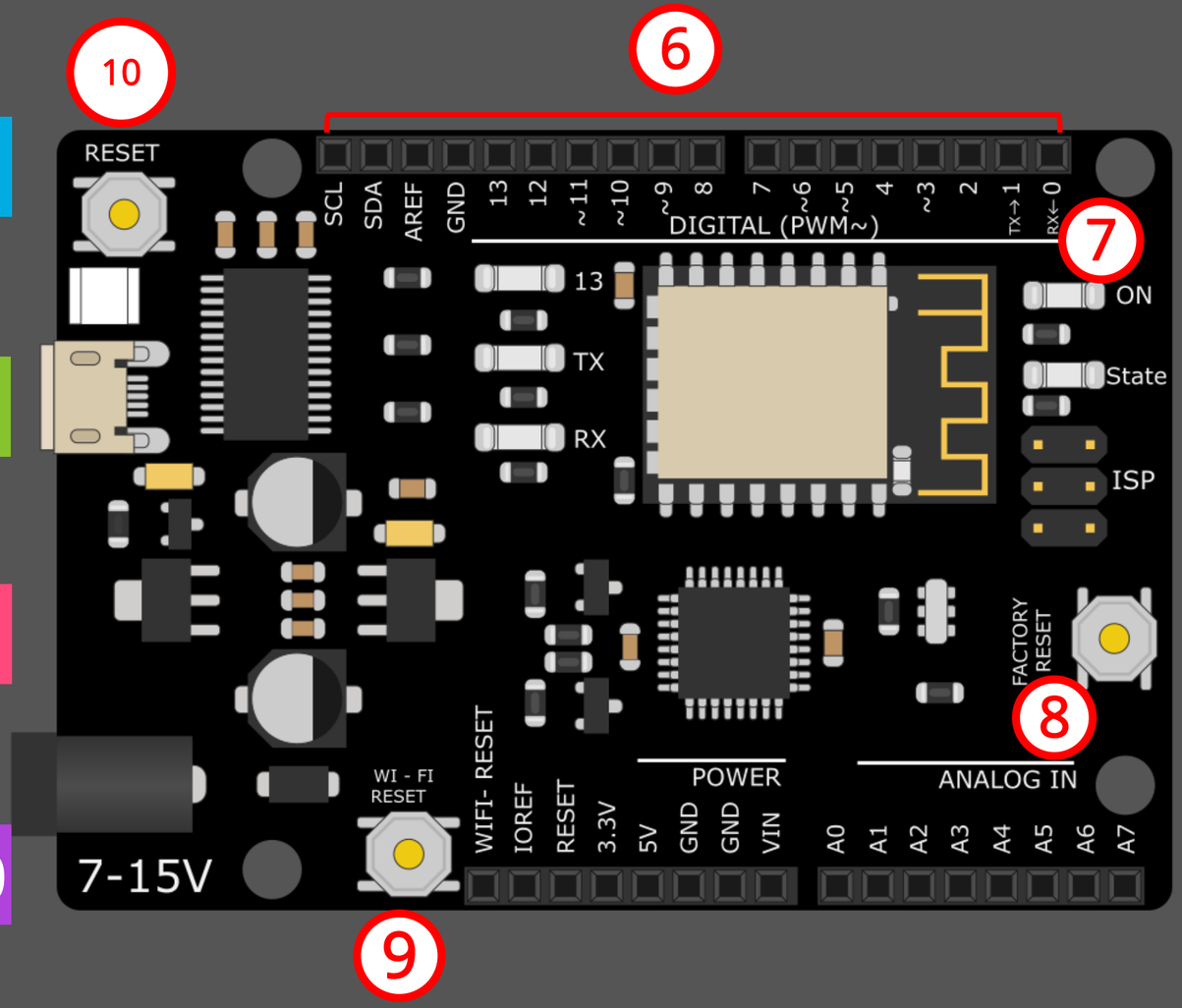
يعتبر الصمام الأحمر مؤشرا للطاقة. بينما يشير الأزرق إلى حالة اللوحة (مهيئة لتعمل أم لا)

8 زر إعادة ضبط "المصنع"

يعيد ضبط اللوحة إلى إعدادات المصنع، حيث يحذف بيانات الواي فاي وبيانات الحساب من اللوحة. (اضغط لمدة 5 ثوان لإعادة الضبط)

9-10 زر إعادة تشغيل الواي فاي وزر إعادة التشغيل

يقوم الأول بإعادة تشغيل وحدة الواي فاي الخاصة باللوحة ويعيد الاتصال بالخادم (الإنترنت)، والثاني يعيد تشغيل المتحكم المصغر (إعادة تشغيل الكود) (نقرة واحدة).



لوحة AFF IoT

المكونات

11 مآخذ (المدخل التماثلي, الطاقة)

تستعمل مآخذ مختلفة للمدخلات التماثلية, للمخرج (3,3 فولت و5 فولت) و المأخذ الأرضي.

12 وحدة الواي فاي ESP

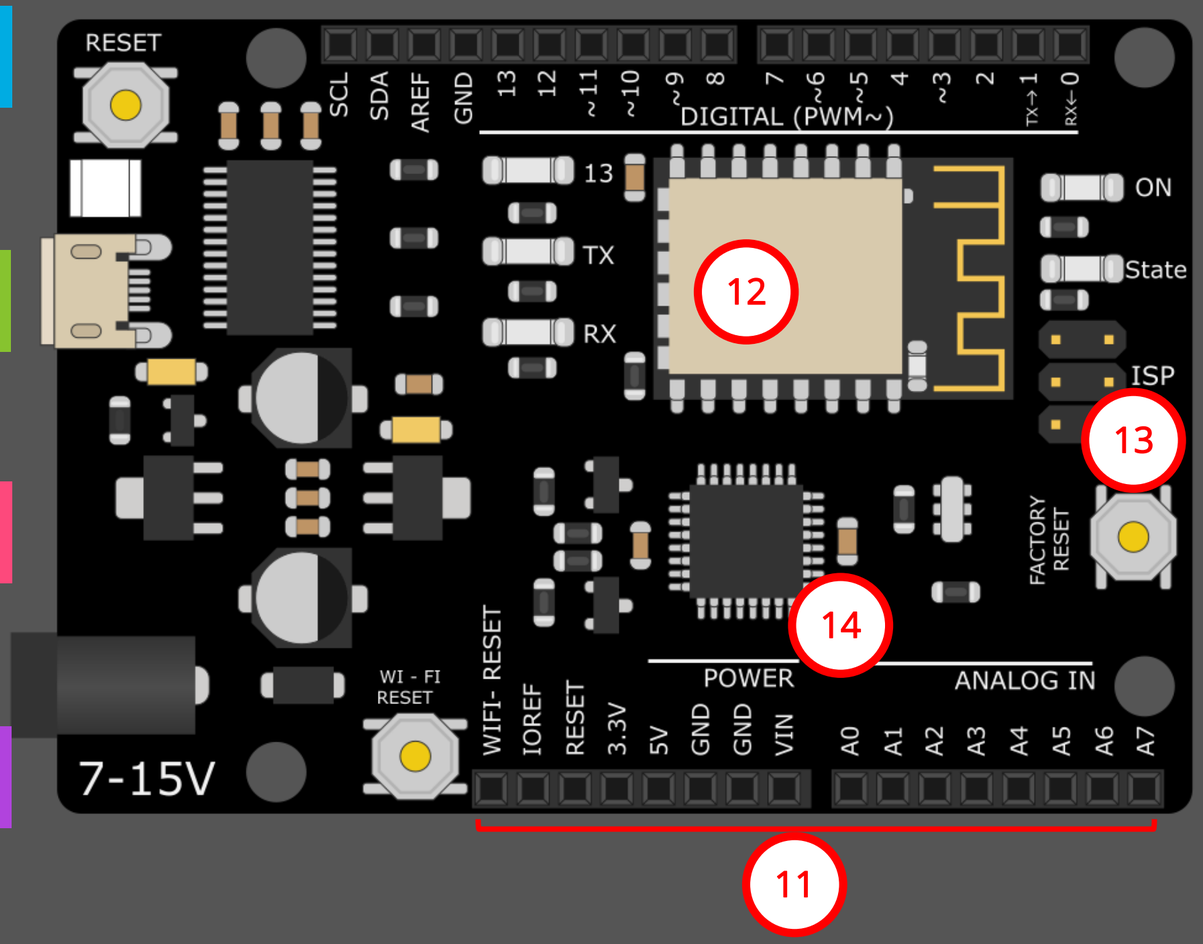
عبارة عن متحكم دقيق مع بروتوكول TCP/IP مدمج فيه لتمكين اللوحة من الإتصال بالإنترنت.

13 مآخذ مدخلات دائرة البرمجة التسلسلية

تستعمل لبرمجة اللوحة عبر مجموعة أدوات البرمجة.

14 المتحكم المصغر AVR

وهو المتحكم المصغر للوحة AFF IoT, وهي مماثلة للوحة الأردوينو أونو التقليدية.



تحميل برنامج الأردوينو

الاتصال بالإنترنت

1

قم بتحميل آخر إصدار لبرنامج الأردوينو (مجانا) من الموقع www.arduino.cc

أكتب الرابط التالي في شريط العناوين للمتصفح

https://www.arduino.cc/en/Main/Software

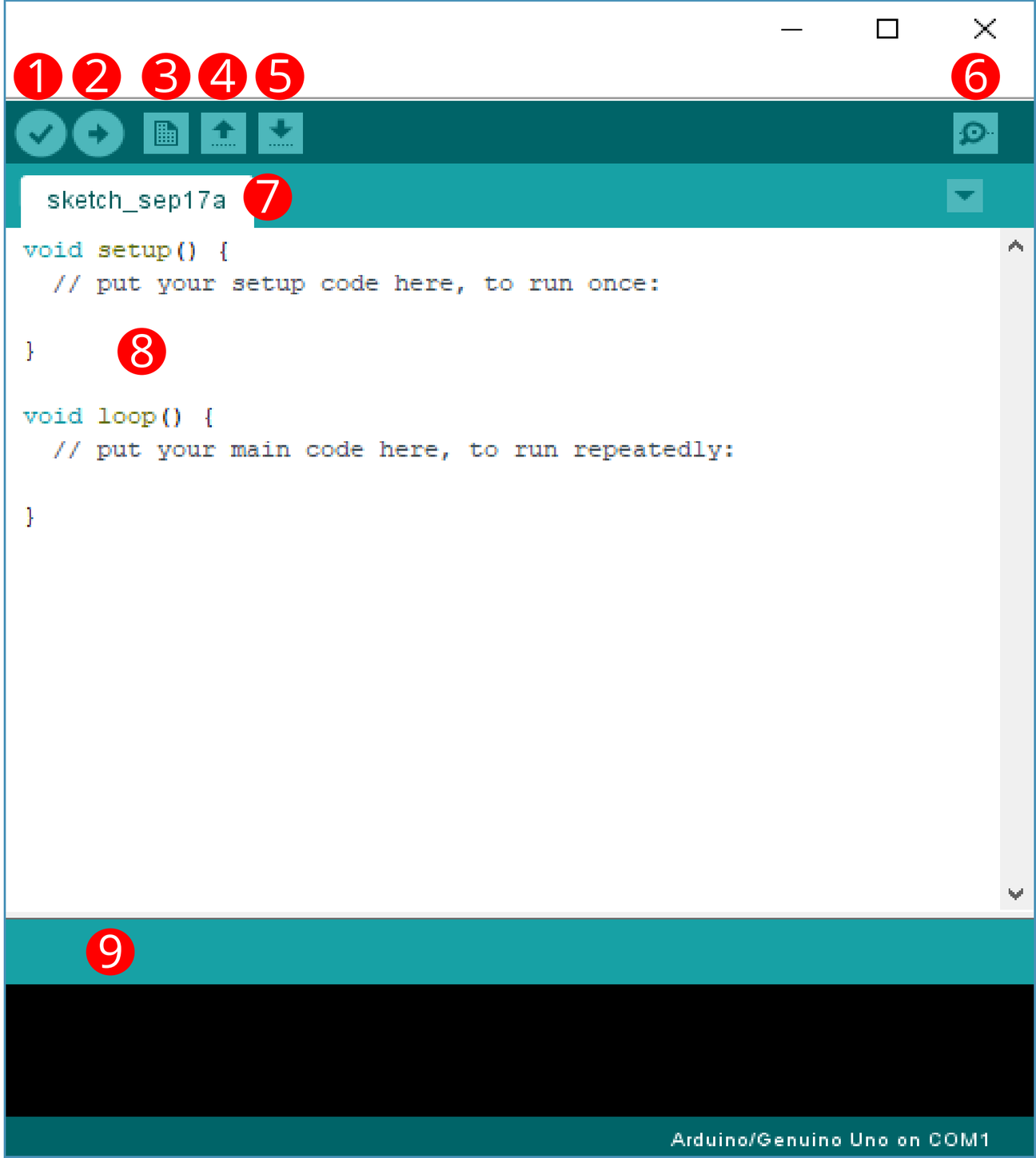
2

تحميل

قم باختيار حزمة تثبيت نظام التشغيل المناسبة (الويندوز, الماك أو لينوكس).

حول برنامج الأردوينو

قم بفتح برنامج الأردوينو في حاسوبك وسترى صفحة مماثلة لهذه التي على الجانب.



حول برنامج الأردوينو

1 تحقق

الترجمة والتحقق من الكودات البرمجية، لاستكشاف الأخطاء كنسيان الأقواس أو الفاصلة المنقوطة.

2 رفع

إرسال الكودات البرمجية الى لوحة AFF. عند الضغط عليه يومض الصمامان الأصفران (الأرسال و الاستقبال) بسرعة.

3 جديد

فتح نافذة تعليمات برمجية جديدة والتي تضم الوظيفتين: `setup()` و `loop()`، و سنتطرق إلى شرحهما فيما بعد.

4 فتح

فتح مسودة قديمة.

5 حفظ

حفظ المسودة الحالية.

6 الشاشة التسلسلية

فتح نافذة تظهر كل المعلومات التسلسلية التي ترسلها لوحة AFF IoT. وهي مهمة جدا لتصحيح كودات البرمجة.

7 إسم المسودة

عرض اسم المسودة.

8 منطقة البرمجة

وهي المنطقة المخصصة لتركيب الكودات البرمجية في المسودة.

9 مساحة الرسائل

تقوم بإعلام المبرمج بوجود أخطاء في كودات البرمجة, كما تخبره بنجاح عملية التحقق أو الرفع.

```
void setup() {
  // put your setup code here, to run once:

}
```

وظيفة setup()

تستعمل هذه الوظيفة لإعداد جميع ترتيبات المآخذ والواجهات وكذا القيم البدائية للمتغيرات المستعملة.

تعمل هذه الوظيفة عند تشغيل اللوحة و عند الضغط على زر إعادة التشغيل.

```
void loop() {
  // put your main code here, to run repeatedly:

}
```

وظيفة loop()

تعمل كل الكودات البرمجية المكتوبة داخل هذه الوظيفة مرارا وتكرارا طالما أن اللوحة مزودة بالطاقة.

تستعمل هذه الوظيفة لتنفيذ الكودات البرمجية بدون توقف.

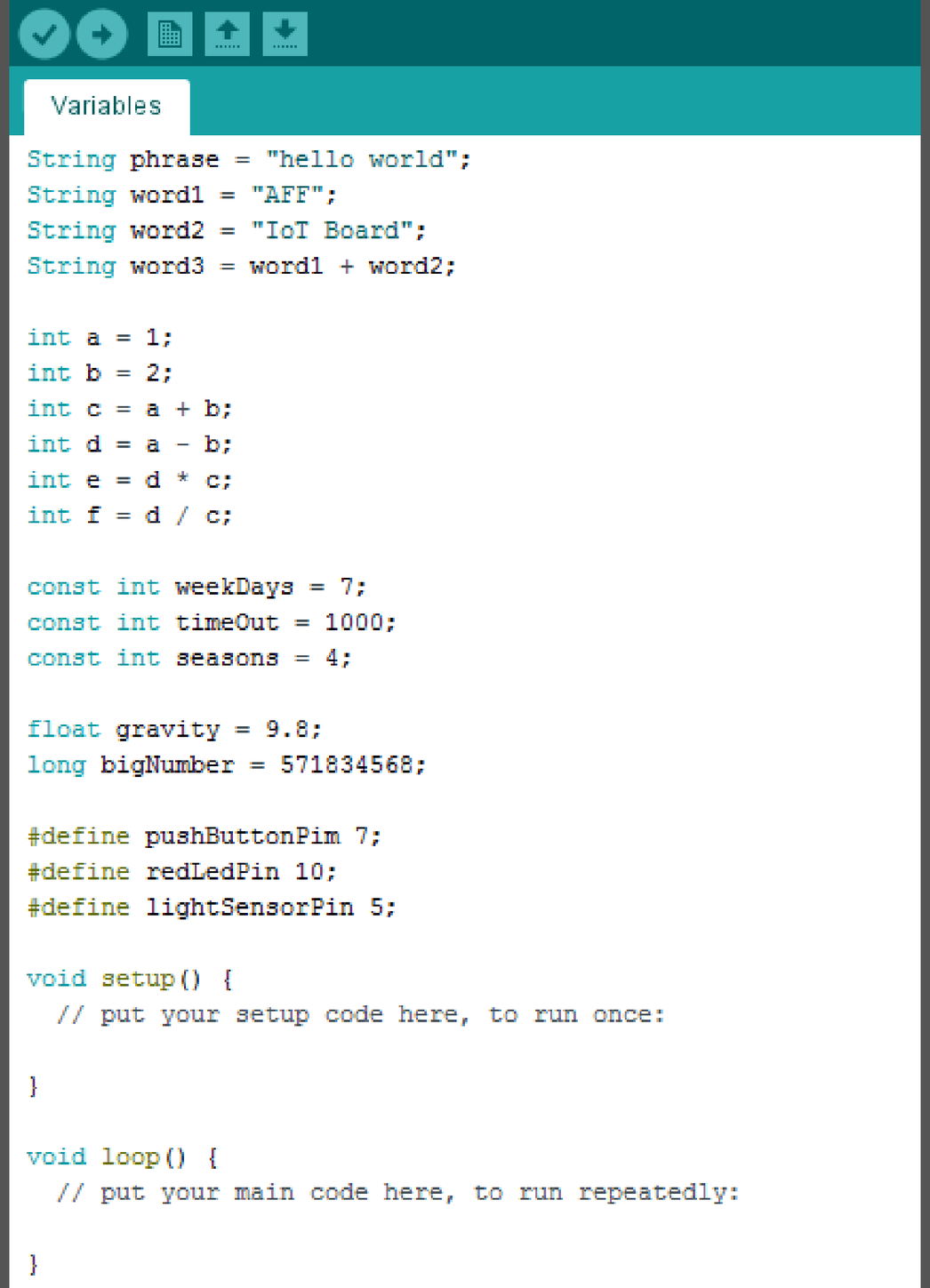
الوظائف الأساسية للأردوينو

```
//هذه من أجل التعليق في السطر الواحد
//من الجيد وضع وصف في البداية
//وقبل أي شيء صعب
```

```
/* هذا للتعليق متعدد الأسطر
   مثل هذا...
   و هذا...*/
```

تعلم أساسيات البرمجة و الكود

نشرح في القسم التالي جميع مفاهيم البرمجة اللازمة للعمل بلوحة AFF IoT بالإضافة إلى بعض الأمثلة التوضيحية.



المتغيرات العشرية و الطويلة

- تعرف المتغيرات العشرية باستعمال النوع "float".
- المتغيرات الطويلة مماثلة للمتغيرات الصحيحة غير أنها تحمل قيما كبيرة.
- يتراوح مجال تعريف المتغيرات الطويلة ما بين 2147483647- و 2147483646+

```
float gravity = 9.8;

long bigNumber = 571834568;
```

المتغيرات المعرفة

- هي مماثلة للمتغيرات الثابتة.
- تستعمل عادة لربط أرقام المآخذ بالمكونات الفيزيائية.
- لا حاجة للتعريف بنوعها, يمكن ربطها مباشرة.
- وهي تعرف دون استعمال الرمز "=".

```
#define pushButtonPim 7;

#define redLedPin 10;

#define lightSensorPin 5;
```

المتغيرات الثابتة

- هذا النوع من المتغيرات يحمل قيمة ثابتة لا يمكن تغييرها في أي مكان من الكودات البرمجية.
- يمكن التعريف بها بإضافة الإسم "const" قبل نوعها.
- يتسبب تغيير قيمة المتغيرات الثابتة بخطأ في التحقق.

```
const int weekDays = 7;

const int timeOut = 1000;

const int seasons = 4;
```

المتغيرات السلسلة

- هذا النوع من المتغيرات معد لتخزين حروف و رموز تكون كلمة أو جملة.
- ومثل المتغيرات الصحيحة, يمكن دمج المتغيرات السلسلة معا لتكوين سلسلة أخرى باستعمال الرمز "+".

```
String phrase = "hello world";

String word1 = "AFF";

String word2 = "IoT Board";

String word3 = word1 + word2;
```

المتغيرات الصحيحة

- المتغيرات عبارة عن أوعية تحمل الأعداد أو القيم مؤقتا.
- كل متغير يمكن أن يحمل قيمة معينة باستعمال الرمز "=".
- كل العمليات الحسابية يمكن استعمالها, و يمكن أن توضع نتائجها في متغير آخر.

```
int a = 1;

int b = 2;

int c = a + b;

int d = a - b;

int e = d * c;

int f = d / c;
```

مفاهيم عامة

وظائف و مكتبات

المكتبات

```
#include <Servo.h>

#include <EEPROM.h>

#include <NeoSWSerial.h>
```

المكتبات هي عبارة عن ملفات مكتوبة بلغة C أو C++ توفر لكوداتك وظائف إضافية (مثل القدرة على التحكم في محرك السيرفو، أو قراءة حساس، إلخ). تجعل هذه العبارات الوظائف والمتغيرات الثابتة التي تحددها المكتبة متاحة للإستخدام في المسودة.

- لاستخدام مكتبة موجودة في المسودة، ما عليك سوى الانتقال إلى قائمة "Sketch"، واختر "Import Library" ، واختر من المكتبات المتاحة. سيؤدي ذلك إلى إدراج عبارة #include في الجزء العلوي من المسودة لكل ملف h. (في مجلد المكتبة).

الوظائف

```
void function(){
  // read sensors
  // test conditions
  // perform actions
}

int add(int a, int b){
  int c = a + b;
  return c;
}
```

- الوظيفة عبارة عن مجموعة من التعليمات التي تنفذ مهمة.
- وهي تأخذ بعض العوامل (a و b كما في المثال) و تنجز بعض العمليات (الجمع في المثال) ثم تخرج النتيجة (وهي جمع a و b في c).
- كما أنها من الممكن ألا تأخذ أي عوامل وألا تخرج أي قيمة (وتعرف ب void function) مثل الوظيفة print.

تعليمات التحكم

البيان الأكثر أهمية

تتحقق عبارة `if` من الشرط وتنفذ العبارة التي تليه إذا كان الشرط صحيحا.
يمكن حذف الأقواس بعد عبارة `if` فيصبح السطر التالي (المعرّف بواسطة الفاصلة المنقوطة) هو البيان الشرطي الوحيد.
البيانات التي يتم تقييمها داخل الأقواس تتطلب استخدام عامل واحد أو أكثر كما هو موضح أدناه.
عوامل المقارنة:

```
x==y   (x تساوي y)
x ! =y  (x تساوي y)
x <y   (x أقل من y)
x > y   (x أكبر من y)
x <= y  (x أقل من أو تساوي y)
x > = y  (x أكبر من أو تساوي y)
```

```
if (condition)
{
  //statement(s)
}

if (x > 120) digitalWrite(LEDpin, HIGH);

if (x > 120)
  digitalWrite(LEDpin, HIGH);

if (x > 120) {
  digitalWrite(LEDpin, HIGH);
}

if (x > 120) {
  digitalWrite(LEDpin1, HIGH);
  digitalWrite(LEDpin2, HIGH);
}
```

احذر من استخدام علامة المساواة الفردية عن طريق الخطأ (على سبيل المثال `if(x = 10)` علامة المساواة الفردية هي عامل تعيين `x` داخل `10` (تضع القيمة `10` في المتغير x).

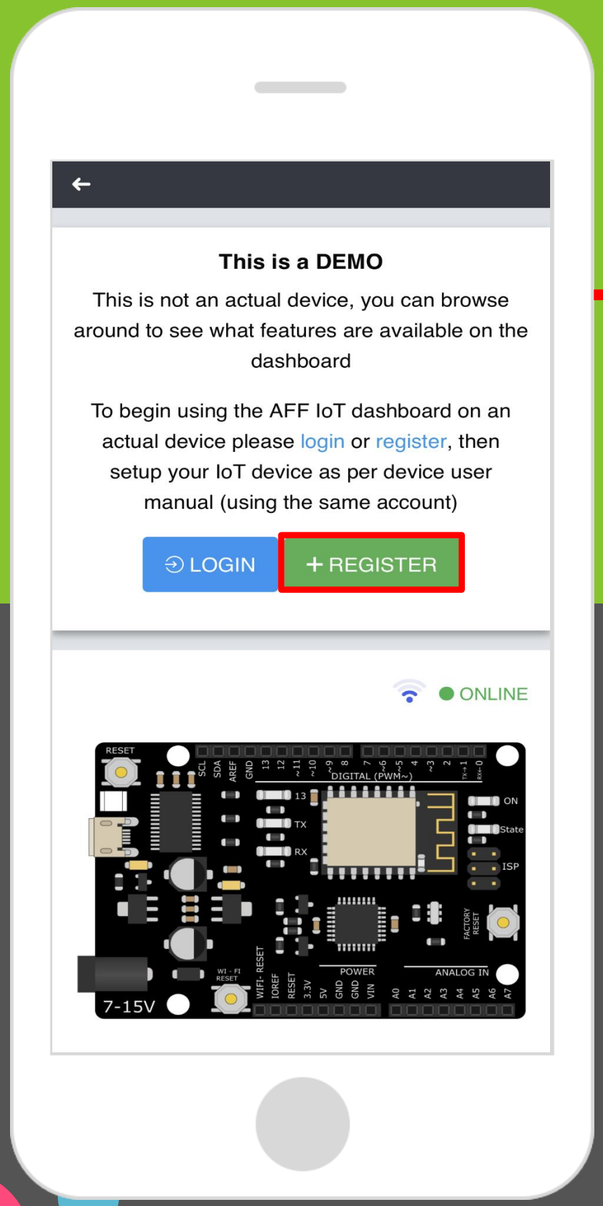
بدلاً من ذلك ، استخدم علامة المساواة المزدوجة (على سبيل المثال `if(x == 10)` و هو عامل المقارنة ، ويختبر ما إذا كانت x تساوي `10` أم لا.

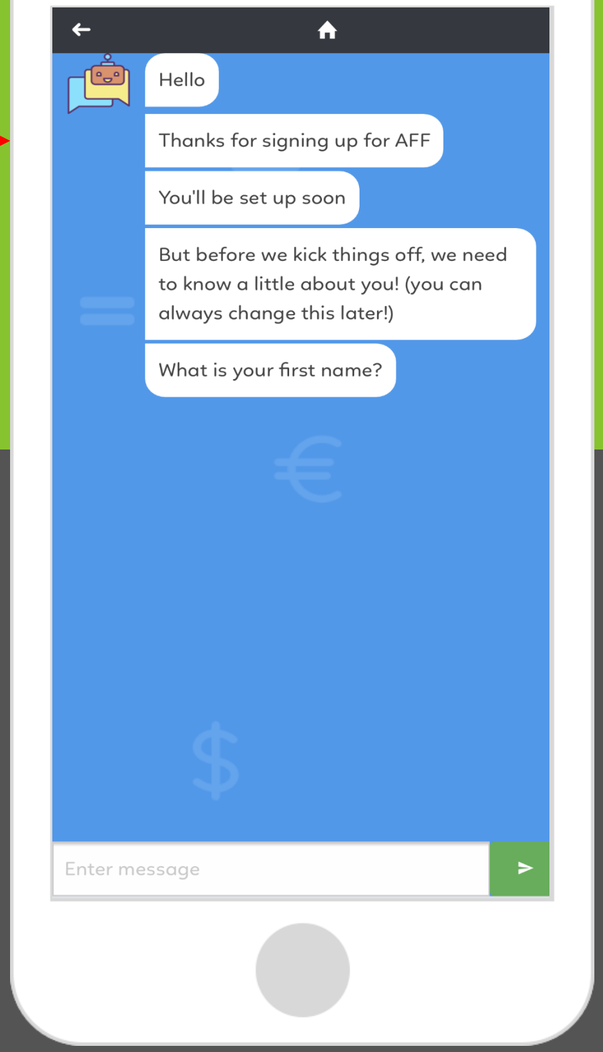
العبارة الأخيرة صحيحة فقط إذا كانت x تساوي `10` ولكن العبارة السابقة ستكون صحيحة دائمًا.

التسجيل في تطبيق AFF IoT

بعد تحميل تطبيق (AFF IoT), قم بإنشاء حساب جديد.
يحتاج إنشاء الحساب عشرة خطوات ولا يستغرق إلا دقيقتين.
إذا كنت لا تملك هاتفا ذكيا, يمكنك استعمال متصفح الأنترنيت في الكمبيوتر مثل كروم أو فايرفوكس, إلخ...

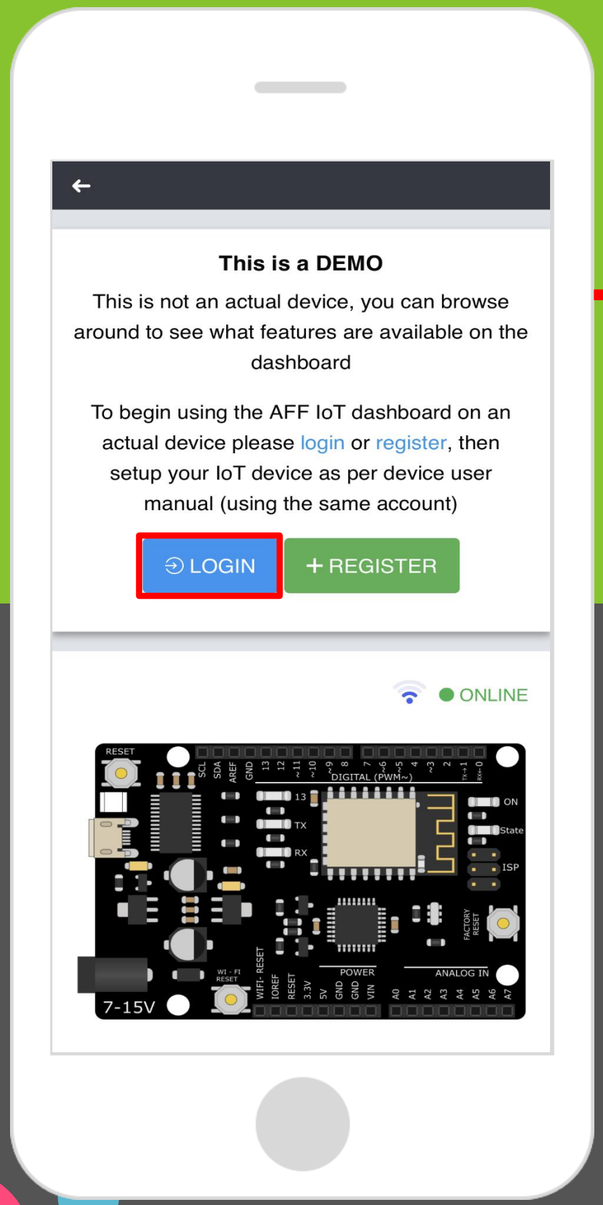
* إذا واجهتك مشكلة عند إنشاء حساب جديد, أو عند الدخول إليه,
فيمكنك مراسلتنا على *support@affcity.com*

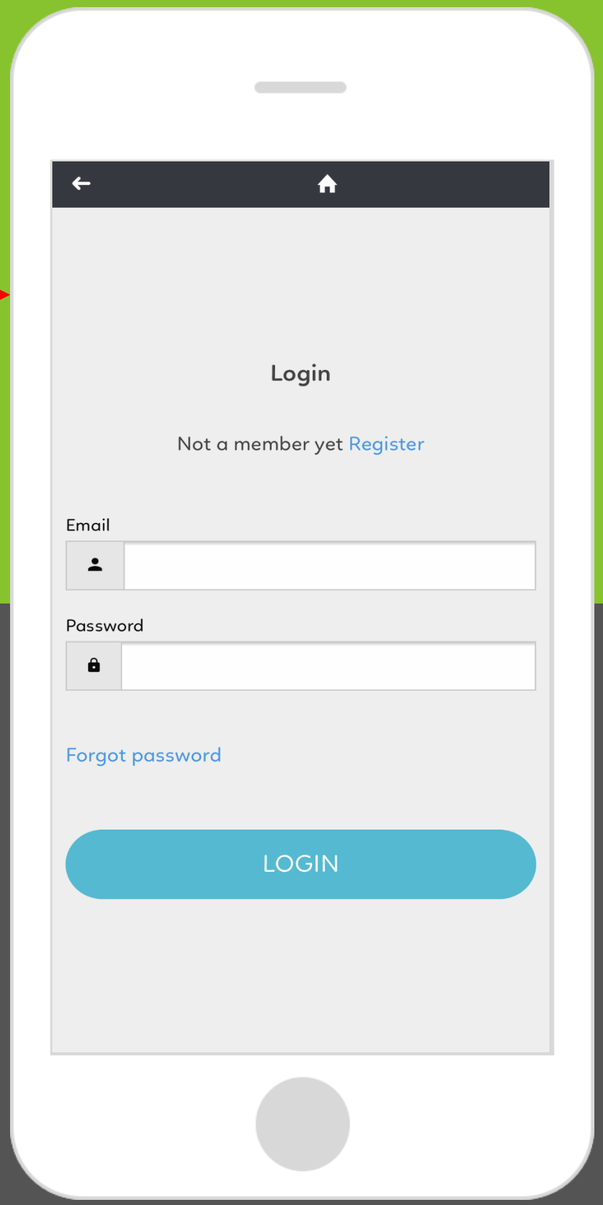




تطبيق AFF IoT

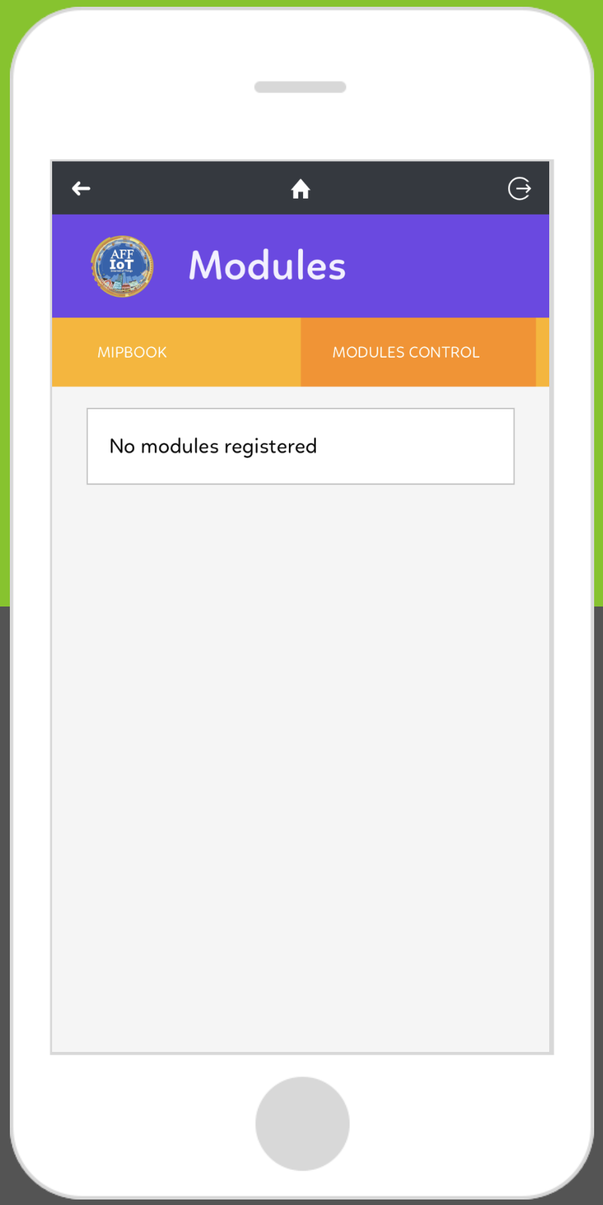
1. أنشئ حسابا بالضغط على تسجيل "REGISTER".
2. قم بملئ المعلومات المطلوبة.
3. بعدها قم بالدخول باستعمال البريد الإلكتروني وكلمة السر.





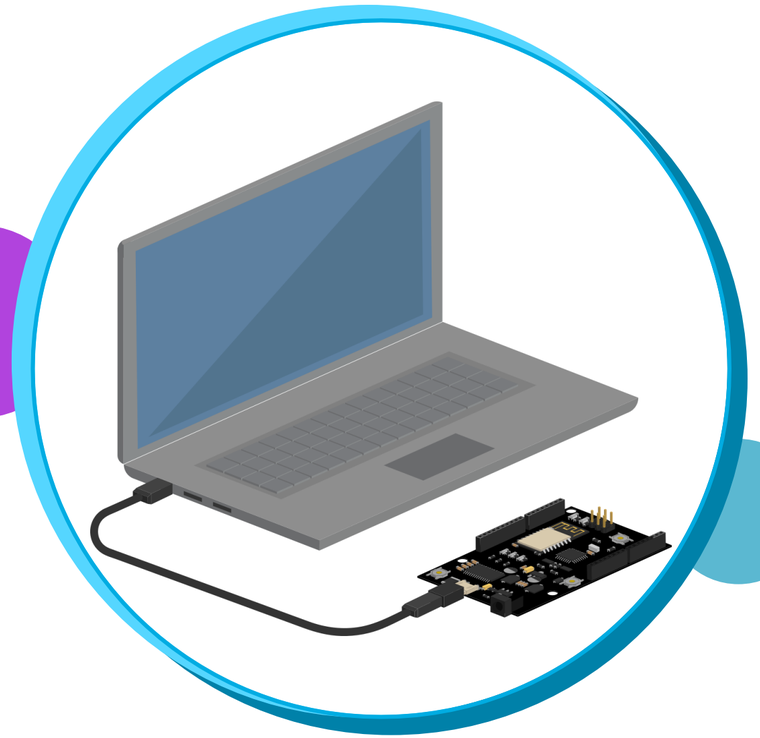
تطبيق AFF IoT

1. إذا كنت تملك حسابا مسبقا, إضغط على الزر الأزرق "LOGIN"
2. بعدها قم بالدخول باستعمال البريد الإلكتروني وكلمة المرور.



تطبيق AFF IoT

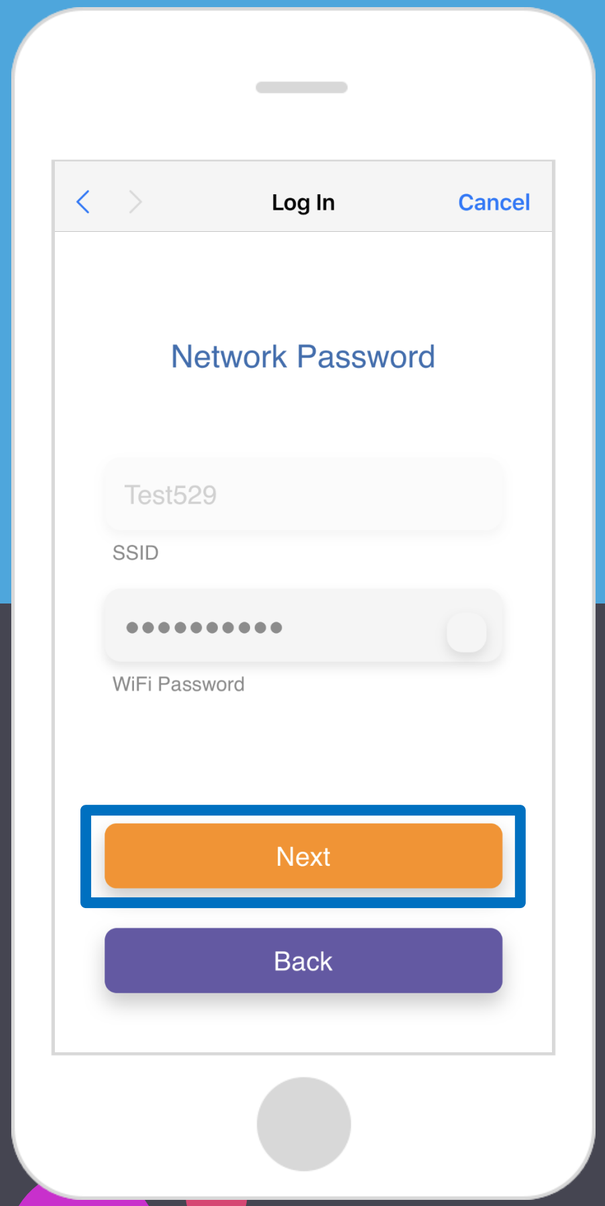
بعدها, قم بتسجيل لوحة AFF IoT (الشرح في الفقرة التالية). بعدها ستكون مستعدا لإنشاء مشاريعك !



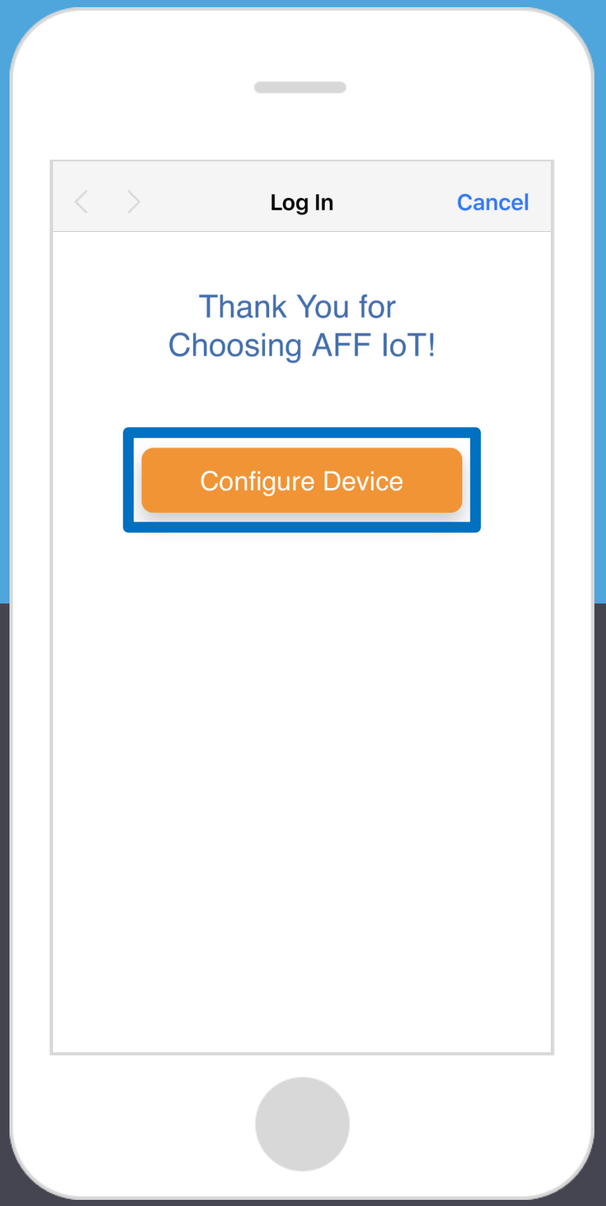
قم بتوصيل لوحة AFF IoT بالكمبيوتر

إستعمل كابل micro USB لتوصيل لوحة AFF IoT بأي منفذ من منافذ USB بالكمبيوتر.

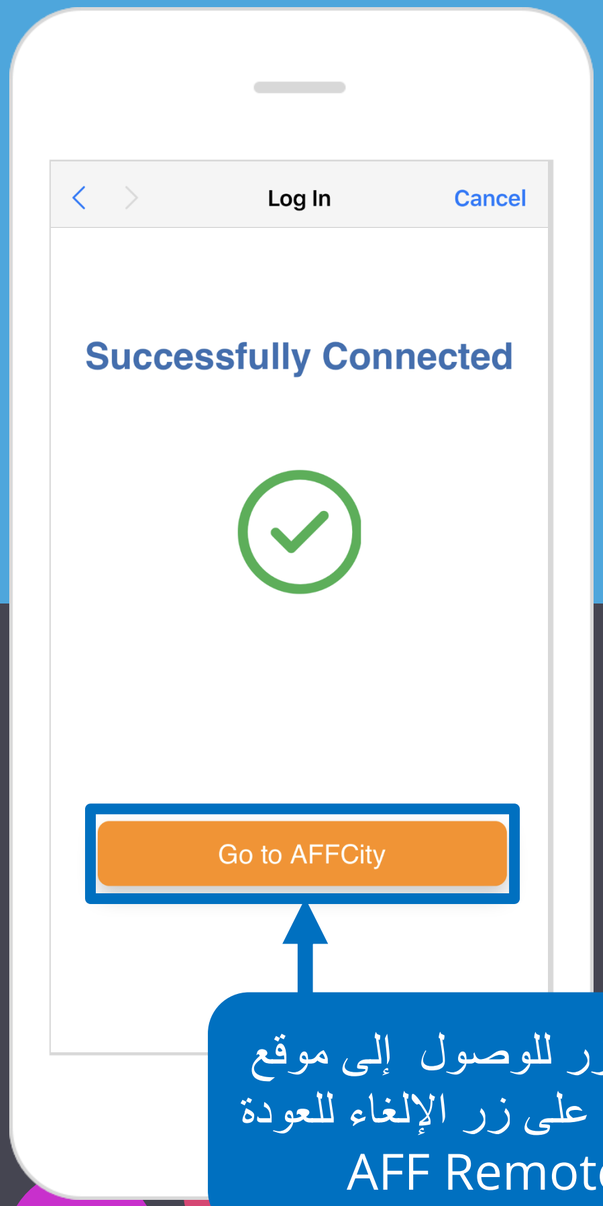
ثم قم بفتح صفحة الوايفاي بهاتفك الذكي أو الكمبيوتر.

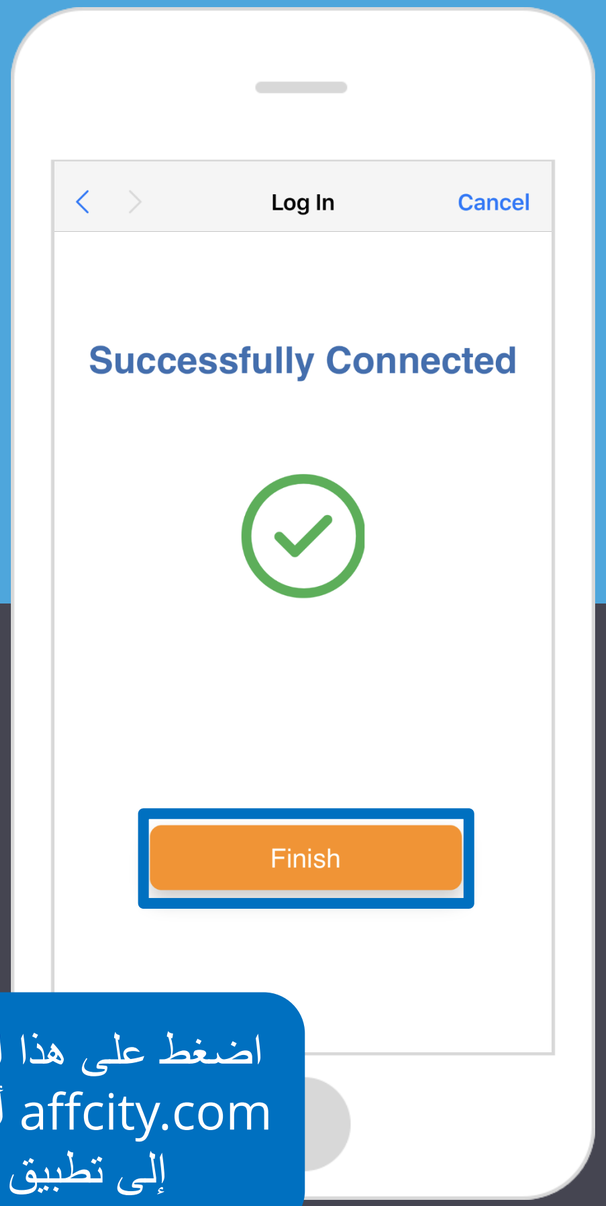


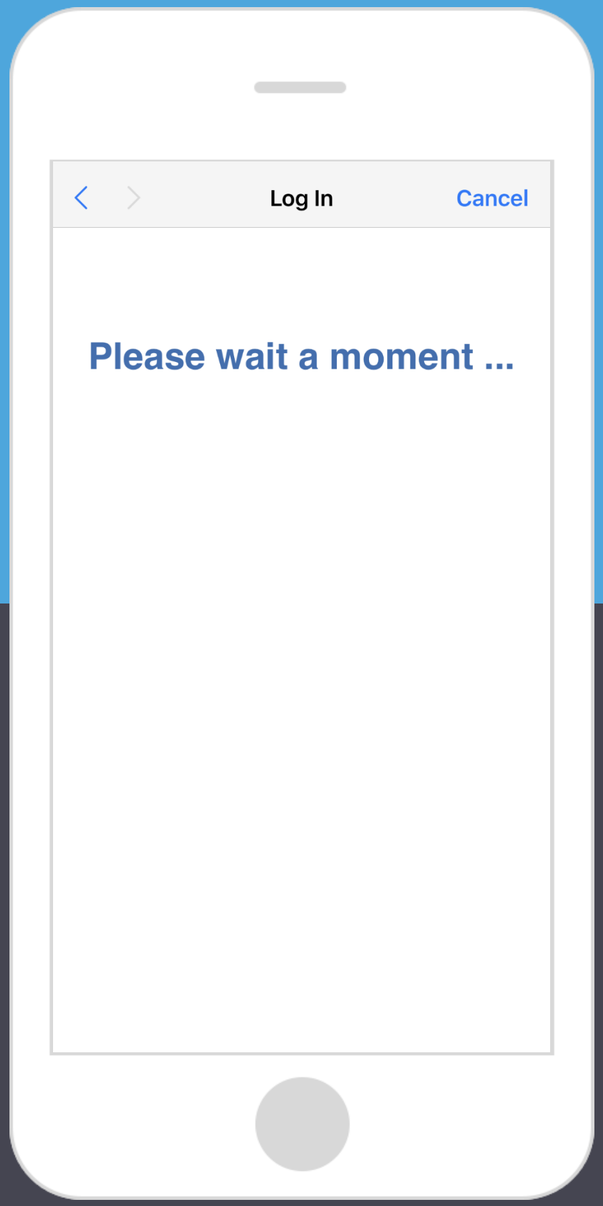
إذا لم تكن شبكتك في هذه اللائحة
فاضغط على هذا الزر

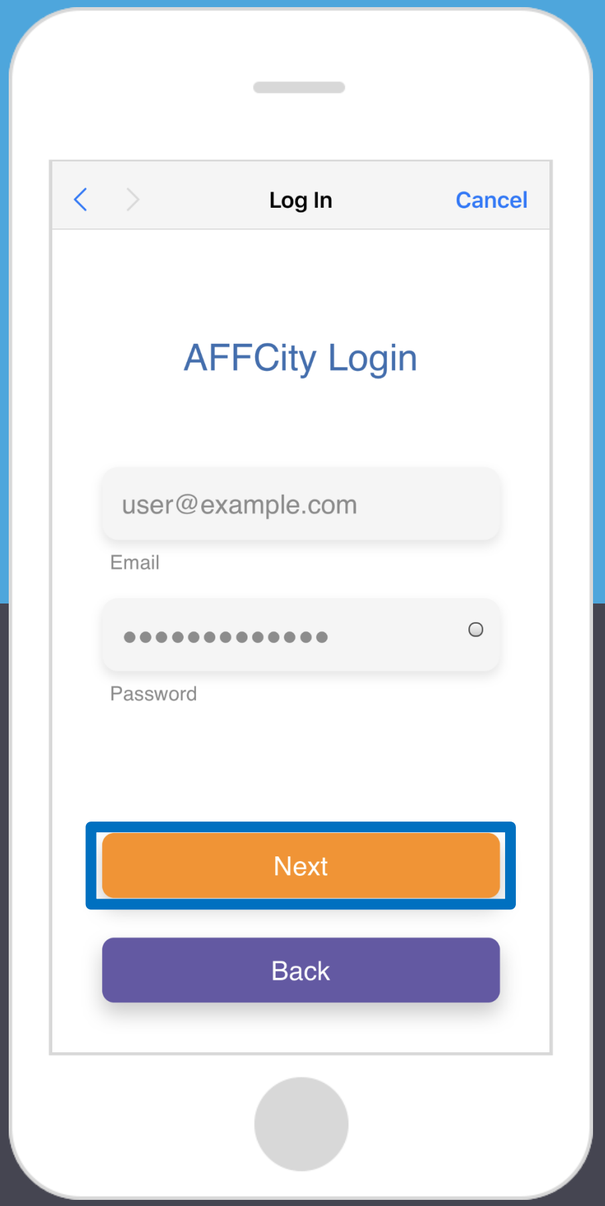


أختر لوحة AFF IoT
خاصتك من قائمة الوايفاي







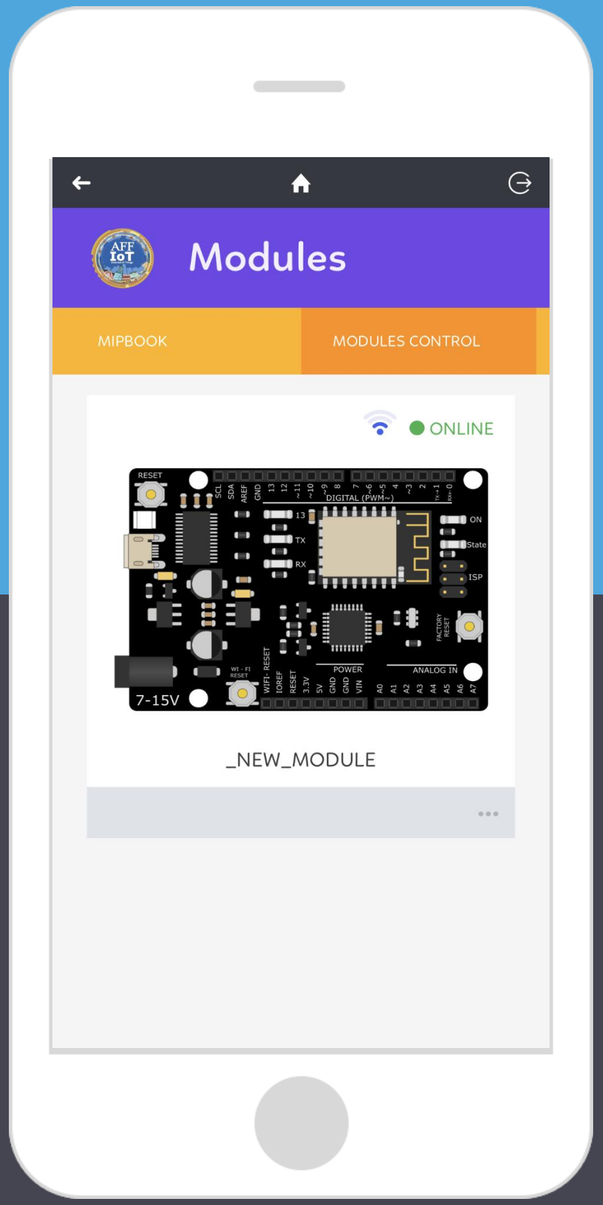


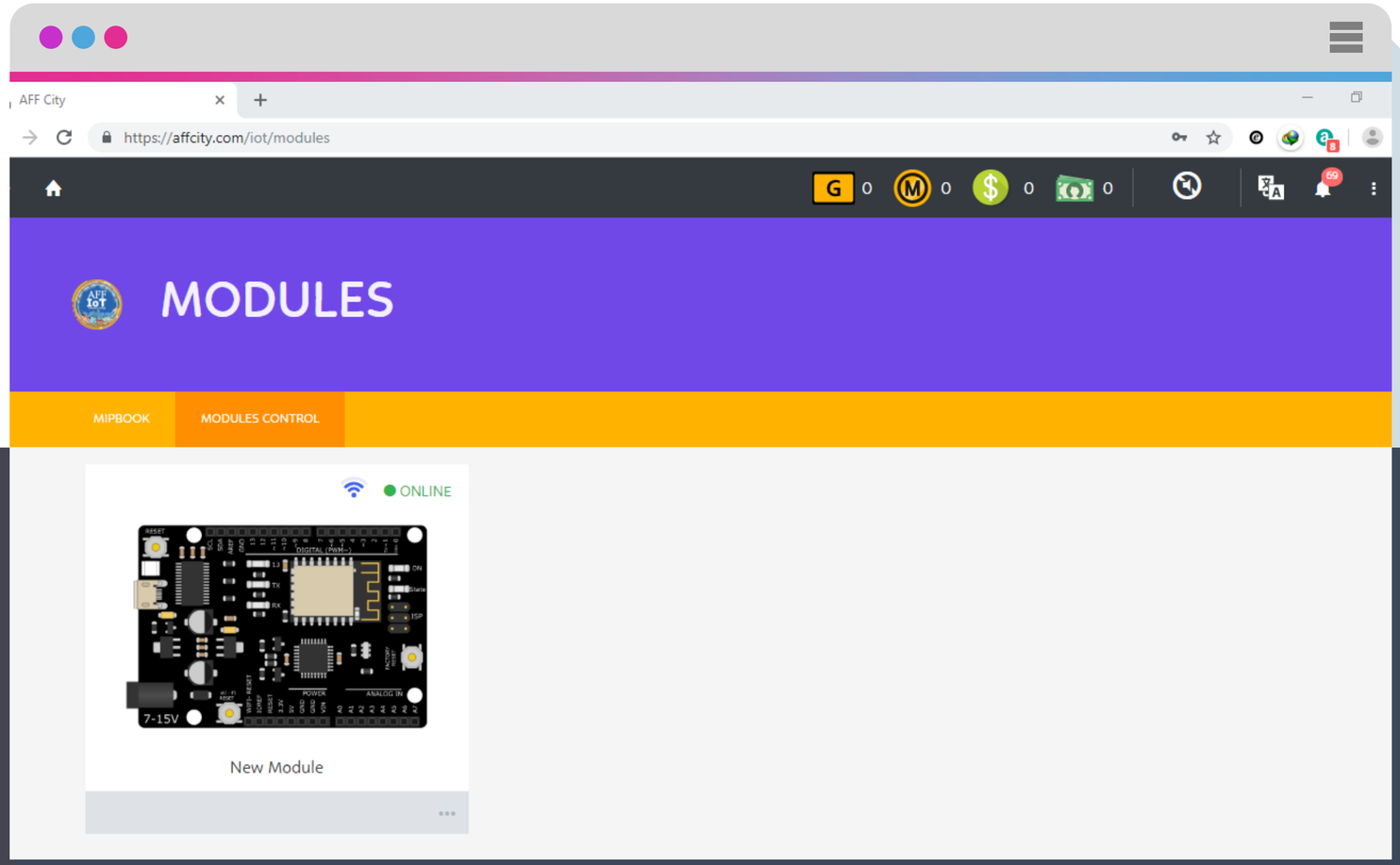
اضغط على هذا الزر للوصول إلى موقع affcity.com أو على زر الإلغاء للعودة إلى تطبيق AFF Remote

وأخيرا...

إفتح تطبيق AFF IoT, يجب أن تظهر لك لوحة AFF IoT متصلة بالإنترنت.

تسمى اللوحة تلقائيا NEW MODULE و يمكن تغيير هذا الاسم.





بالنسبة لمستخدمي الكمبيوتر

تتبع نفس الطريقة المشروحة سابقا, إلا أنه بدل استعمال التطبيق AFF IoT تنتقل إلى الموقع https://affcity.com/iot/modules و تكمل كما هو مبين باستعمال التطبيق.

إنجاز 12 مشروعا

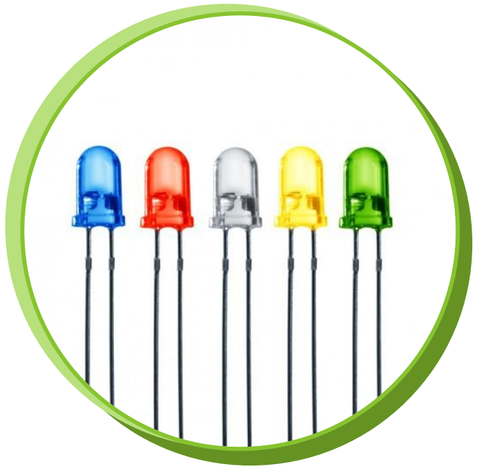
تضم مجموعة AFF IoT:

مقاومة متغيرة 4x

مقاومة متغيرة تتراوح بين 0Ω و 5kΩ

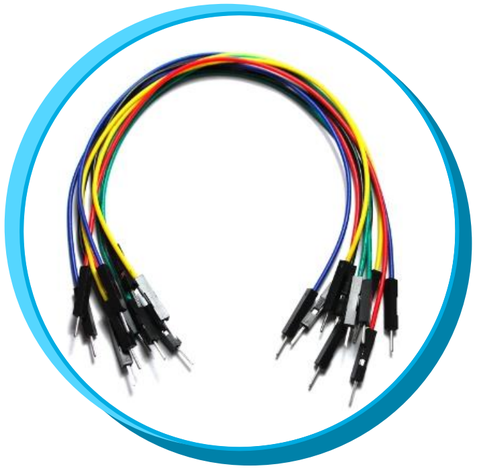
أزرار 9x

أزرار مربعة الشكل



صمامات مضيئة 27x

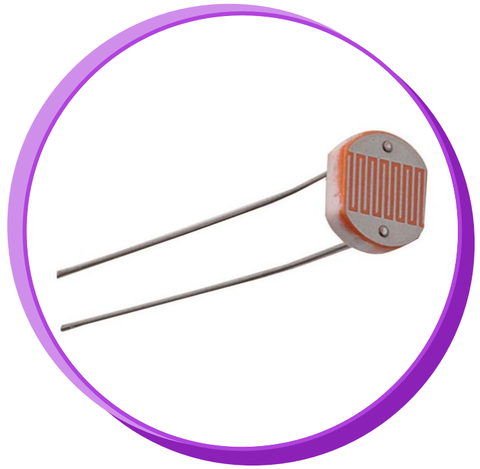
x5 حمراء, x5 خضراء, x5 زرقاء,
x5 صفراء, x5 بيضاء و اثنتان RGB



أسلاك التوصيل 20x

ألوان مختلفة

تضم مجموعة AFF IoT:

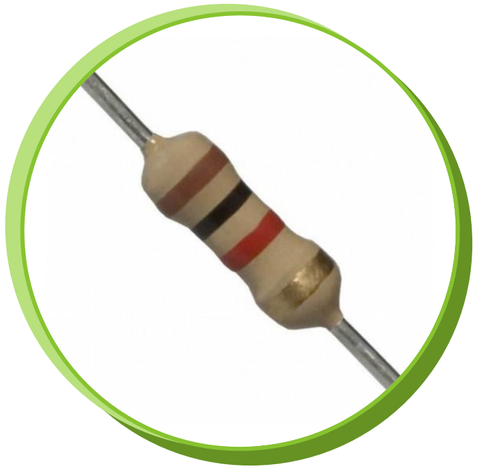


خلايا ضوئية 2x

مقاومة معتمدة على الضوء (LDR)

330Ω مقاومات x25

0.25W 5%



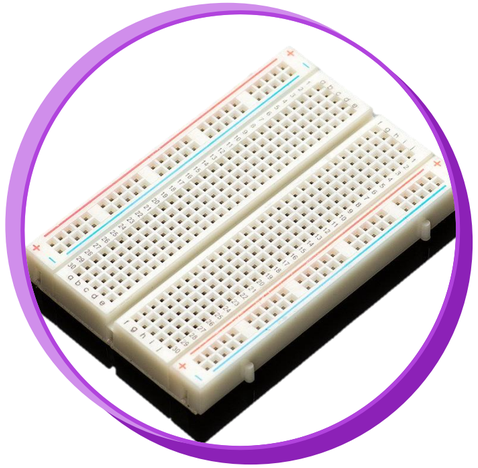
1KΩ مقاومات x4

0.5W 5%

10KΩ مقاومات x6

0.5W 5%

تضم مجموعة AFF IoT:



1x لوح التجارب

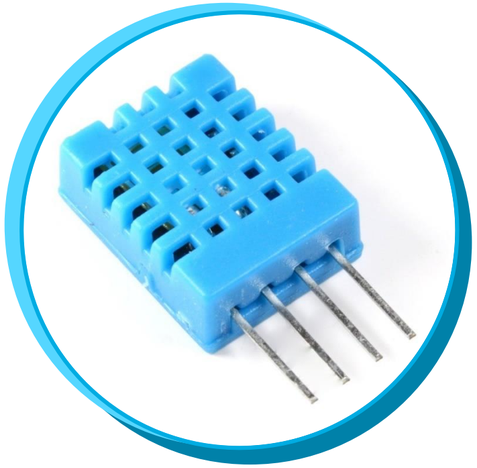
لوح التجارب ذو 400 مأخذ

2x جرس

جرس ذو قطبين

3x مقاومة حرارية

1KΩ على درجة حرارة 25°C

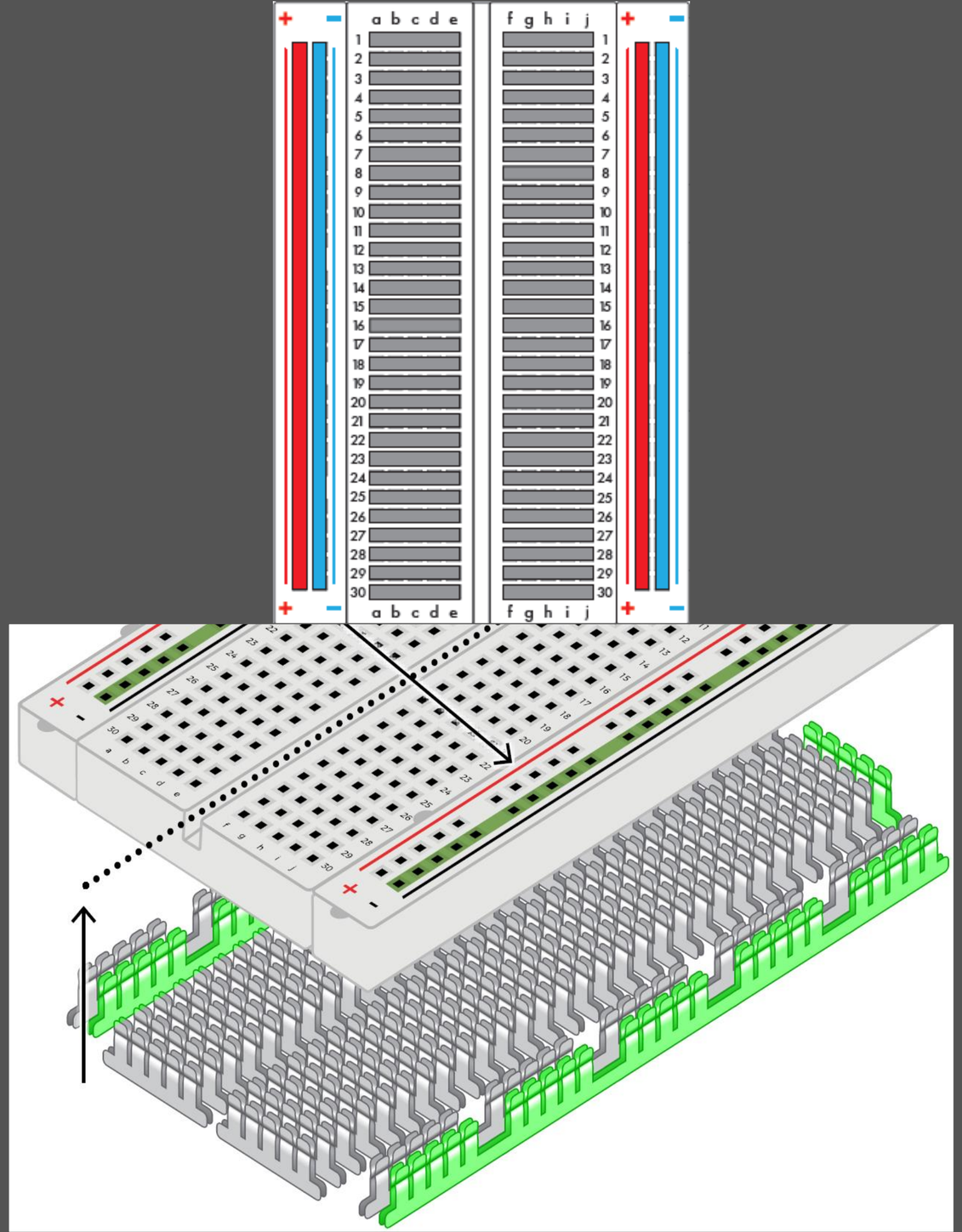


1x DHT11

حساس الحرارة و الرطوبة

لوح التجارب

لوح التجارب هو قاعدة البناء للنماذج الأولية للإلكترونيات.



الموجب:
تعمل كل علامة + على تزويد أي مكان في العمود الرأسي بالطاقة.

الأرضي:
يمتد في أي مكان في العمود الرأسي على طول علامة - .

الصفوف الأفقية : (a-e & f-j)
يتكون كل من هذه الصفوف المرقمة من 1-30 من خمسة مآخذ أفقية. كل المكونات الموصولة على المآخذ في نفس الصف تصبح موصولة ببعضها البعض.

1 وميض الصمام الثنائي المضيئ المدمج
الصفحة 32

2 التحكم في الصمام الثنائي المضيئ بالأزرار
الصفحة 42

3 خفت الصمام المضيئ باستخدام المقاومة المتغيرة
الصفحة 51

4 إستعمال الشاشة التسلسلية
الصفحة 61

5 التحكم في الصمام الثنائي المضيئ (عبر "الإنترنت")
الصفحة 67

6 قراءة شدة الضوء (عبر "الإنترنت")
الصفحة 78

7 مراقبة درجة الحرارة (عبر "الإنترنت")
الصفحة 86

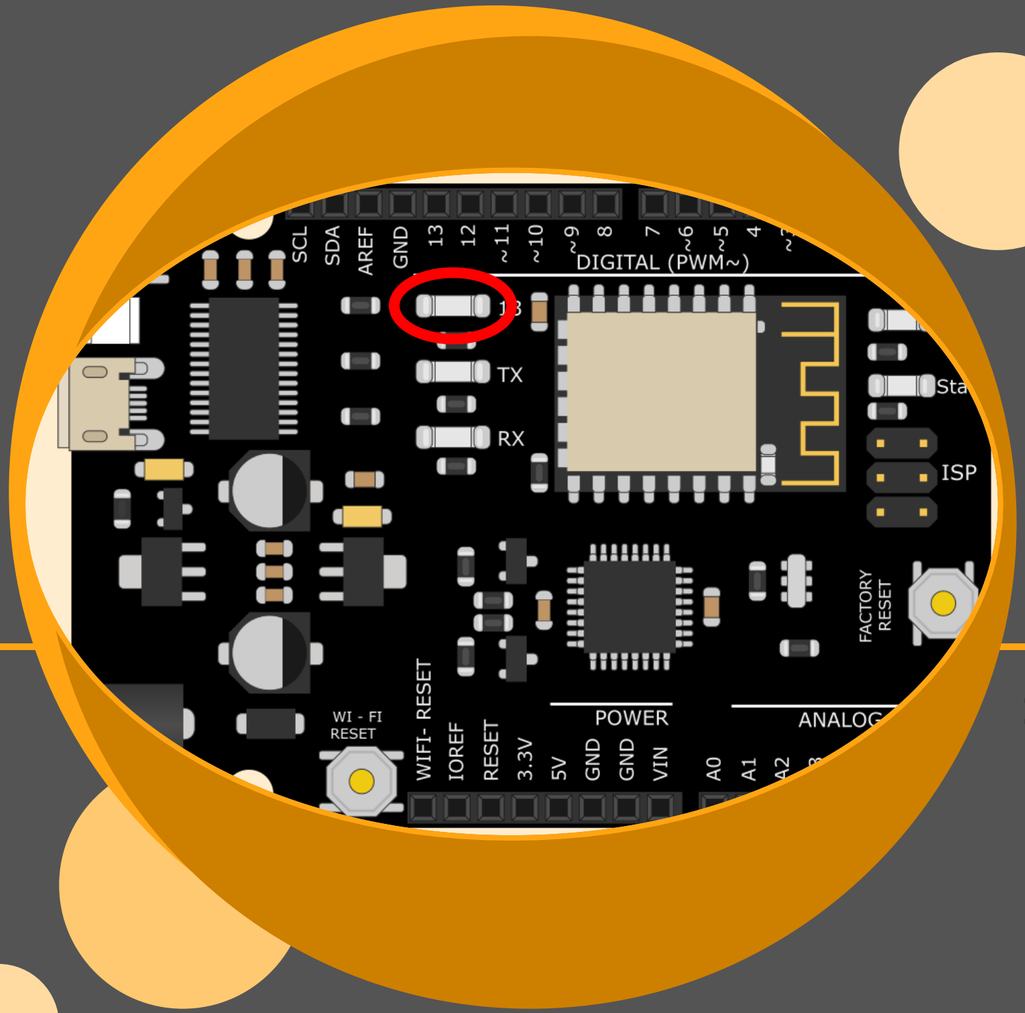
8 قياس الحرارة والرطوبة (عبر "الإنترنت")
الصفحة 94

9 ضبط لون الصمام المضيئ RGB (عبر "الإنترنت")
الصفحة 107

10 جدولة التحكم
الصفحة 117

11 محاكاة نظام الإضاءة الآلي
الصفحة 128

12 بناء دارة إنذار إرتفاع درجة الحرارة
الصفحة 141



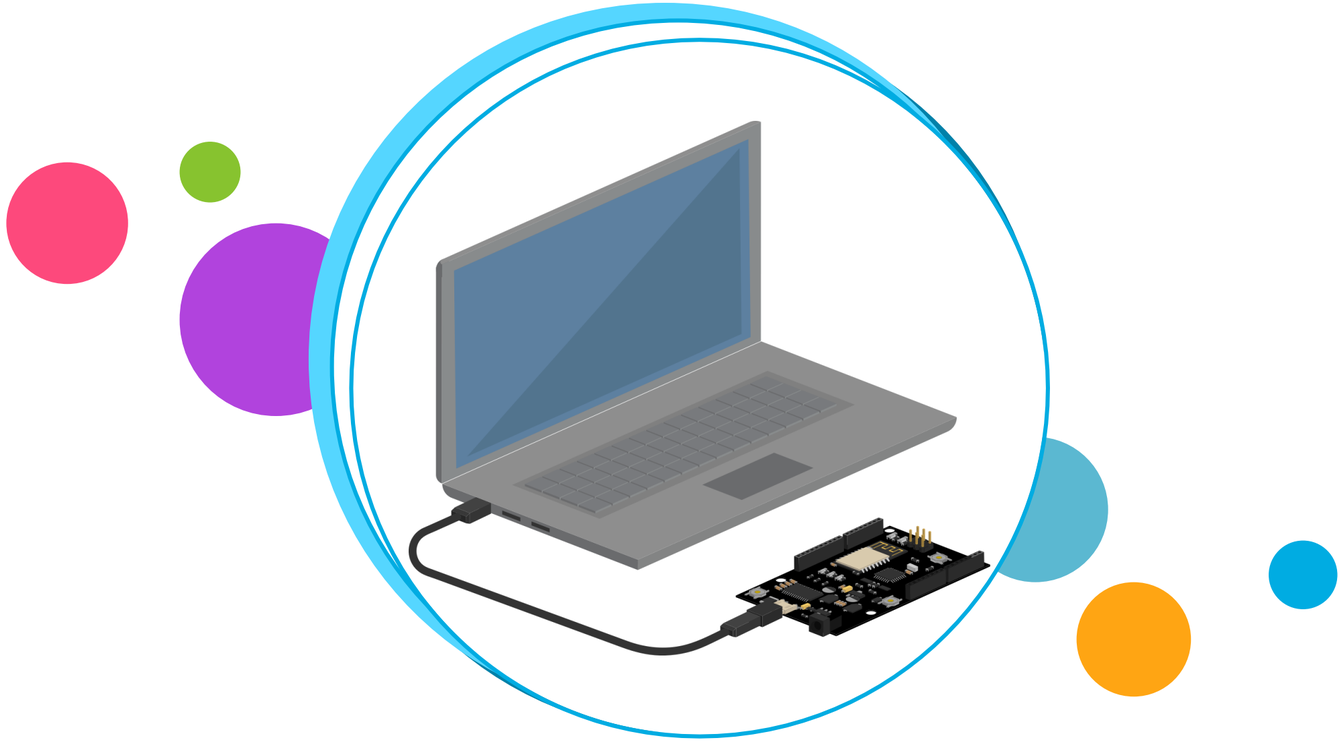
1 وميض الصمام الثنائي المضيئ المدمج

أول ما يتعلمه أي مبرمج هو كتابة كودات برمجية كافية لجعل برنامجه يظهر الجملة: "Hello World!" (مرحبا بالعالم!) على الشاشة.

وميض الصمام المدمج هو مرادف "مرحبا بالعالم!" في عالم الكمبيوتر.

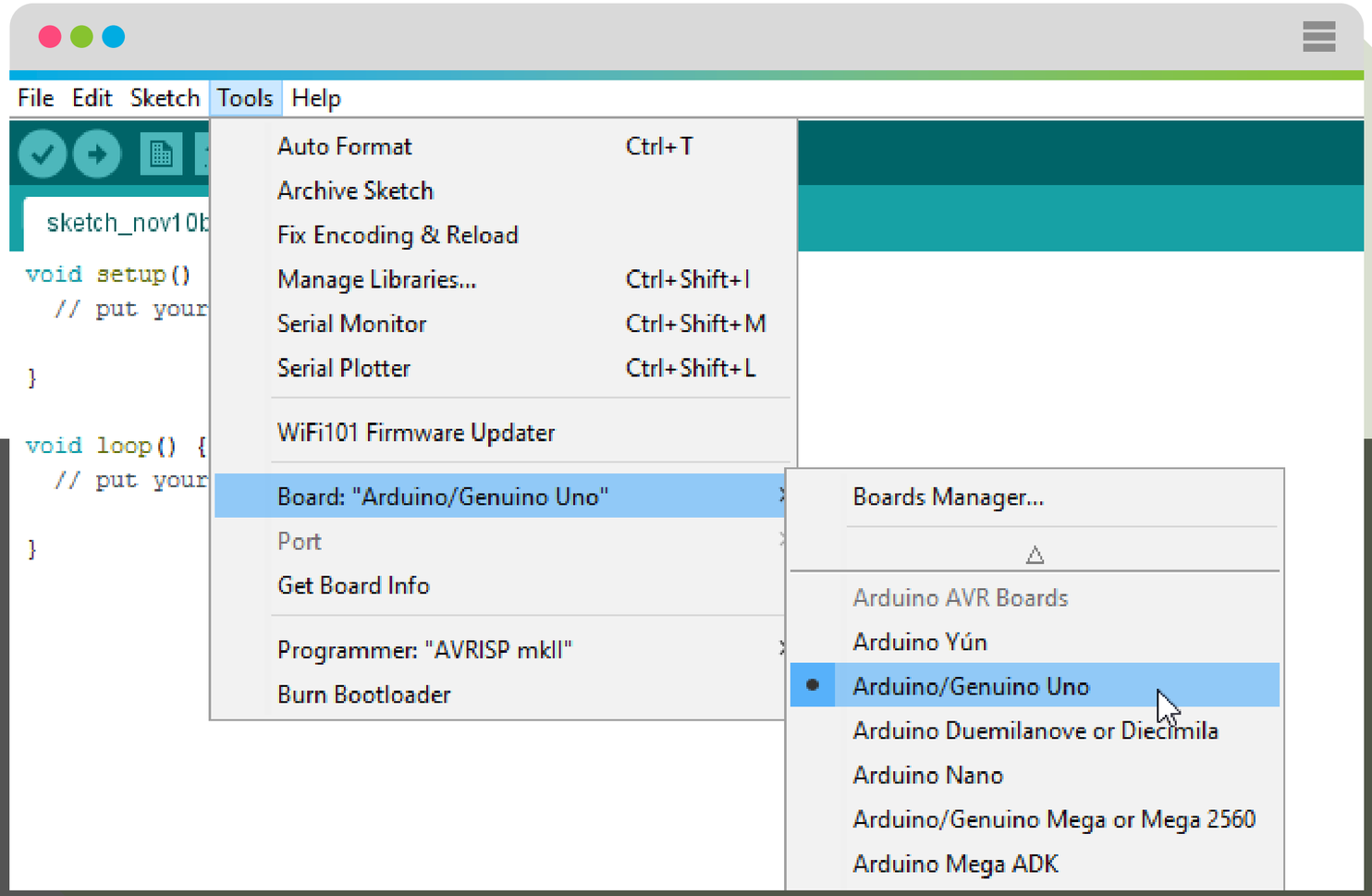
في هذا المثال لن تحتاج إلا إلى لوحة AFF IoT و كابل micro USB.

الهدف من هذا المثال هو جعل الصمام الثنائي المضيئ 13 يضئ لمدة ثانية وينطفئ لمدة ثانية واحدة بشكل متكرر.



قم بربط لوحة AFF IoT بالكمبيوتر

إستعمل كابل micro USB لتوصيل لوحة AFF IoT بأي منفذ من منافذ USB بالكمبيوتر.

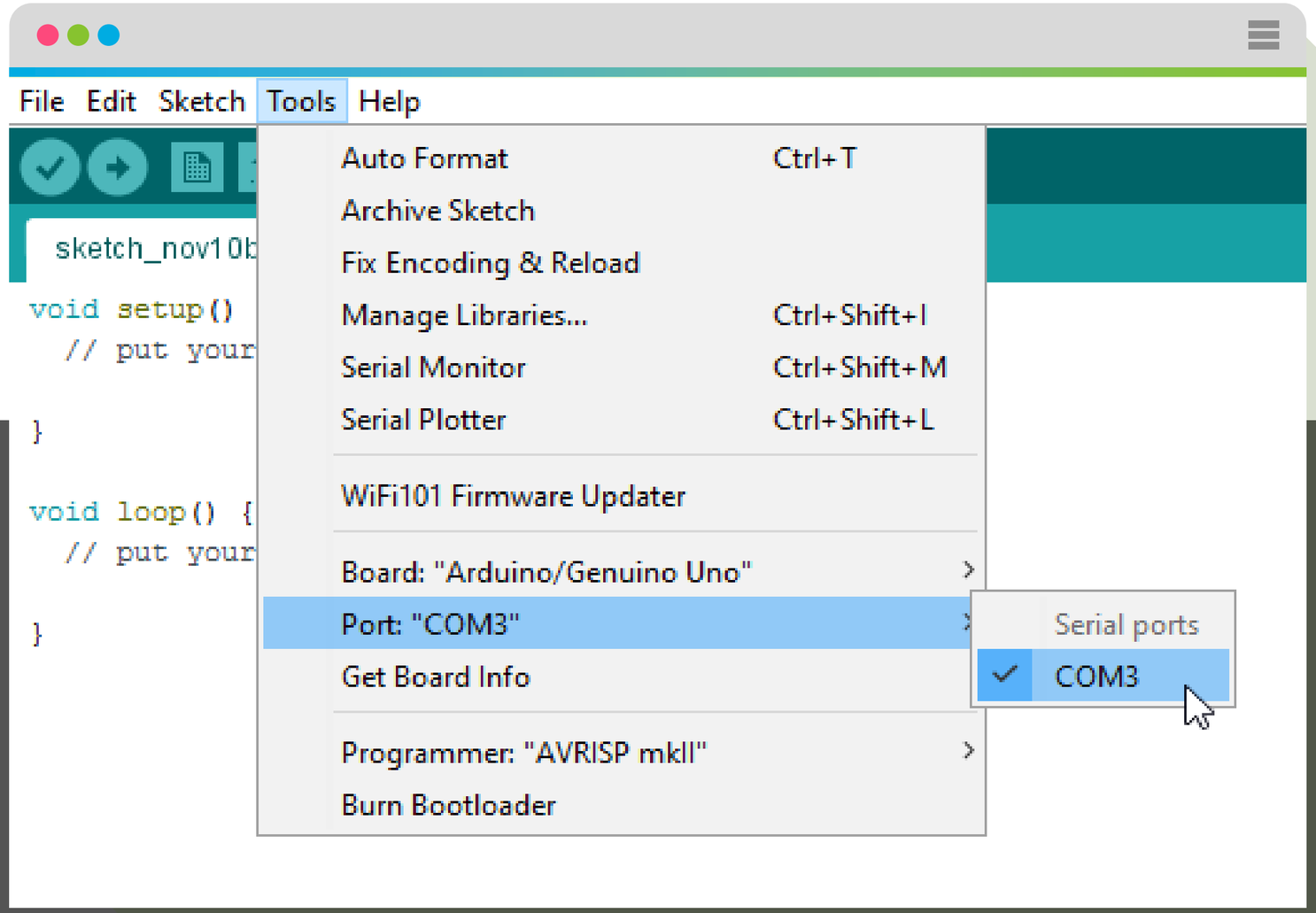


اختر اللوحة :
Arduino/Genuino Uno

ملاحظة :

لن تجد لوحة AFF IoT في لائحة برنامج الأردوينو, يمكنك اختيار بدلا عنها "Arduino/Genuino UNO".

قم باختيار جهازك التسلسلي (الويندوز)



إختر المنفذ التسلسلي للوحة عبر الضغط على اللائحة Tools → Port.

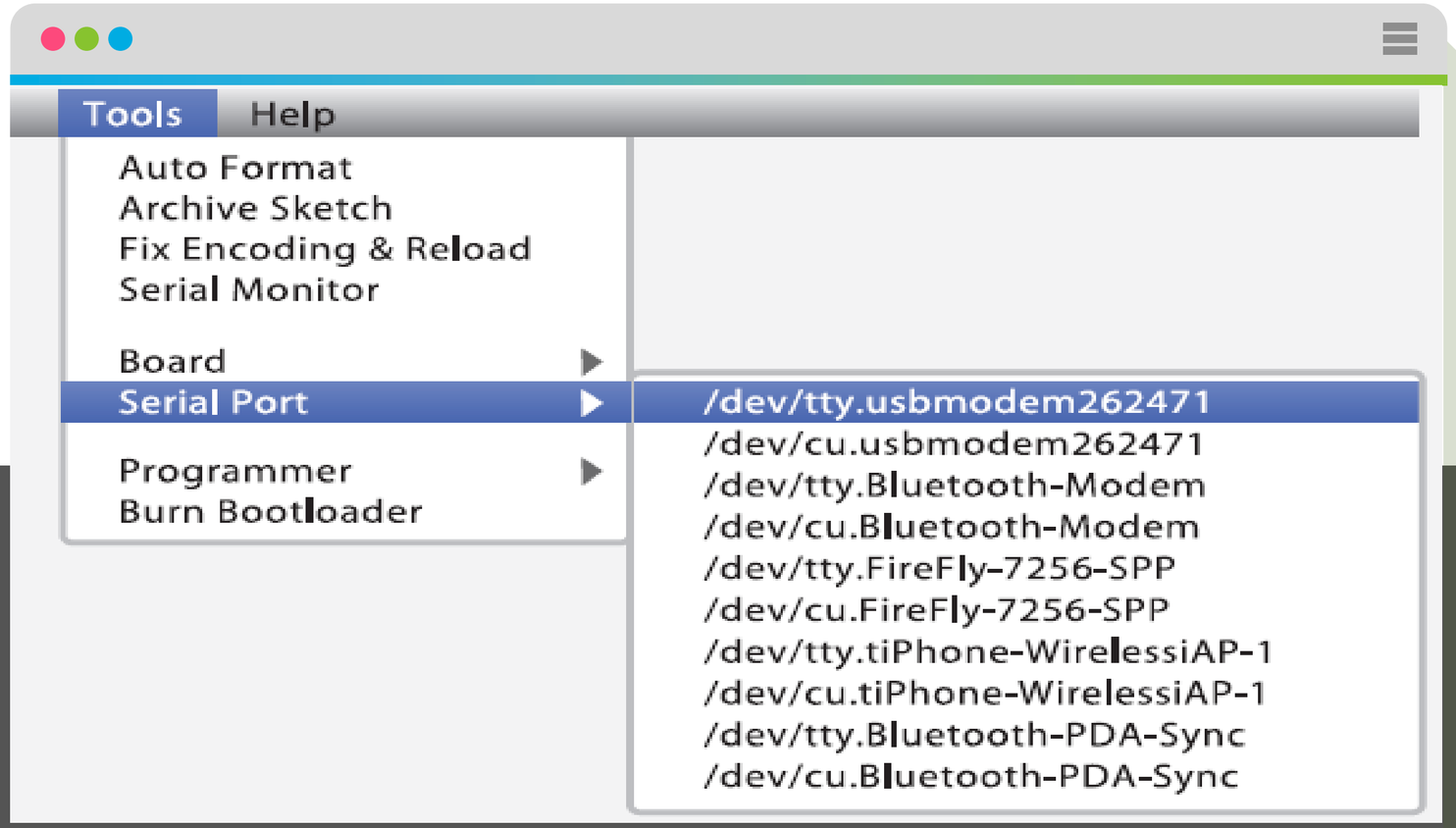
لمعرفة المأخذ بالضبط, يمكنك فصل اللوحة وإعادة فتح اللائحة, المأخذ الذي يختفي هو حتما اللوحة.

قم بإعادة توصيل اللوحة واختر هذا المأخذ.

أن رقم المنفذ قد يختلف عن المثال المطروح.

ملاحظة: إذا لم تستطع إيجاد اللوحة, قم بزيارة هذا الرابط :

https://learn.sparkfun.com/tutorials/how-to-install-ftdi-drivers/windows---in-depth



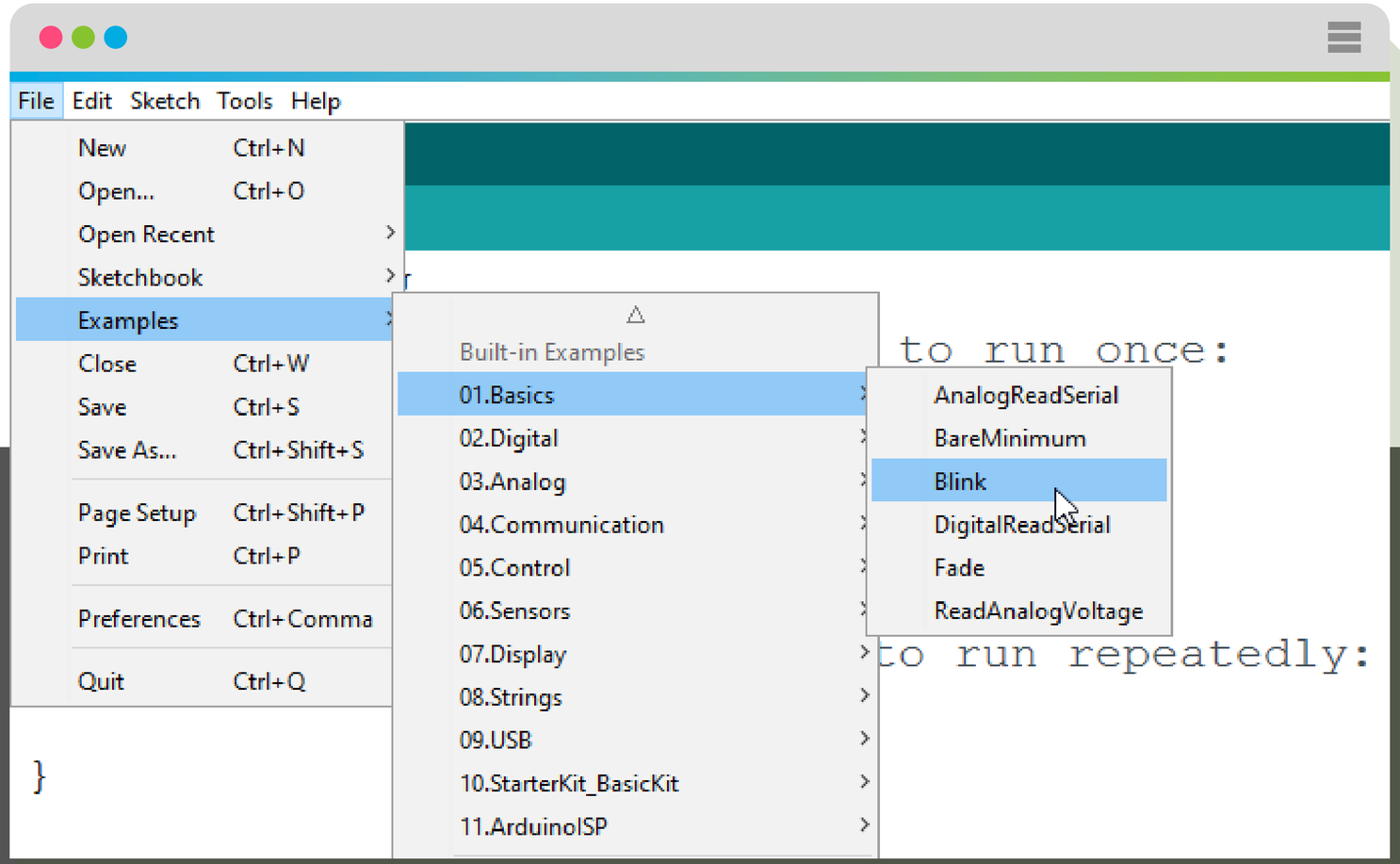
قم باختيار جهازك التسلسلي (الماك)

اختر الجهاز التسلسلي للوحة عبر الضغط على اللائحة

Tools → Serial Port ثم قم باختيار

/dev/tty.usbmodem أو

/dev/tty.usbserial



إفتح المثال Blink

بعد توصيل لوحة AFF IoT بالكمبيوتر, قم بفتح المثال Blink الموجود في لائحة الأمثلة في برنامج الأردوينو.

المثال Blink

يعد هذا المثال بمثابة الأبجدية لعالم المتحكم المصغر!

Blink

```
/*
  Blink

  Turns an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
  the correct LED pin independent of which board is used.
  If you want to know what pin the on-board LED is connected to on your Arduino
  model, check the Technical Specs of your board at:
  https://www.arduino.cc/en/Main/Products

  modified 8 May 2014
  by Scott Fitzgerald
  modified 2 Sep 2016
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman

  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

تعليمات برمجية يجب تدوينها

pinMode(LED_BUILTIN, OUTPUT):

قبل استعمال أي مأخذ من اللوحة, يجب تعريفه كمدخل INPUT أو مخرج OUTPUT.

نستعمل الوظيفة pinMode() لتعريف المأخذ 13 (المرتبط بالصمام 13) كمخرج.

digitalWrite(LED_BUILTIN, HIGH):

المخرج الرقمي يمكن تعريفه إما ب HIGH(5 فولط) أو LOW (0 فولط).

delay(1000):

تقوم بتوقيف البرنامج لمدة معينة (بأجزاء الثانية) محددة كعامل .(تضم الثانية 1000 من أجزاء الثانية)

وأخيرا...

تتم معالجة البرنامج, ويقوم النظام باستبداله بتعليمات يفهمها المتحكم المصغر.

يتم إرسال التعليمات عبر USB إلى المتحكم المصغر داخل اللوحة, التي تبدأ بتطبيق البرنامج أوتوماتيكيا.

ماذا ستلاحظ ؟

ستلاحظ أن الصمام يضيء لثانية واحدة وينطفيء لثانية مرارا وتكرارا.

إذا لم يتم ذلك, تأكد أن التعليمات قد حملت بشكل صحيح الى اللوحة, وأن لا وجود لأخطاء المعالجة, أو ألق نظرة على تعليمات المساعدة أسفله.

مساعدة

مازال لا يحمل؟
الحل المشترك لكل الأجهزة الإلكترونية هو إطفاؤها ثم إعادة تشغيلها من جديد.

البرنامج لا يحمل؟
يحدث هذا أحيانا, وغالبا ما يكون السبب هو اختيار المأخذ الخاطئ, يمكن تغيير ذلك في tools -> serial port

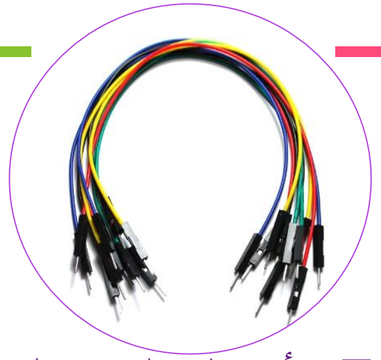
الصمام 13 لا يضيء
تأكد من اتصال اللوحة بالكمبيوتر بشكل صحيح.

2 التحكم في الصمام بالأزرار

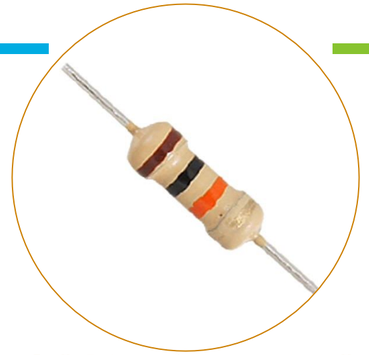
x2 أزرار

x1 مقاومة 330Ω

x1 صمام مضيء

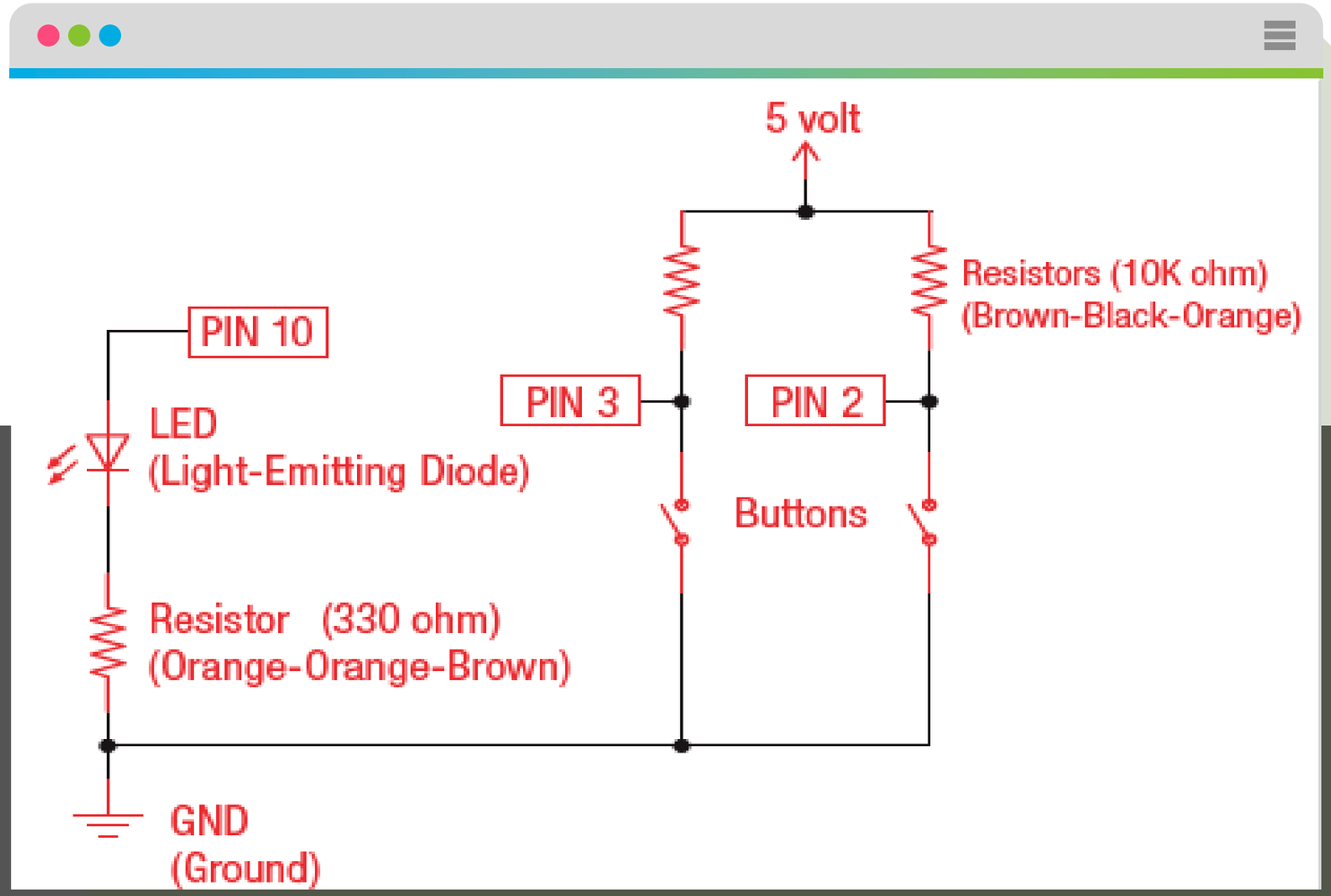


x7 أسلاك التوصيل



x2 مقاومة 10KΩ

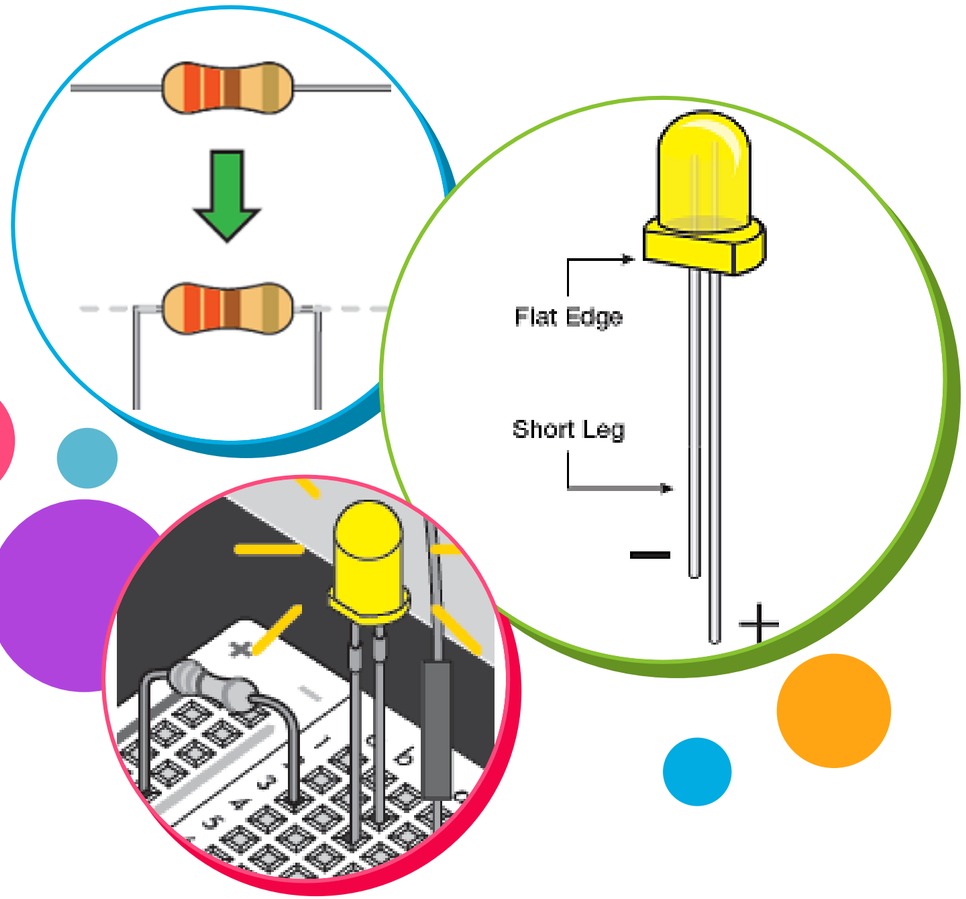
الأزرار



تعتبر الأزرار من بين أهم المداخل البسيطة, تعمل الأزرار مع اللوحة كما يلي, عندما يكون الزر مضغوطا تنخفض الطاقة. تقرأ اللوحة هذا فتتصرف تبعا لذلك.

في هذه الدائرة, نستعمل مقاومة pull-up للحفاظ على الطاقة في المستوى الأعلى عندما لا يكون الزر مضغوطا.

انتبه!



صمام مضيء :

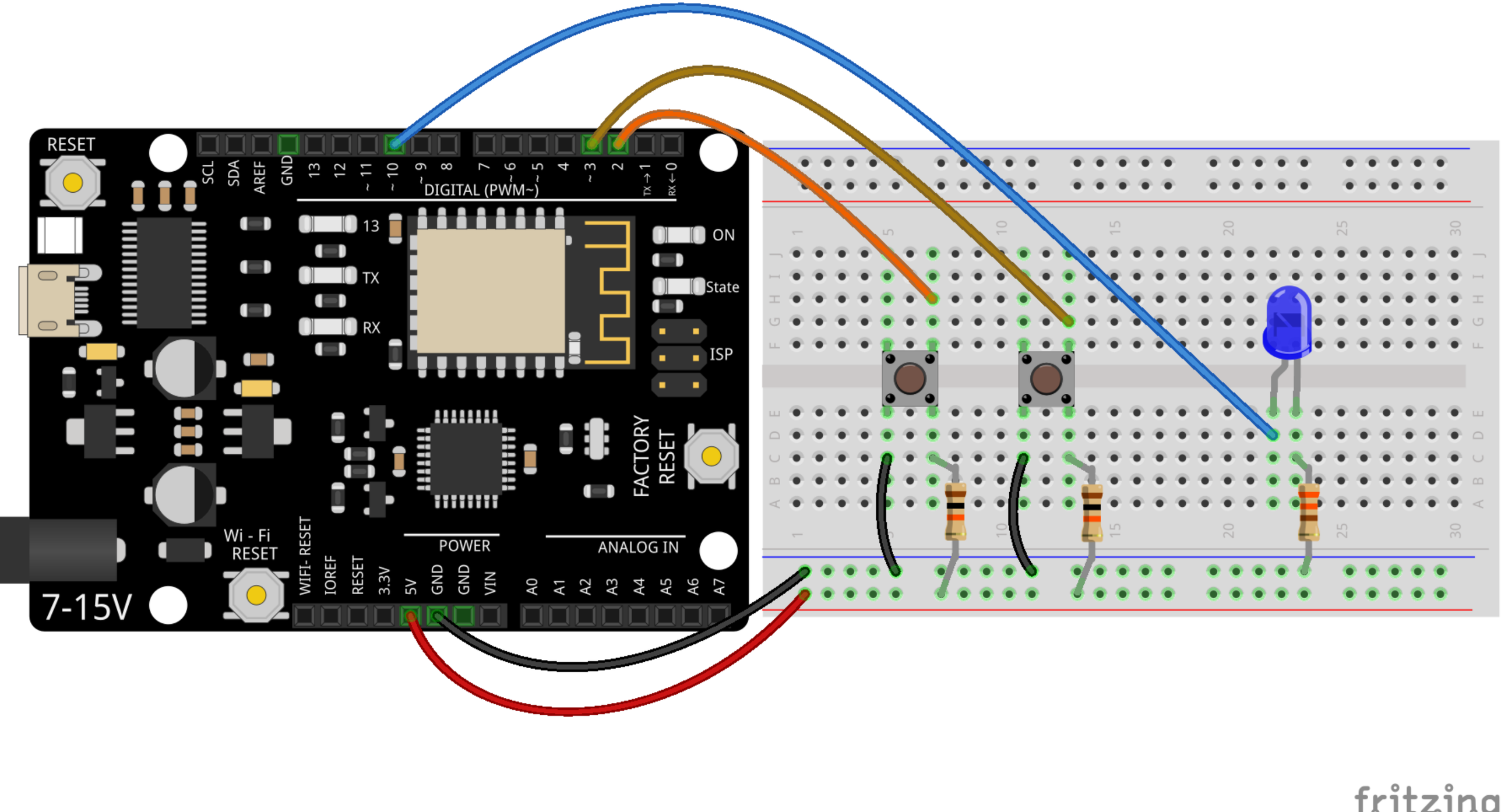
الساق القصيرة ، مع الجانب المسطح ، يتصل بالقطب السلبي.(-)

مقاومة 330Ω:

يجب أن يقرأ النطاق اللوني كما يلي :

البرتقالي-البرتقالي-البني-الذهبي.

يمكن لكلا الساقين أن يتصل بأي مأخذ.


RESET
SCL
SDA
AREF
GND
13
12
~ 11
~ 10
~ 9
8
7
~ 6
~ 5
4
~ 3
2
TX → 1
RX ← 0
DIGITAL (PWM~)
13
TX
RX
ON
State
ISP
FACTORY RESET
Wi - Fi RESET
WIFI- RESET
IOREF
RESET
3.3V
5V
GND
GND
VIN
POWER
ANALOG IN
A0
A1
A2
A3
A4
A5
A6
A7
7-15V

fritzing

==	EQUIVALENCE	A == B is true if A and B are the SAME.
!=	DIFFERENCE	A != B is true if A and B are NOT THE SAME.
&&	AND	A && B is true if BOTH A and B are TRUE.
||	OR	A || B is true if A or B or BOTH are TRUE.
!	NOT	!A is TRUE if A is FALSE. !A is FALSE if A is TRUE.

كيف نستعمل المنطق؟

يمكنك تشغيل التدفئة إذا أصبح الجو باردا ، أوالمروحة إذا كان الجو حارا، وسقي النباتات إذا جفت التربة ، إلخ.

من أجل اتخاذ مثل هذه القرارات ، هناك مجموعة من العمليات المنطقية التي يمكن دمجها لبناء عبارات شرطية معقدة.

مثال :

```
if ((mode == heat) && ((temperature < threshold) || (time == “night”)))
        digitalWrite(HEATER, HIGH);
```

AFF IoT Board folder > Circuits > Using_PushButtons

Using_Pushbuttons

```
#define Button1Pin  2
#define Button2Pin  3
#define BlueLedPin  10

void setup() {
  // Set up the pushbutton pins to be an input:
  pinMode(Button1Pin, INPUT); // configure the pin connected to the pushbutton 1 to be an input
  pinMode(Button2Pin, INPUT); // configure the pin connected to the pushbutton 2 to be an input
  pinMode(BlueLedPin, OUTPUT); // configure the pin connected to the Blue Led to be an output
}

void loop() {

  int Button1State, Button2State;  // variables to hold the pushbutton states
  Button1State = digitalRead(Button1Pin); // read the state of Button1 (LOW if pressed and HIGH if released)
  Button2State = digitalRead(Button2Pin); // read the state of Button2 (LOW if pressed and HIGH if released)

  if (((Button1State == LOW) || (Button2State == LOW))  // if we're pushing button 1 OR button 2
      && !                                               // AND we're NOT
      ((Button1State == LOW) && (Button2State == LOW))) // pushing button 1 AND button 2 simultaneously
                                                         // then...
  {
    digitalWrite(BlueLedPin, HIGH);  // turn the Blue LED on
  }
  else
  {
    digitalWrite(BlueLedPin, LOW);  // turn the Blue LED off
  }
}
```

افتح المسودة :

Using_PushButtons

تعليمات برمجية يجب تدوينها..

pinMode(Button1Pin, INPUT) : يمكن استخدام المآخذ الرقمية للمدخلات وكذلك للمخرجات. قبل البدء ، يجب تحديد المأخذ المستخدم.

Button1State = digitalRead(Button1Pin) : لقراءة مدخلات رقمية ، نستخدم وظيفة digitalRead () .وهي ستخرج قيمة "HIGH" إذا كانت الطاقة في المأخذ هي 5 فولط، أو LOW إذا كانت الطاقة في المأخذ هي 0 فولط.

if (Button1State == LOW) : نظرًا لأننا قمنا بتوصيل الزر بـ GND فسيتم قراءة LOW عند الضغط عليه. يتم استخدام عامل "التكافؤ" ("==") لمعرفة ما إذا تم الضغط على الزر.

Else : تعني أنه إذا كان الشرط في عبارة if غير صحيح ، فسيتم تنفيذ الكودات البرمجية الموجودة بعد **Else** بين الأقواس.

ماذا ستلاحظ؟

ستلاحظ إضاءة الصمام إذا ضغطت على واحد من الأزرار ، ثم ينطفئ إذا ضغطت على الزرين معا. (انظر الكود) إذا كان لا يعمل ، فتأكد من قيامك بتجميع الدائرة بشكل صحيح وتحقق من الكود وتحميله على شريحتك أو قم بمراجعة نصائح المساعدة أدناه.

مساعدة

مازال لايعمل؟

تعمل المصابيح في اتجاه واحد فقط . لذا حاول إخراجها ولفها بزاوية 180 درجة (لا تقلق ، تثبيتها بالطريقة المعاكسة لا يلحق الضرر بها).

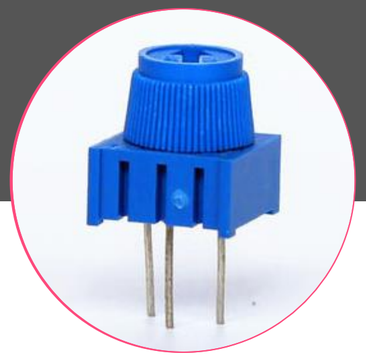
الصمام لايضيء؟

بما أن الزر مربع ، فمن السهل وضعه بطريقة خاطئة. قم بليه ب 90 درجة وجرب ما إذا كان يعمل.

تطبيقها في الحياة اليومية

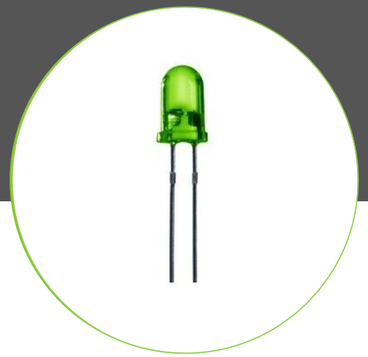
الأزرار في معظم أجهزة التحكم لألعاب الفيديو.

3 تعتيم الصمام باستخدام مقياس الجهد



مقياس الجهد

مقاومة بجهد 330Ω



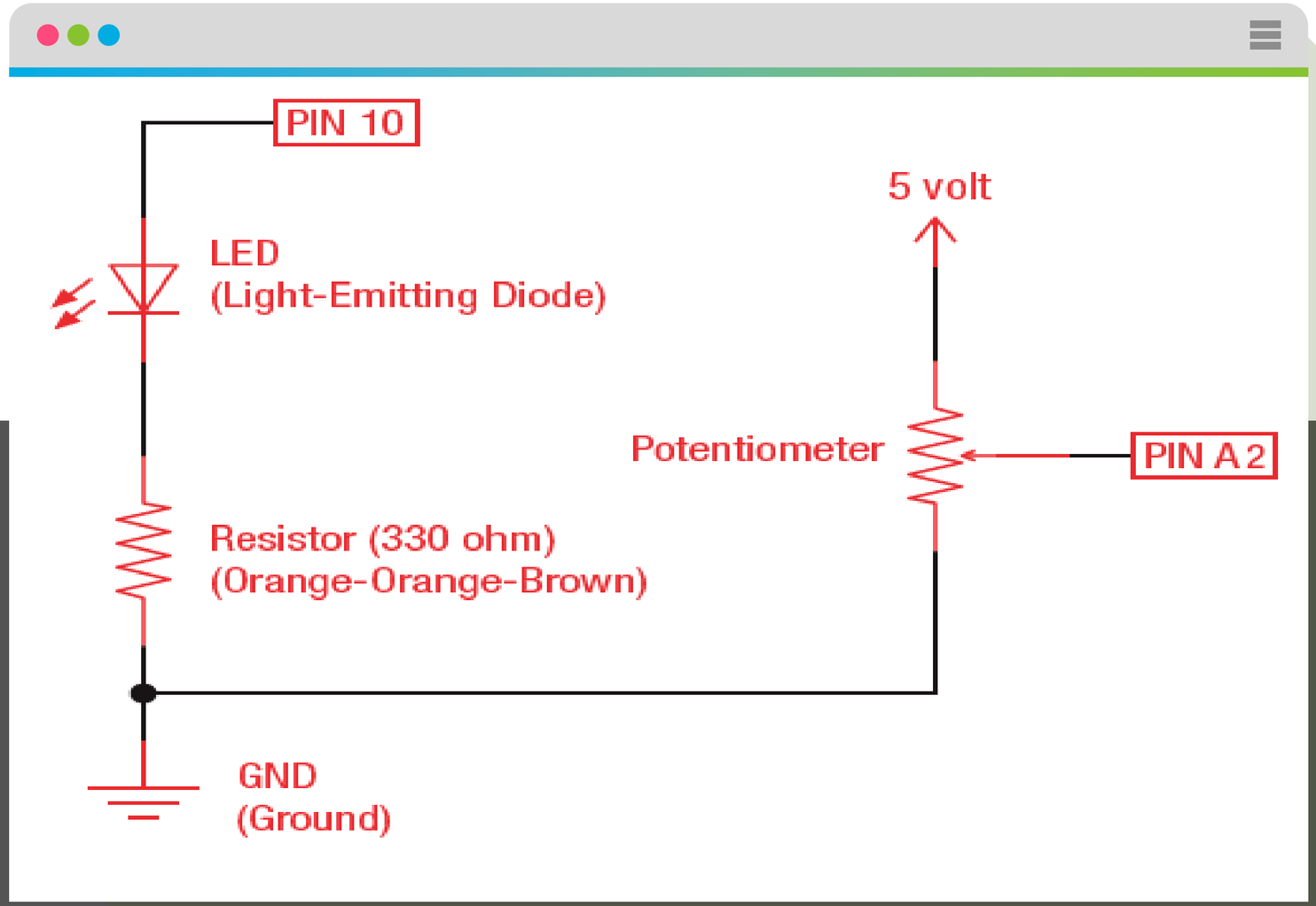
صمام مضيء أخضر

7 من أسلاك التوصيل

لاحظ أن المآخذ التماثلية A0 و A1 محجوزة للاتصال بوحدة الواي فاي لذا يرجى عدم استخدامها ، وإلا فلن تعمل اللوحة بشكل صحيح!

مقياس الجهد

(مقاومة متغيرة)



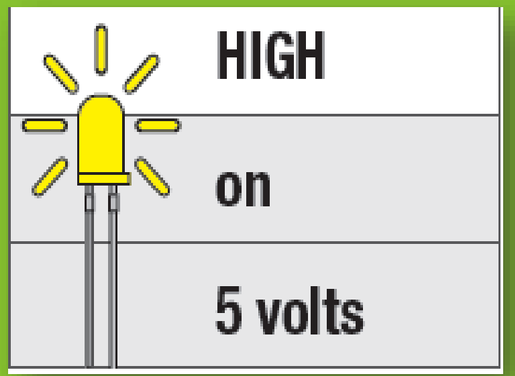
يحتوي مقياس الجهد على 3 مآخذ 2 خارجيان و 1 داخلي.

عندما يكون متصلاً بتيارخمسة فولتات عبر المآخذ الخارجية ، يخرج الدبوس الأوسط جهدًا بين 0 فولط و 5 فولط ، اعتمادًا على موضع المقبض على مقياس الجهد.

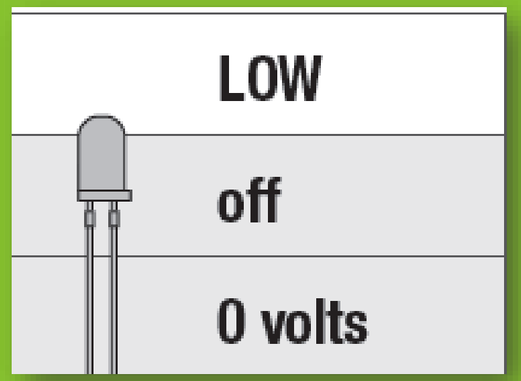
مقياس الجهد هو عبارة على دائرة للجهد المتغير.

في هذه المثال ، سوف تتعلم كيفية استخدام مقياس الجهد للسيطرة على إضاءة الصمام.

رقمي أو تماثلي



أو



تحتوي العديد من الأجهزة ، مثل الصمامات المضيئة وأزرار الضغط ، على حالتين فقط محتملتين: "مشغل" أو "غير مشغل" كما تعرف ب "HIGH " (5 volts) أو "LOW" (0 volts) .

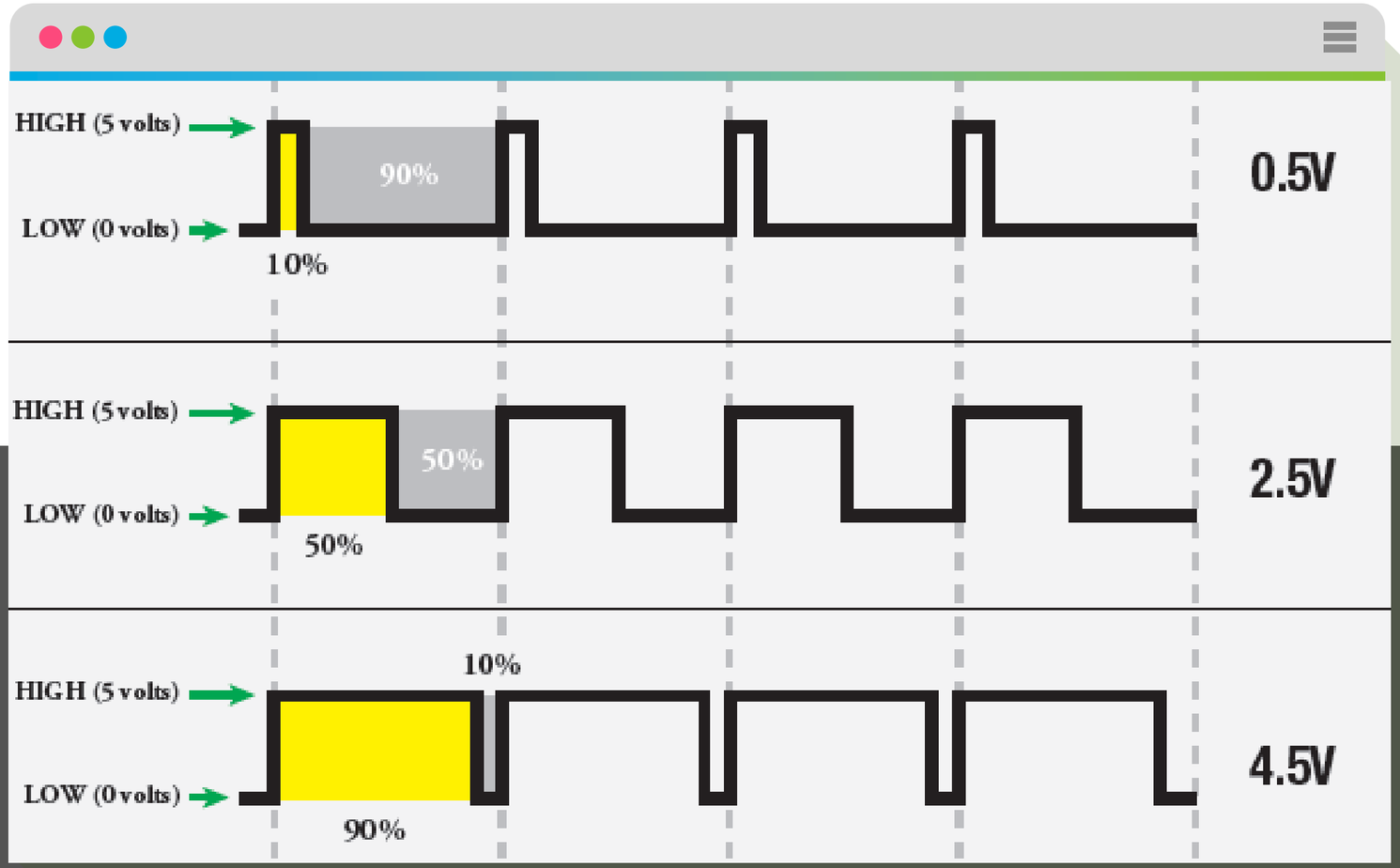
تعتبر المآخذ الرقمية للوحة مفيدة في إرسال أو استقبال الإشارات من وإلى العالم الخارجي، ويمكنها حتى القيام ببعض الحيل مثل محاكاة التعتيم (من خلال تشغيل وإيقاف الصمام بسرعة) ، وكذا في الاتصالات التسلسلية (نقل البيانات إلى جهاز آخر عن طريق ترميزها بالقيم HIGH و LOW).

0 volts	إلى	5 volts
0		1023

ولكن هناك أيضًا الكثير من الأشياء التي ليست مجرد "تشغيل" أو "إيقاف". كمستويات درجة الحرارة ، شدة الضوء ، إلخ.

جميعها لها مجموعة مستمرة من القيم بين "عالي" و "منخفض". بالنسبة لهذه ، توفر اللوحة ستة مدخلات تماثلية تترجم الجهد الداخل إلى قيمة تتراوح من 0 (0 فولت) إلى 1023 (5 فولت). وتعتبر المآخذ التماثلية مثالية لقياس جميع قيم "العالم الحقيقي".

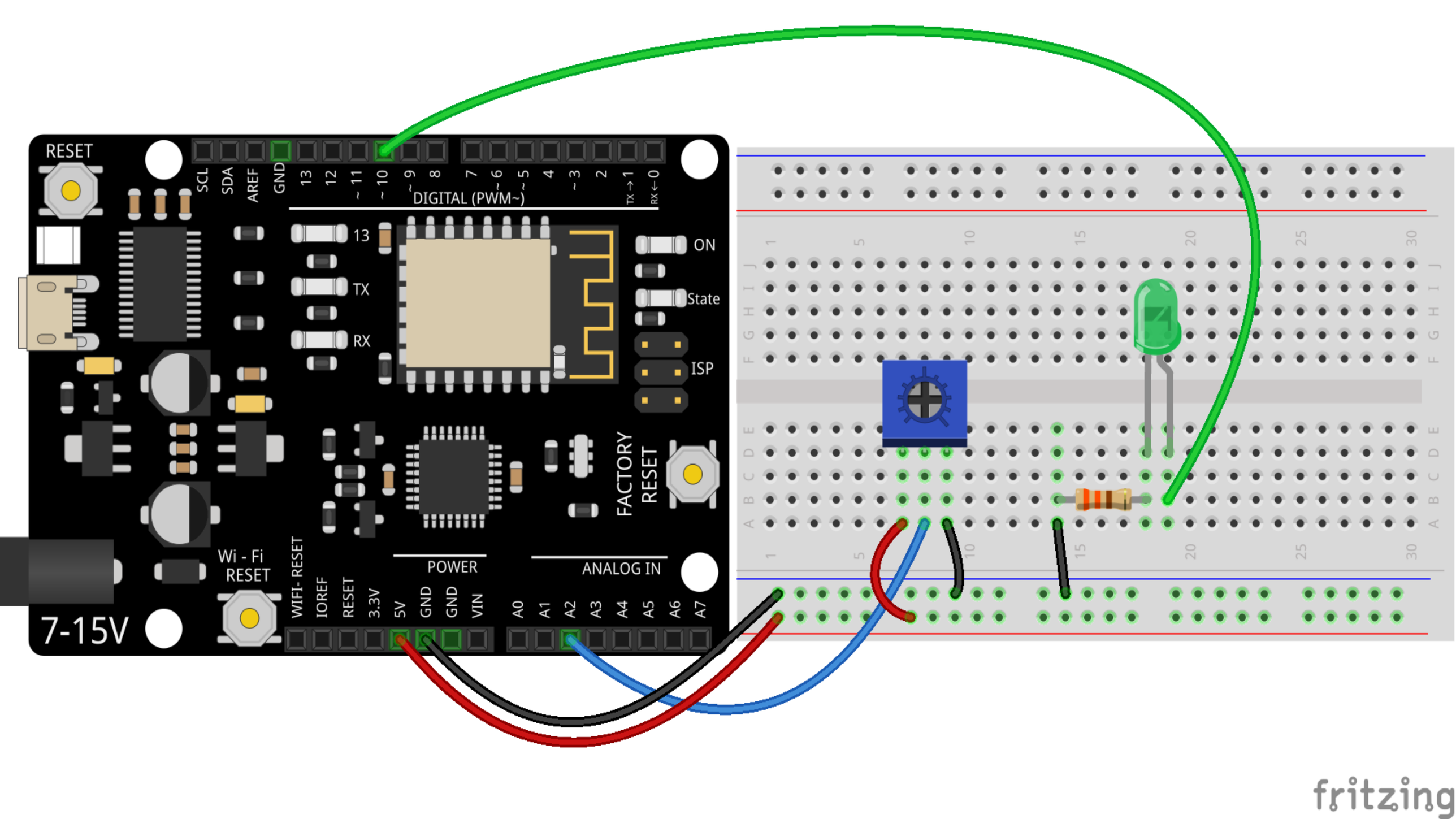
هل يمكن للوحة IoT إخراج جهد تناظري



لا تستطيع لوحة IoT إخراج جهد تناظري حقيقي. ولكن نظرًا لأنها سريعة للغاية ، يمكنها إخراج جهد مزيف باستخدام ما يسمى PWM ("تعديل عرض النبض").

المآخذ ذات العلامة "~" الموجودة على اللوحة متوافقة مع PWMأو المخارج التناظرية. المتحكم المصغر سريع للغاية بحيث يمكن أن يجعل مأخذا يوميض وويتوقف ما يقرب من 1000 مرة / ثانية.

يمضي PWM خطوة إلى الأمام عن طريق تغيير مقدار الوقت الذي يقضيه المأخذ في الجهد "عالي" مقابل الوقت الذي يمضيه في الجهد "منخفض". إذا كان يقضي معظم وقته "عالي" ، فسيظهر الصمام الضوئي المتصل بهذا المأخذ مشرقًا. إذا كان يقضي معظم وقته في " منخفض" فإن الصمام سوف يبدو باهتا. ونظرًا لأن الدمأخذ يومض بسرعة أكبر بكثير مما يمكن أن تكتشفه عينيك ، فإن اللوحة تخلق الوهم التماثلي "الحقيقي".


RESET
SCL
SDA
AREF
GND
13
12
~ 11
~ 10
~ 9
8
7
~ 6
~ 5
4
~ 3
2
TX → 1
RX ← 0
DIGITAL (PWM~)
13
TX
RX
ON
State
ISP
FACTORY RESET
Wi - Fi RESET
WIFI- RESET
IOREF
RESET
3.3V
5V
GND
GND
VIN
POWER
A0
A1
A2
A3
A4
A5
A6
A7
ANALOG IN
7-15V
fritzing

AFF IoT Board folder > Circuits > Dimming_LED_using_Potentiometer

افتح المسودة :

Dimming_LED_using_Potentiometer

Dimming_LED_using_Potentiometer

```
#define GreenLedPin       10
#define PotentiometerPin  A2

void setup() {
  // put your setup code here, to run once:
  pinMode(GreenLedPin, OUTPUT); // configure the Green Led pin to be an output
}

void loop() {
  // put your main code here, to run repeatedly:
  int analogValue = analogRead(PotentiometerPin); // read the actual converted value (between 0 and 1023)
  // Note the analog pin can be read without declaring it as input

  analogValue = map(analogValue, 0, 1023, 0, 255); // transform the scale from 0 and 1023 to 0 and 255
  // becuase the analogWrite() function take parameter from 0 (0%) to 255 (100%) of duty cycle

  analogWrite(GreenLedPin, analogValue); // generate the PWM duty cycle according the needed intensity
}
```

تعليمات برمجية يجب تدوينها..

int analogValue : يجب التعريف بالمتغيرات قبل استخدامها؛ نحن هنا نعلن عن analogValue كمتغير من النوع "int" (عدد صحيح). لا تنس أن الأسماء المتغيرة حساسة لحالة الأحرف!

analogValue = analogRead(PotentiometerPin) يتم استخدام وظيفة () analogRead لقراءة القيمة في مأخذ إدخال تمثيلي. تأخذ ()analogRead معاملا واحدا، و هو هنا "**PotentiometerPin**"، وتُرجع قيمة "analogValue" بين 0 (0 فولط) و 1023 (5 فولط).

analogWrite(GreenLEDPin, analogValue): تستعمل الوظيفة analogWrite لتوليد إشارة PWM في مأخذ الصمام الأخضر مع دورة العمل لanalogValue (وهي نسبة التيار العالي(5فولط) في الدورة) .

تعليمات برمجية يجب تدوينها..

```
map(value, fromLow, fromHigh, toLow, toHigh)
```

Value : العدد الذي يتم تحويله

fromLow : الحد الأدنى لقيمة النطاق الحالي

fromHigh : الحد الأعلى لقيمة النطاق الحالي

toLow : الحد الأدنى لقيمة النطاق المستهدف

toHigh : الحد الأعلى لقيمة النطاق المستهدف

```
analogValue= map(analogValue, 0, 1023, 0, 255)
```

قراءة إشارة تماثلية باستعمال `analogRead()`, ترجع عددا من 0 إلى 1023. ولكن قيادة مأخذ PWM باستعمال `analogWrite()` تتطلب رقما من 0 إلى 255. لذلك يجب "الضغط" على النطاق الأكبر في النطاق الأصغر باستخدام الوظيفة `map()`.

ماذا ستلاحظ

ستلاحظ أن الصمام يصبح أكثر إضاءة أو تعتيما وفقًا لمقياس الجهد الذي اخترته. إذا لم تكن تعمل ، فتأكد من قيامك بتجميع الدائرة بشكل صحيح والتحقق من الكود وتحميله على اللوحة الخاصة بك ، أوقم بمراجعة التعليمات أدناه.

مساعدة

الصمام لا يضيء ؟

تعمل المصابيح في اتجاه واحد فقط. لذا حاول إخراجها ولفها بزاوية 180 درجة (لا تقلق ، تثبيتها بالطريقة المعاكسة لا يلحق الضرر بها).

لا يعمل

تأكد من أنك لم تقم بتوصيل المساحات ، العنصر المقاوم في مقياس الجهد ، بالمأخذ الرقمي 2 بدلاً من المأخذ التماثليA2 (صف المآخذ أسفل مآخذ الطاقة)..

تعمل بشكل متقطع

الأرجح أن السبب هو رداءة الاتصال مع مأخذ مقياس الجهد. حاول الضغط على مقياس الجهد.

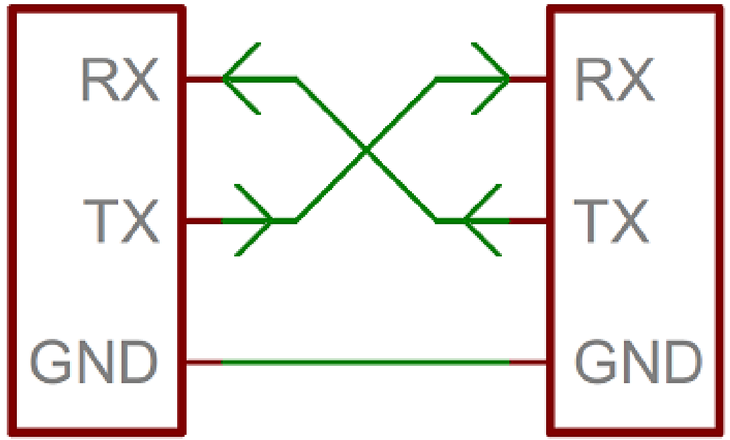
تطبيقها في الحياة اليومية

توظف معظم مقابض الصوت التقليدية مقياس الجهد.

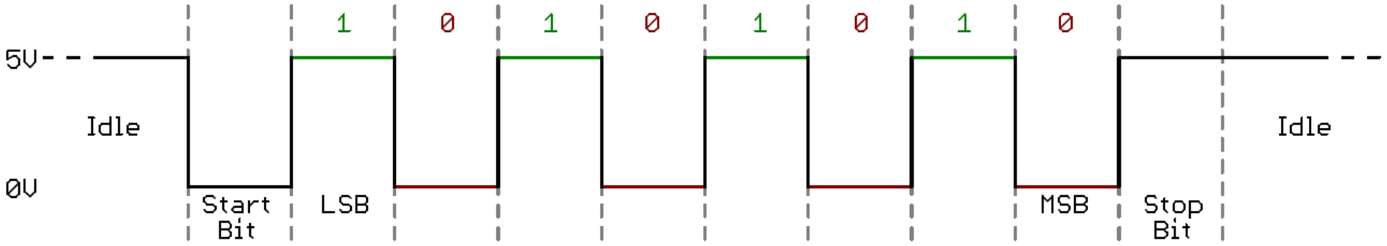
4 استعمال الشاشة التسلسلية

تعتبر الشاشة التسلسلية واحدة من أهم الأمور في نظام الأردوينو.

الاتصال التسلسلي



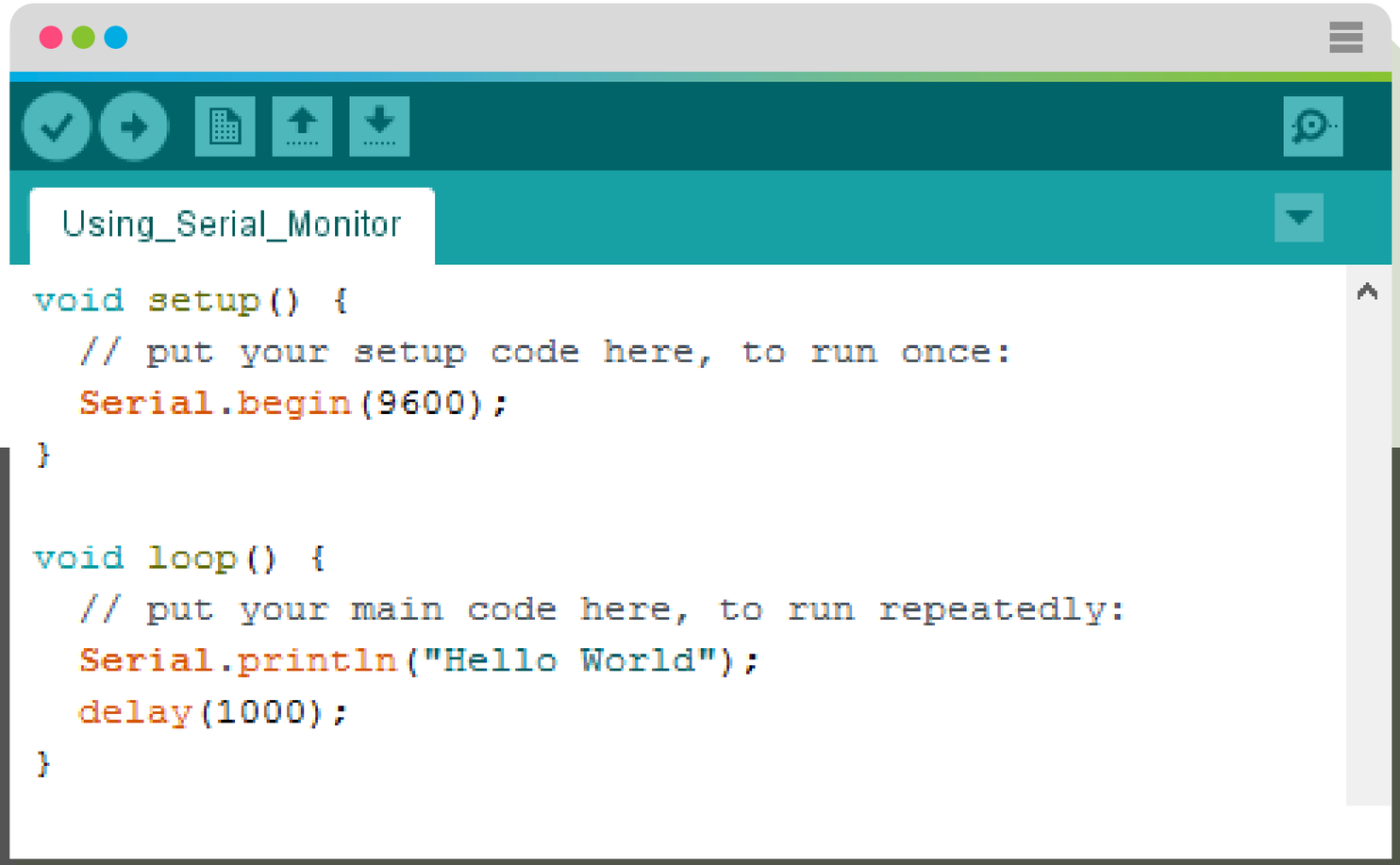
يتكون الناقل التسلسلي من سلكين - أحدهما لإرسال البيانات والآخر للاستلام. على هذا النحو ، يجب أن يكون للأجهزة التسلسلية مآخذ متسلسلة: واحد للاستقبال RX والآخر للإرسال TX.



في الاتصالات التسلسلية ، يتم إرسال البيانات واحدة تلو الأخرى ، حتى النهاية.

يتم إرسال البيانات من المرسل إلى المتلقي مثل سحب كرات التنس من أنبوب الحاوية.

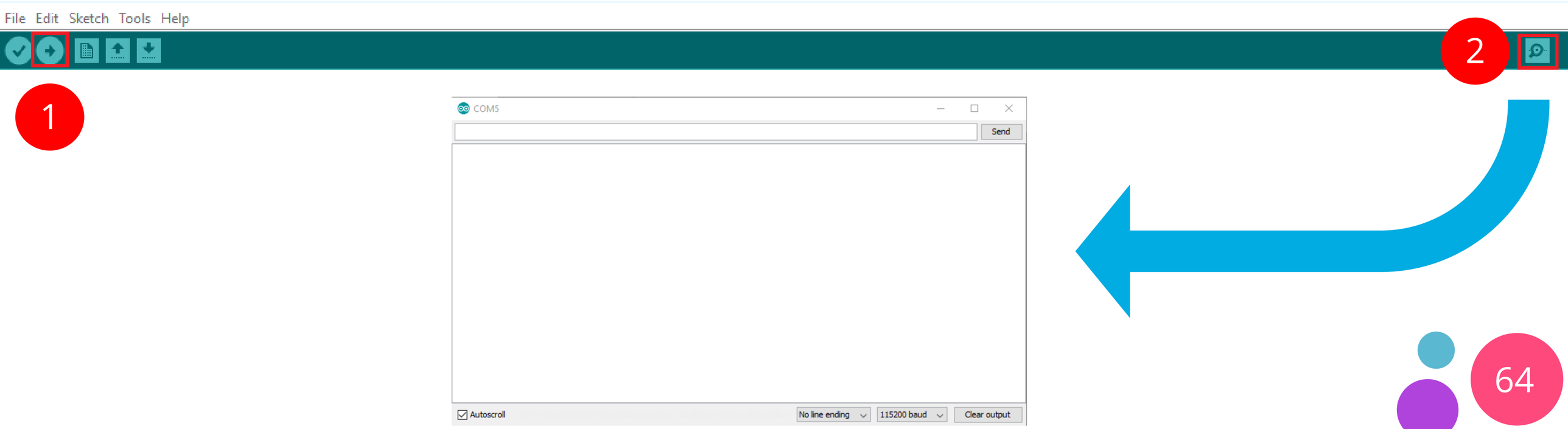
AFF IoT Board folder > Circuits > Using_Serial_Monitor



افتح المسودة :
Using_Serial_Monitor

الشاشة التسلسلية

يستخدم هذا المثال الشاشة التسلسلية لنظام الأردوينو. لفتحها ، قم أولاً بتحميل البرنامج ثم انقر فوق الزر الذي يشبه المكبر داخل مربع. من أجل أن تعمل الشاشة التسلسلية بشكل صحيح ، يجب ضبطها على نفس معدل الباود (السرعة بالبث في الثانية) مثل الشفرة قيد التشغيل. يعمل الرمز بمعدل 9600 باود. إذا لم يكن إعداد معدل البث بالباود هو 9600 ، فيرجى تغييره إلى 9600.



تعليمات برمجية يجب تدوينها..

قبل استخدام الشاشة التسلسلية ، يجب عليك الاتصال بـ **`Serial.begin ()`** لتهيئتها. **`9600`** هو **`"معدل البث بالباود"`** ، أو سرعة الاتصالات. عندما يتصل جهازان ببعضهما البعض ، يجب ضبط كليهما بنفس السرعة.

`Serial.println(“Hello World”)` :

تقوم الوظيفة Serial.println () بطباعة أي شيء مكتوب بداخلها ، بما في ذلك المتغيرات من جميع الأنواع ، ثم تنتقل إلى السطر التالي.

كما تطبع Serial.print () كل شيء على نفس السطر.

لمزيد من المعلومات ، يرجى زيارة : *http://arduino.cc/en/serial/print*

ماذا ستلاحظ ؟

ستلاحظ ظهور عبارة "Hello World" على الشاشة التسلسلية على رأس كل ثانية.

مساعدة

لا يحدث أي شيء عند فتح الشاشة التسلسلية

حاول العثور على نافذة الشاشة التسلسلية في شريط مهام النوافذ عن طريق تمرير الماوس فوق أيقونة Arduino ثم النقر على الشاشة التسلسلية.

يتم عرض رطانة

يحدث هذا لأن الشاشة التسلسلية تتلقى البيانات على
سرعة مختلفة عن المتوقع. انقر على المربع المنسدل الذي يقرأ " baud ***"وقم بتغييره إلى" 9600 baud "

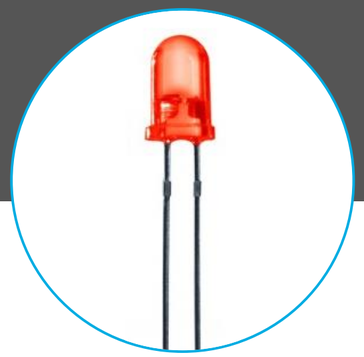
لا شيء معروض

حاول الخروج من الشاشة التسلسلية ثم أعد تشغيلها مرة أخرى.

5 التحكم في الصمام (عبر "الإنترنت")

3 من أسلاك التوصيل

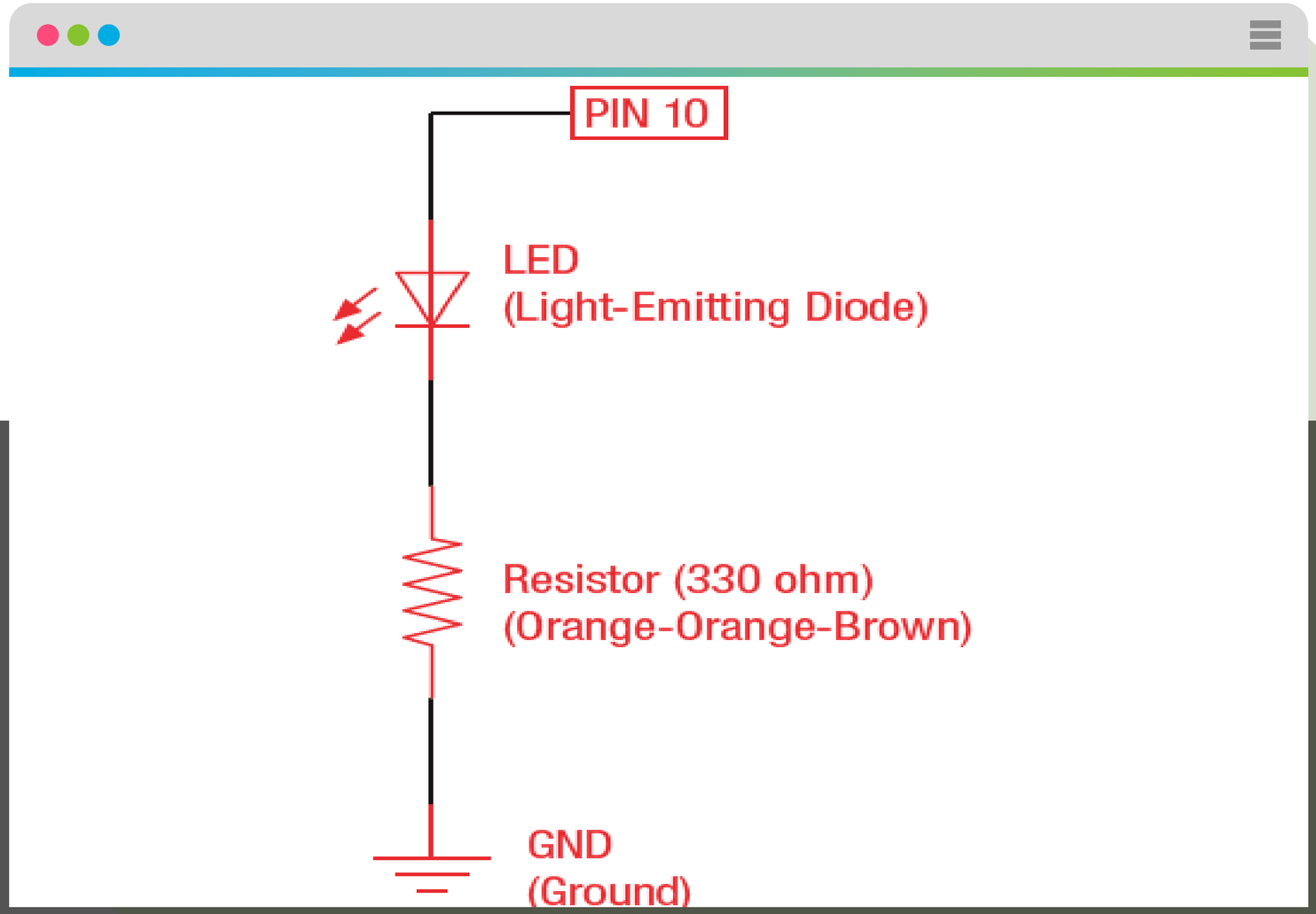
مقاومة ذات الجهد 330Ω

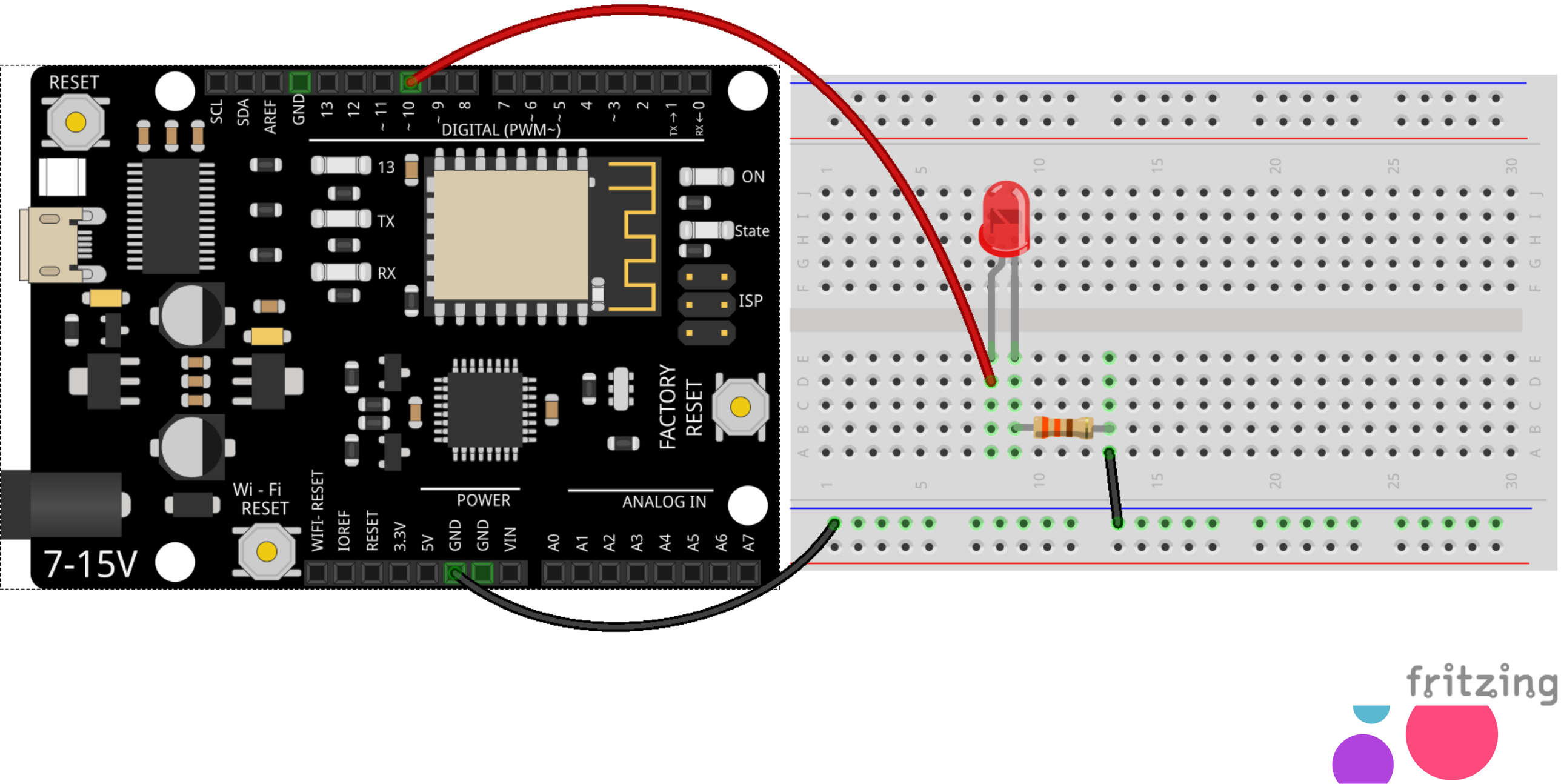


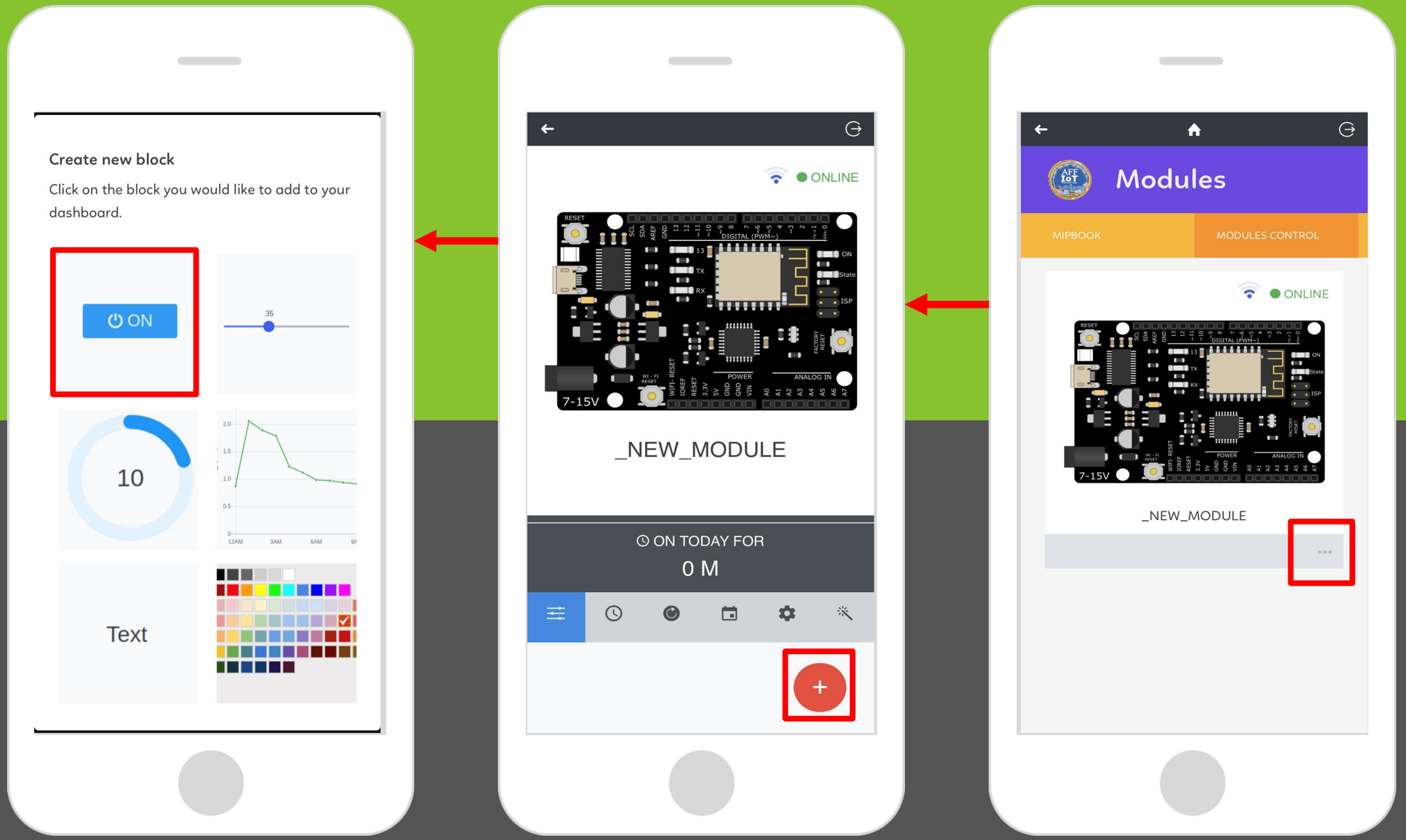
صمام مضيء أحمر

تشغيل أو إيقاف الصمام

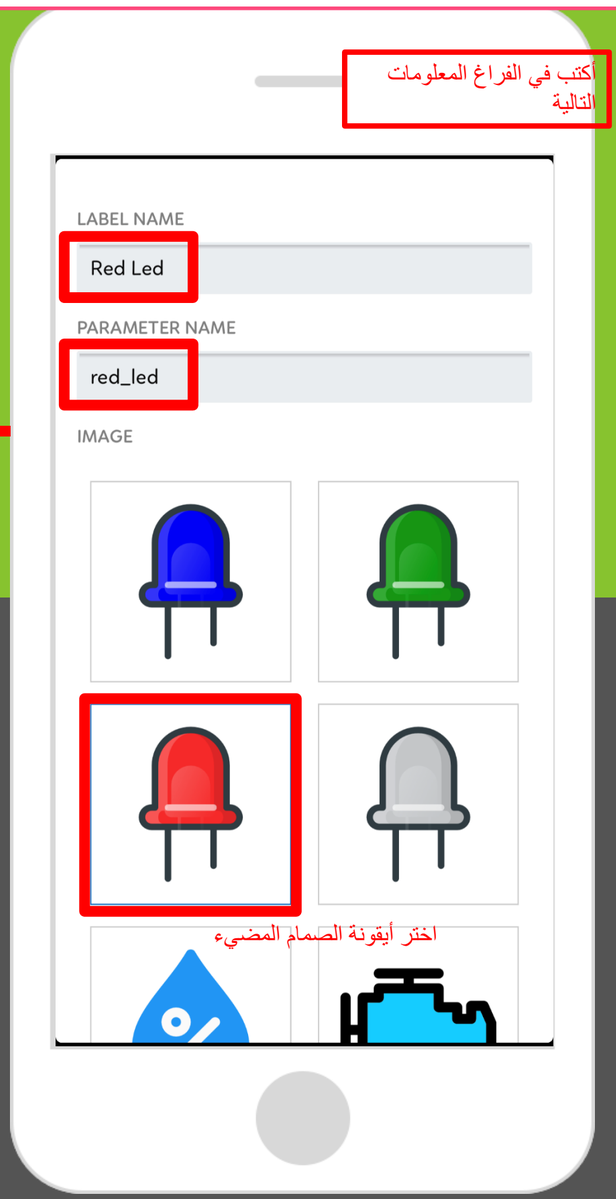
الصمام المضيء هو مصدر ضوئي صغير ولكنه قوي يستخدم في العديد من التطبيقات المختلفة. أولاً سنعمل على جعله يومض, رغم أنها وظيفة بسيطة، إلا أنها الأساس لمزيد من الوظائف المعقدة.

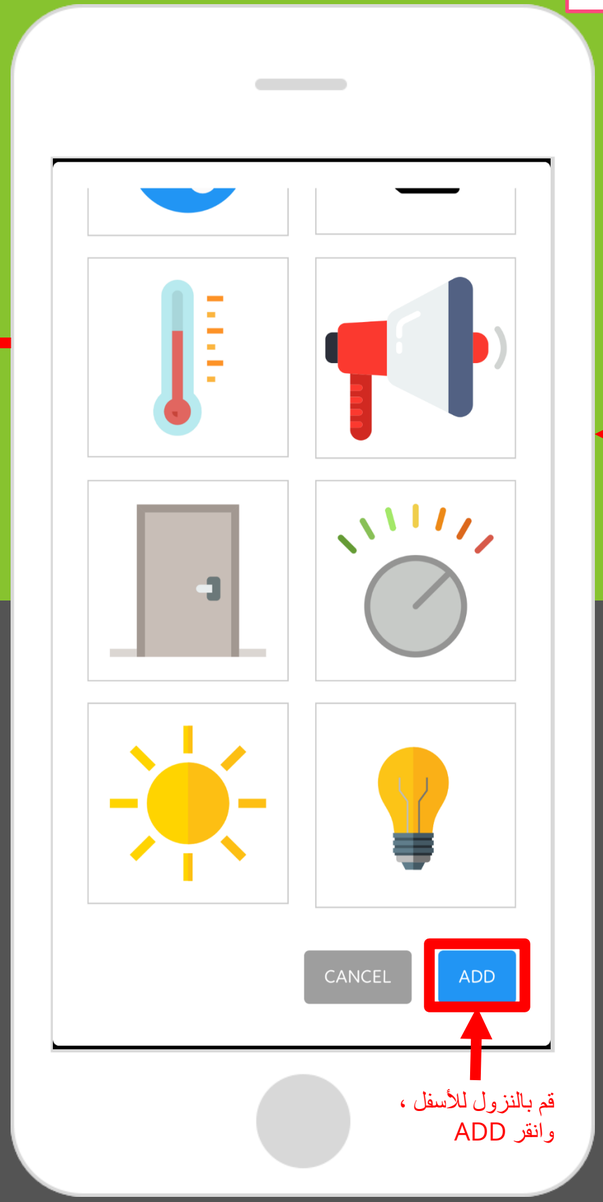


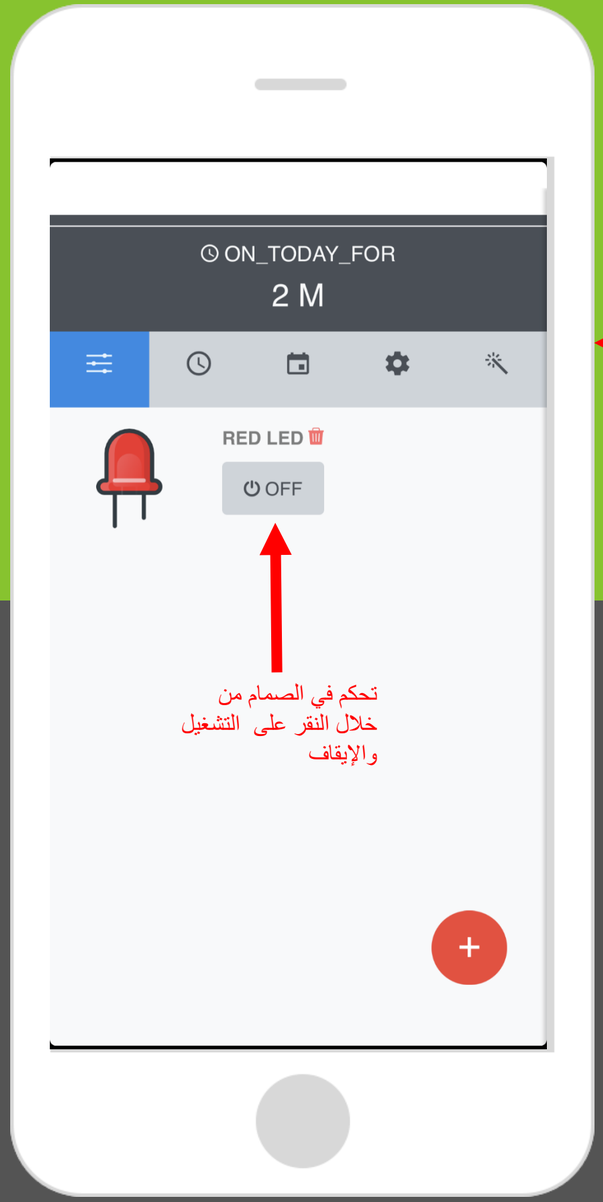

RESET
SCL
SDA
AREF
GND
13
12
~ 11
~ 10
~ 9
8
7
~ 6
~ 5
4
~ 3
2
TX → 1
RX ← 0
DIGITAL (PWM~)
13
TX
RX
ON
State
ISP
FACTORY RESET
Wi - Fi RESET
WIFI- RESET
IOREF
RESET
3.3V
5V
GND
GND
VIN
POWER
ANALOG IN
A0
A1
A2
A3
A4
A5
A6
A7
7-15V
fritzing


Create new block
Click on the block you would like to add to your dashboard.
ON
35
10
Text
ONLINE
_NEW_MODULE
ON TODAY FOR
0 M
Modules
MIPBOOK
MODULES CONTROL
ONLINE
_NEW_MODULE

LABEL NAME : هو الاسم الذي يظهر في التطبيق.
PARAMETER NAME : هو الاسم المستخدم في الكودات البرمجية.

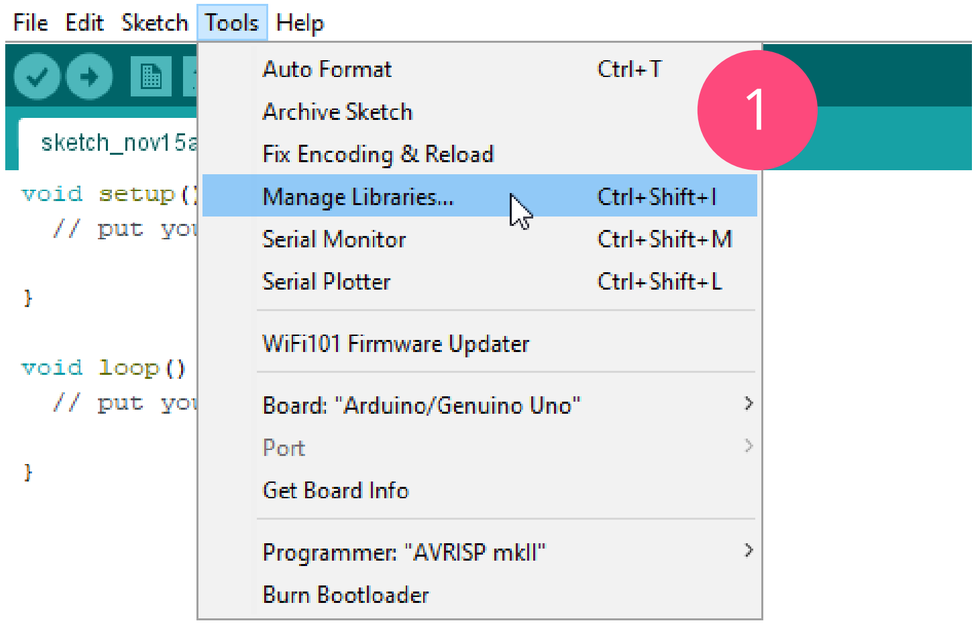


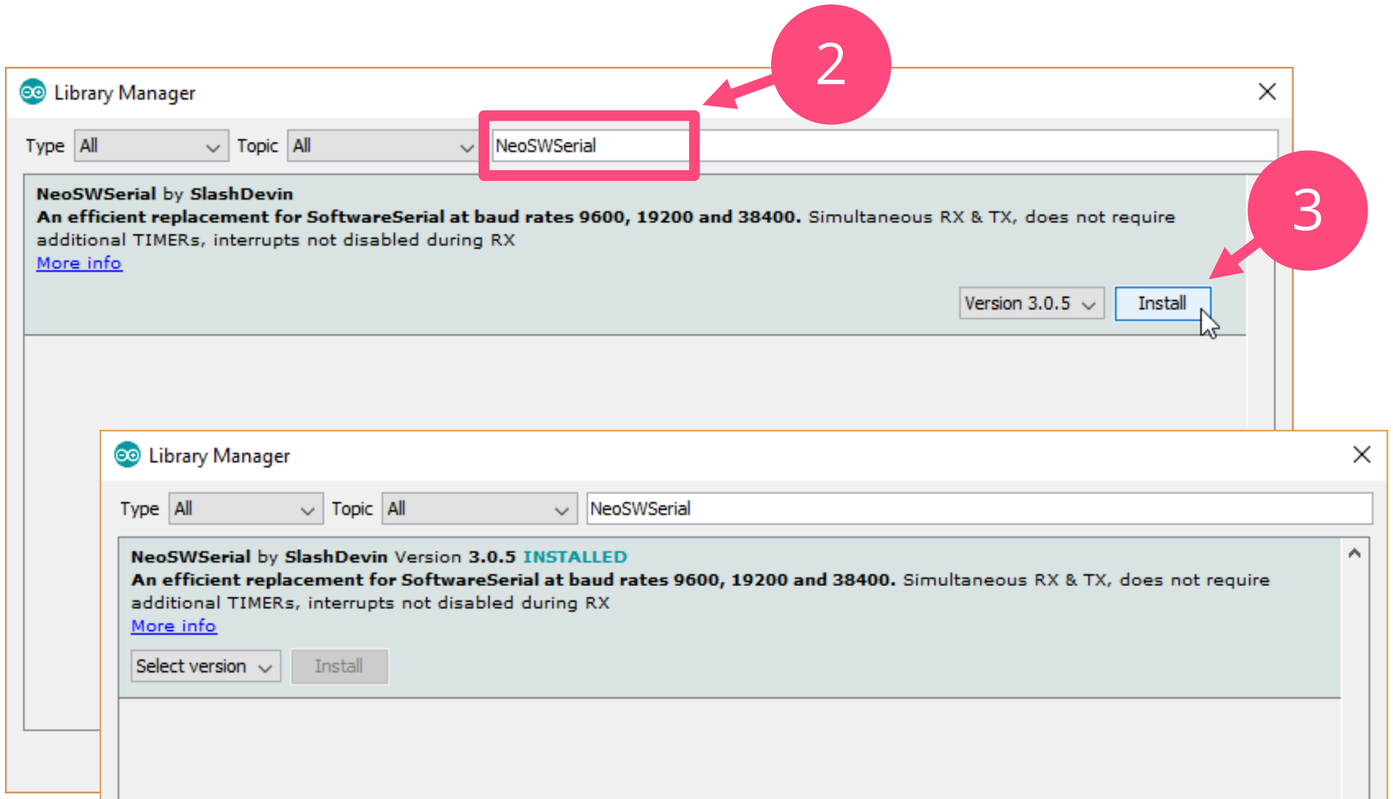


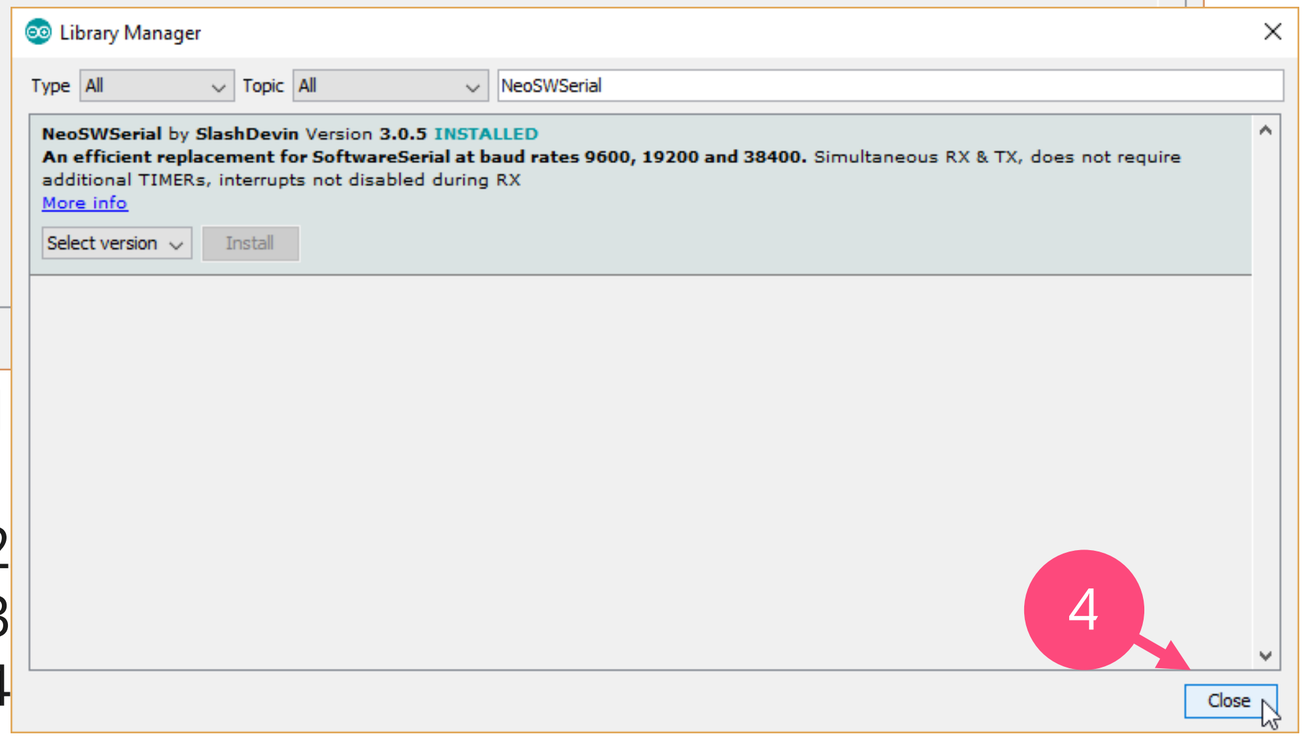


قبل البدء!

تعليمات لإضافة مكتبة جديدة (إلزامية للتواصل مع وحدة الواي فاي)







1) افتح برنامج الأردوينو وانقر فوق أدوات (Tools)، إدارة المكتبات(ManageLibraries).
2) البحث عن «NeoSWSerial».
3) ثم انقر فوق تثبيت.(Install)
4) بمجرد انتهاء التنزيل ، انقر فوق إغلاق(close).

AFF IoT Board folder > Circuits > Controlling_an_LED

ستجد هنا الكودات
البرمجية لكل الدوائر

افتح المسودة :

Controlling_an_LED

Controlling_an_LED

```
/*Start of mandatory lines of codes in each sketch*/
#define RX A0 // define the Receive pin (RX) to communicate with the WiFi module
#define TX A1 // define the Transmit pin (TX) to communicate with the WiFi module
#include <NeoSWSerial.h> // including the library to use the Software Serial rather than the Hardware Serial (Serial)
NeoSWSerial WiFiModule(RX, TX); //initialize the variable to use in communication with the WiFi module
/*End of mandatory lines of code*/

#define RedLedPin   10

void setup() {
  // put your setup code here, to run once:
  WiFiModule.begin(19200); // begin the communication between the WiFi module and the microcontroller on the board
  pinMode(RedLedPin, OUTPUT); // configure the pin connected to the Red Led to be an output
}

void loop() {
  // put your main code here, to run repeatedly:
  if (WiFiModule.available() > 0) // if the WiFi module receive data from the server
  {
    String Command = WiFiModule.readStringUntil('\n'); // read the command sent from the WiFi module to the microcontroller

    if (Command.indexOf("red_led=1") >= 0) // if the received command contains "red_led=1" turn on the LED
    {
      digitalWrite(RedLedPin, HIGH); // turn on the LED
    }
    if (Command.indexOf("red_led=0") >= 0) // if the received command contains "red_led=0" turn it off
    {
      digitalWrite(RedLedPin, LOW); // turn off the LED
    }
  }
}
//(for more info about indexOf function see https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/indexof/)
```

تعليمات برمجية يجب تدوينها..

#include <NeoSWSerial.h> : يمكن هذا السطر من الكودات البرمجية من استيراد مكتبة البرامج التسلسلية ، وتمكن هذه المكتبة من جعل أي من المآخذ بمثابة مأخذ الإرسال (Tx) و مأخذ الاستقبال (Rx) للاتصال التسلسلي بين اللوحة وجهاز آخر (وحدة الواي فاي في لوحة AFF IoT) .

SoftwareSerial WiFiModule(Rx, Tx); : هنا يتم الإعلان عن متغير من النوع « SoftwareSerial» المسمى "WiFiModule" يعين المآخذ A0 و A1 ليكونا مأخذي الإرسال والاستقبال للاتصالات التسلسلية مع وحدة الواي فاي.

تعليمات برمجية يجب تدوينها..

`WiFiModule.available()` : تختبر هذه الوظيفة ما إذا كانت وحدة الواي فاي قد تلقت أمرًا من الإنترنت وتُرجع عدد البايتات المستلمة.

ملاحظة: في حالة عدم وجود بيانات ، تكون قيمة الإرجاع لهذه الوظيفة هي 0.

`String command = WiFimodule.readStringUntil('\n')` : نعرف هنا متغيرا من نوع سلسلة الرموز باسم `command` , وسيتم تعيينه بواسطة نص الأمر الذي تم استلامه من وحدة الوايفاي.

readStringUntil('\n') يعني أن اللوحة تستمر في تلقي البيانات حتى يتم تلقي الحرف "\ n" (وهو تمثيل لمفتاح Enter بلوحة المفاتيح).

ملاحظة: استخدم دائمًا هذا السطر من الكودات البرمجية كما هو مكتوب للحصول على الأمر الذي تم إرساله بواسطة وحدة الوايفاي.

`if(command.indexOf("red_led") >= 0)` : تقوم الوظيفة indexOf باختبار ما إذا كانت السلسلة بين قوسين (كمعامل) موجودة في سلسلة المتصل (سلسلة الأوامر في المثال) وترجع المؤشر حيث يتم الخروج. إذا لم تكن السلسلة موجودة في سلسلة المتصل ، فإن الوظيفة تُرجع -1.

ماذا ستلاحظ ؟

ستلاحظ أن الصمام يضيء و ينطفئ عند الضغط على على زر التبديل في لوحة القيادة. إذا لم تكن كذلك ، فتأكد من قيامك بتجميع الدائرة بشكل صحيح والتحقق من الكود وتحميله على لوحتك أو الاطلاع على نصائح المساعدة أدناه.

مساعدة

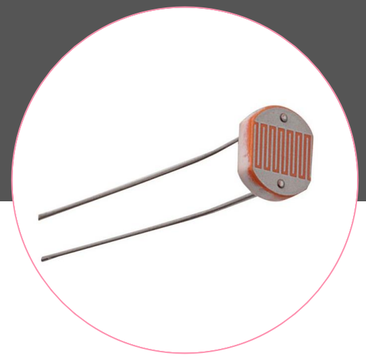
الصمام لا يضيء؟
تعمل المصابيح في اتجاه واحد فقط . لذا حاول إخراجها ولفها بزاوية 180 درجة (لا تقلق ، تثبيتها بالطريقة المعاكسة لا يلحق الضرر بها).

البرنامج لا يحمل؟
يحدث هذا أحيانا, وغالبا ما يكون السبب هو اختيار المأخذ الخاطئ, يمكن تغيير ذلك في tools -> serial port

تطبيقها في الحياة اليومية

مؤشر التشغيل و الإيقاف في معظم الأجهزة الإلكترونية

6 قراءة شدة الضوء (عبر "الإنترنت")



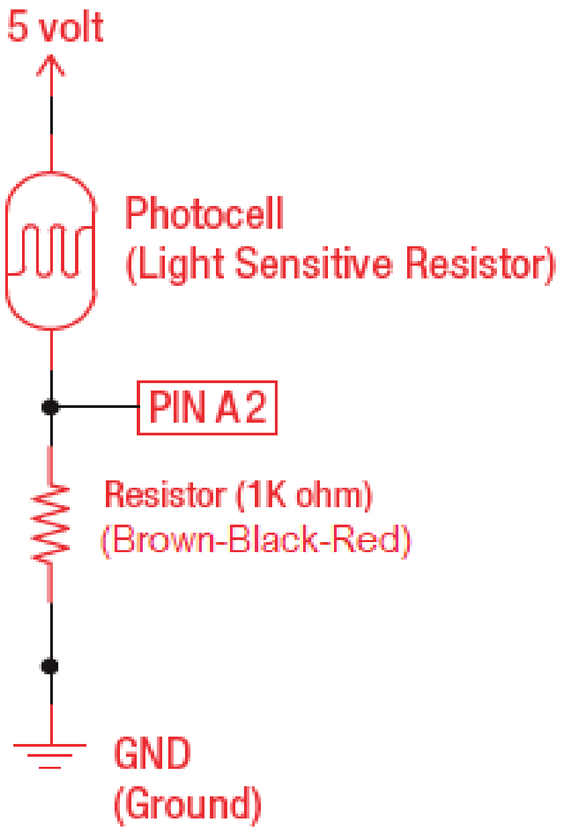
خلية ضوئية

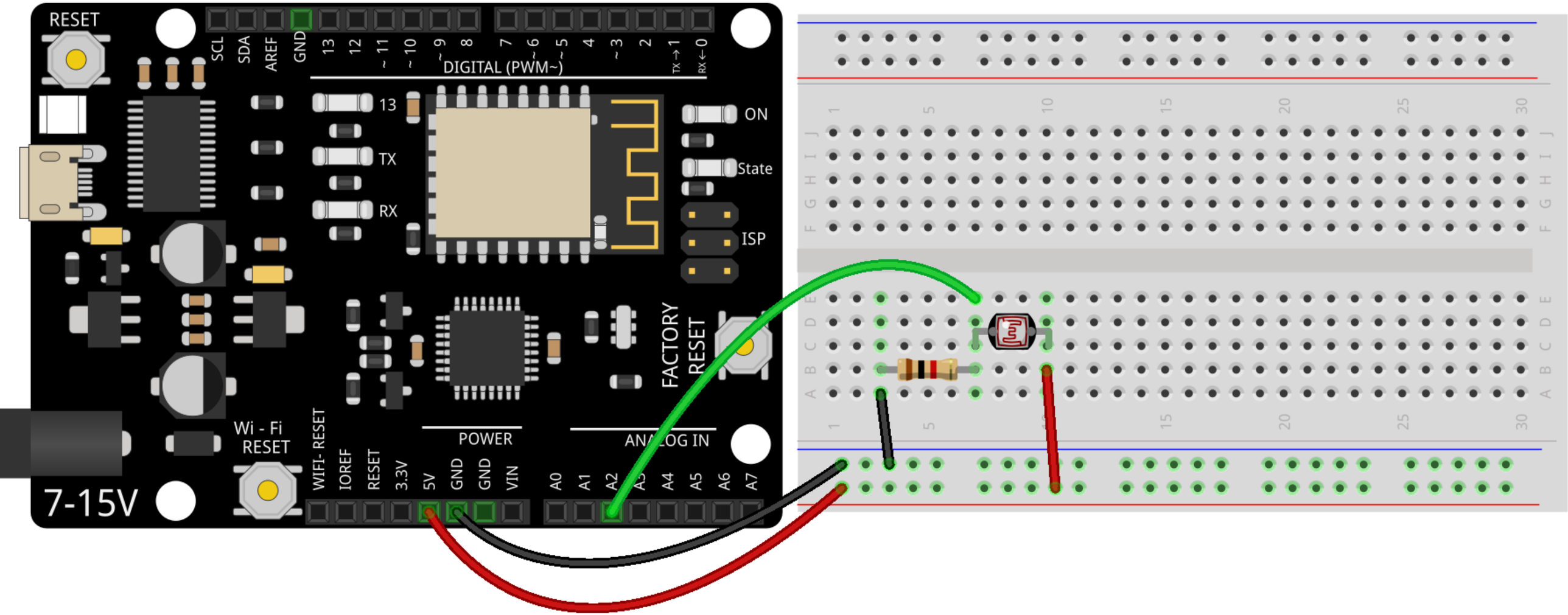
مقاومة ذات جهد 10KΩ

5 من أسلاك التوصيل

خلية ضوئية

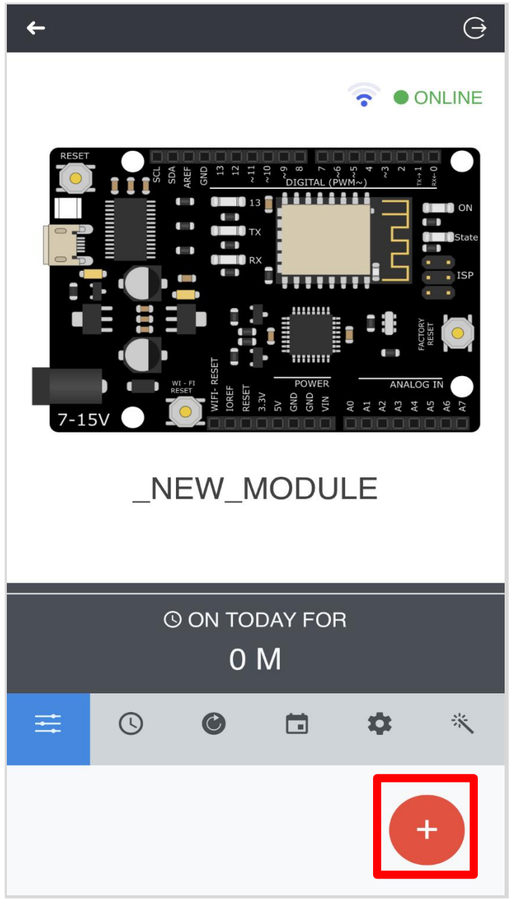
الخلية الضوئية هي عبارة عن مقاومة تختلف قيمتها مع تغير شدة الضوء. عندما يكون الضوء ساطعًا ، يكون جهد الخروج عاليًا ، وعندما يكون الظلام ، يكون جهد الخروج منخفضًا.

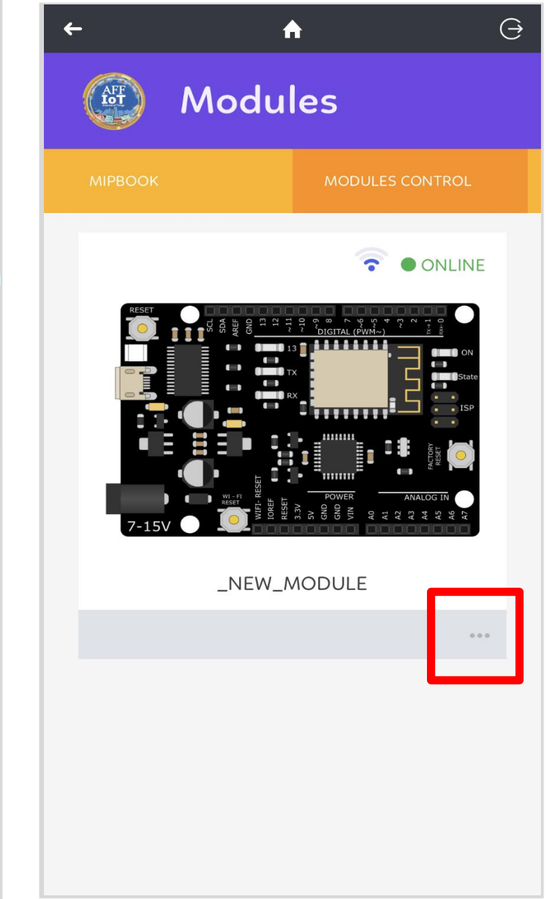



RESET
SCL
SDA
AREF
GND
13
12
~ 11
~ 10
~ 9
8
7
~ 6
~ 5
4
~ 3
2
TX → 1
RX ← 0
DIGITAL (PWM~)
13
TX
RX
ON
State
ISP
FACTORY
RESET
Wi - Fi
RESET
WIFI- RESET
IOREF
RESET
3.3V
5V
GND
GND
VIN
POWER
A0
A1
A2
A3
A4
A5
A6
A7
ANALOG IN
7-15V

fritzing

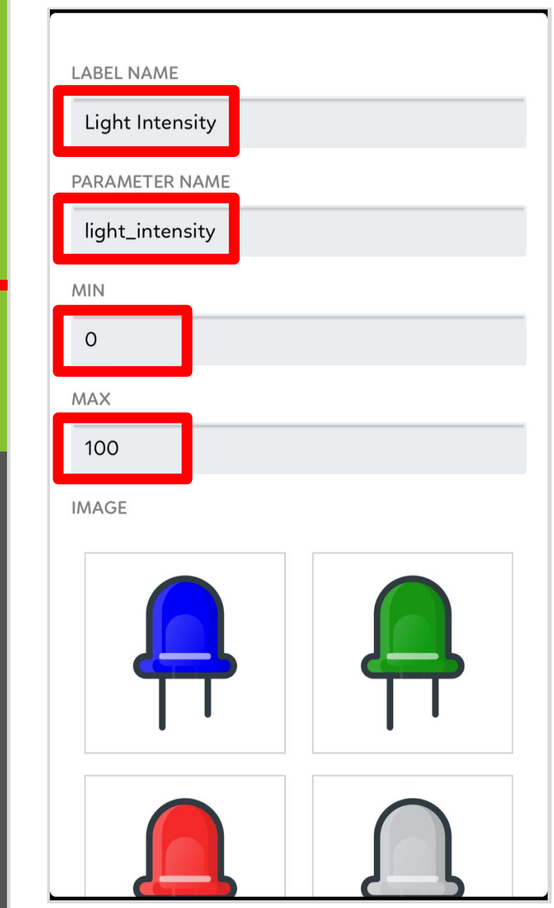
Create new block
Click on the block you would like to add to your dashboard.
ON
35
10
Text

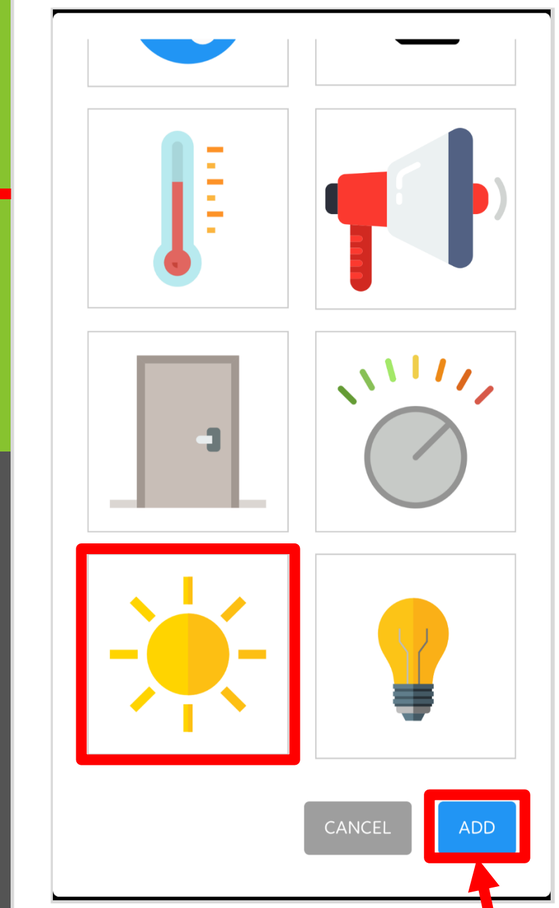

ONLINE
_NEW_MODULE
ON TODAY FOR
0 M


Modules
MIPBOOK
MODULES CONTROL
ONLINE
_NEW_MODULE

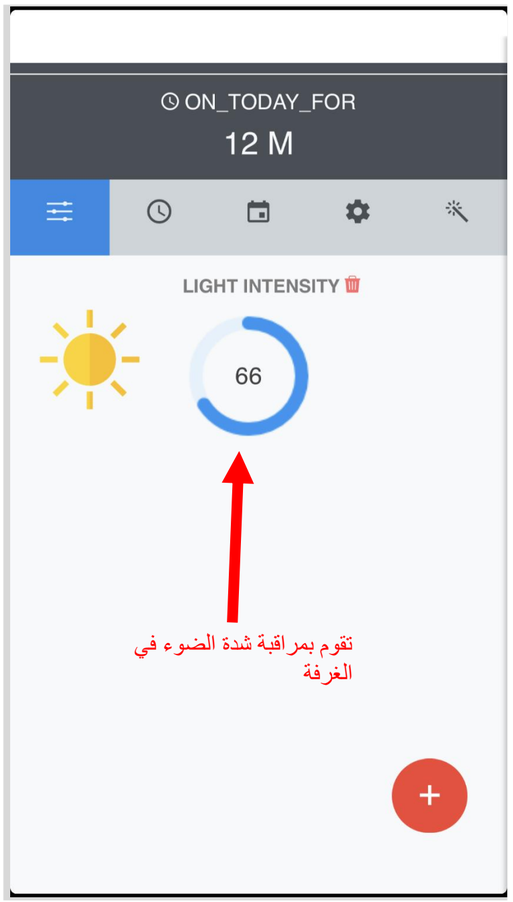
LABEL NAME : هو الاسم الذي يظهر في التطبيق.
PARAMETER NAME : هو الاسم المستخدم في الكودات البرمجية.

أكتب في الفراغ المعلومات التالية





قم بالنزول للأسفل ، ثم انقر ADD



تقوم بمراقبة شدة الضوء في الغرفة

AFF IoT Board folder > Circuits > Reading_Light_Intensity

Reading_Light_Intensity

```
/*Start of mandatory lines of codes in each sketch*/
#define RX A0 // define the Receive pin (RX) to communicate with the WiFi module
#define TX A1 // define the Transmit pin (TX) to communicate with the WiFi module
#include <NeoSWSerial.h> // including the library to use the Software Serial rather than the Hardware Serial (Serial)
NeoSWSerial WiFiModule(RX, TX); //initialize the variable to use in communication with the WiFi module
/*End of mandatory lines of code*/

#define LightSensorPin   A2

void setup() {
  // put your setup code here, to run once:
  WiFiModule.begin(19200); // begin the communication between the WiFi module and the microcontroller on the board
}

void loop() {
  // put your main code here, to run repeatedly:
  int lightIntensity; // declare an integer to read the voltage

  lightIntensity = analogRead(LightSensorPin); // read the actual converted value (between 0 and 1023)
  lightIntensity = map(lightIntensity, 0, 1023, 0, 100); // transform the scale from 0 and 1023 to 0% and 100%

  WiFiModule.println("light_intensity=" + String(lightIntensity)); // send the light intensity value to the server

  delay(3000); // wait for 3 second (3000ms = 3s)
}
```

افتح المسودة :

Reading_Light_Intensity

ماذا ستلاحظ ؟

ستلاحظ زيادة أو نقصان قيمة المقياس وفقًا لشدة الضوء على الخلية الضوئية. إذا لم تكن تعمل ، فتأكد من قيامك بتجميع الدائرة بشكل صحيح والتحقق من الكود وتحميله على اللوحة الخاصة بك أو الاطلاع على نصائح المساعدة أدناه.

مساعدة

مازال لا يعمل
قد تكون في غرفة إما مشرقة أو مظلمة للغاية. حاول تشغيل الأنوار أو إيقاف تشغيلها لمعرفة ما إذا كان هذا يساعد. أو إذا كان لديك مصباح يدوي فيمكنك الاستعانة به.

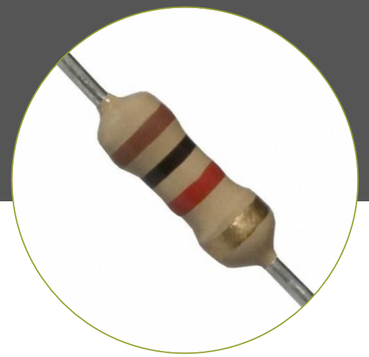
لا تستجيب الخلية لتغيرات الضوء
بما أن تباعد الأسلاك على الخلية الكهروضوئية غير قياسي ، فمن السهل أن تحل محلها على لوح التجارب. تأكد من أنه في المكان المناسب.

تطبيقها في الحياة اليومية

يستخدم مصباح الشارع مستشعر الضوء للكشف عن وقت تشغيله

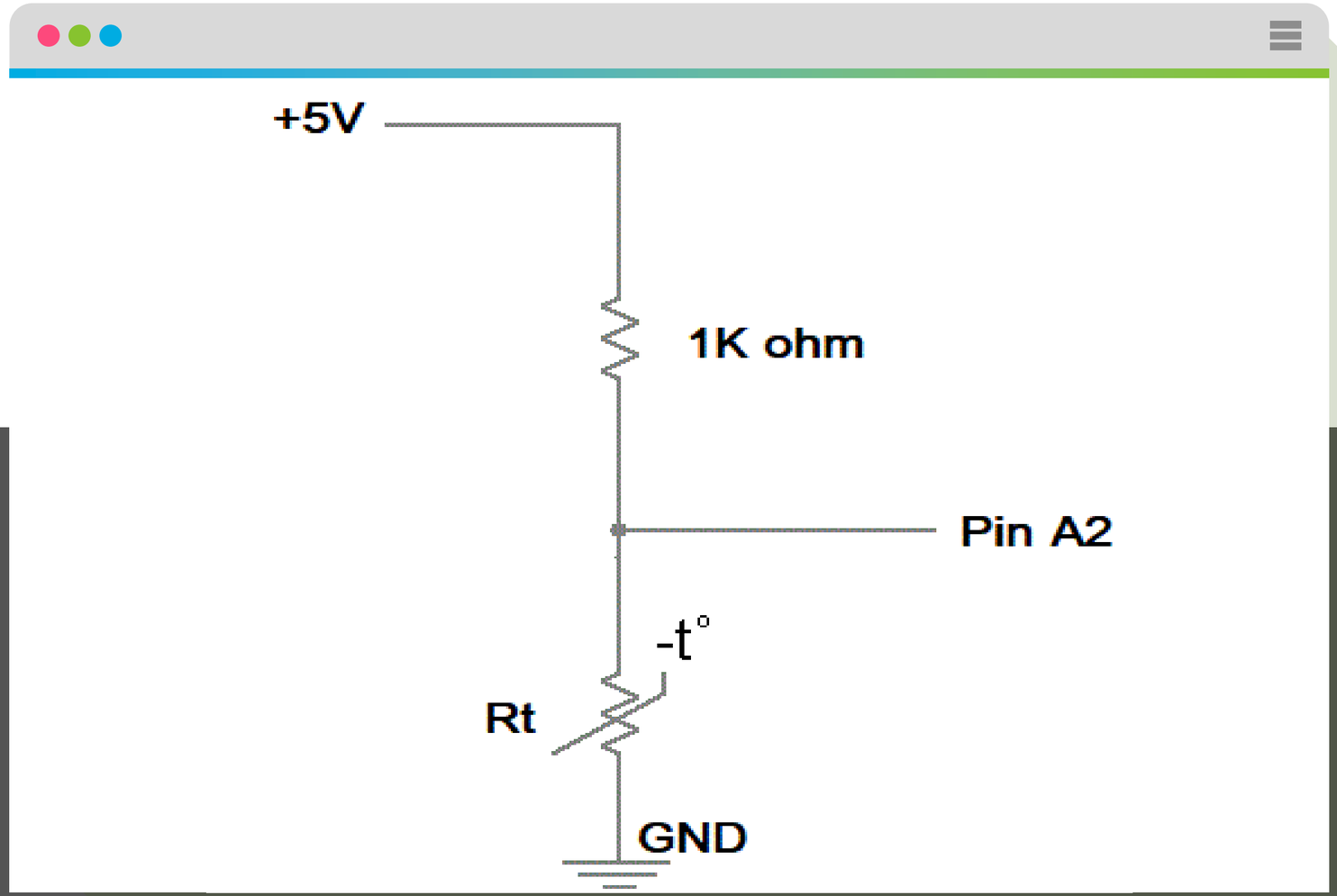
7 مراقبة درجة الحرارة (عبر "الإنترنت")

5 من أسلاك التوصيل



مقاومة ذات جهد 1KΩ

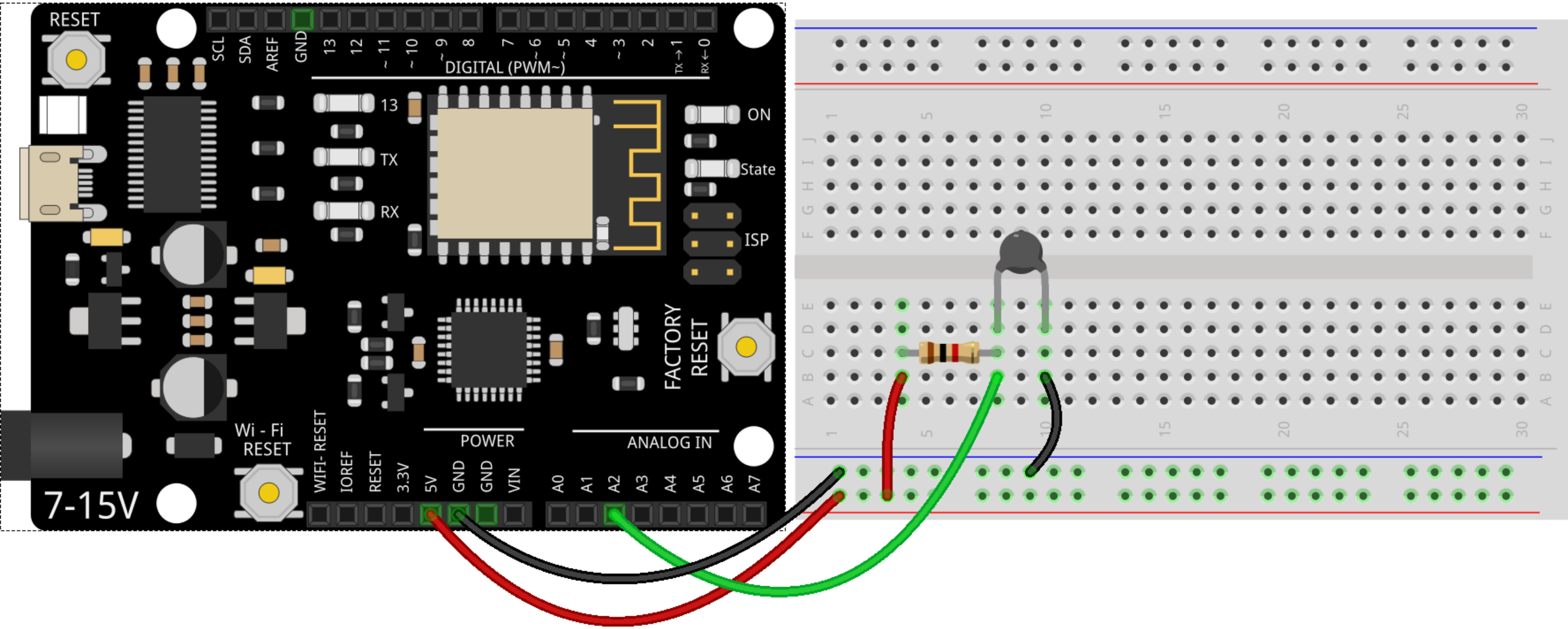
مقاومة حرارية



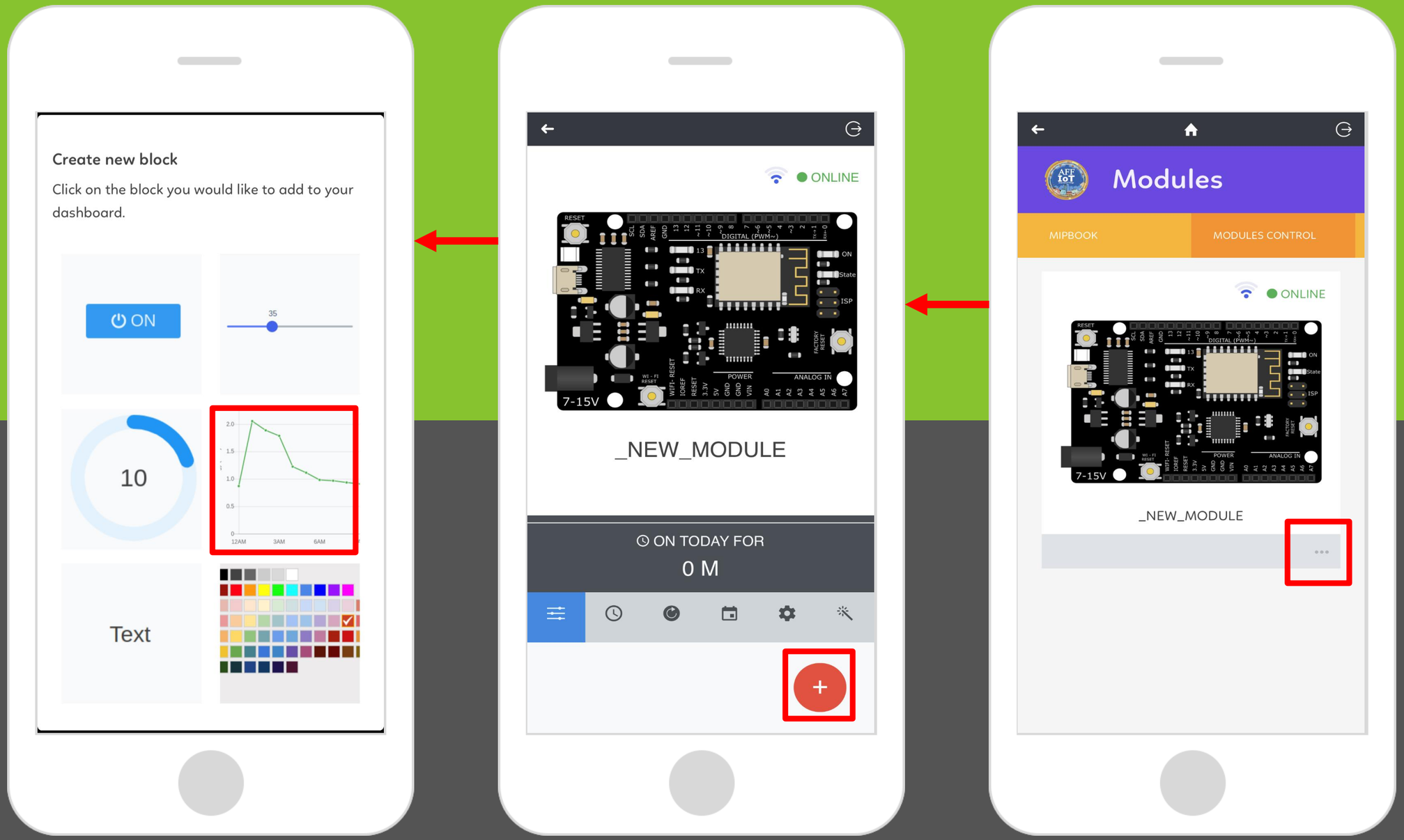
المقاومة الحرارية

المقاومة الحرارية هي نوع من المقاومة الذي تعتمد مقاومته على درجة الحرارة ، وهي أكثر حساسية من المقاومات الإعتيادية.

تستخدم المقاومة الحرارية على نطاق واسع كمستشعرات لدرجة الحرارة (معامل درجة الحرارة السالبة NTC) ، عناصر التسخين ذاتية التنظيم (معامل درجة الحرارة الموجبة PTC). مع المقاومة الحرارية NTC ، تنخفض المقاومة كلما زادت درجة الحرارة.


RESET
SCL
SDA
AREF
GND
13
12
~ 11
~ 10
~ 9
8
7
~ 6
~ 5
4
~ 3
2
TX → 1
RX ← 0
DIGITAL (PWM~)
13
TX
RX
ON
State
ISP
FACTORY
RESET
Wi - Fi
RESET
WIFI- RESET
IOREF
RESET
3.3V
5V
GND
GND
VIN
POWER
A0
A1
A2
A3
A4
A5
A6
A7
ANALOG IN
7-15V

fritzing


Create new block
Click on the block you would like to add to your dashboard.
ON
35
10
Text
ONLINE
_NEW_MODULE
ON TODAY FOR
0 M
Modules
MIPBOOK
MODULES CONTROL
ONLINE
_NEW_MODULE

LABEL NAME : هو الاسم الذي يظهر في التطبيق.
PARAMETER NAME : هو الاسم المستخدم في الكودات البرمجية.

أكتب في الفراغ المعلومات التالية

LABEL NAME
Temperature
PARAMETER NAME
temperature
UNIT
*C
IMAGE

CANCEL
ADD

قم بالنزول للأسفل ،
ثم انقر ADD

ON_TODAY_FOR
17 M

TEMPERATURE

Temperature
30
20
10
0
7:41:28 pm
7:41:58 pm
7:42:28 pm
time

تقوم بمراقبة شدة الضوء في الغرفة

WEATHER Show More
Beirut
Partly Cloudy
Chance of Rain: 0%
28°
28° / 24°

AFF IoT Board > Circuits > Monitoring_Temperature

Monitoring_Temperature

```
/*Start of mandatory lines of codes in each sketch*/
#define RX A0 // define the Receive pin (RX) to communicate with the WiFi module
#define TX A1 // define the Transmit pin (TX) to communicate with the WiFi module
#include <NeoSWSerial.h> // including the library to use the Software Serial rather than the Hardware Serial (Serial)
NeoSWSerial WiFiModule(RX, TX); //initialize the variable to use in communication with the WiFi module
/*End of mandatory lines of code*/

#define TemperatureSensorPin   A2

void setup() {
  // put your setup code here, to run once:
  WiFiModule.begin(19200); // begin the communication between the WiFi module and the microcontroller on the board
  Serial.begin(9600); // begin the communication between the board and the PC
}

void loop() {
  // put your main code here, to run repeatedly:
  float Voltage; // declare a decimal variable to store the read voltage
  int Temperature; // declare an integer variable to calculate the temperature
  Voltage = analogRead(TemperatureSensorPin) * 0.0048828125; // voltage = analog_value * (5/1024);

  Temperature = -21.231 * (Voltage - 3.765);
  // this equation is the linearized form of the temperature equation between 0 and 50 degrees (specific to the thermistor in the kit)

  Serial.print("temperature: "); // print on the serial monitor "Temperature: " and stay on the same line
  Serial.println(Temperature); // print on the serial monitor the actual temperature then go to the next line

  WiFiModule.println("temperature=" + String(Temperature)); // send the Temperature value to the server

  delay(3000); // wait for 3 second (3000ms = 3s)
}
```

افتح المسودة : Monitoring_Temperature

ماذا ستلاحظ ؟

يجب أن تكون قادرًا على قراءة درجة الحرارة التي يكتشفها مستشعر درجة الحرارة لديك ، على الشاشة التسلسلية في نظام الأردوينوعلى تطبيقك. إذا لم تكن تعمل ، فتأكد من قيامك بتجميع الدائرة بشكل صحيح والتحقق من الكود وتحميله على اللوحة الخاصة بك أو الاطلاع على نصائح المساعدة.

مساعدة

قيمة درجة الحرارة لا تتغير

حاول فرك المستشعر بأصابعك لتسخينه أو الضغط عليه بكيس من الثلج لتبريده.

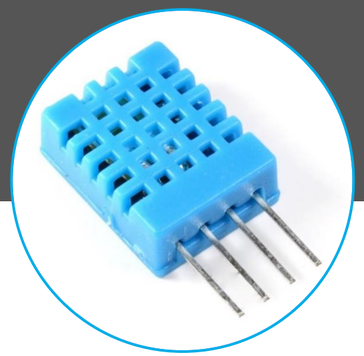
يتم عرض رطانة

يحدث هذا لأن الشاشة التسلسلية تتلقى البيانات على سرعة مختلفة عن المتوقع. انقر على المربع المنسدل الذي يقرأ " baud ***"وقم بتغييره إلى" 9600 baud "

تطبيقها في الحياة اليومية

أنظمة التحكم في المناخ تستخدم جهاز استشعار درجة الحرارة
لمراقبة والحفاظ على إعدادات التبريد أو التدفئة.

8 قياس الحرارة والرطوبة (عبر "الإنترنت")



DHT11 x1

"Humidity and Temperature sensor"

x1 مقاومة 10KΩ

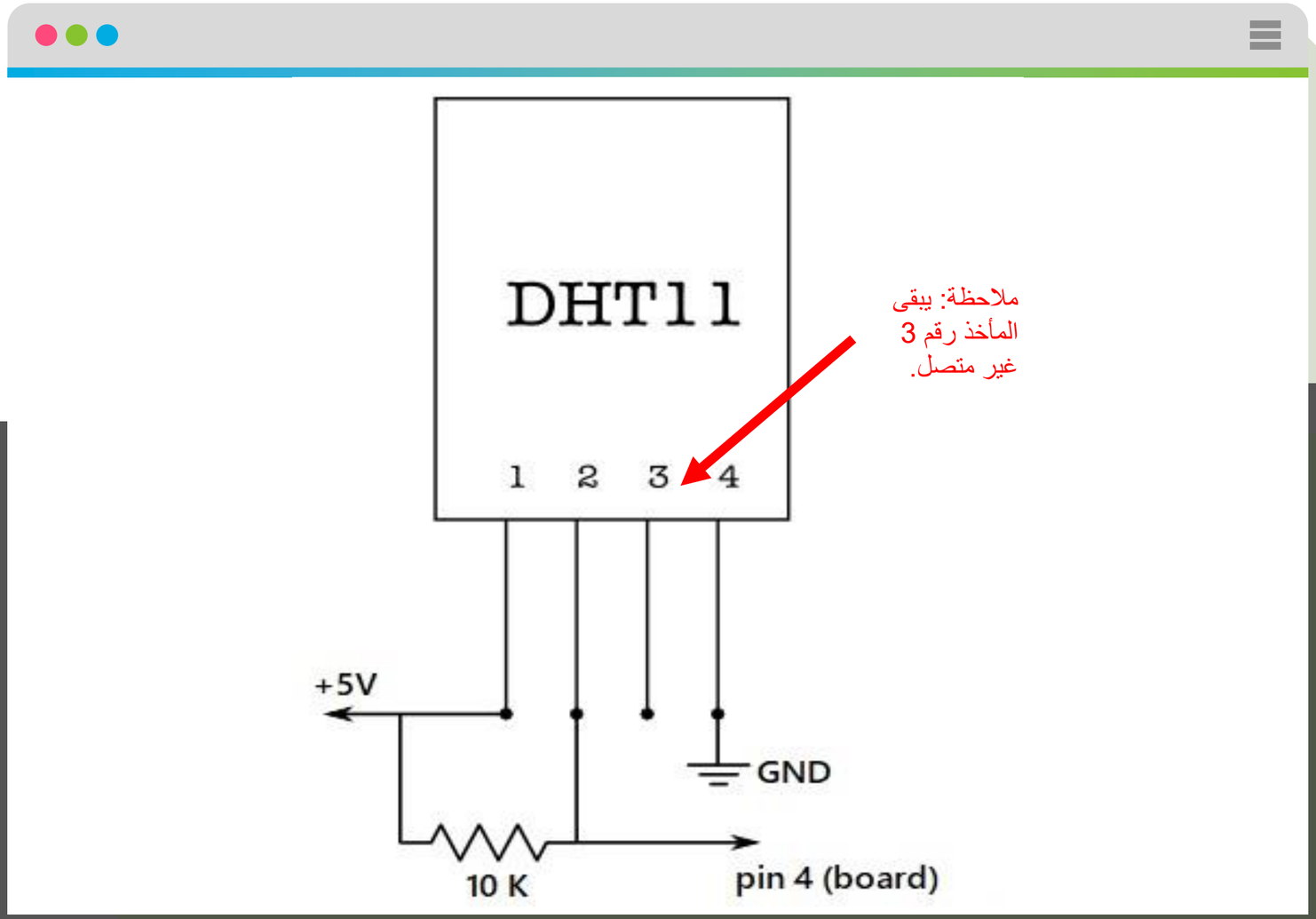
x6 أسلاك التوصيل

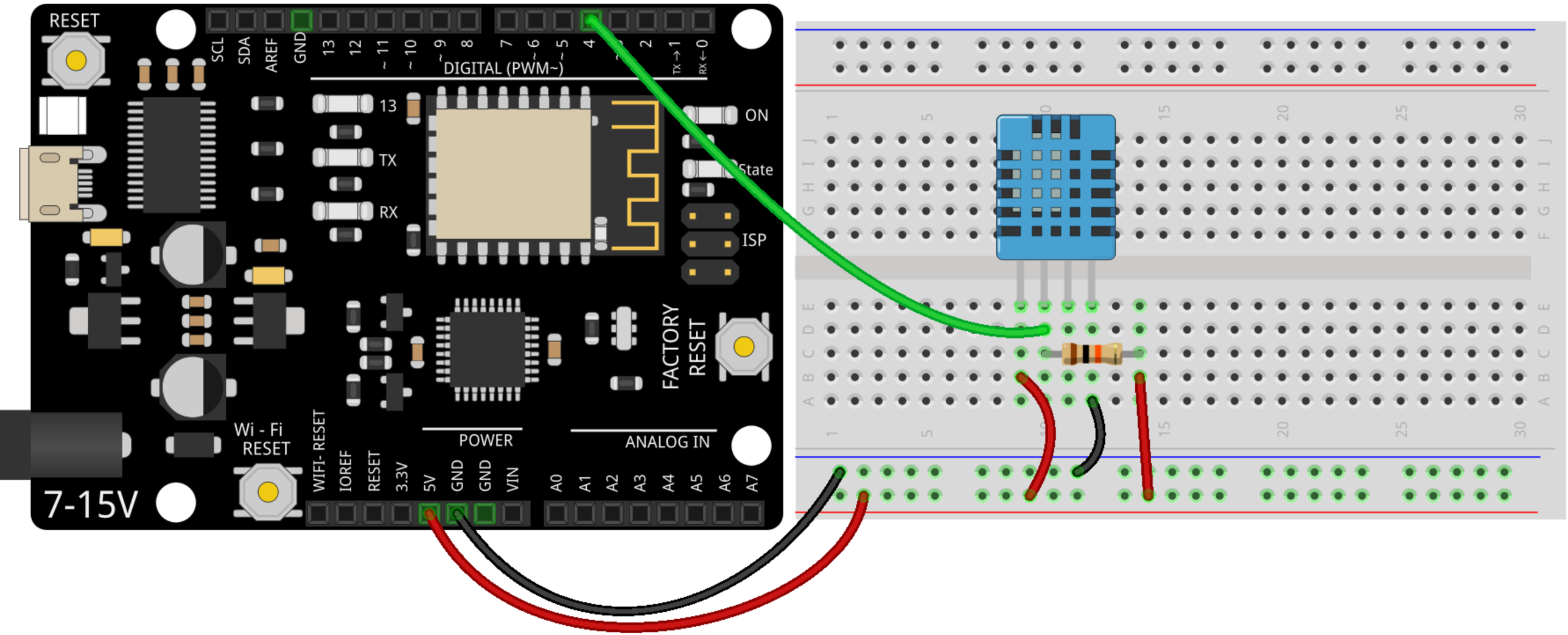
مستشعر الحرارة و الرطوبة

DHT11 مستشعر الحرارة و الرطوبة

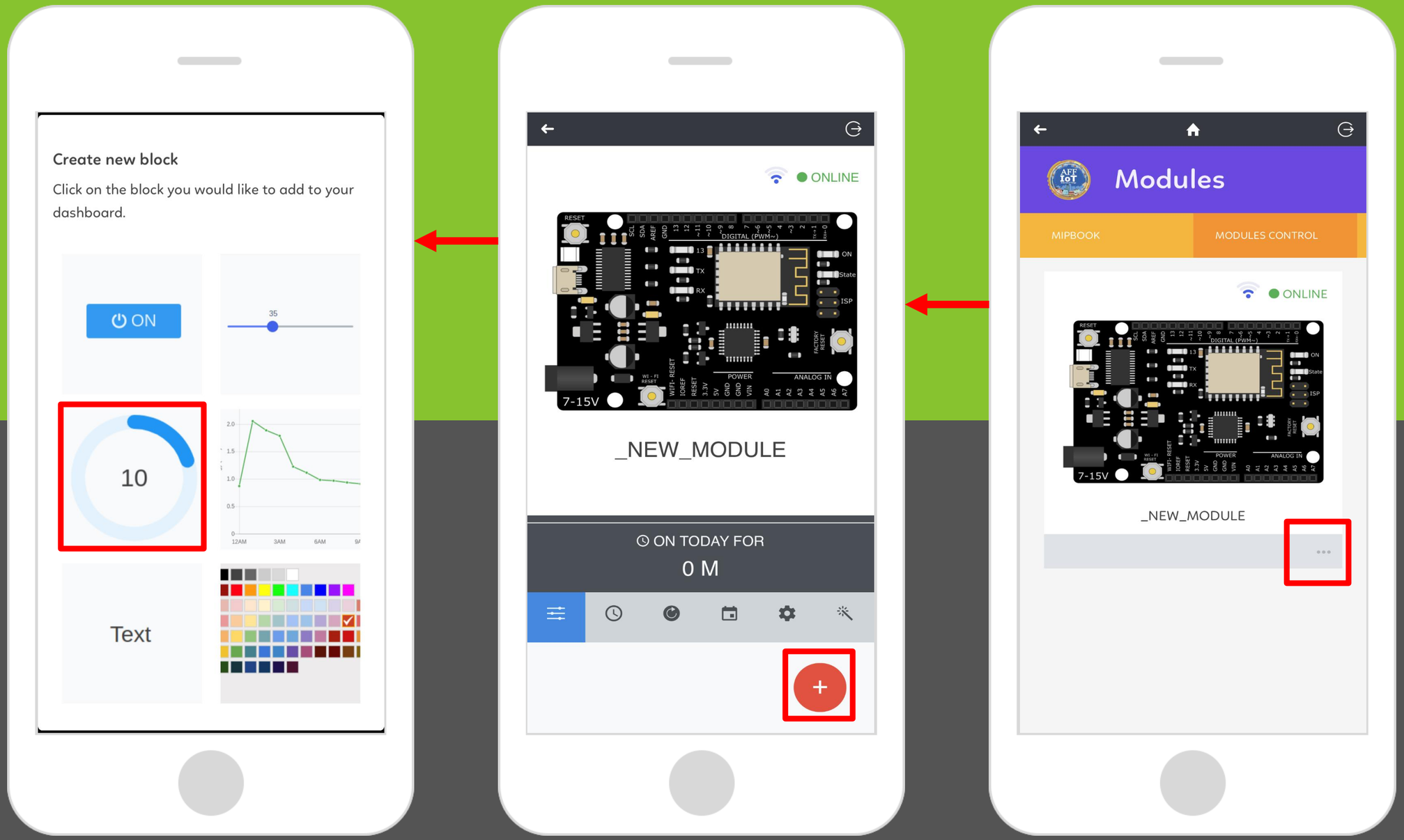
يتميز بمجموعة استشعار لدرجة الحرارة والرطوبة مع إخراج إشارة رقمية معايرة.

يشمل المستشعر مكونا لقياس الرطوبة ومكونا لقياس درجة الحرارة.



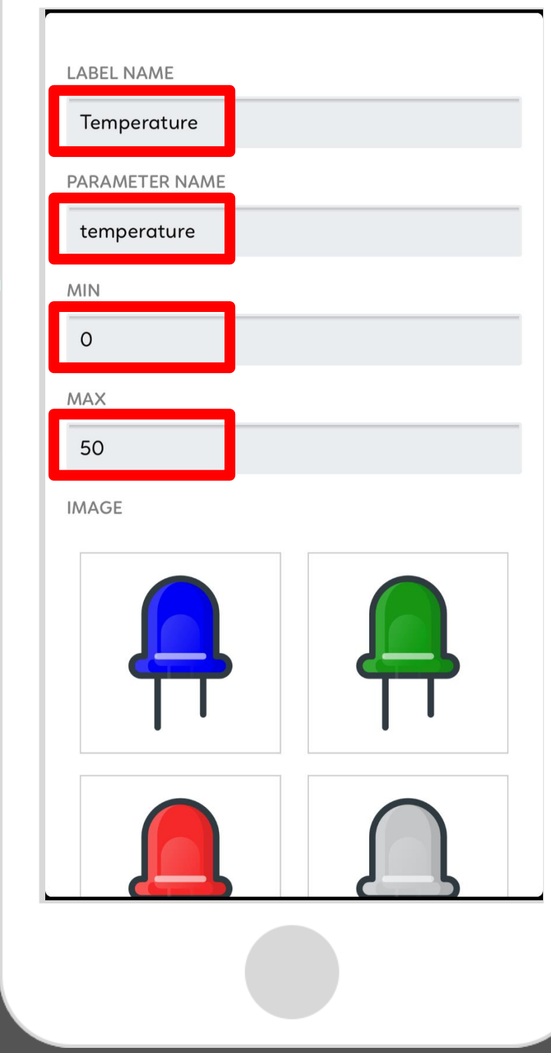

RESET
SCL
SDA
AREF
GND
13
12
~ 11
~ 10
~ 9
8
7
~ 6
~ 5
4
2
TX → 1
RX ← 0
DIGITAL (PWM~)
13
TX
RX
ON
State
ISP
FACTORY RESET
Wi - Fi RESET
WIFI- RESET
IOREF
RESET
3.3V
5V
GND
GND
VIN
POWER
A0
A1
A2
A3
A4
A5
A6
A7
ANALOG IN
7-15V

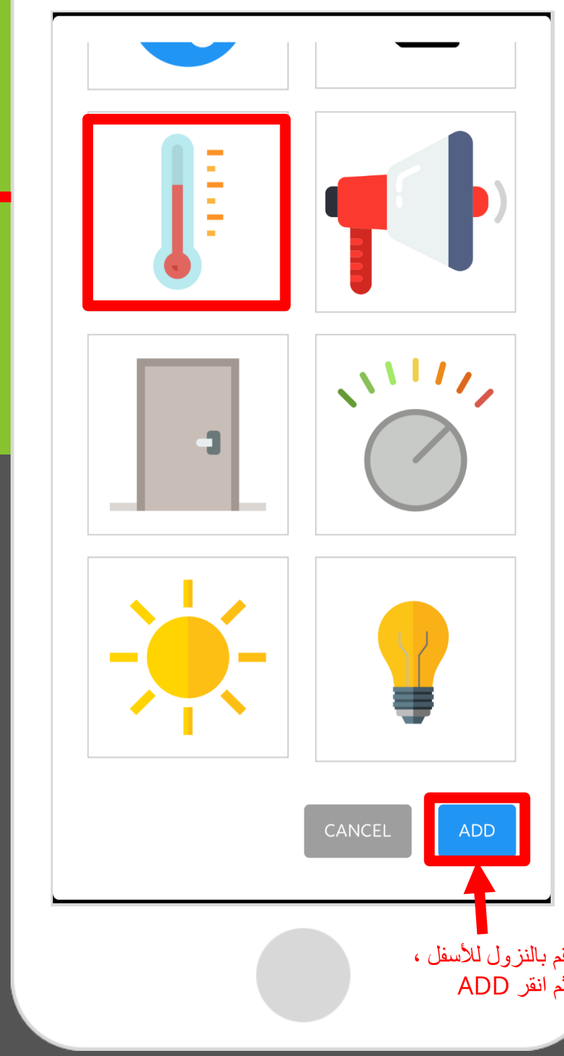
fritzing


Create new block
Click on the block you would like to add to your dashboard.
ON
35
10
Text
ONLINE
_NEW_MODULE
ON TODAY FOR
0 M
Modules
MIPBOOK
MODULES CONTROL
ONLINE
_NEW_MODULE

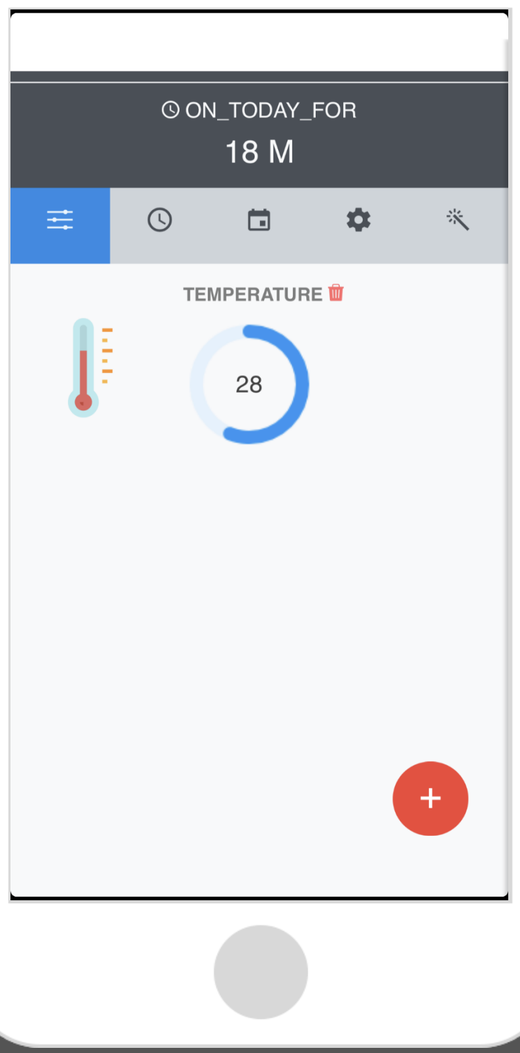
LABEL NAME : هو الاسم الذي يظهر في التطبيق.
PARAMETER NAME : هو الاسم المستخدم في الكودات البرمجية.

أكتب في الفراغ المعلومات التالية



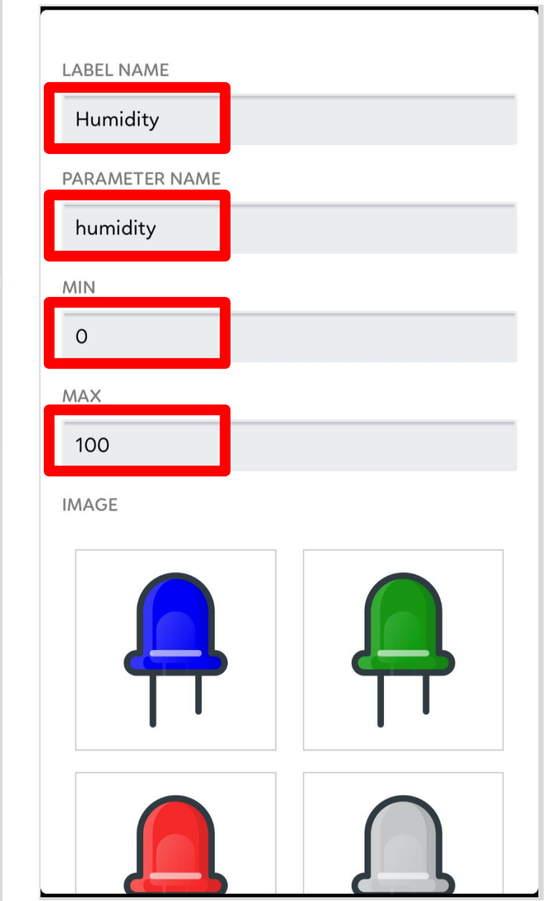


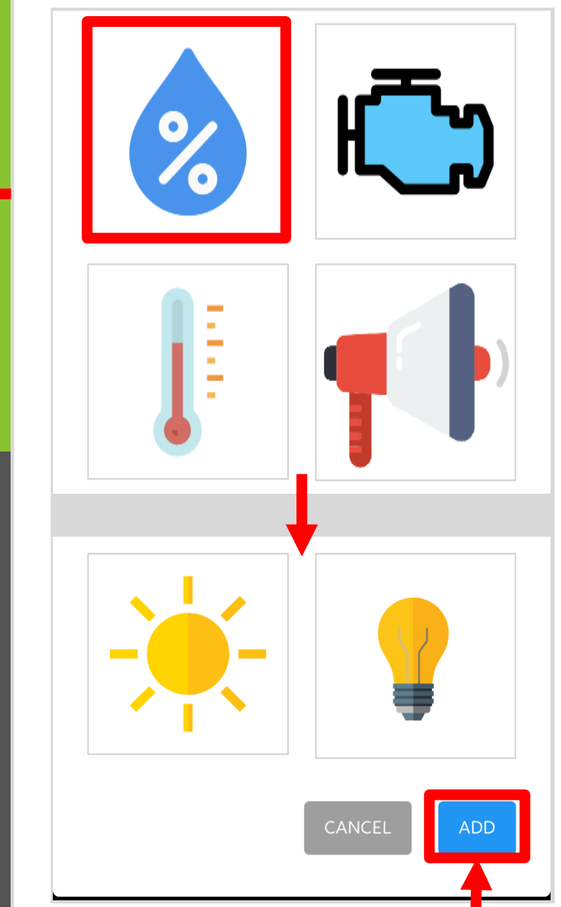
قم بالنزول للأسفل ،
ثم انقر ADD



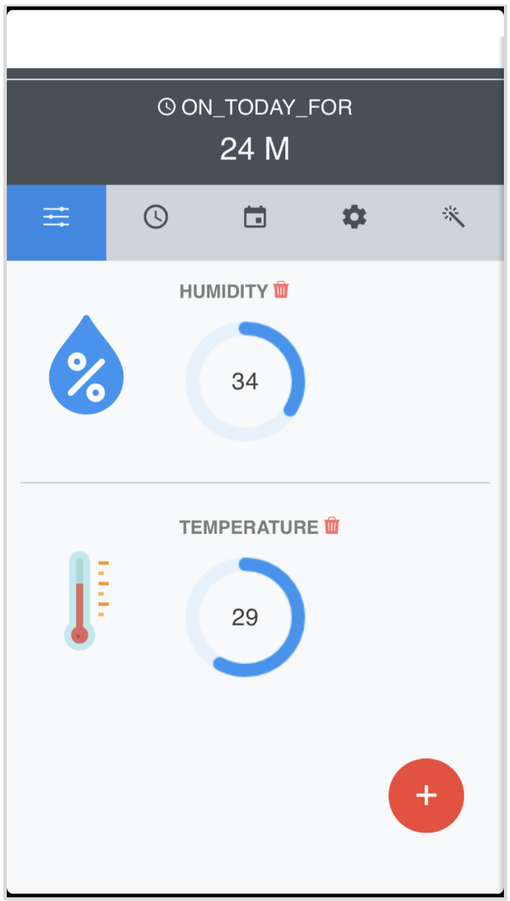
LABEL NAME : هو الاسم الذي يظهر في التطبيق.
PARAMETER NAME : هو الاسم المستخدم في الكودات البرمجية.

أكتب في الفراغ المعلومات التالية





قم بالنزول للأسفل ،
ثم انقر ADD



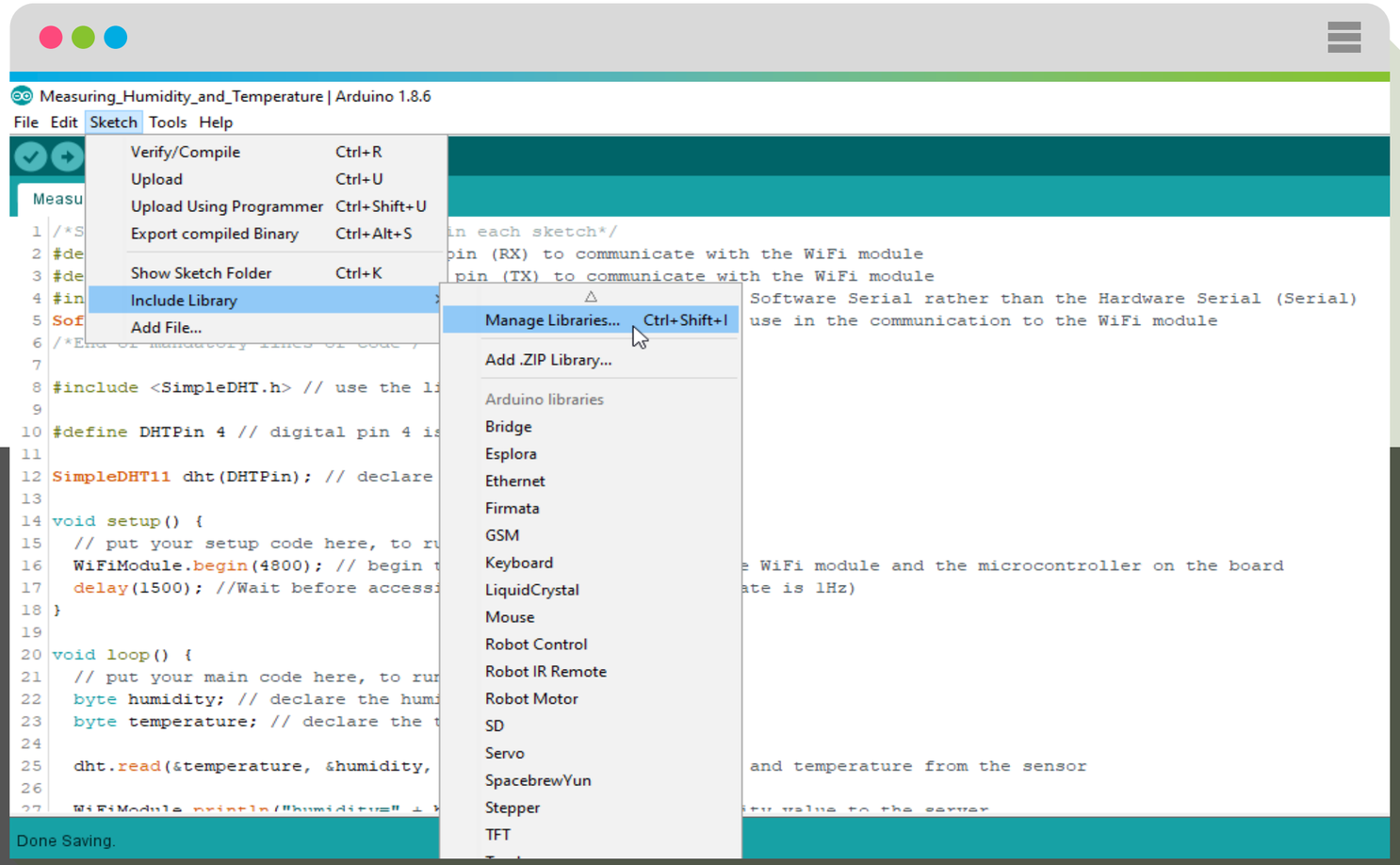
AFF IoT Board > Circuits > Measuring_Humidity_and_Temperature

Measuring_Humidity_and_Temperature

```
/*Start of mandatory lines of codes in each sketch*/
#define RX A0 // define the Receive pin (RX) to communicate with the WiFi module
#define TX A1 // define the Transmit pin (TX) to communicate with the WiFi module
#include <NeoSWSerial.h> // including the library to use the Software Serial rather than the Hardware Serial (Serial)
NeoSWSerial WiFiModule(RX, TX); //initialize the variable to use in communication with the WiFi module
/*End of mandatory lines of code*/

#include <SimpleDHT.h> // use the library functions in the code

#define DHTPin 4 // digital pin 4 is connected to the sensor

SimpleDHT11 dht(DHTPin); // declare the DHT11 instance

void setup() {
  // put your setup code here, to run once:
  WiFiModule.begin(19200); // begin the communication between the WiFi module and the microcontroller on the board
  delay(1500); //Wait before accessing Sensor (DHT11 sampling rate is 1Hz)
}

void loop() {
  // put your main code here, to run repeatedly:
  byte humidity; // declare the humidity variable
  byte temperature; // declare the temperature variable

  dht.read(&temperature, &humidity, NULL); // read both humudity and temperature from the sensor

  String data = "";
  data += "humidity=" + String(humidity); //example: humidity=70
  data += ","; // all parameters should be comma delimited
  data += "temperature=" + String(temperature); //example: temperature=27
  //data = "humidity=70,temperature=27"

  WiFiModule.println(data);// send the Humidity and Temperature values to the server

  delay(3000);//Wait 3 seconds before accessing sensor again.
}
```

افتح المسودة :

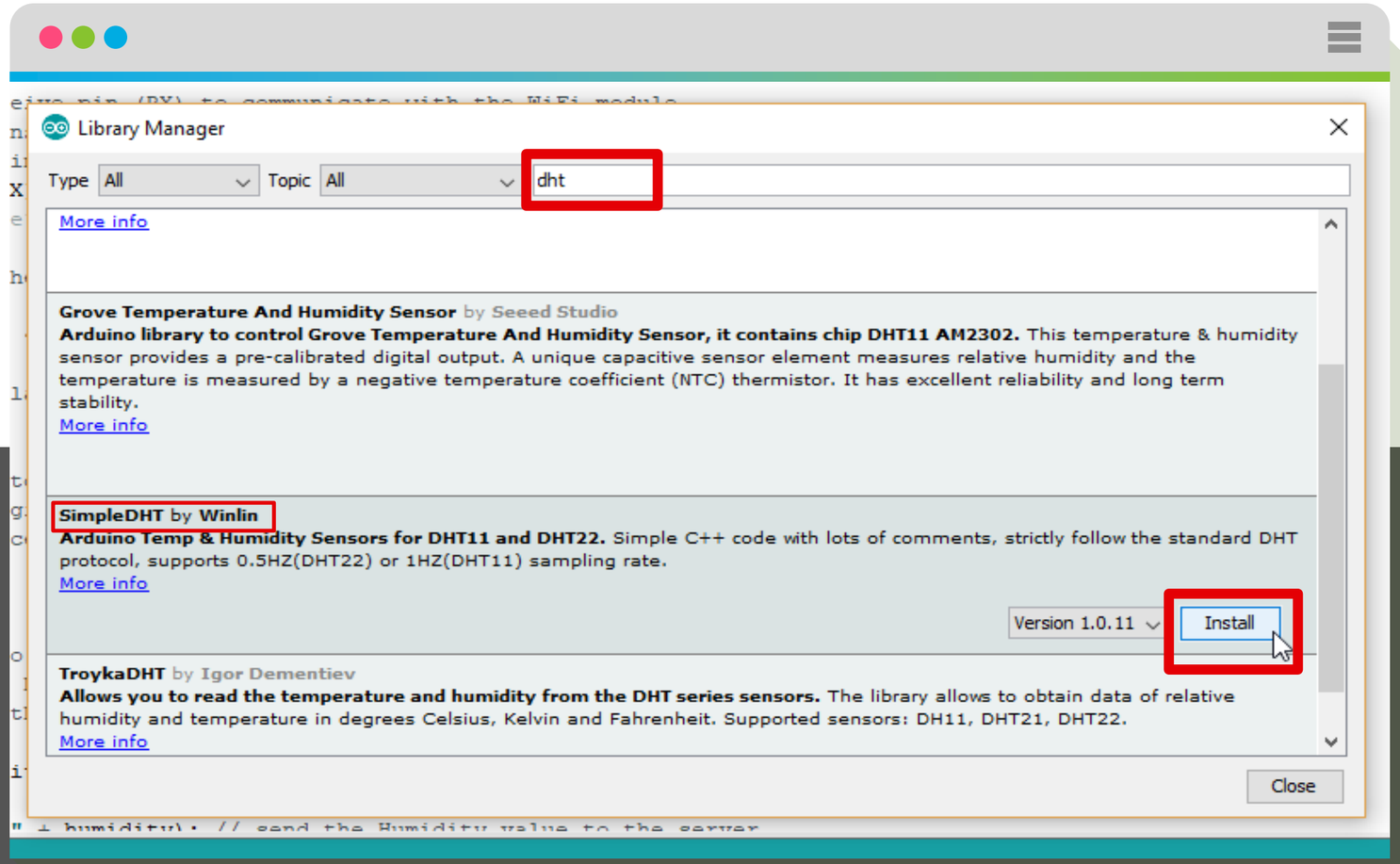
Measuring_Humidity_
and_Temperature



قبل معالجة البيانات

قم بتنزيل مكتبة DHT

Sketch > Include Library > Manage Libraries



قبل معالجة البيانات

قم بتنزيل مكتبة DHT

ابحث عن « dht » وحدد الإصدار الأحدث من مكتبة « SimpleDHT »وانقر فوق "تثبيت".

قبل معالجة البيانات

قم بتنزيل مكتبة DHT

بعد النقر فوق الزر "إغلاق" ، يمكننا أن نرى أن نظام الاردوينو يتعرف على نوع SimpleDHT ويغير لونه إلى اللون البرتقالي.

تعليمات برمجية يجب تدوينها..

```
dht.read(&temperature, &humidity, NULL);
```

يستدعي هذا السطر من الكودات البرمجية وظيفة القراءة للحصول على الرطوبة ودرجة الحرارة من جهاز الاستشعار DHT11, ويضع القيم في متغيرات "درجة الحرارة" و "الرطوبة".

ماذا ستلاحظ ؟

تظهرلك القيم الحقيقية للرطوبة و درجة الحرارة. إذا لم يتم ذلك، فتأكد من قيامك بتجميع الدائرة بشكل صحيح والتحقق من الكود وتحميله على اللوحة الخاصة بك أو الاطلاع على نصائح المساعدة أدناه.

مساعدة

القيم لا تتغير
حاول فرك المستشعر بأصابعك لتسخينه أو الضغط على كيس من الثلج بالقرب منه لتبريده.

لا شيء يحدث
هذا البرنامج لا يوجد لديه إشارة إلى الخارج أنه يعمل. للاطلاع على النتائج ، يجب عليك فتح شاشة نظام الأردوينو التسلسلية.

تطبيقها في الحياة اليومية

قياس الرطوبة في غرف التخزين لمنع تلف العناصر المخزنة

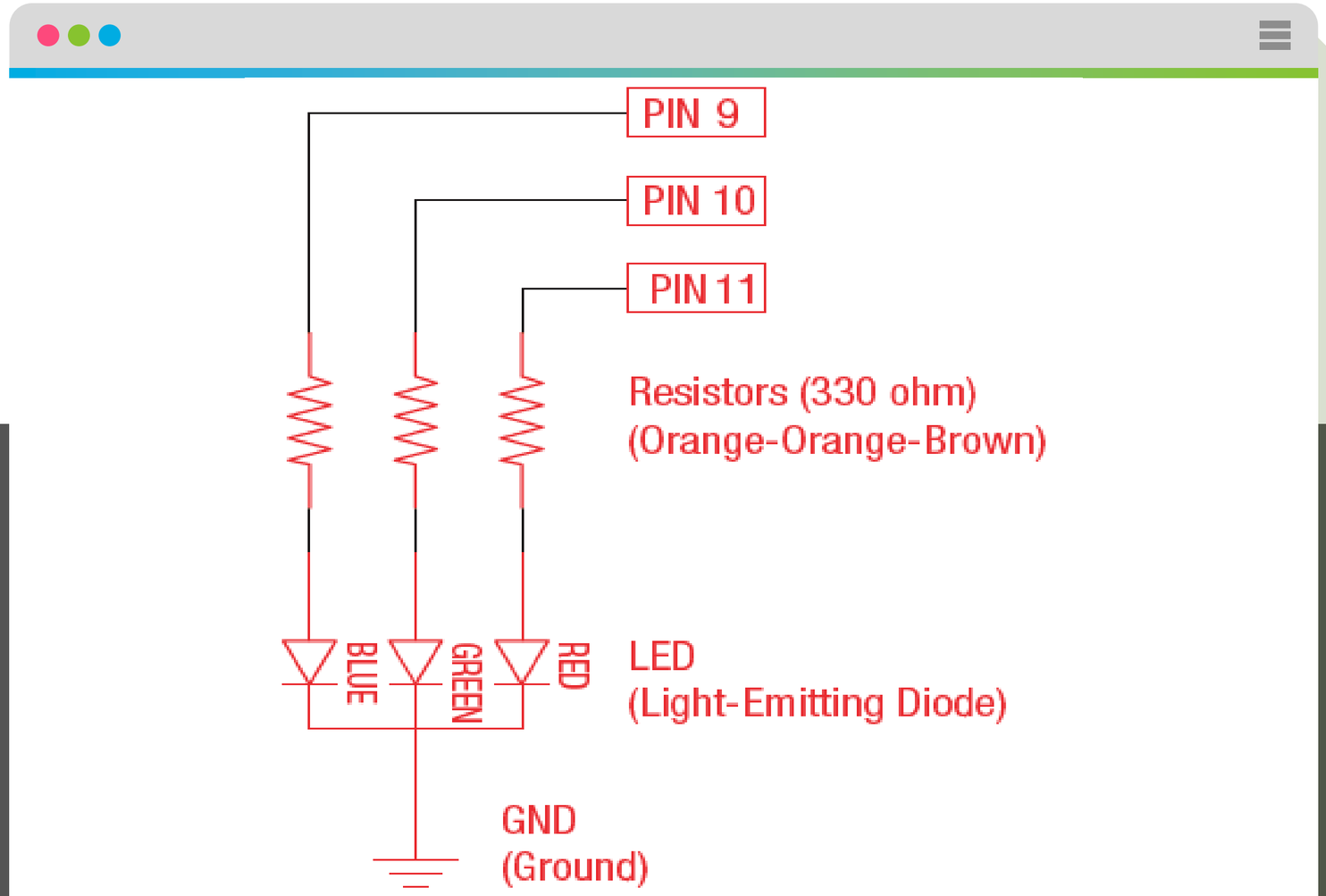
9 ضبط لون الصمام RGB (عبر "الإنترنت")

5 من أسلاك التوصيل

3 من المقاومة ذات الجهد 330Ω

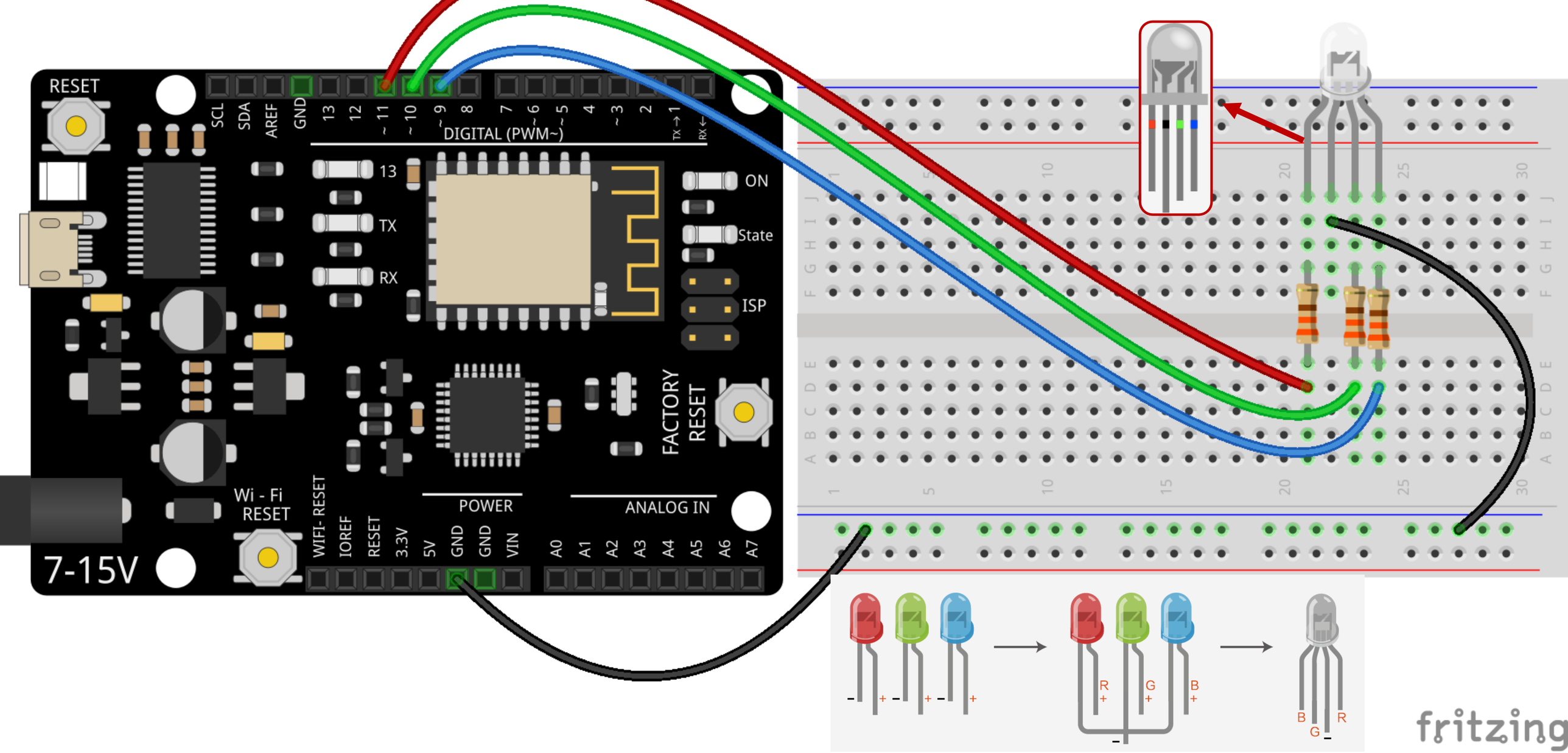
صمام مضيء RGB

الصمام المضيء RGB

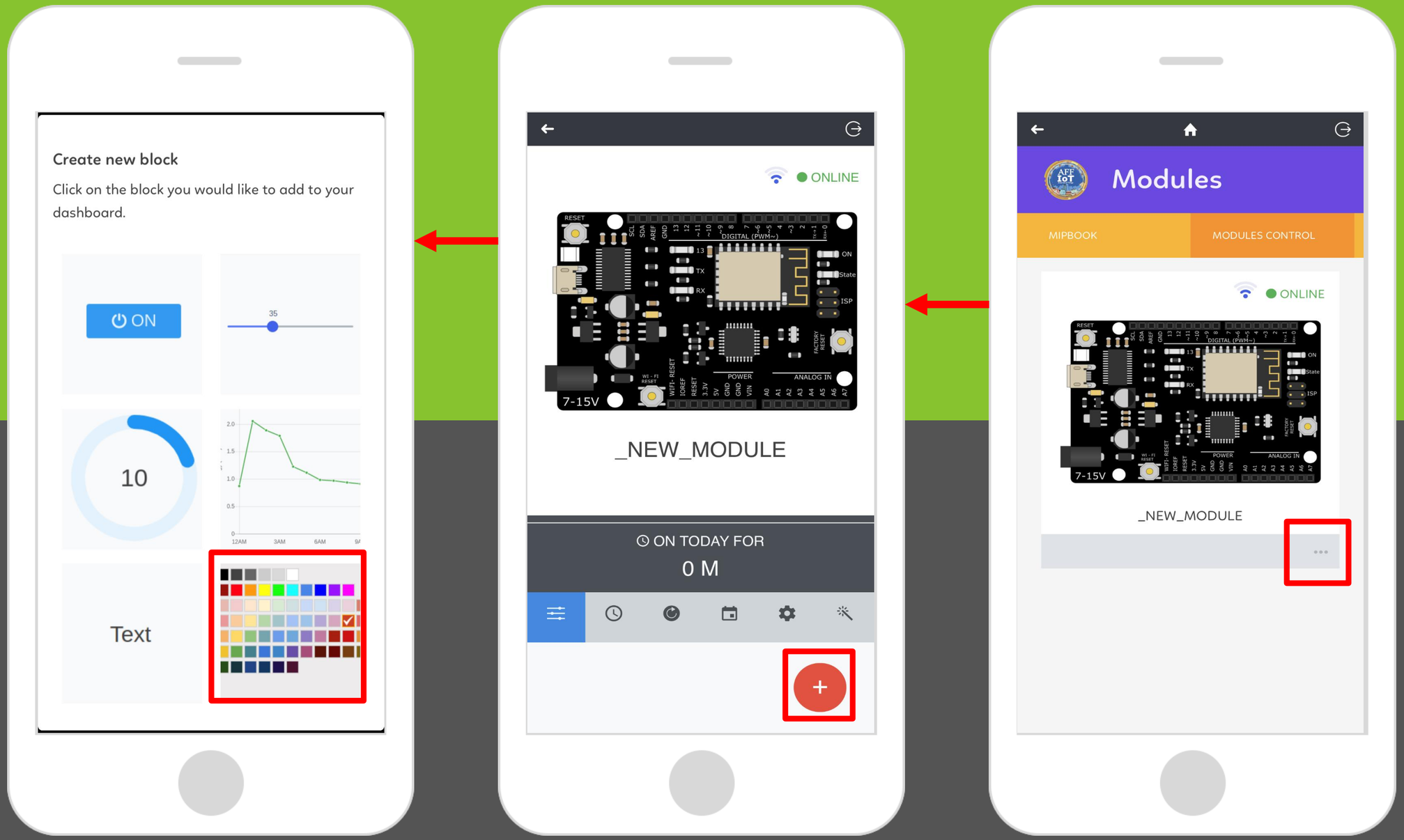


تحتوي مصابيح RGB (الأحمر والأخضر والأزرق) ، على ثلاث ثنائيات مختلفة ينبعث منها اللون ويمكن دمجها لإنشاء كل الألوان. في هذه المثال ، ستتعلم كيفية استخدام صمام RGB للحصول على ألوان فريدة من نوعها.

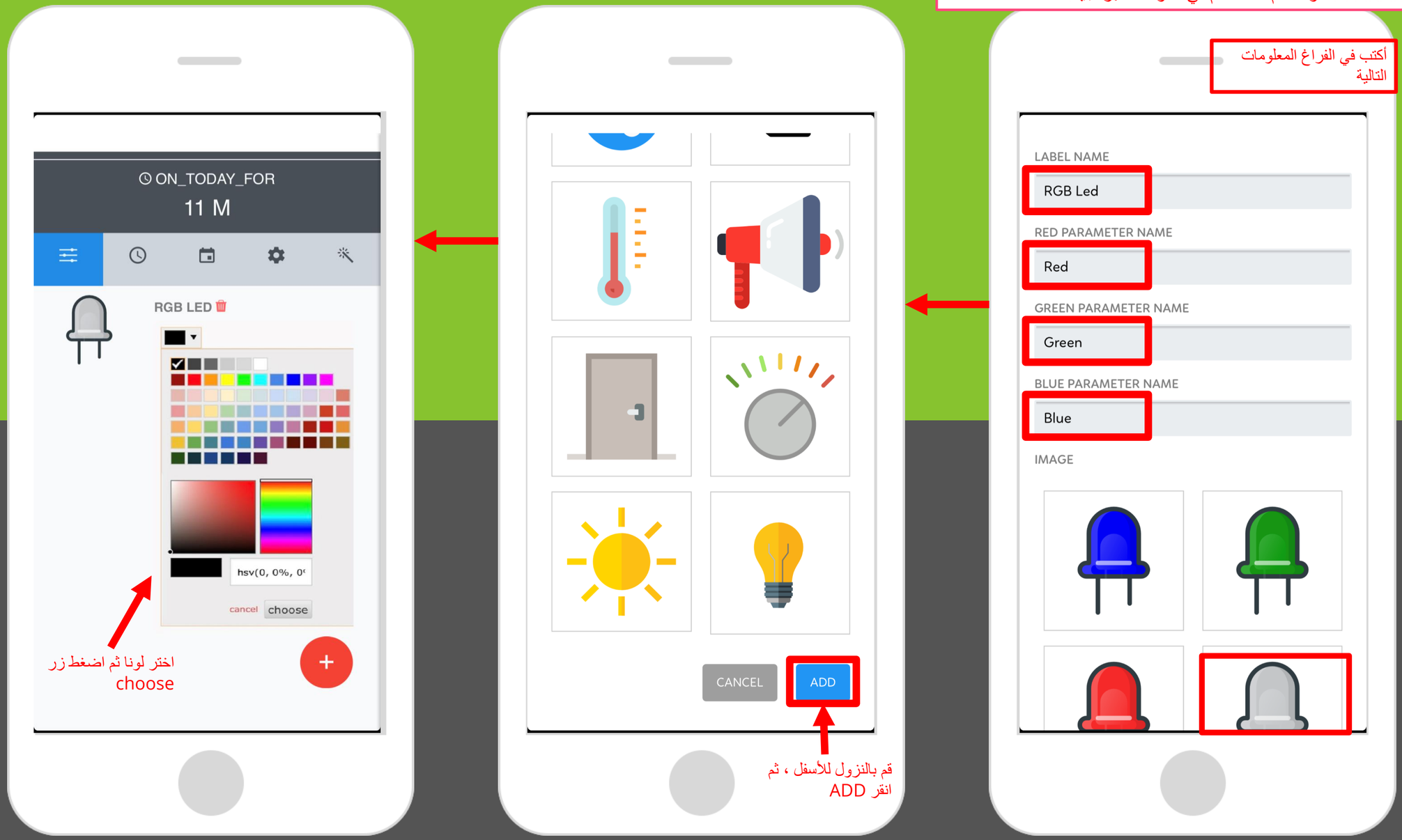
اعتمادًا على مدى سطوع كل صمام ثنائي ، يكون تركيب أي لون تقريبًا ممكنًا!


RESET
SCL
SDA
AREF
GND
13
12
~ 11
~ 10
~ 9
8
7
~ 6
~ 5
4
~ 3
2
TX → 1
RX ←
DIGITAL (PWM~)
13
TX
RX
ON
State
ISP
FACTORY RESET
Wi - Fi RESET
WIFI- RESET
IOREF
RESET
3.3V
5V
GND
GND
VIN
POWER
A0
A1
A2
A3
A4
A5
A6
A7
ANALOG IN
7-15V
R
G
B
B
G
R

fritzing


Create new block
Click on the block you would like to add to your dashboard.
ON
35
10
Text
ONLINE
_NEW_MODULE
ON TODAY FOR
0 M
Modules
MIPBOOK
MODULES CONTROL
ONLINE
_NEW_MODULE

LABEL NAME : هو الاسم الذي يظهر في التطبيق.
PARAMETER NAME : هو الاسم المستخدم في الكودات البرمجية.



AFF IoT Board folder > Circuits > Adjusting_RGB_led_Color

1

افتح المسودة :

Adjusting_RGB_LED_Color

Adjusting_RGB_LED_Color

```
/*Start of mandatory lines of codes in each sketch*/
#define RX A0 // define the Receive pin (RX) to communicate with the WiFi module
#define TX A1 // define the Transmit pin (TX) to communicate with the WiFi module
#include <NeoSWSerial.h> // including the library to use the Software Serial rather than the Hardware Serial (Serial)
NeoSWSerial WiFiModule(RX, TX); //initialize the variable to use in communication with the WiFi module
/*End of mandatory lines of code*/

#define RedLedPin    11
#define GreenLedPin  10
#define BlueLedPin   9

String RedStringValue = "";
String GreenStringValue = "";
String BlueStringValue = "";

int RedValue = 0;
int GreenValue = 0;
int BlueValue = 0;

void setup() {
  // put your setup code here, to run once:
  WiFiModule.begin(19200); // begin the communication between the WiFi module and the microcontroller on the board
  pinMode(RedLedPin, OUTPUT); // configure the pin connected to the Red Led to be an output
  pinMode(GreenLedPin, OUTPUT); // configure the pin connected to the Green Led to be an output
  pinMode(BlueLedPin, OUTPUT); // configure the pin connected to the Blue Led to be an output
  Serial.begin(9600);
}
```

AFF IoT Board folder > Circuits > Adjusting_RGB_led_Color

```
void loop() {
  // put your main code here, to run repeatedly:
  if (WiFiModule.available() > 0) // if the WiFi module receive data from the server
  {
    String Command = WiFiModule.readStringUntil('\n'); // read the command sent from the WiFi module to the microcontroller
    Serial.println(Command);
    if (Command.indexOf("Red=") >= 0) // if the received command contains "Red=" so we should change the Red intensity
    {
      RedStringValue = Command.substring(Command.indexOf('=') + 1); // extract the Red intensity string value
      RedValue = RedStringValue.toInt(); // transform the string to an integer value
      RedValue = map(RedValue, 0, 100, 0, 255); // this function transform the intensity percentage to a PWM ratio
      analogWrite(RedLedPin, RedValue); // generate the PWM duty cycle according to the needed intensity
    }
    else if (Command.indexOf("Green=") >= 0) // if the received command contains "Green=" so we should change the Green intensity
    {
      GreenStringValue = Command.substring(Command.indexOf('=') + 1); // extract the Green intensity string value
      GreenValue = GreenStringValue.toInt(); // transform the string to an integer value
      GreenValue = map(GreenValue, 0, 100, 0, 255); // this function transform the intensity percentage to a PWM ratio
      analogWrite(GreenLedPin, GreenValue); // generate the PWM duty cycle according to the needed intensity
    }
    else if (Command.indexOf("Blue=") >= 0) // if the received command contains "Blue=" so we should change the Blue intensity
    {
      BlueStringValue = Command.substring(Command.indexOf('=') + 1); // extract the Blue intensity string value
      BlueValue = BlueStringValue.toInt(); // transform the string to an integer value
      BlueValue = map(BlueValue, 0, 100, 0, 255); // this function transform the intensity percentage to a PWM ratio
      analogWrite(BlueLedPin, BlueValue); // generate the PWM duty cycle according to the needed intensity
    }
  }
}
//(for more info about indexOf function see https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/indexof/)
//(for more info about substring function see https://www.arduino.cc/en/Tutorial/StringSubstring)
```

2

افتح المسودة :

Adjusting_RGB_LED_Color

تعليمات برمجية يجب تدوينها..

RedValue = RedStringValue.toInt() : تحويل سلسلة صالحة إلى عدد صحيح. يجب أن تبدأ سلسلة الإدخال برقم صحيح. إذا كانت السلسلة تحتوي على أرقام غير عدد صحيح ، فستتوقف الوظيفة عن إجراء التحويل.

Command.substring(index) : بمعامل واحد فقط ، يبحث عن سلسلة فرعية معينة من مؤشرمعين إلى نهاية السلسلة (Command في هذا المثال).

ماذا ستلاحظ ؟

ستلاحظ أن الصمام يضئ, لكن هذه المرة بألوان جديدة ومرحة, إذا لم يتم ذلك ، فتأكد من قيامك بتجميع الدائرة بشكل صحيح والتحقق من الكود وتحميله على لوحتك أو الاطلاع على نصائح المساعدة أدناه.

مساعدة

الصمام لايضيء أو يظهر ألوانا خاطئة

مع وجود أربعة مسامير من الصمام قريبة جدًا من بعضها البعض ، يكون من السهل أحيانًا استبدال أحدها. تحقق من المآخذ حيث يتم إدراج كل منها.

الـضوء أحمـر

قد يكون الصمام الثنائي الأحمر داخل الصمام RGB أكثر إشراقًا قليلاً من الاثنين الأخريين. لجعل ألوانك أكثر توازناً ، استخدم مقاومة ذات أوم أعلى للون الأحمر. أو اضبطه في الكودات البرمجية كالآتي :

```
analogWrite(RedledPin, RedValue);
To analogWrite(RedledPin, RedValue/3);
```

تطبيقها في الحياة اليومية

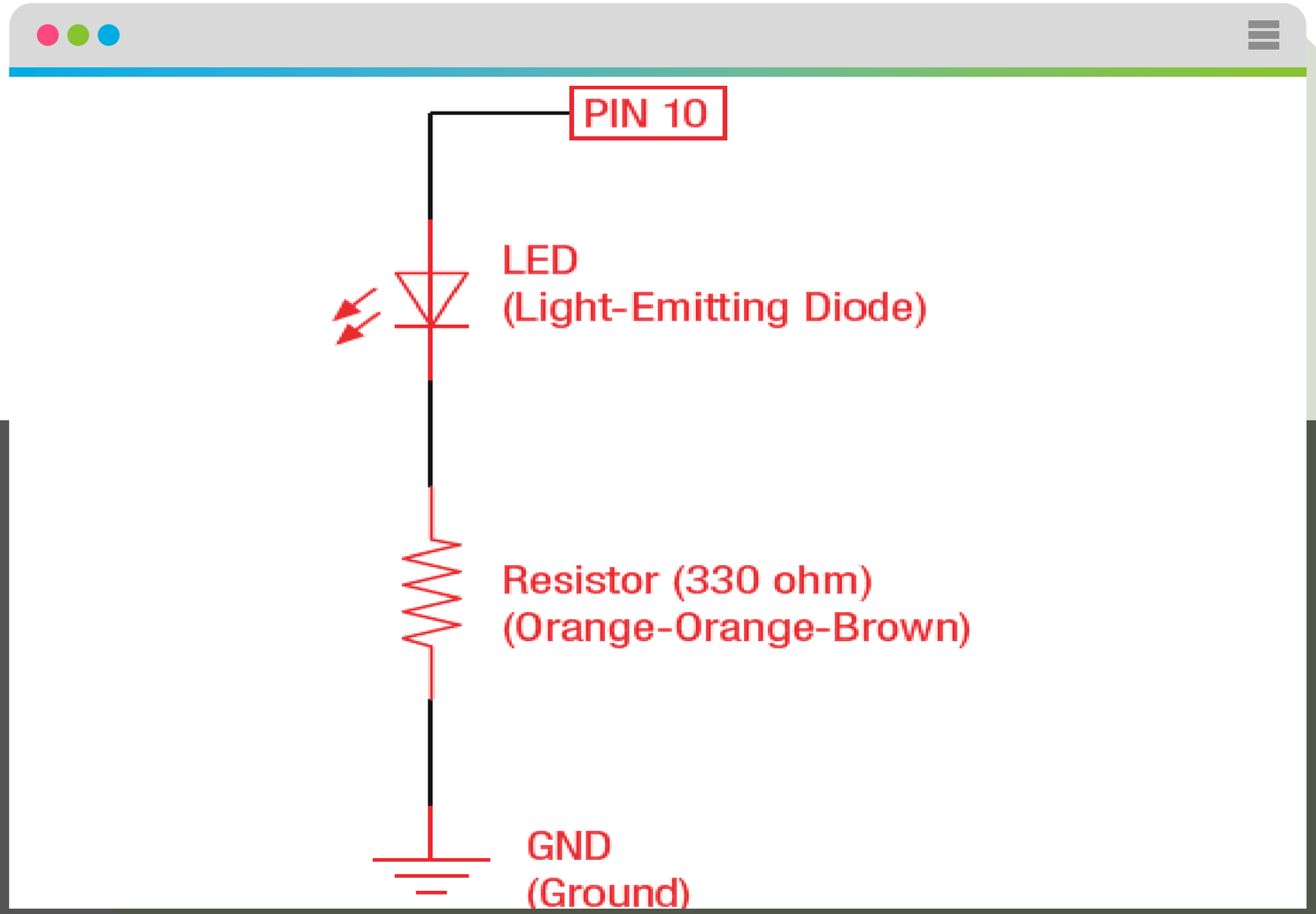
تستخدم العديد من الأجهزة الإلكترونية مثل وحدات التحكم في ألعاب الفيديو الصمام المضيء RGB لإظهار ألوان مختلفة داخل نفس مصدر الضوء.

10 جدولة المراقبة

" من أسلاك التوصيل

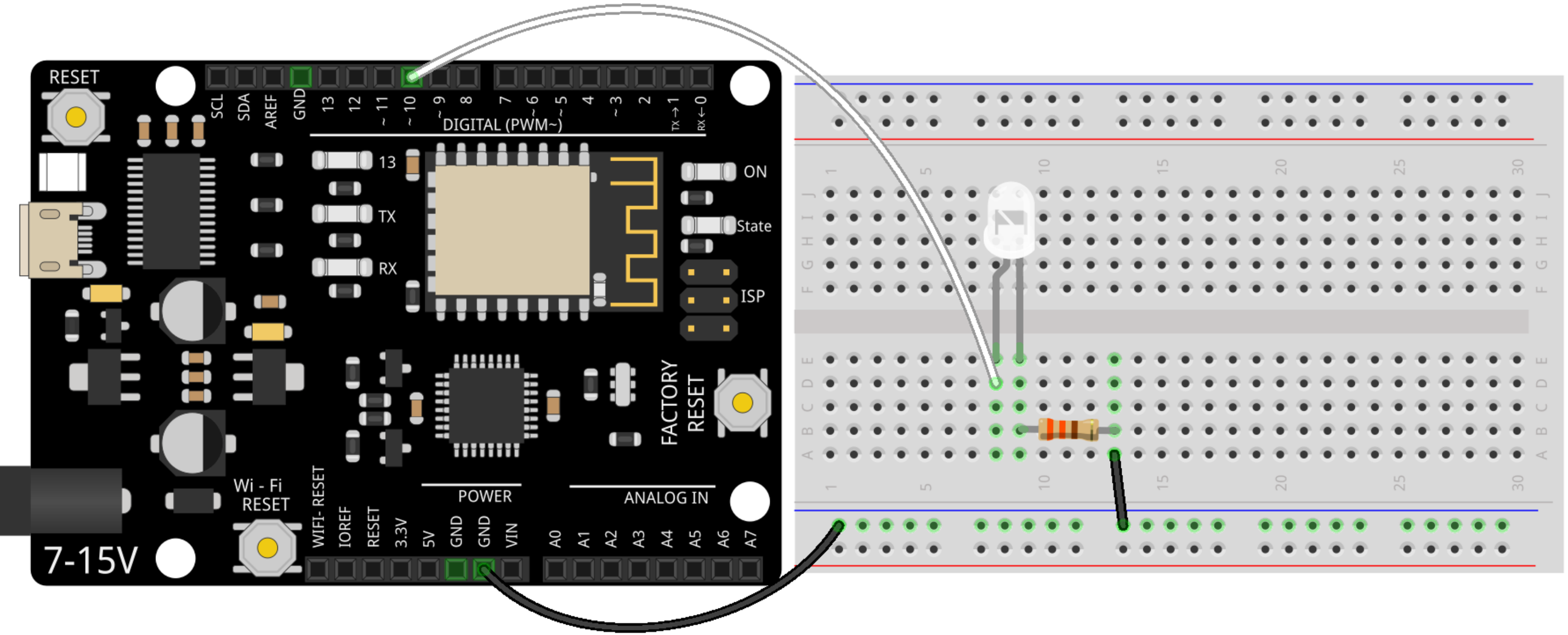
مقاومة ذات جهد 330Ω

صمام مضيء White

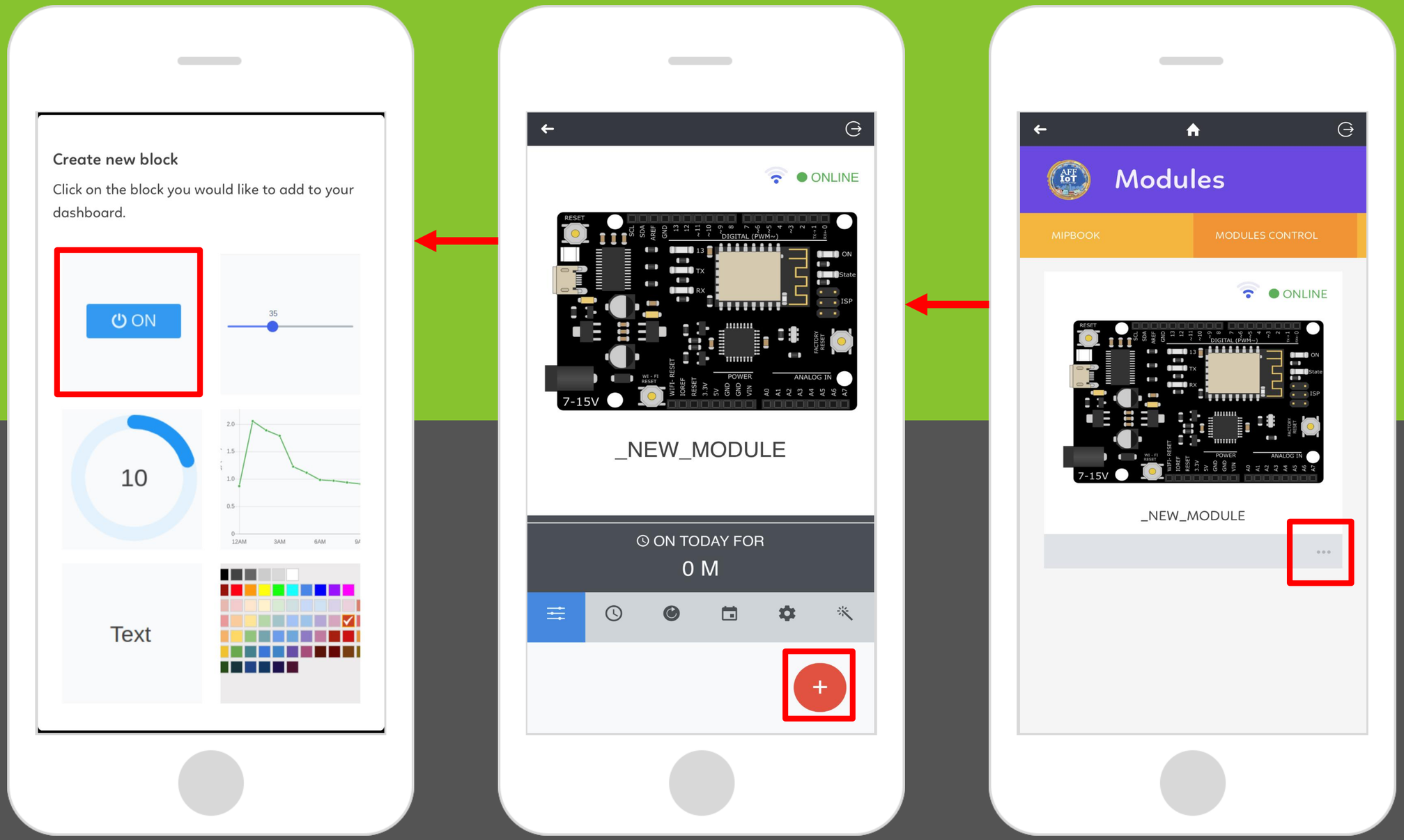


تشغيل وإطفاء الصمام

الصمام المضيء هو مصدر ضوئي صغير ولكنه قوي يستخدم في العديد من التطبيقات المختلفة. أولاً سنعمل على جعله يومض, رغم أنها وظيفة بسيطة، إلا أنها الأساس لمزيد من الوظائف المعقدة.


RESET
SCL
SDA
AREF
GND
13
12
~ 11
~ 10
~ 9
8
7
~ 6
~ 5
4
~ 3
2
TX → 1
RX ← 0
DIGITAL (PWM~)
13
TX
RX
ON
State
ISP
FACTORY
RESET
Wi - Fi
RESET
WIFI- RESET
IOREF
RESET
3.3V
5V
GND
GND
VIN
POWER
A0
A1
A2
A3
A4
A5
A6
A7
ANALOG IN
7-15V

fritzing


Create new block
Click on the block you would like to add to your dashboard.
ON
35
10
Text
ONLINE
_NEW_MODULE
ON TODAY FOR
0 M
Modules
MIPBOOK
MODULES CONTROL
ONLINE
_NEW_MODULE

LABEL NAME : هو الاسم الذي يظهر في التطبيق.
PARAMETER NAME : هو الاسم المستخدم في الكودات البرمجية.

أكتب في الفراغ المعلومات التالية

LABEL NAME
White Led
PARAMETER NAME
white_led
IMAGE

أختر أيقونة الصمام المضيئ

CANCEL
ADD

قم بالنزول للأسفل ، ثم انقر ADD

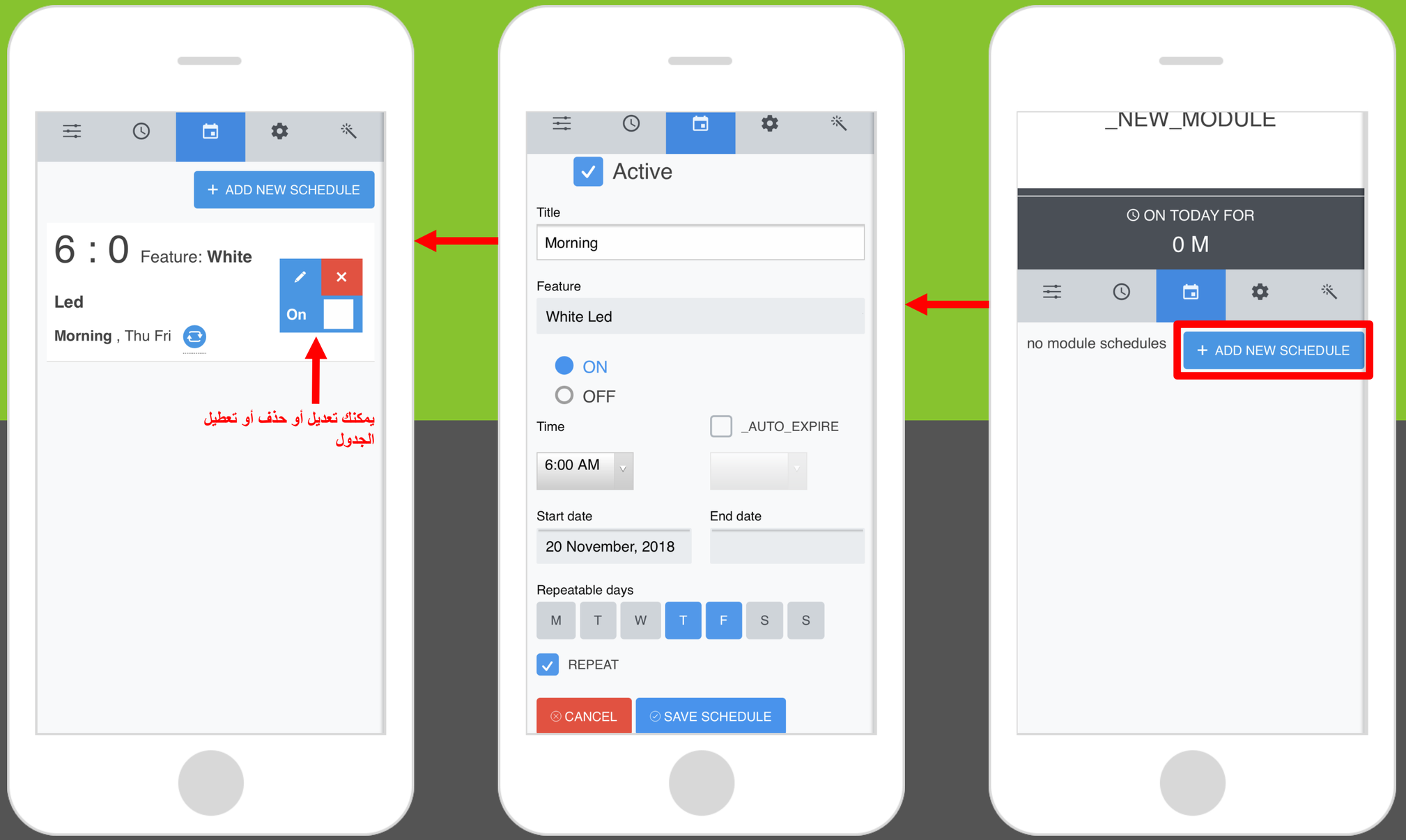
_NEW_MODULE

ON TODAY FOR
0 M

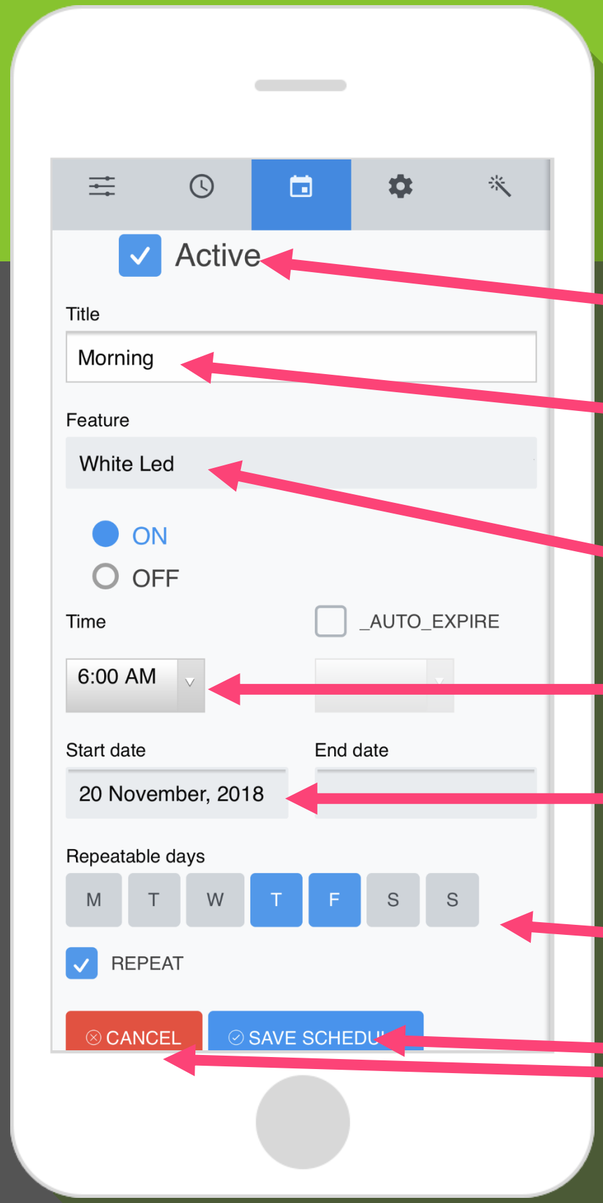
White Led
OFF

الخطوة القادمة : وضع جدول زمني

تحكم في الصمام بالضغط على ON و OFF


+ ADD NEW SCHEDULE
6 : 0 Feature: White
Led
Morning , Thu Fri
On
يمكنك تعديل أو حذف أو تعطيل
الجدول
Active
Title
Morning
Feature
White Led
ON
OFF
Time
_AUTO_EXPIRE
6:00 AM
Start date
End date
20 November, 2018
Repeatable days
M T W T F S S
REPEAT
CANCEL
SAVE SCHEDULE
_NEW_MODULE
ON TODAY FOR
0 M
no module schedules
+ ADD NEW SCHEDULE

تفسير

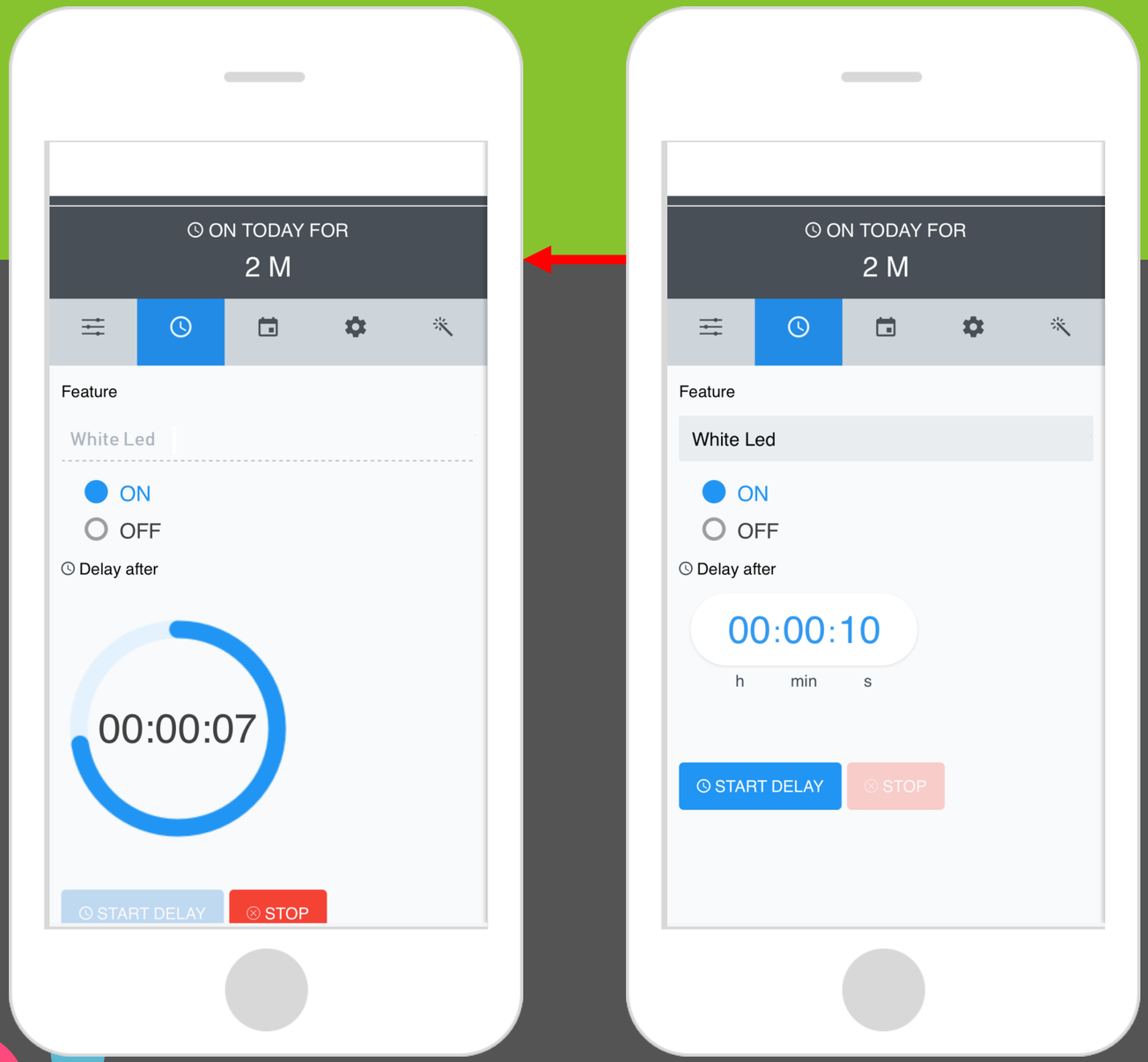


- فعال : يحدد حالة الجدول (فعال أم لا).
- العنوان: هو عنوان الجدول الذي تم إنشاؤه.
- الميزة : هي عنصر التحكم الذي ستتم جدولته (صمام مضيءاختر ON أو OFF).
- الوقت: هو الوقت الذي سيتم فيه تنفيذ الإجراء.
- تاريخ البدء: هو التاريخ الذي سيتم فيه تفعيل الجدول.
- الأيام المتكررة: إذا تم تحديد خانة الاختيار تكرار ، فأنت بذلك يمكن أن تختار ما هي الأيام التي يتم فيها التفعيل.
- حفظ الجدول الزمني أو إلغائه.

المؤقت

يمكن خيار المؤقت من اختيار العنصرالمتحكم فيه وتحديد وقت تأخير. بعد هذا التأخير سيتم تنفيذ الأمر.

يمكنك أيضًا إيقاف المؤقت عن طريق النقر فوق الزر "إيقاف" ، وسيتم تجاهل الإجراء المتأخر.



AFF IoT Board folder > Circuits > Examples > Scheduling_Controls

Scheduling_Controls

```
/*Start of mandatory lines of codes in each sketch*/
#define RX A0 // define the Receive pin (RX) to communicate with the WiFi module
#define TX A1 // define the Transmit pin (TX) to communicate with the WiFi module
#include <NeoSWSerial.h> // including the library to use the Software Serial rather than the Hardware Serial (Serial)
NeoSWSerial WiFiModule(RX, TX); //initialize the variable to use in communication with the WiFi module
/*End of mandatory lines of code*/

#define WhiteLedPin   10

void setup() {
  // put your setup code here, to run once:
  WiFiModule.begin(19200); // begin the communication between the WiFi module and the microcontroller on the board
  pinMode(WhiteLedPin, OUTPUT); // configure the pin connected to the White Led to be an output
}

void loop() {
  // put your main code here, to run repeatedly:
  if (WiFiModule.available() > 0) // if the WiFi module receive data from the server
  {
    String Command = WiFiModule.readStringUntil('\n'); // read the command sent from the WiFi module to the microcontroller

    if (Command.indexOf("white_led=1") >= 0) // if the received command contains "white_led=1" turn on the LED
    {
      digitalWrite(WhiteLedPin, HIGH); // turn on the LED
    }
    if (Command.indexOf("white_led=0") >= 0) // if the received command contains "white_led=0" turn it off
    {
      digitalWrite(WhiteLedPin, LOW); // turn off the LED
    }
  }
}
//(for more info about indexOf function see https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/indexof/)
```

افتح المسودة : Scheduling_Controls

سيعمل هذا المثال كما في المثال 5 مراقبة الصمام المضيء مع تغيير في اسم الصمام.

ماذا ستلاحظ ؟

ستلاحظ أن الصمام يضيء بعد عشر ثواني. إذا اخترت زر الإيقاف ستلاحظ انطفاء الصمام بعد مرور الوقت المحدد.

مساعدة

الصمام لا يضيء؟
تعمل المصابيح في اتجاه واحد فقط . لذا حاول إخراجها ولفها بزاوية 180 درجة (لا تقلق ، تثبيتها بالطريقة المعاكسة لا يلحق الضرر بها).

تطبيقها في الحياة اليومية

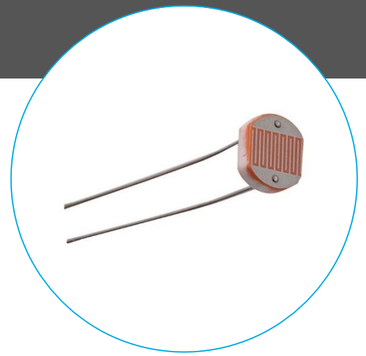
معظم أجهزة أنترنيت الأشياء تضم مؤقتا

11 محاكاة نظام الإضاءة الآلي

مقاومة ذات الجهد 330Ω

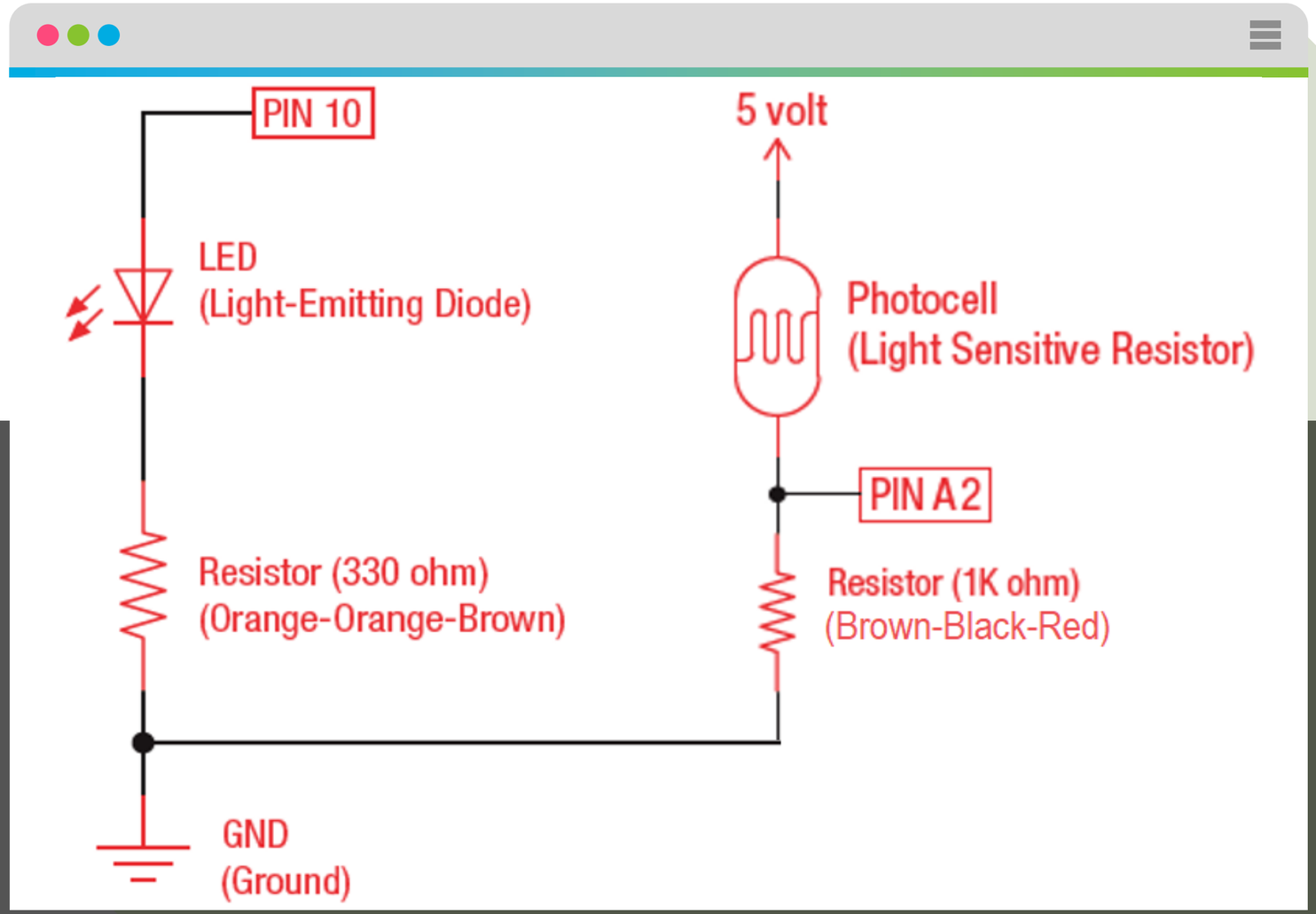
مقاومة ذات الجهد 10KΩ

صمام مضيء أصفر



خلية ضوئية

5 من أسلاك التوصيل

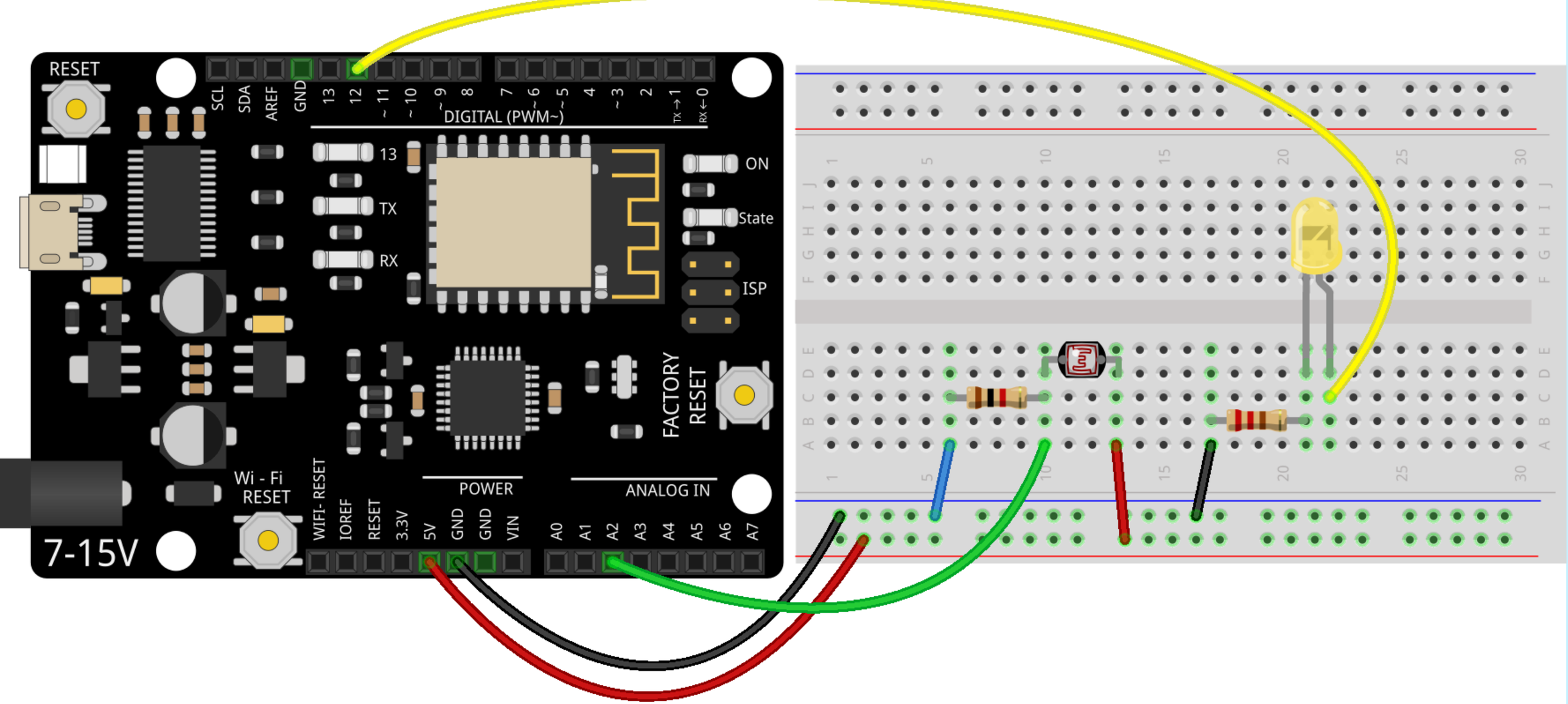


محاكاة النظام الحقيقي

المحاكاة هي تقليد لعملية حقيقية.

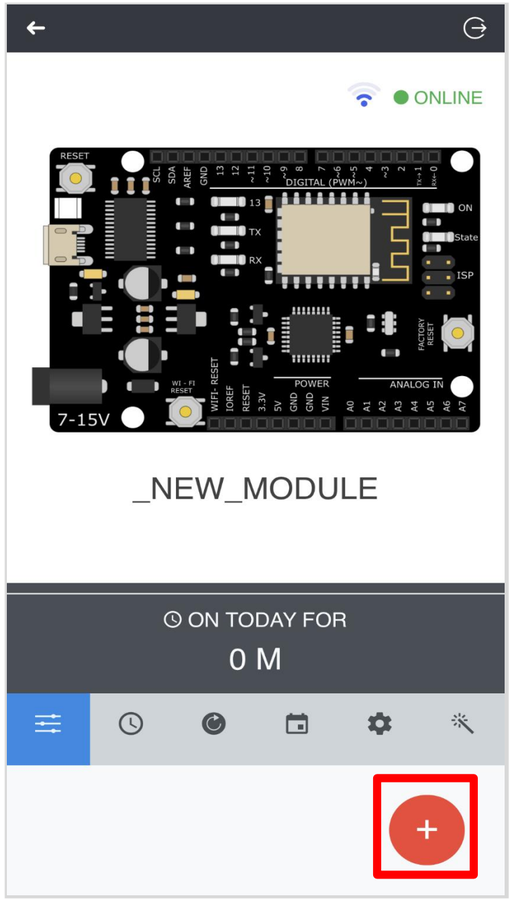
لذلك في العالم الحقيقي ، بدلا من الصمام ، يتم استخدام المصباح الكهربائي.

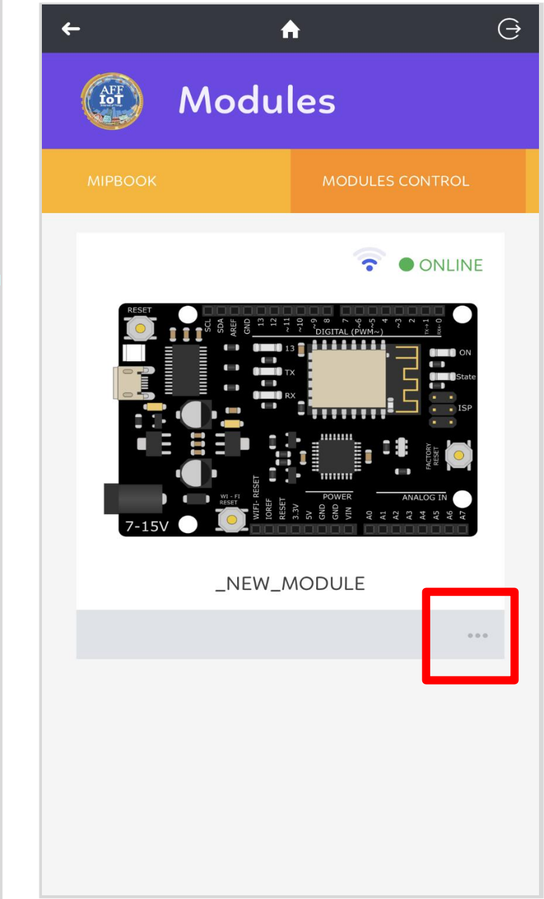
يتم استخدام الصمام المضيء بدلاً من المصباح لأسباب تتعلق بالسلامة أثناء تطبيق العديد من التجارب.


RESET
SCL
SDA
AREF
GND
13
12
~ 11
~ 10
~ 9
8
7
~ 6
~ 5
4
~ 3
2
TX → 1
RX ← 0
DIGITAL (PWM~)
13
TX
RX
ON
State
ISP
FACTORY RESET
Wi - Fi RESET
WIFI- RESET
IOREF
RESET
3.3V
5V
GND
GND
VIN
POWER
A0
A1
A2
A3
A4
A5
A6
A7
ANALOG IN
7-15V

fritzing

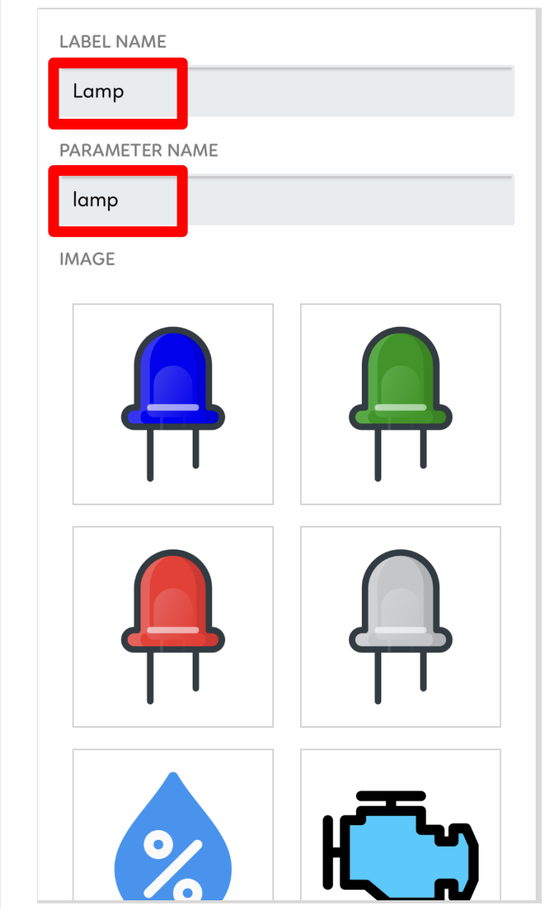
Create new block
Click on the block you would like to add to your dashboard.
ON
35
10
Text

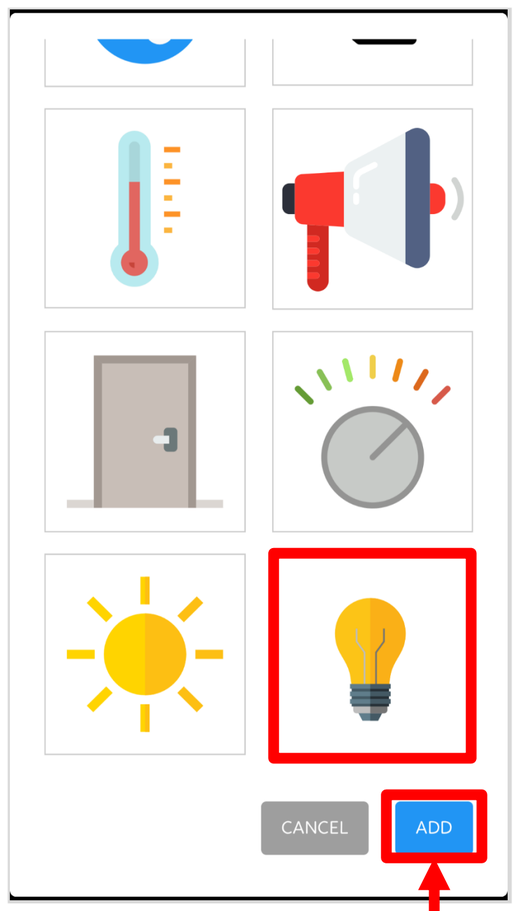

ONLINE
_NEW_MODULE
ON TODAY FOR
0 M


Modules
MIPBOOK
MODULES CONTROL
ONLINE
_NEW_MODULE

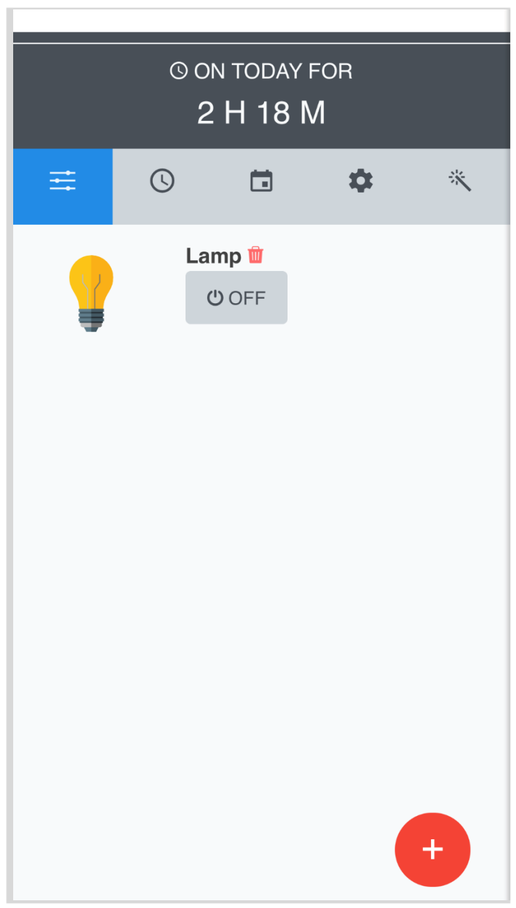
LABEL NAME : هو الاسم الذي يظهر في التطبيق.
PARAMETER NAME : هو الاسم المستخدم في الكودات البرمجية.

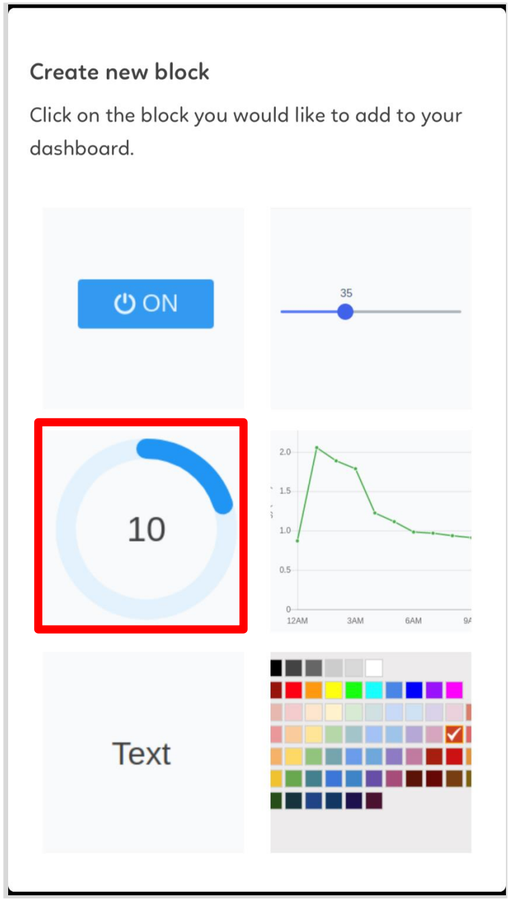
أكتب في الفراغ المعلومات التالية

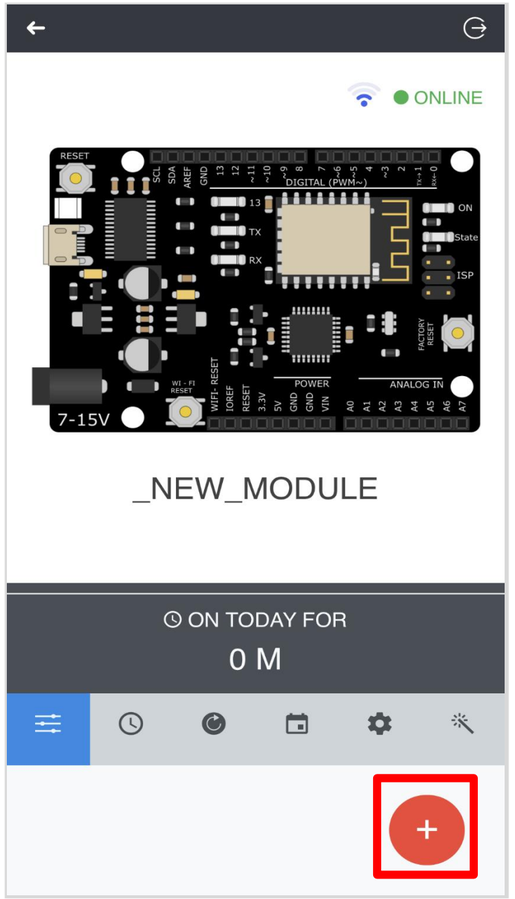


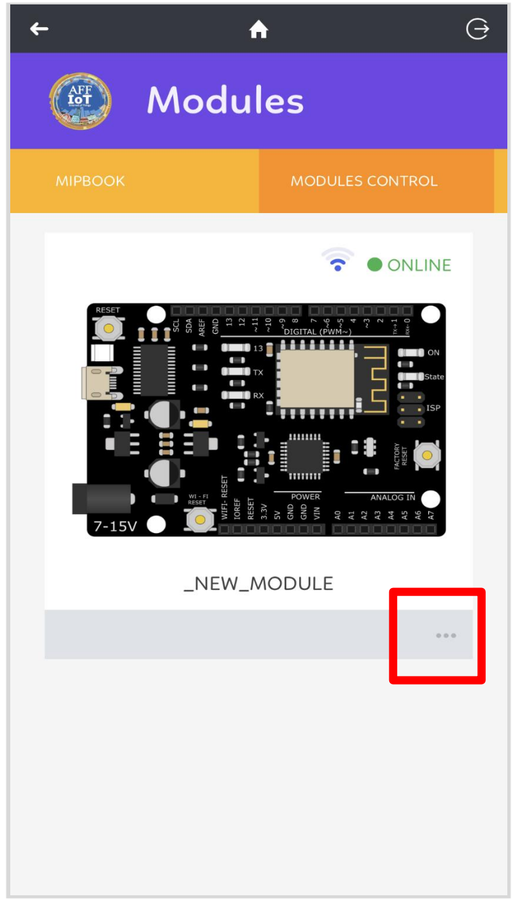


قم بالنزول للأسفل ،
ثم انقر ADD



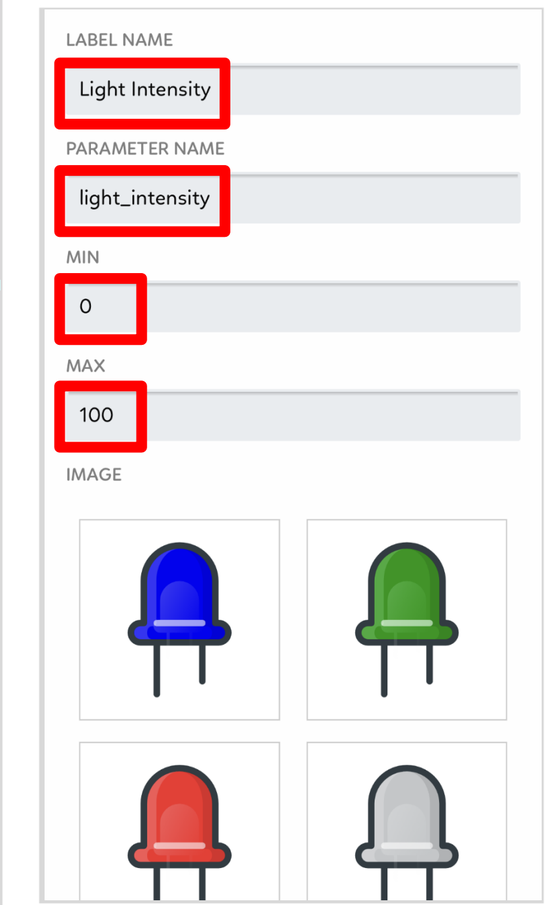

Create new block
Click on the block you would like to add to your dashboard.
ON
35
10
Text

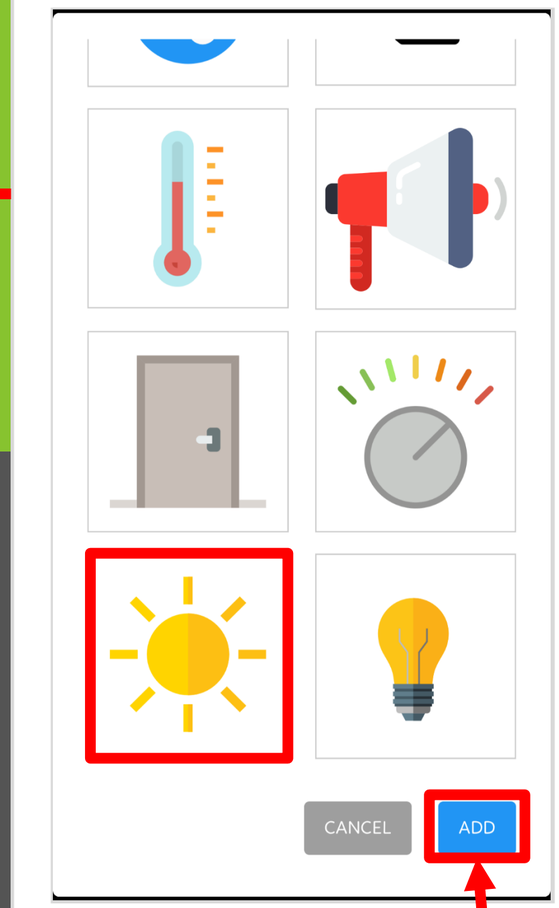

ONLINE
7-15V
_NEW_MODULE
ON TODAY FOR
0 M


Modules
MIPBOOK
MODULES CONTROL
ONLINE
7-15V
_NEW_MODULE

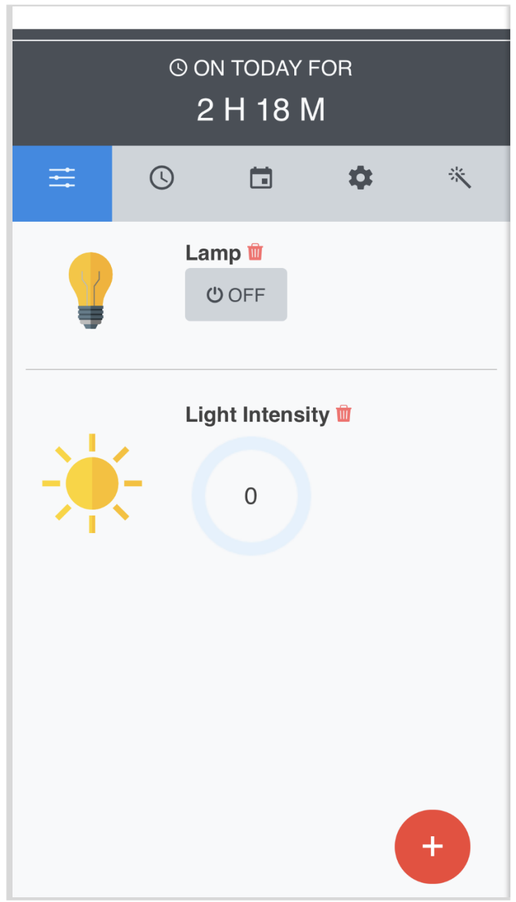
LABEL NAME : هو الاسم الذي يظهر في التطبيق.
PARAMETER NAME : هو الاسم المستخدم في الكودات البرمجية.

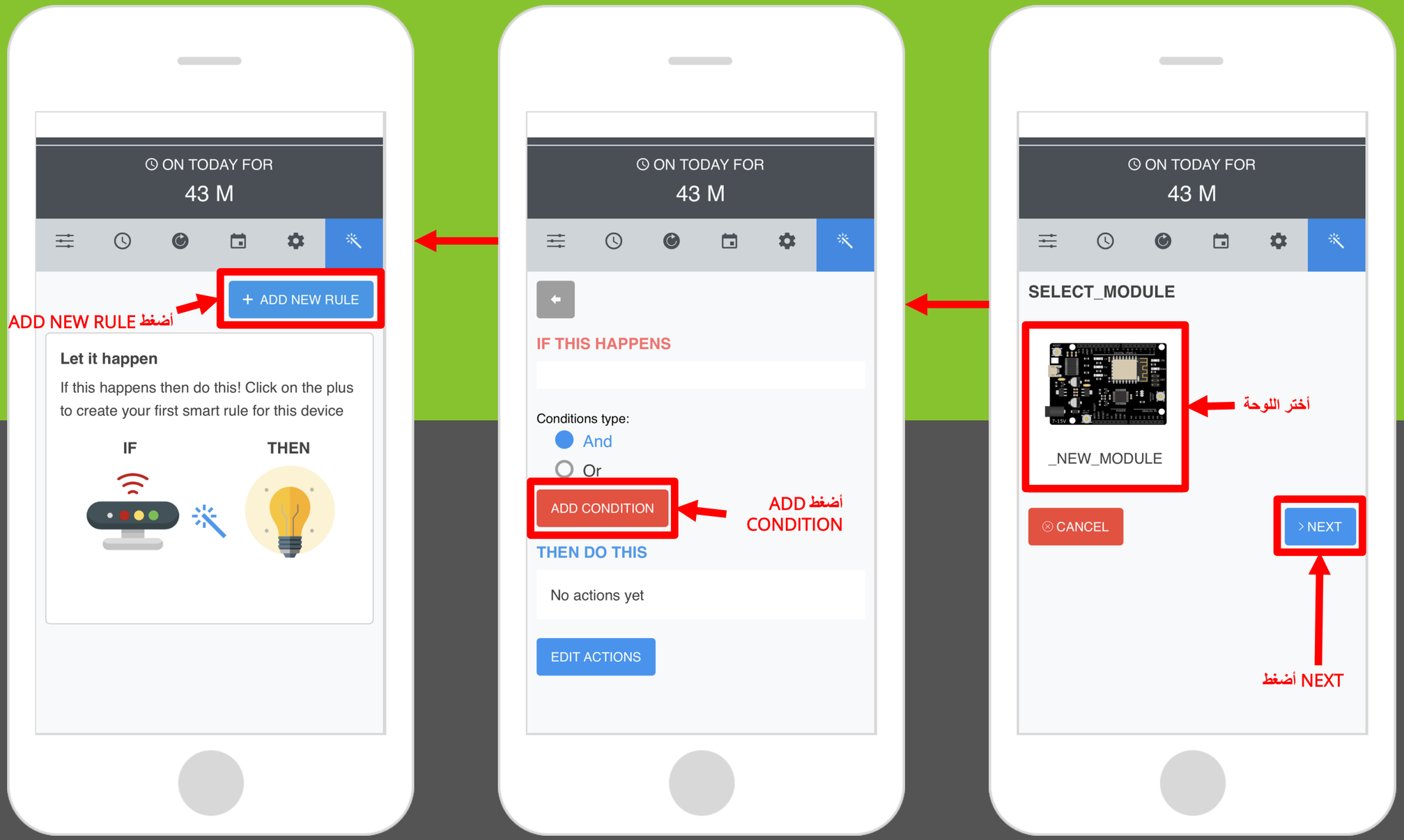
أكتب في الفراغ المعلومات التالية

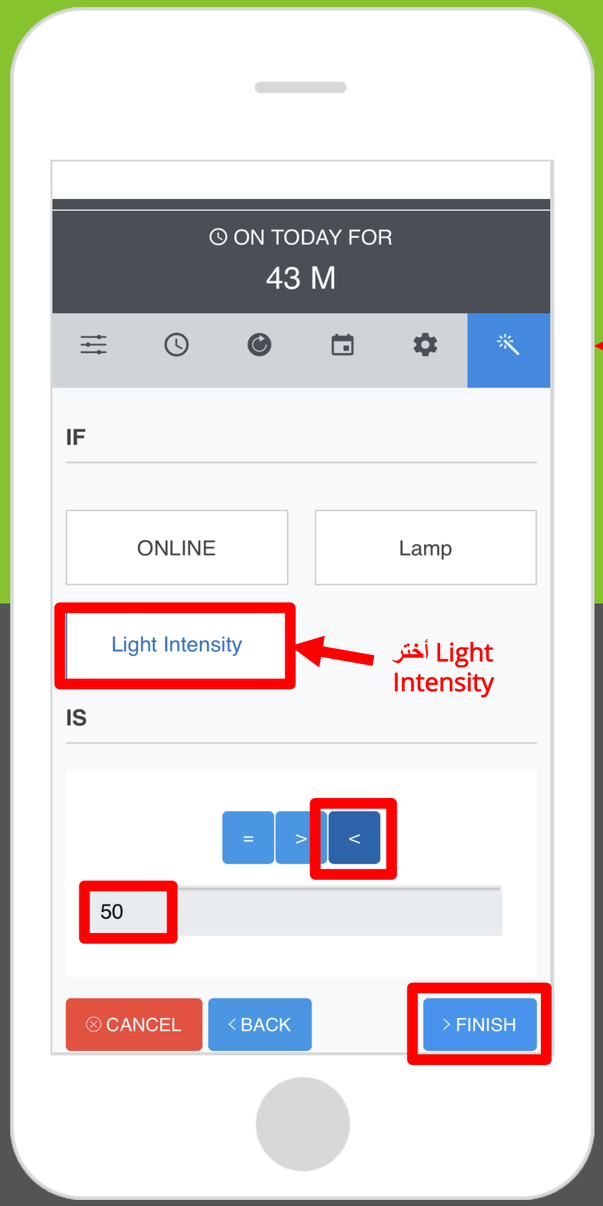


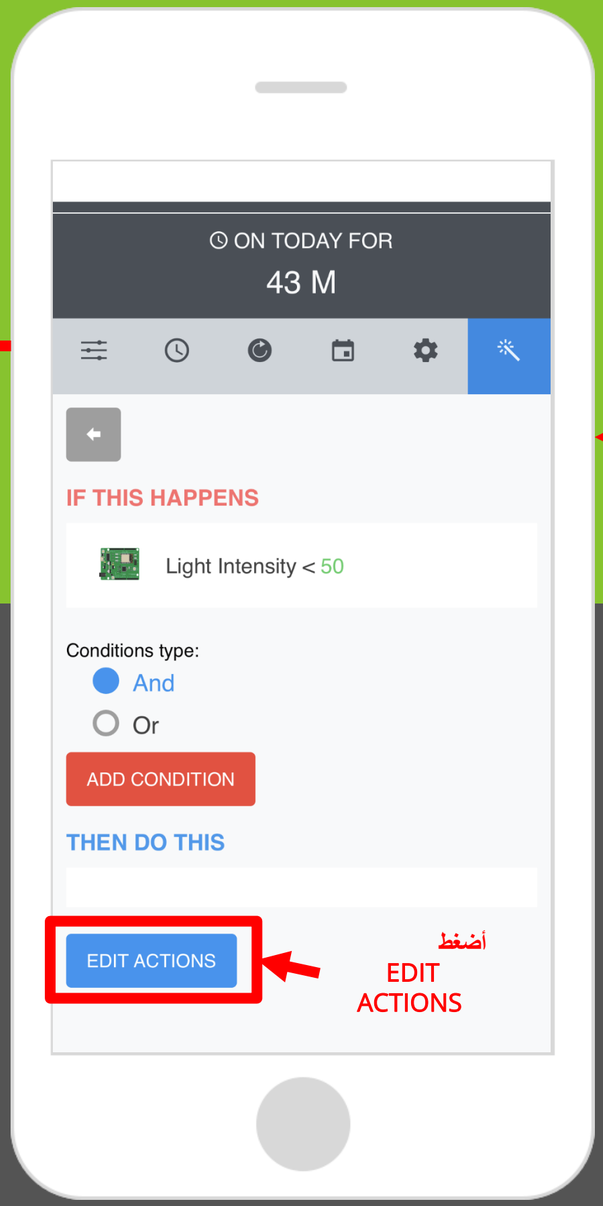


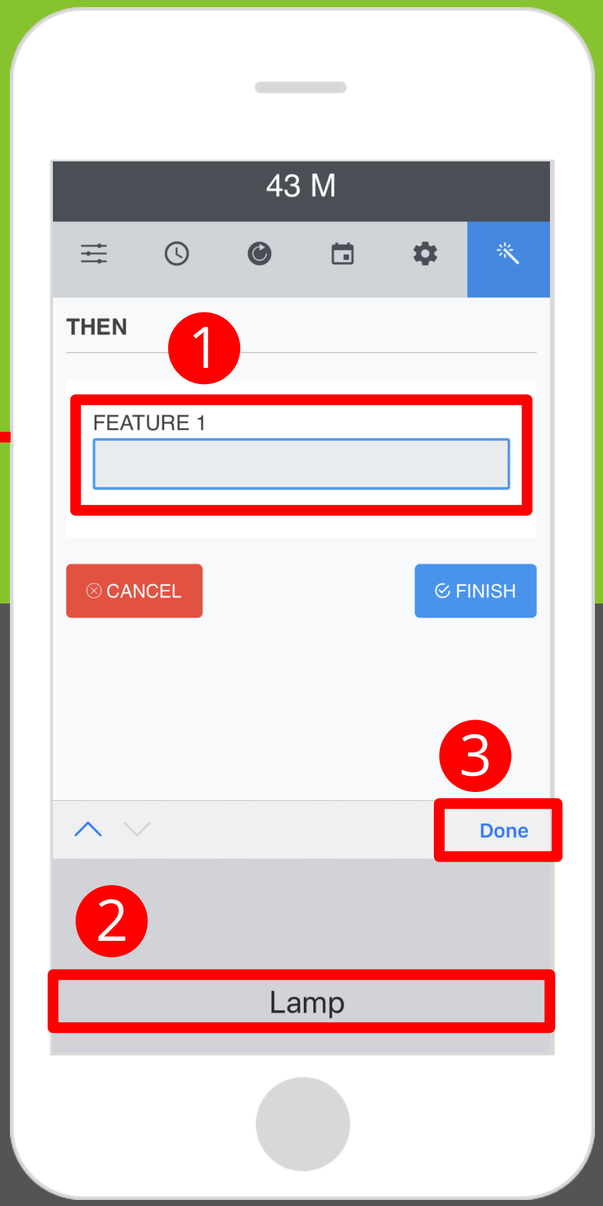
قم بالنزول للأسفل ،
ثم انقر ADD




ON TODAY FOR
43 M
+ ADD NEW RULE
ADD NEW RULE أضغط
Let it happen
If this happens then do this! Click on the plus to create your first smart rule for this device
IF
THEN
ON TODAY FOR
43 M
IF THIS HAPPENS
Conditions type:
And
Or
ADD CONDITION
ADD أضغط CONDITION
THEN DO THIS
No actions yet
EDIT ACTIONS
ON TODAY FOR
43 M
SELECT_MODULE
_NEW_MODULE
أختر اللوحة
CANCEL
NEXT
أضغط NEXT


ON TODAY FOR
43 M
IF
ONLINE
Lamp
Light Intensity
أختر Light Intensity
IS
=
>
<
50
CANCEL
BACK
FINISH


ON TODAY FOR
43 M
IF THIS HAPPENS
Light Intensity < 50
Conditions type:
And
Or
ADD CONDITION
THEN DO THIS
EDIT ACTIONS
أضغط EDIT ACTIONS

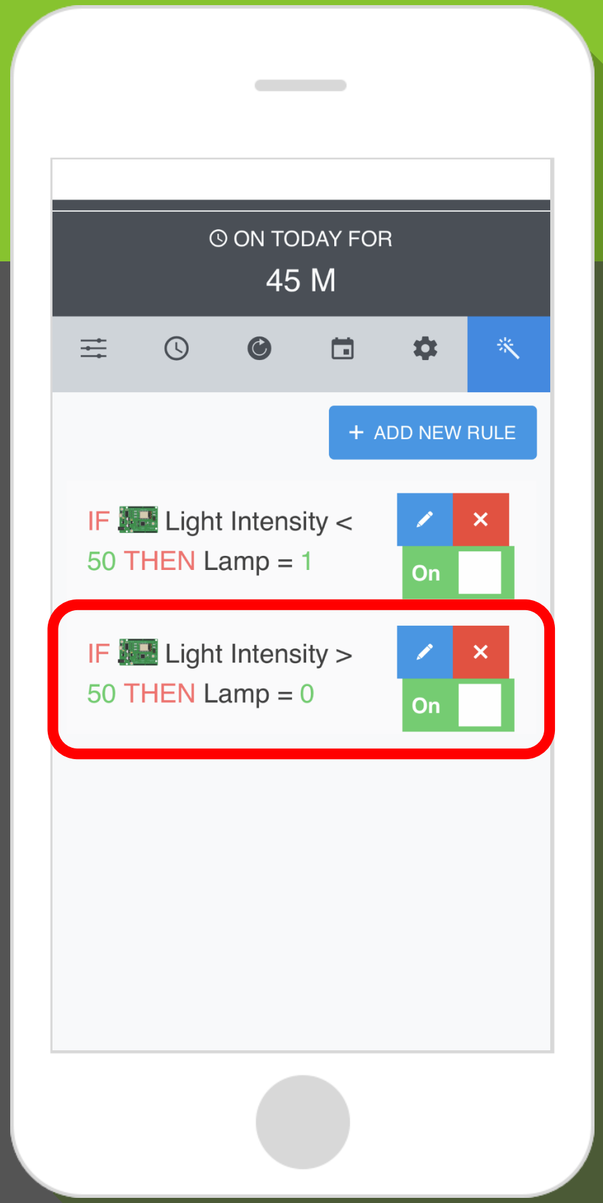

43 M
THEN
1
FEATURE 1
CANCEL
FINISH
3
Done
2
Lamp

ON TODAY FOR
45 M
+ ADD NEW RULE
IF Light Intensity < 50 THEN Lamp = 1
On
يمكنك تحرير القاعدة أو حذفها
أو إلغاء تنشيطها
الآن عندما تكون شدة
الإضاءة أقل من 50٪ ، يتم
تشغيل الصمام الأصفر
(مصباح في الحياة الحقيقية)
ON TODAY FOR
45 M
انقر فوق BACK للعودة
IF THIS HAPPENS
Light Intensity < 50
Conditions type:
And
Or
ADD CONDITION
THEN DO THIS
Lamp 1
EDIT ACTIONS
ON TODAY FOR
43 M
THEN
قم بتشغيل المصباح
FEATURE 1
Lamp
ON
OFF
FEATURE 2
CANCEL
FINISH
أضغط FINISH

وأخيرا

كرر العملية ولكن بدلاً من ذلك حدد شدة الإضاءة لتكون أقل بنسبة 50٪ والهدف هو إيقاف التشغيل.

يمكنك ضبط نسبة شدة الإضاءة لتتناسب مع إضاءة غرفتك.



AFF IoT Board folder > Circuits > Examples > Automated_Lighting_System

Automated_Lighting_System

```
/*Start of mandatory lines of codes in each sketch*/
#define RX A0 // define the Receive pin (RX) to communicate with the WiFi module
#define TX A1 // define the Transmit pin (TX) to communicate with the WiFi module
#include <NeoSWSerial.h> // including the library to use the Software Serial rather than the Hardware Serial (Serial)
NeoSWSerial WiFiModule(RX, TX); //initialize the variable to use in communication with the WiFi module
/*End of mandatory lines of code*/

#define LightSensorPin   A2
#define YellowLedPin     10

void setup() {
  // put your setup code here, to run once:
  WiFiModule.begin(19200); // begin the communication between the WiFi module and the microcontroller on the board
}

void loop() {
  // put your main code here, to run repeatedly:
  if (WiFiModule.available() > 0) // if the WiFi module receive data from the server
  {
    String Command = WiFiModule.readStringUntil('\n'); // read the command sent from the WiFi module to the microcontroller

    if (Command.indexOf("lamp=1") >= 0) // if the received command contains "lamp=1" turn on the LED
    {
      digitalWrite(YellowLedPin, HIGH); // turn on the LED
    }
    if (Command.indexOf("lamp=0") >= 0) // if the received command contains "lamp=0" turn it off
    {
      digitalWrite(YellowLedPin, LOW); // turn off the LED
    }
  }

  int lightIntensity; // declare an integer to read the voltage

  lightIntensity = analogRead(LightSensorPin); // read the actual converted value (between 0 and 1023)
  lightIntensity = map(lightIntensity, 0, 1023, 0, 100); // transform the scale from 0 and 1023 to 0% and 100%

  WiFiModule.println("light_intensity=" + String(lightIntensity)); // send the light intensity value to the server

  delay(3000); // wait for 3 second (3000ms = 3s)
}
```

افتح المسودة :

Automated_Lighting_System

ماذا ستلاحظ ؟

ستلاحظ أن الصمام الأصفر يضيء عندما تنخفض إضاءة الغرفة, و ينطفئ عندما تزداد إضاءتها.

مساعدة

بإمكانك العودة إلى مثالي التحكم في الصمام و قراءة شدة الضوء.

تطبيقها في الحياة اليومية

في الآونة الأخيرة ، تأتي معظم المصابيح مع مستشعر إضاءة مدمج. يتم تشغيلها عندما يكون المصباح أقل من العتبة المحددة مسبقًا ، ويتم إيقاف التشغيل بطريقة أخرى.

12 إنذار ارتفاع درجة الحرارة

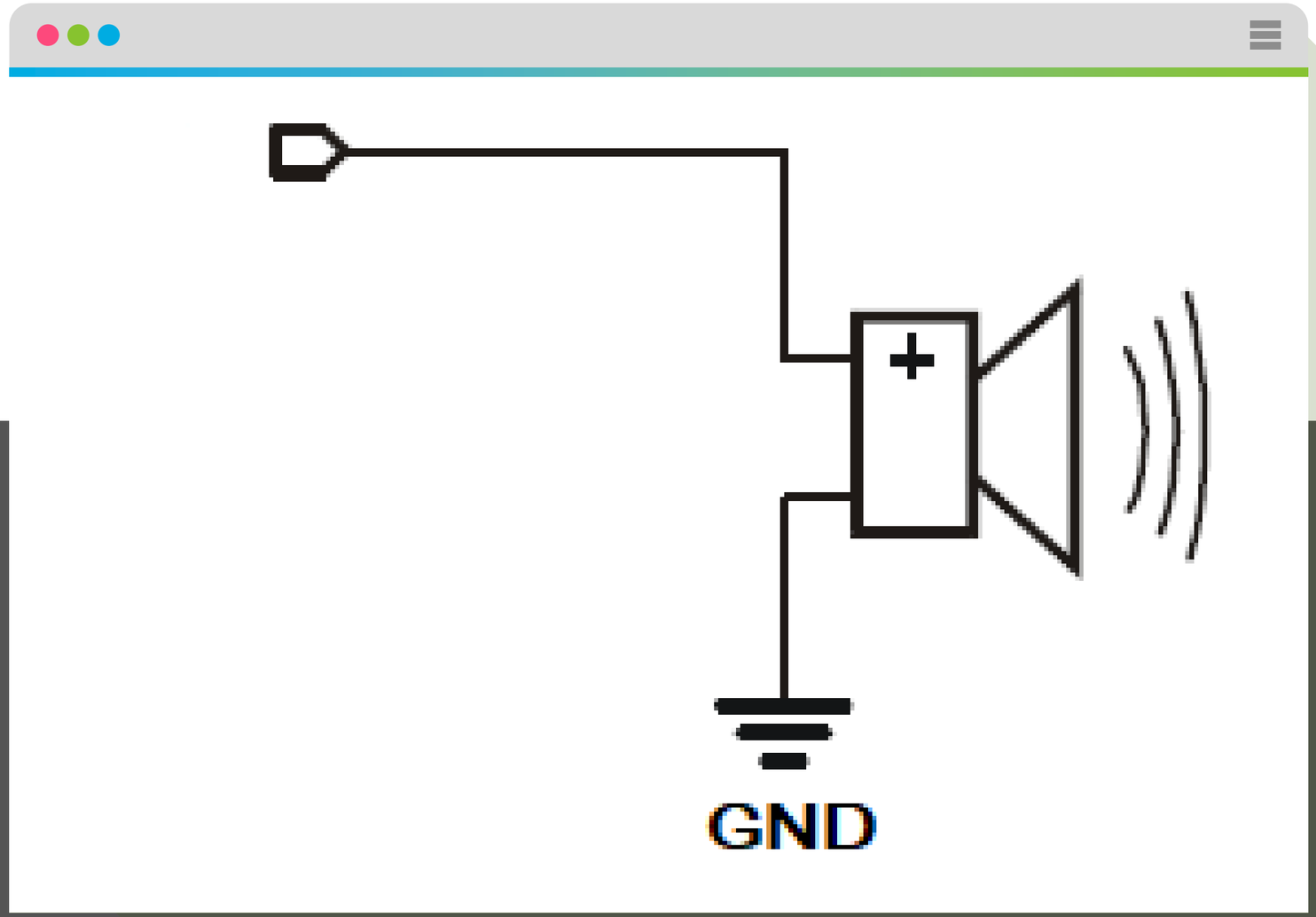
7 من أسلاك التوصيل

مقاومة ذات جهد 1KΩ

مقاومة حرارية

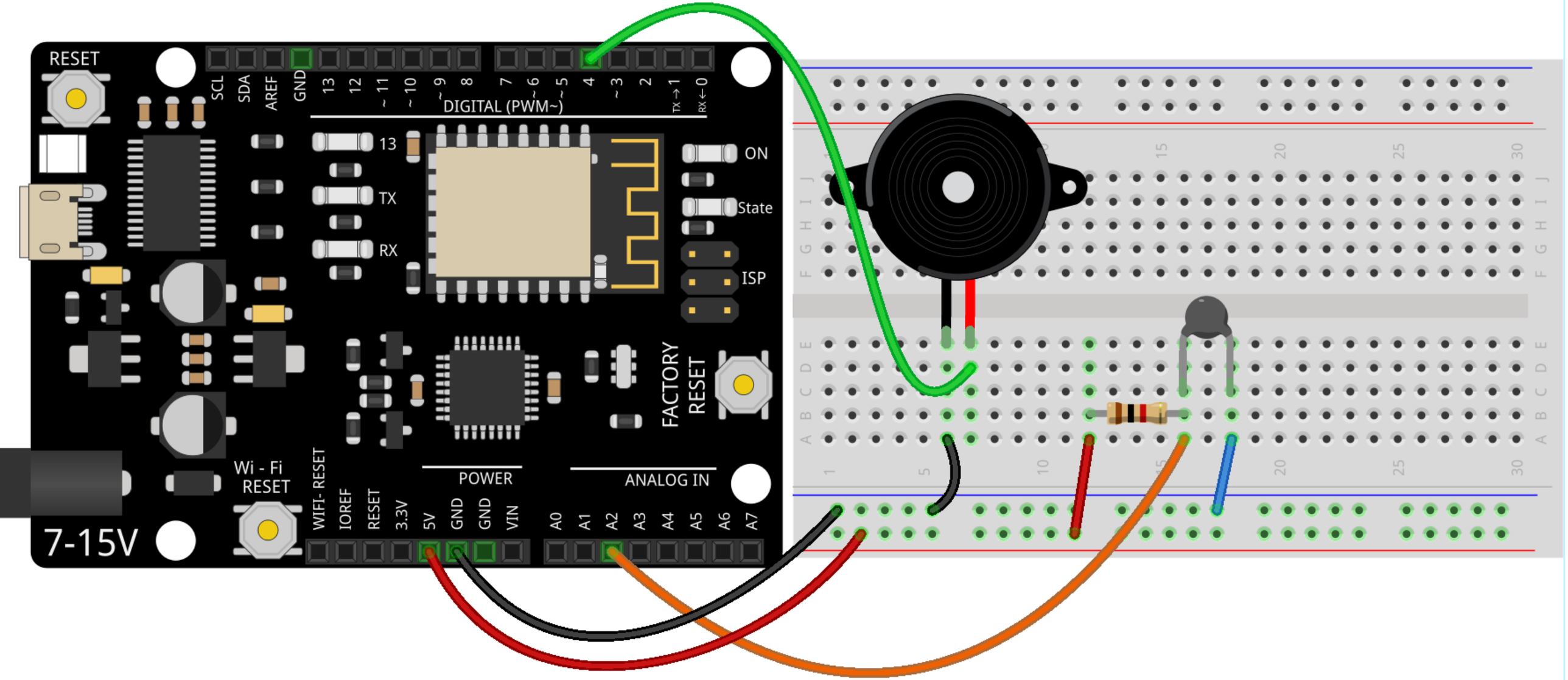
جرس

الجرس



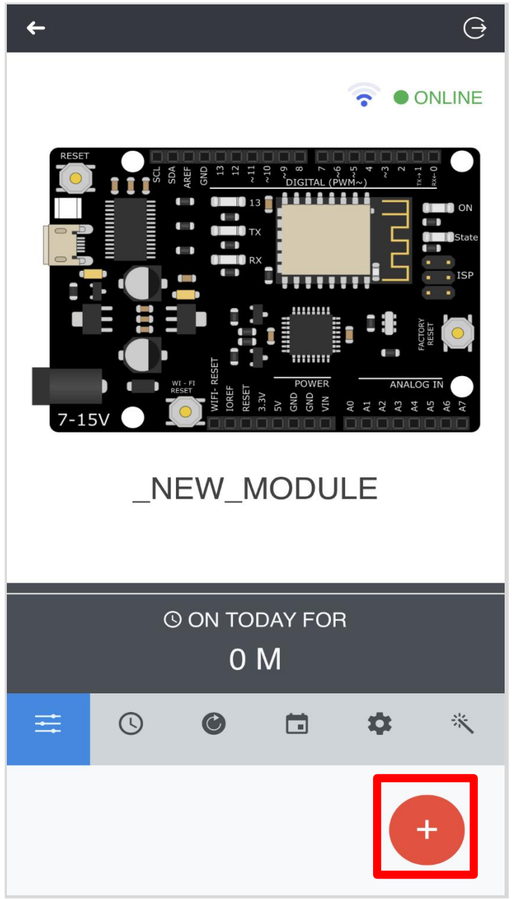
سنستخدم جرسا يصدر صوتًا عند تطبيق الجهد عليه!

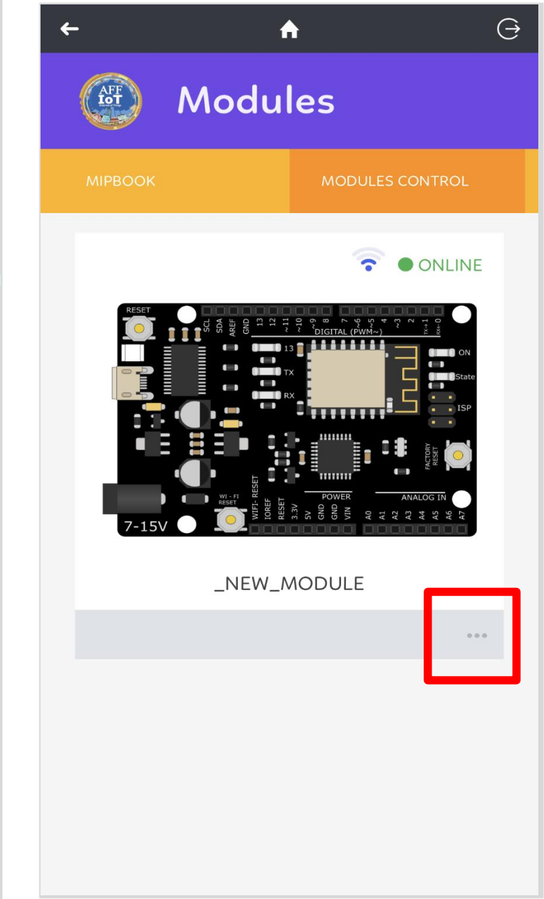
هذا المثال هو مقدمة لأنظمة الإنذار.


RESET
SCL
SDA
AREF
GND
13
12
~ 11
~ 10
~ 9
8
7
~ 6
~ 5
4
~ 3
2
TX → 1
RX ← 0
DIGITAL (PWM~)
13
TX
RX
ON
State
ISP
FACTORY RESET
Wi - Fi RESET
WIFI- RESET
IOREF
RESET
3.3V
5V
GND
GND
VIN
POWER
ANALOG IN
A0
A1
A2
A3
A4
A5
A6
A7
7-15V

fritzing

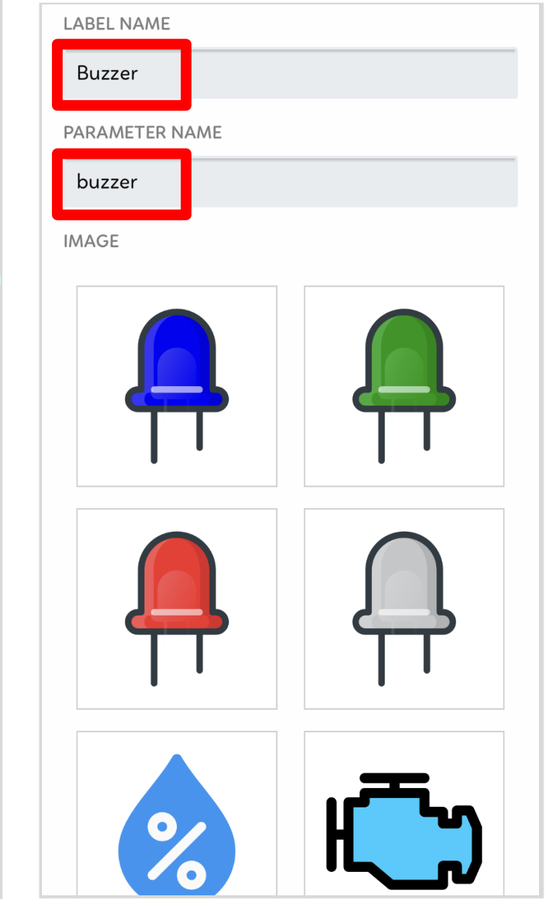
Create new block
Click on the block you would like to add to your dashboard.
ON
35
10
Text

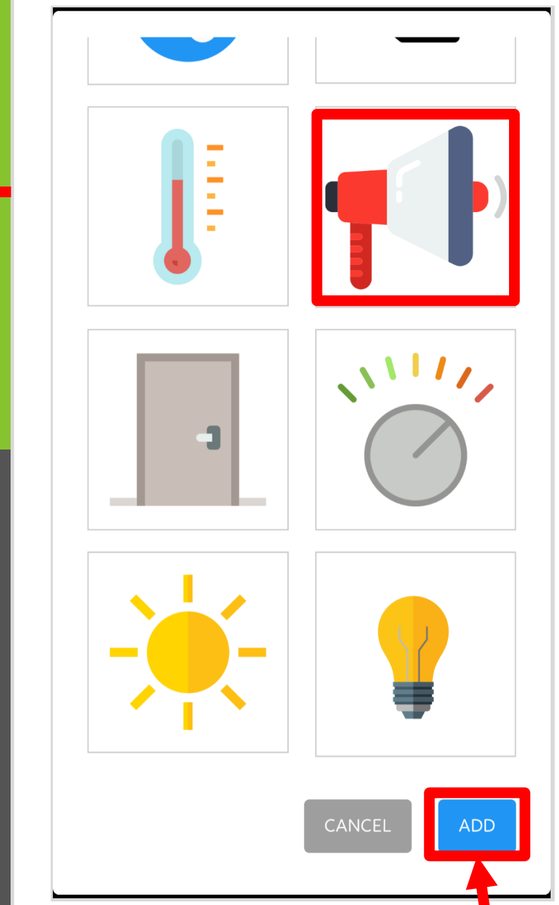

ONLINE
_NEW_MODULE
ON TODAY FOR
0 M


Modules
MIPBOOK
MODULES CONTROL
ONLINE
_NEW_MODULE

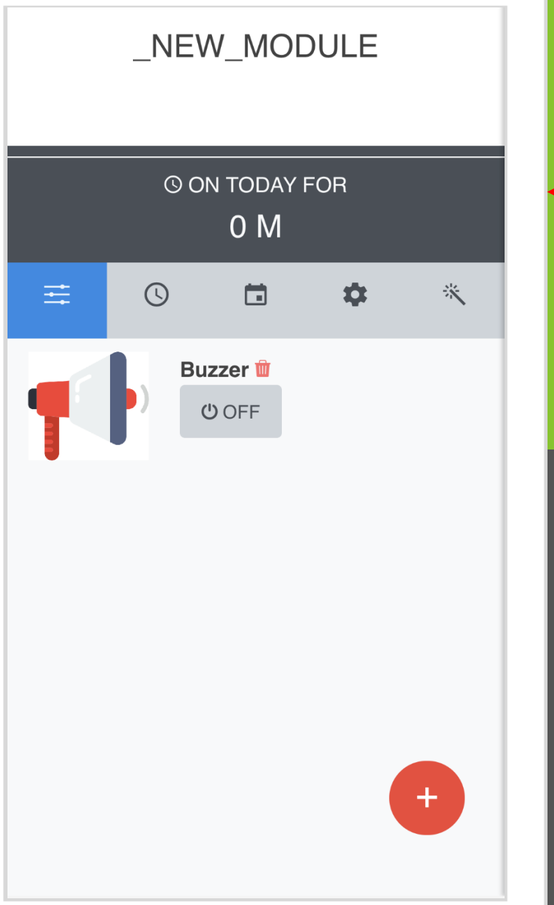
LABEL NAME : هو الاسم الذي يظهر في التطبيق.
PARAMETER NAME : هو الاسم المستخدم في الكودات البرمجية.

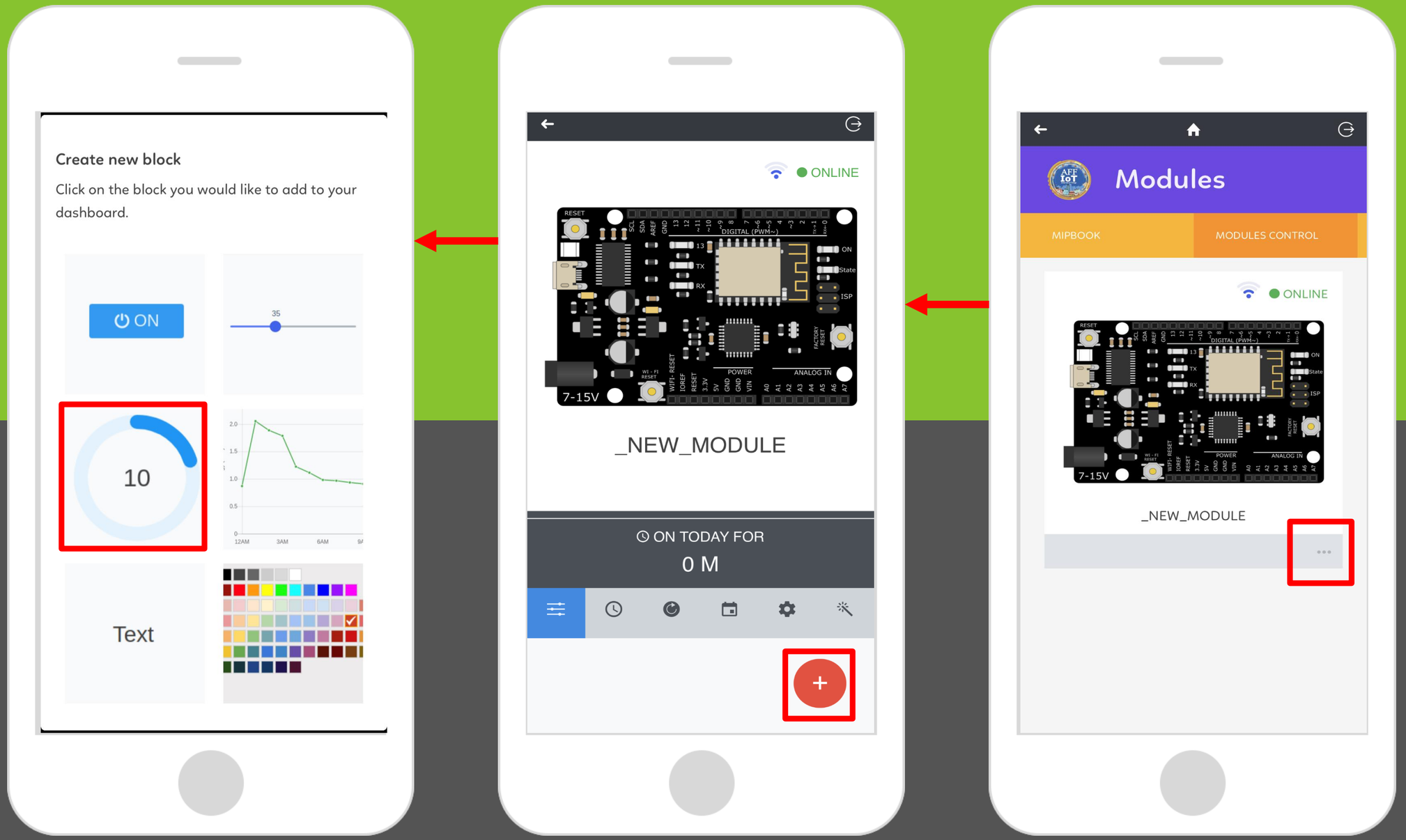
أكتب في الفراغ المعلومات التالية





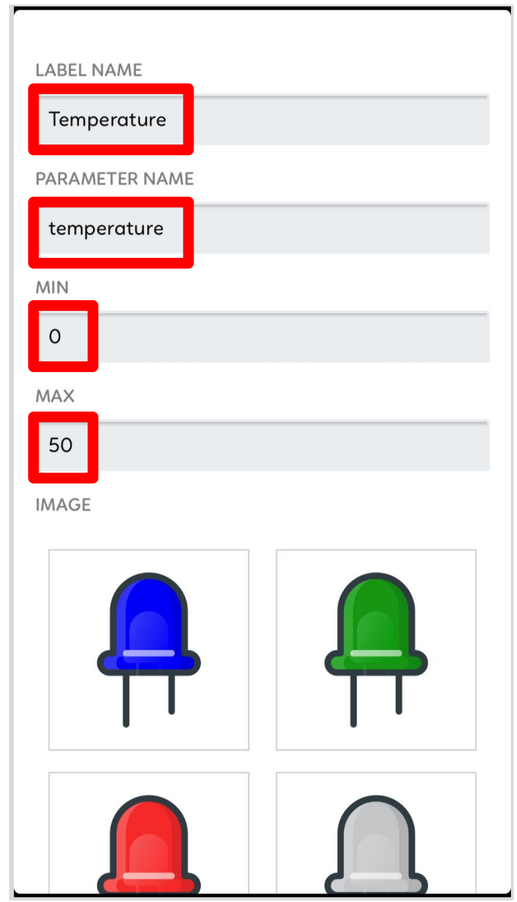
قم بالنزول للأسفل ، ثم انقر ADD

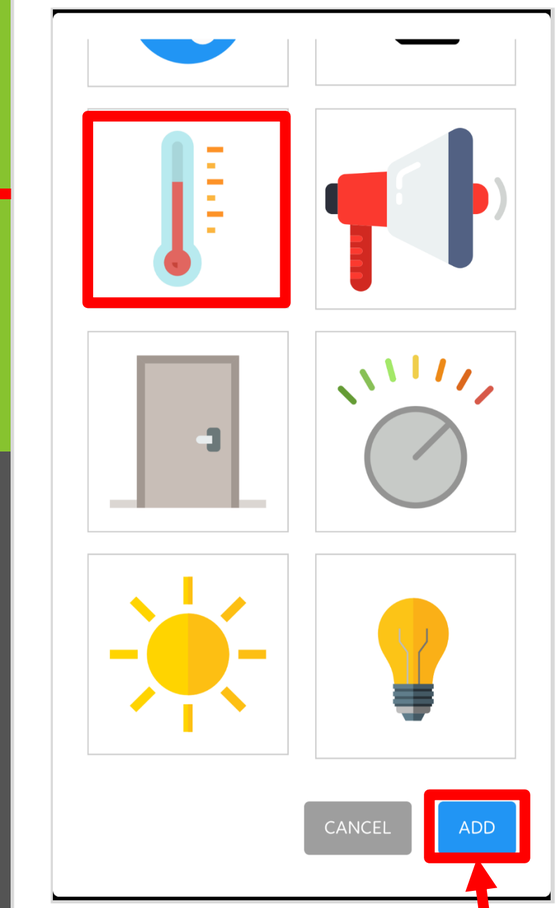



Create new block
Click on the block you would like to add to your dashboard.
ON
35
10
Text
ONLINE
_NEW_MODULE
ON TODAY FOR
0 M
Modules
MIPBOOK
MODULES CONTROL
ONLINE
_NEW_MODULE

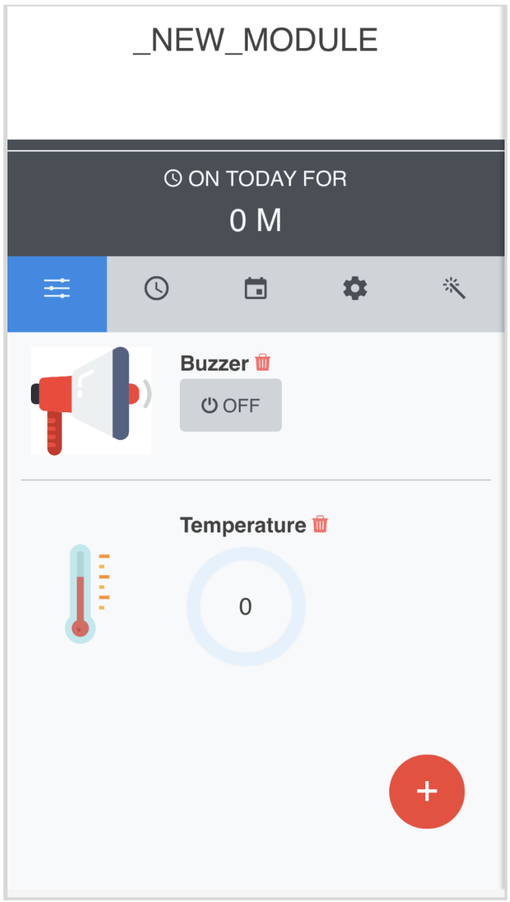
LABEL NAME : هو الاسم الذي يظهر في التطبيق.
PARAMETER NAME : هو الاسم المستخدم في الكودات البرمجية.

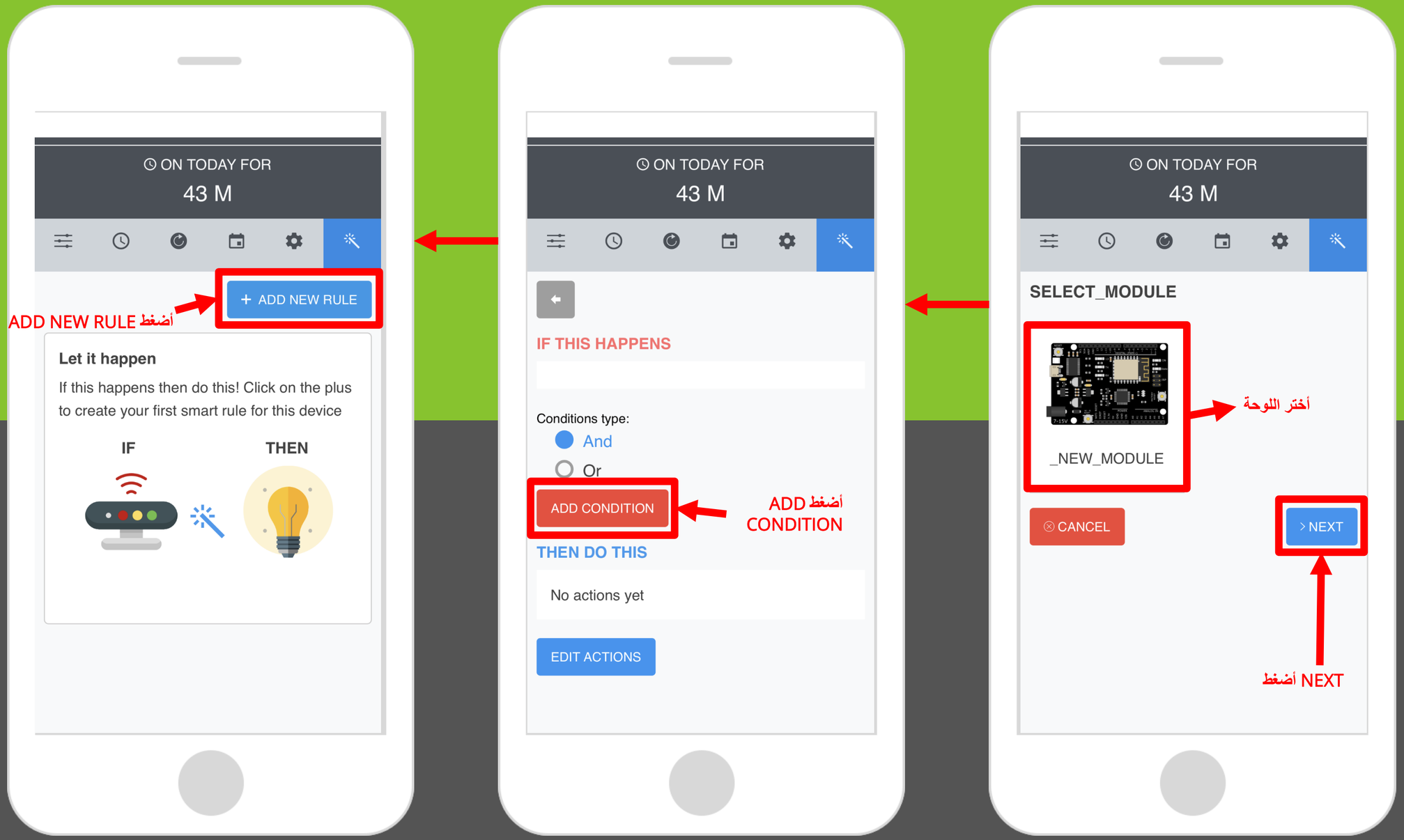
أكتب في الفراغ المعلومات التالية

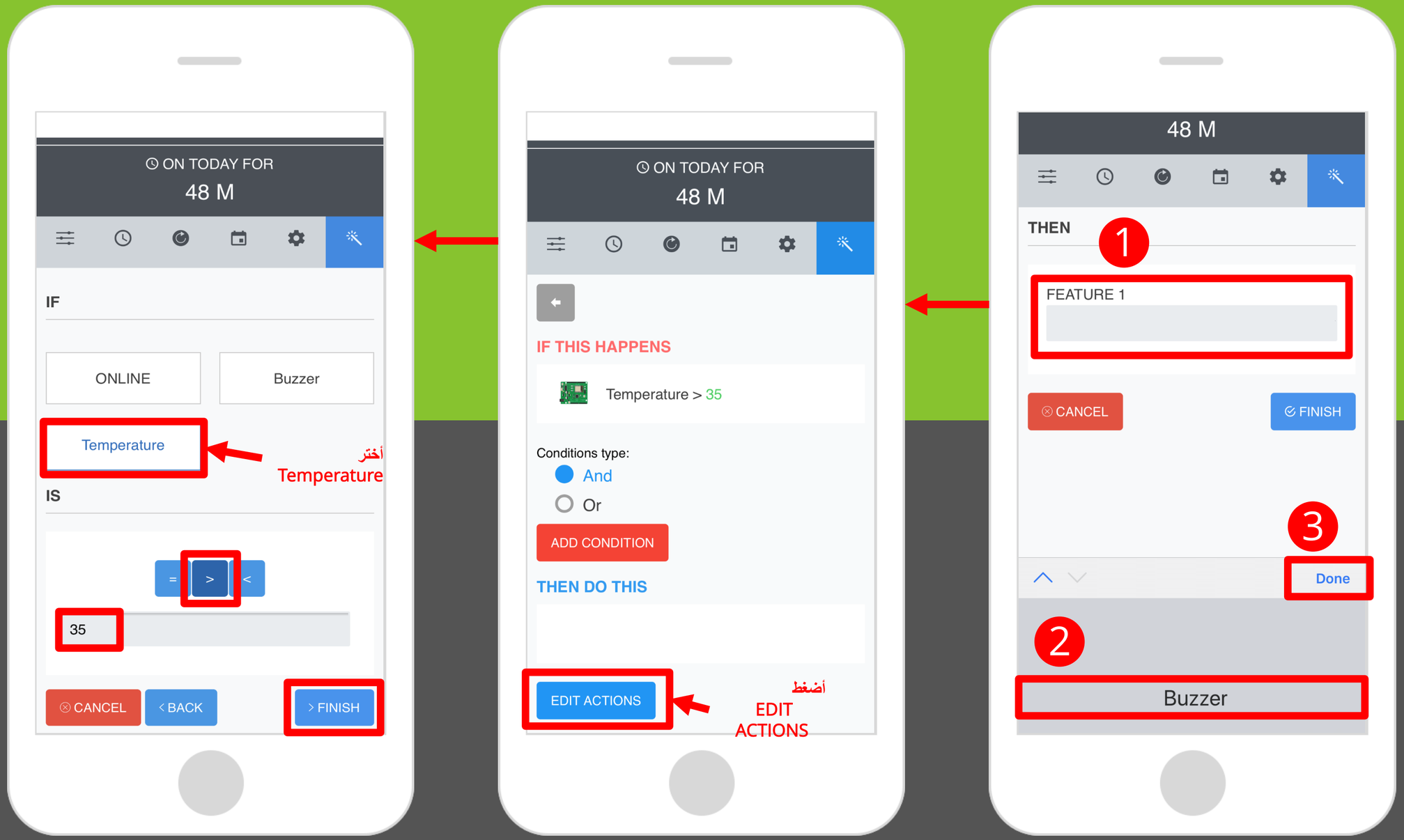




قم بالنزول للأسفل ، ثم انقر ADD




ON TODAY FOR
43 M
+ ADD NEW RULE
أضغط ADD NEW RULE
Let it happen
If this happens then do this! Click on the plus to create your first smart rule for this device
IF
THEN
ON TODAY FOR
43 M
IF THIS HAPPENS
Conditions type:
And
Or
ADD CONDITION
أضغط ADD CONDITION
THEN DO THIS
No actions yet
EDIT ACTIONS
ON TODAY FOR
43 M
SELECT_MODULE
_NEW_MODULE
أختر اللوحة
CANCEL
NEXT
أضغط NEXT


ON TODAY FOR
48 M
IF
ONLINE
Buzzer
Temperature
أختر
Temperature
IS
=
>
<
35
CANCEL
BACK
FINISH
ON TODAY FOR
48 M
IF THIS HAPPENS
Temperature > 35
Conditions type:
And
Or
ADD CONDITION
THEN DO THIS
EDIT ACTIONS
أضغط
EDIT
ACTIONS
48 M
THEN
1
FEATURE 1
CANCEL
FINISH
3
Done
2
Buzzer

ON TODAY FOR
48 M
IF THIS HAPPENS
Temperature > 35
Conditions type:
And
Or
ADD CONDITION
THEN DO THIS
Buzzer =1
EDIT ACTIONS
ON TODAY FOR
48 M
+ ADD NEW RULE
IF Temperature >
35 THEN Buzzer = 1
On
يمكنك تحرير القاعدة أو
حذفها أو إلغاء تنشيطها

Open AFF IoT Board// Circuits > Examples >Overheat_Alarm

Overheat_Alarm

```
/*Start of mandatory lines of codes in each sketch*/
#define RX A0 // define the Receive pin (RX) to communicate with the WiFi module
#define TX A1 // define the Transmit pin (TX) to communicate with the WiFi module
#include <NeoSWSerial.h> // including the library to use the Software Serial rather than the Hardware Serial (Serial)
NeoSWSerial WiFiModule(RX, TX); //initialize the variable to use in communication with the WiFi module
/*End of mandatory lines of code*/

#define BuzzerPin   4
#define TemperatureSensorPin A2


void setup() {
  // put your setup code here, to run once:
  WiFiModule.begin(19200); // begin the communication between the WiFi module and the microcontroller on the board
  pinMode(BuzzerPin, OUTPUT); // configure the pin connected to the Buzzer to be an output
}
void loop() {
  // put your main code here, to run repeatedly:
  float Voltage; // declare a decimal variable to store the read voltage
  int Temperature; // declare an integer variable to calculate the temperature
  Voltage = analogRead(TemperatureSensorPin) * 0.0048828125; // voltage = analog_value * (5/1024);

  Temperature = -21.231 * (Voltage - 3.765);
  // this equation is the linearized form of the temperature equation between 0 and 50 degrees (specific to the thermistor in the kit)

  WiFiModule.println("temperature=" + String(Temperature)); // send the Temperature value to the server

  delay(3000); // wait for 3 second (3000ms = 3s)

  if (WiFiModule.available() > 0) // if the WiFi module receive data from the server
  {
    String Command = WiFiModule.readStringUntil('\n'); // read the command sent from the WiFi module to the microcontroller
    if (Command.indexOf("buzzer=1") >= 0) // if the received command contains "buzzer=1" trigger on the Buzzer
    {
      digitalWrite(BuzzerPin, HIGH); // turn on the LED
    }
    if (Command.indexOf("buzzer=0") >= 0) // if the received command contains "buzzer=0" trigger it off
    {
      digitalWrite(BuzzerPin, LOW); // turn off the LED
    }
  }
}
```

افتح المسودة :
Overheat_Alarm

في هذا المثال، افترضنا أن مستشعر درجة الحرارة هو تركيب صمام مضيء في الغرفة. عندما ترتفع درجة الحرارة لتصل إلى 35 °C درجة مئوية ، يجب على الجرس إصدار صوت إنذار.

هذه الكودات البرمجية هي مزيج من الكودات البرمجية للتحكم في الجرس و مراقبة درجة الحرارة .

ماذا ستلاحظ ؟

لا شيء! ولكن يجب أن تسمع صوتًا عند تقريب الضوء باتجاه المستشعر. إذا لم يعمل ، فتأكد من قيامك بتجميع الدائرة بشكل صحيح وتحقق من الكود وتحميله على لوحتك أو راجع نصائح المساعدة أدناه.

مساعدة

لا صوت أيضا

يتميز هذا الجرس بأنه مستقطب، لذا احرص على توصيل المأخذ الذي يحمل العلامة "+" بالمأخذ الرقمي والآخر بالأرضي.

لا يسمع أي صوت

نظرًا لحجم وشكل الجرس، فمن السهل تفويت الثقوب اليمنى على اللوح. تحقق جيدا من موضعها.

تطبيقها في الحياة اليومية

تستخدم جميع أنظمة الأمان أجراسا (أو صفارات الإنذار) عند إطلاق إنذار.

تنويه

حقوق النشر © 2019 بواسطة AFF Asset Switzerland S.A., جميع الحقوق محفوظة.
تخضع مجموعة أدوات AFF IoT, ولوحة AFF IoT ومواصفاتها ، ومتطلبات النظام ، ومدى توفرها للتغيير دون إشعار.
جميع العلامات التجارية الأخرى الواردة هنا هي ملك لأصحابها.

دليل AFF IoT الخاص بمجموعة AFF IoT لخاص بلوحة AFF IoT مرخص بموجب ترخيص Creative Commons Attribution-ShareAlike 4.0 الدولي. لعرض نسخة من هذا الترخيص ، تفضل بزيارة
http://creativecommons.org/licenses/by-sa/4.0/
أو أرسل خطابًا إلى Creative Commons، P.O. Box 1866، Mountain View، CA 94042، USA.

تم إنشاء جميع الرسوم البيانية باستخدام Fritzing.

الشكر موصول لدليل Sparkfun Inventor على صور التوضيح.