برمجة لوحة اردوينو

المتحكمات الدقيقة

م. غنام الجعبري

برنامج اردوينو

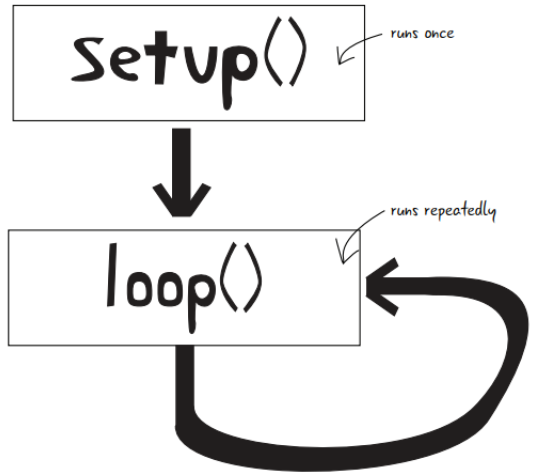
- برنامج اردوينو (Arduino sketch) هو سلسلة من التعليمات المكتوبة باستخدام لغة اردوينو لتنفيذ مجموعة من المهام على لوحة اردوينو
- لغة اردوينو هي لغة برمجة مشتقة من لغة C++ مع اضافة بعض الدوال الخاصة باللوحة
- برمجة لوحة اردوينو تشير الى كتابة التعليمات او الكود وتحميل البرنامج على اللوحة
- يجب على المبرمج قبل كتابة الكود ان يقوم بتحديد الخطوات التي تؤدي الى الحل او صياغة الخوارزمية (algorithm)

عناصر لغة البرمجة

- تتألف لغة البرمجة من مجموعة من العناصر اهمها:
  - بناء البرنامج (Basic Syntax)
  - المتغيرات (Variables)
  - الثوابت (Constants)
  - انواع البيانات (Data Types)
  - الدوال (Functions)
  - العوامل (Operators)
  - اتخاذ القرار (Decision Making)
  - الحلقات (Loops)
  - المصفوفات (Arrays)

بناء البرنامج

- يتكون برنامج اردوينو (sketch) من دالتين هما:
  - دالة ()setup تقوم بتنفيذ التعليمات مرة واحدة بعد تشغيل لوحة اردوينو او بعد إعادة تشغيلها وتستخدم في اعداد اللوحة وتعيين القيم التمهيدية
  - دالة ()loop تقوم بتكرار تنفيذ التعليمات بعد تنفيذ دالة ()setup حتى إيقاف تشغيل لوحة اردوينو وتستخدم في استقبال المدخلات والتحكم بالمخرجات



التعليقات

- تستخدم التعليقات (Comments) لتوضيح ما يقوم به جزء من الكود او لعدم تنفيذ الكود اثناء تنفيذ البرنامج
- يمكن تحديد التعليقات في الكود بطريقتين:
  - تعليق مكون من سطر واحد (single line comment) بإضافة الرمزين // الى بداية السطر
  - تعليق مكون من عدة اسطر (block comment) بإضافة الرمزين */ في بداية السطر والرمزين /* في نهاية السطر

```
// This is a valid comment

/*
  This example code is in the public domain.
  (Another valid comment)
*/
```

المتغيرات والثوابت

- المتغيرات (Variables) هي مواقع لتخزين القيم في الذاكرة بشكل مؤقت
- الثوابت (Constants) هي متغيرات قيمتها ثابتة مثل: LOW, HIGH, INPUT, OUTPUT
- يجب تسمية المتغيرات وفق القواعد التالية:
  - ان يحتوي اسم المتغير على احرف صغيرة او كبيرة او ارقام وعلامة الشرطة السفلية فقط (_)
  - ان لا يبدأ اسم المتغير برقم وانما حرف او علامة الشرطة السفلية
  - يفضل ان لا يزيد طول اسم المتغير عن 31 حرف

```
int i;
float temp;
const int ledPin = 13;
```

```
int number1, number2;
int sum = 0;
sum = number1 + number2;
```

نطاق المتغيرات

- يتم التصريح عن المتغيرات في برنامج اردوينو ضمن نطاق (Scope) او موقع محدد:
  - داخل الدالة وتدعى متغيرات محلية (local variables)
  - خارج كافة الدالات او في بداية البرنامج وتدعى متغيرات عامة (global variables)
- يمكن استخدام المتغيرات المحلية في داخل الدالة فقط ولا يتم التعرف عليها خارج الدالة بينما تبقى المتغيرات العامة طيلة وقت تنفيذ البرنامج ويمكن الوصول اليها من اي دالة

```
float temp;  // any function will see this variable

void setup() {
  // ...
}

void loop() {
  int i;    // "i" is only "visible" inside of "loop"
  float f;  // "f" is only "visible" inside of "loop"
  // ...
}
```

أنواع البيانات

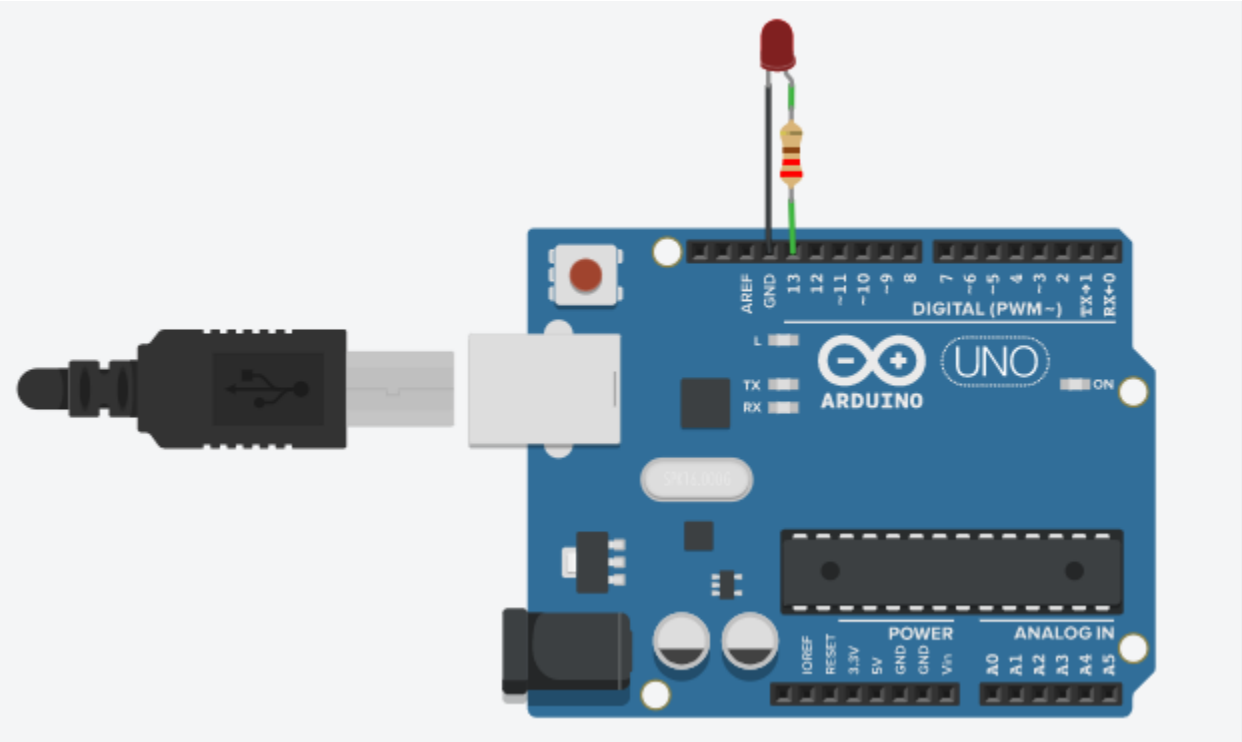
- تستخدم انواع البيانات (Data Types) في التصريح عن المتغيرات وتحديد حجم البيانات
- أنواع البيانات الاساسية:
  - int عدد صحيح مثل 12
  - long عدد صحيح مضاعف مثل 123456
  - float عدد حقيقي مثل 12.1
  - double عدد حقيقي مضاعف مثل 1234.12
  - char حرف مثل A
  - bool منطقي اما true او false

```
int i = 0;
float f = 12.1;
char c = 'A'
bool running = false;
```

الدوال

- الدالة (Function) هي كود مخصص لأداء مهمة محددة مثل طباعة رسالة او قراءة قيمة
- تستخدم الدوال في لوحة اردوينو للتحكم بمنافذ الادخال والاخراج واجراء العمليات الحسابية
- تحتوي لغة برمجة اردوينو على مجموعة من الدوال التي تم بناؤها مسبقا تشمل:
  - دوال الادخال والإخراج الرقمي (Digital I/O)
  - دوال الادخال والإخراج التماثلي (Analog I/O)
  - دوال الوقت (Time)
  - دوال الاتصال (Communication)
  - دوال الرياضيات (Math)

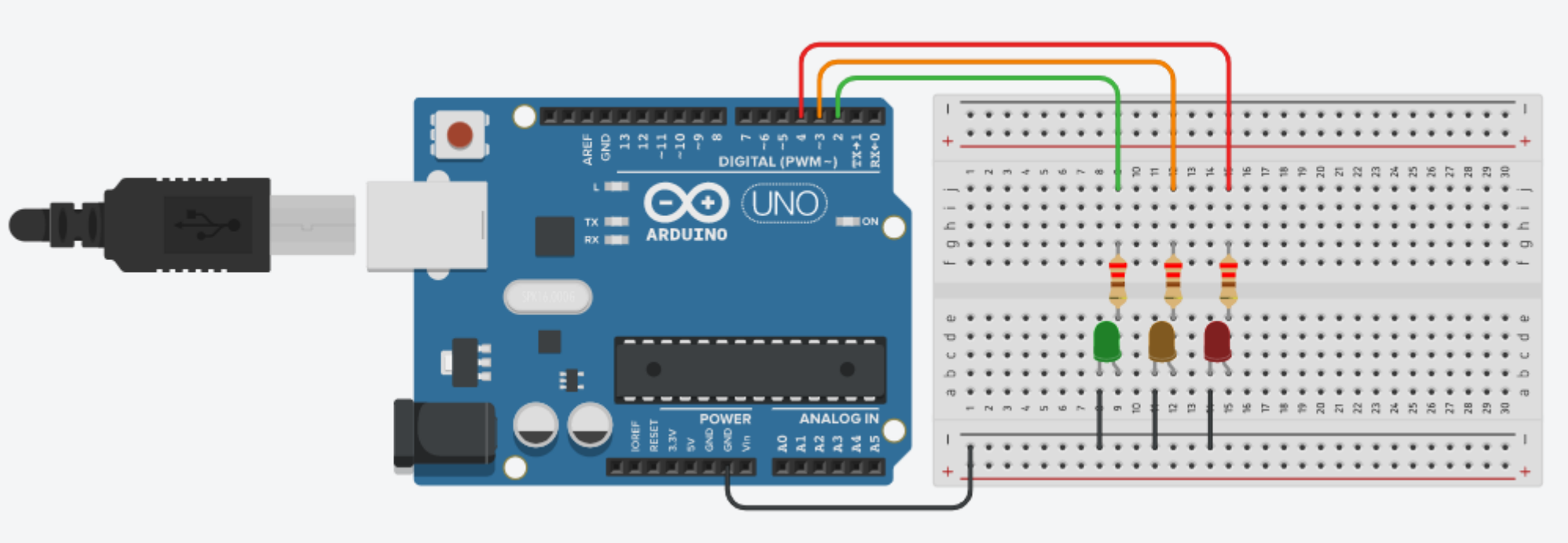
مثال: اختبار اللوحة




```
1  void setup()
2  {
3    pinMode(13, OUTPUT);
4  }
5
6  void loop()
7  {
8    digitalWrite(13, HIGH);
9    delay(1000); // Wait for 1000 millisecond(s)
10   digitalWrite(13, LOW);
11   delay(1000); // Wait for 1000 millisecond(s)
12 }
```

تمرين: الإشارة الضوئية

- اكتب برنامج لتشغيل الإشارة الضوئية باستخدام لوحة اردوينو عن طريق تشغيل الضوء الاحمر لمدة 5 ثوان ثم الضوء البرتقالي لمدة ثانية ثم الضوء الأخضر لمدة 5 ثوان ثم الضوء البرتقالي لمدة ثانية؟



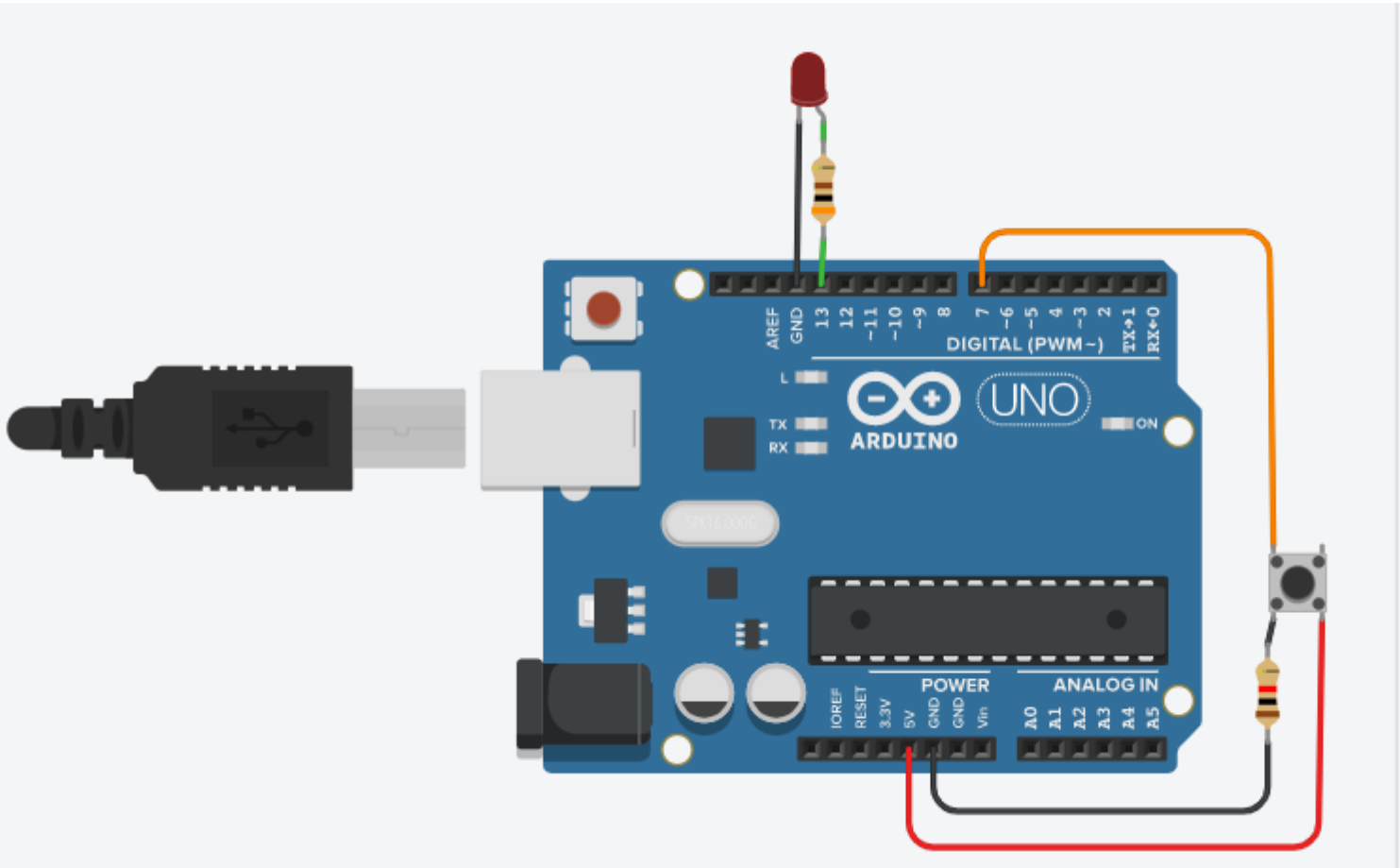
دوال الادخال والإخراج

- تستخدم دالة pinMode() لتحديد الدبوس اما مدخل (INPUT) او مخرج (OUTPUT)
- تستخدم دالة digitalWrite() لكتابة قيمة رقمية على المخرج اما LOW او HIGH
- تستخدم دالة digitalRead() لقراءة قيمة رقمية من المدخل اما LOW او HIGH

```
int ledPin = 13;  // LED connected to digital pin 13
int inPin = 7;    // pushbutton connected to digital pin 7
int val = 0;      // variable to store the read value

void setup() {
  pinMode(ledPin, OUTPUT);  // sets the digital pin 13 as output
  pinMode(inPin, INPUT);    // sets the digital pin 7 as input
}

void loop() {
  val = digitalRead(inPin);   // read the input pin
  digitalWrite(ledPin, val);  // sets the LED to the button's value
}
```

مثال: زر Pushbutton

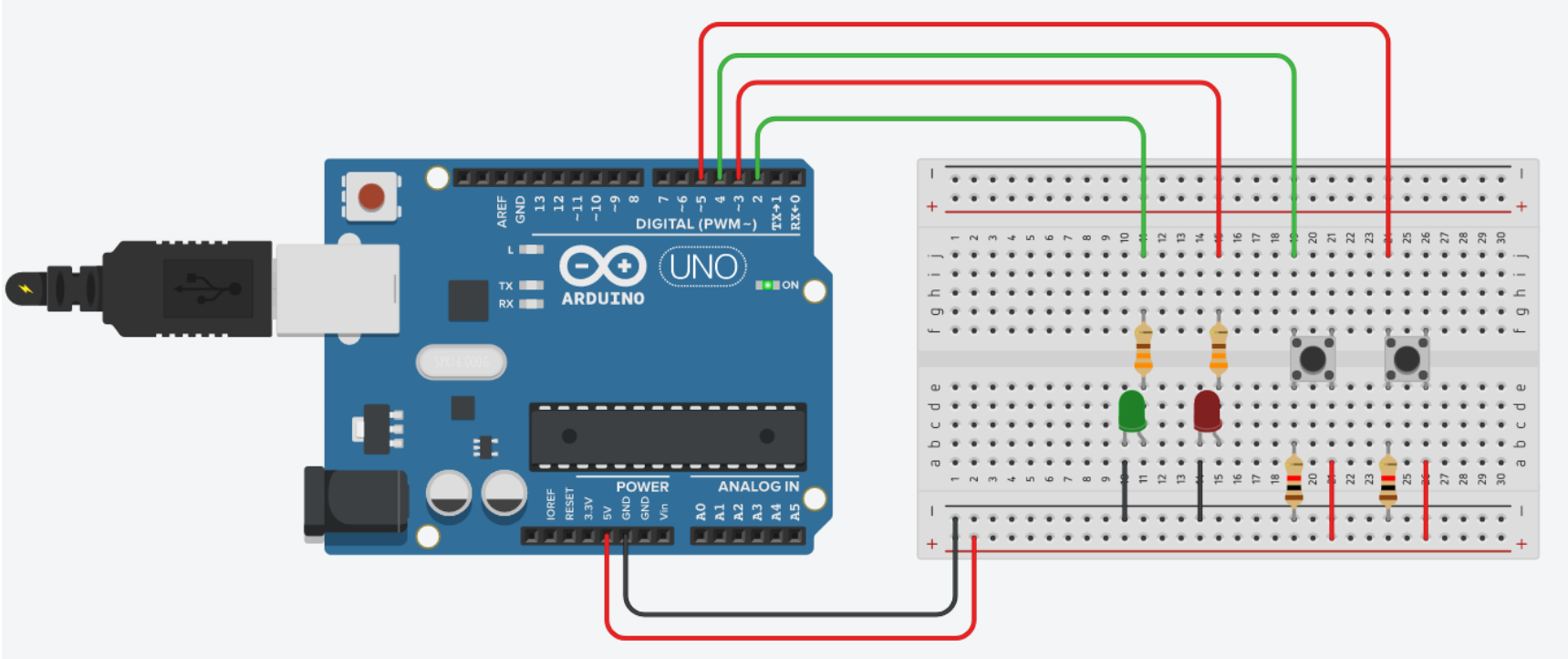
- برنامج لقراءة قيمة رقمية من المدخل وتشغيل او اطفاء LED




```
int x;

void setup()
{
  pinMode(13, OUTPUT);
  pinMode(7, INPUT);
}

void loop()
{
  x = digitalRead(7);
  digitalWrite(13, x);
}
```

تمرين: زر تشغيل

- اكتب برنامج لتشغيل الضوء الاخضر عند الضغط على الزر الاول، وتشغيل الضوء الاحمر عند الضغط على الزر الثاني؟



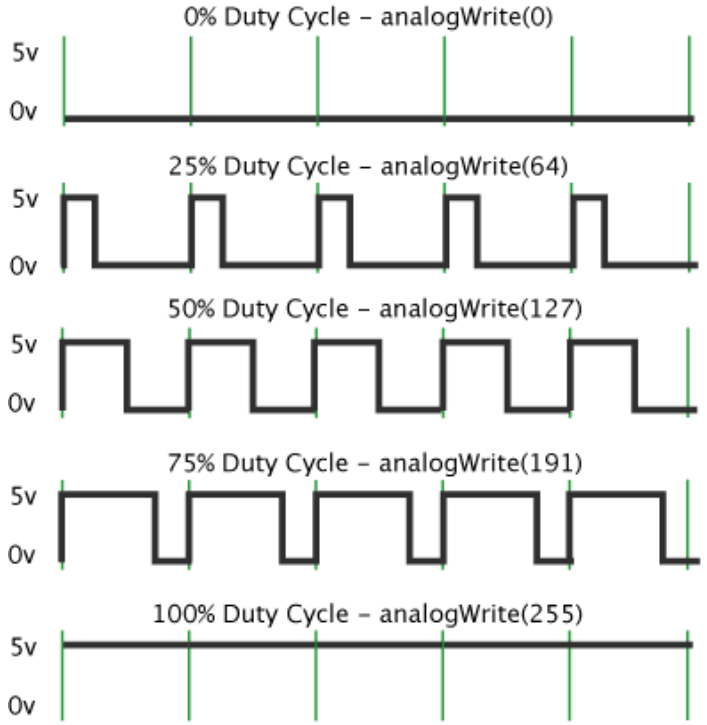
دوال الادخال والاخراج

- تستخدم دالة analogWrite() لكتابة قيمة تماثلية على المخرج على شكل موجة PWM وهذه القيمة تتراوح من 0 الى 255
- تستخدم دالة analogRead() لقراءة قيمة تماثلية من المدخل وهذه القيمة تتراوح من 0 فولت الى 5 فولت وتحويل هذه القيمة الى ما يقابلها رقميا من 0 الى 1023

```
int ledPin = 9;      // LED connected to digital pin 9
int analogPin = 3;   // potentiometer connected to analog pin 3
int val = 0;         // variable to store the read value

void setup() {
  pinMode(ledPin, OUTPUT);  // sets the pin as output
}

void loop() {
  val = analogRead(analogPin);  // analogRead values from 0 to 1023
  analogWrite(ledPin, val / 4); // analogWrite values from 0 to 255
}
```

تقنية PWM

- تستخدم تقنية Pulse Width Modulation للحصول على قيمة تماثلية رقميا
- يتم تحديد القيمة التماثلية عبر التحكم بعرض النبضة (الوقت الذي تكون به الاشارة في حالة HIGH في الدورة الواحدة) أي حاصل ضرب 5 فولت في دورة التشغيل (Duty Cycle)



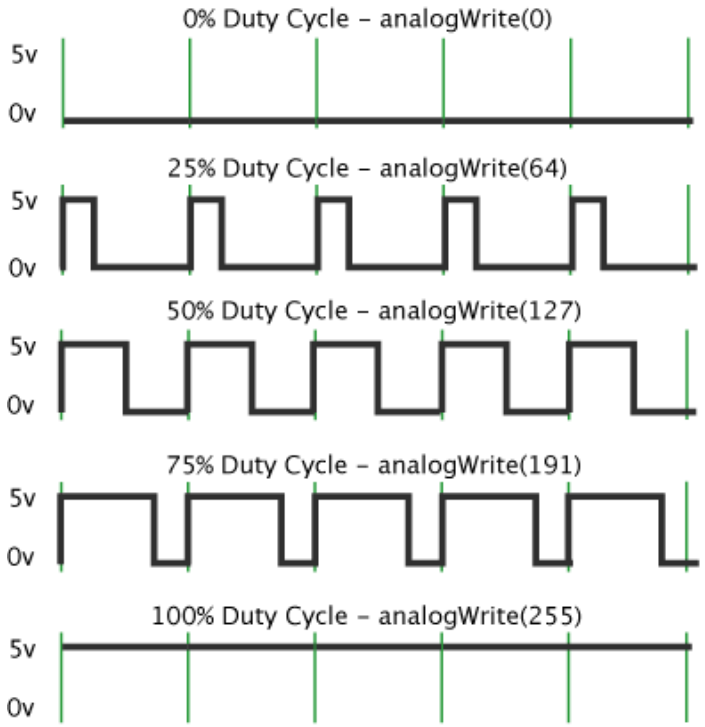
مثال: اخراج قيمة تماثلية

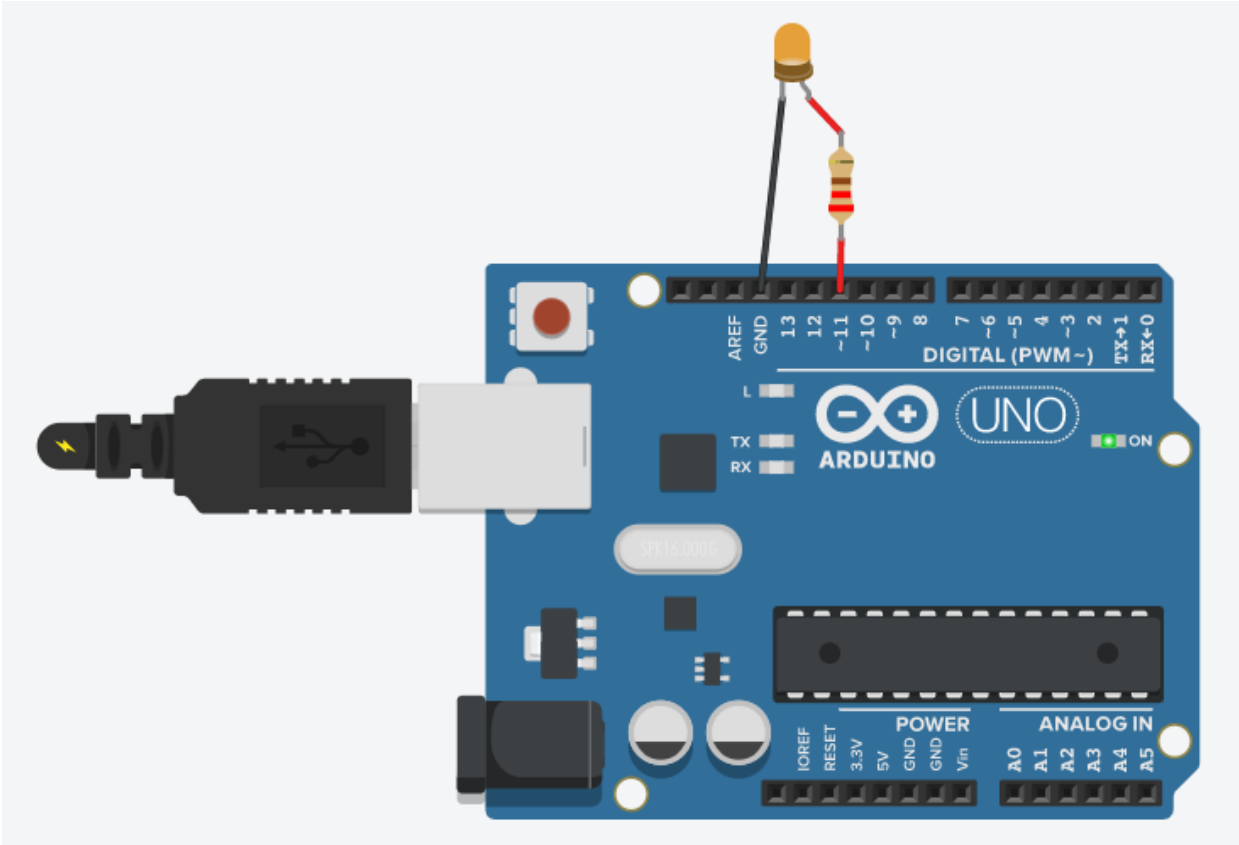
- برنامج لكتابة قيمة تماثلية على المخرج بمقدار 2.5 فولت باستخدام تقنية PWM
- دورة التشغيل = 2.5/5 * %100 = %50
- القيمة التماثلية رقميا = 255 * %50 = 127


```
1  void setup()
2  {
3    pinMode(3, OUTPUT);
4  }
5
6  void loop()
7  {
8    analogWrite(3, 127);
9  }
```

تمرين: شدة الاضاءة

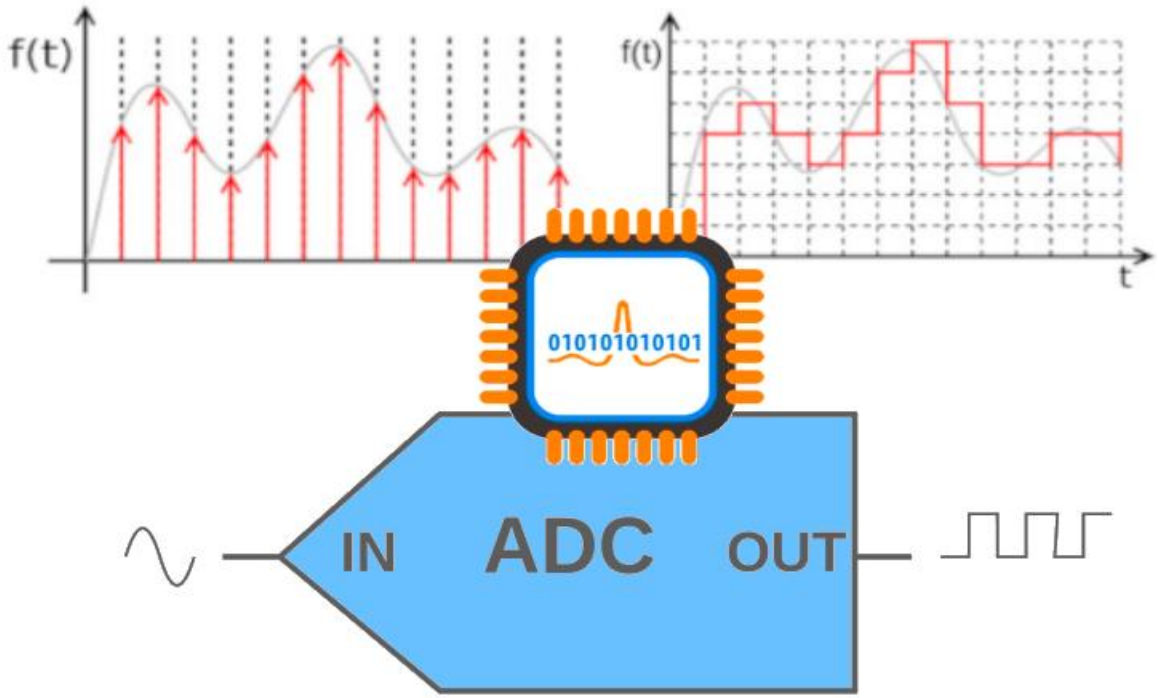
- اكتب برنامج لزيادة شدة الاضاءة تدريجيا باستخدام تقنية PWM وبفاصل زمني 1 ثانية؟





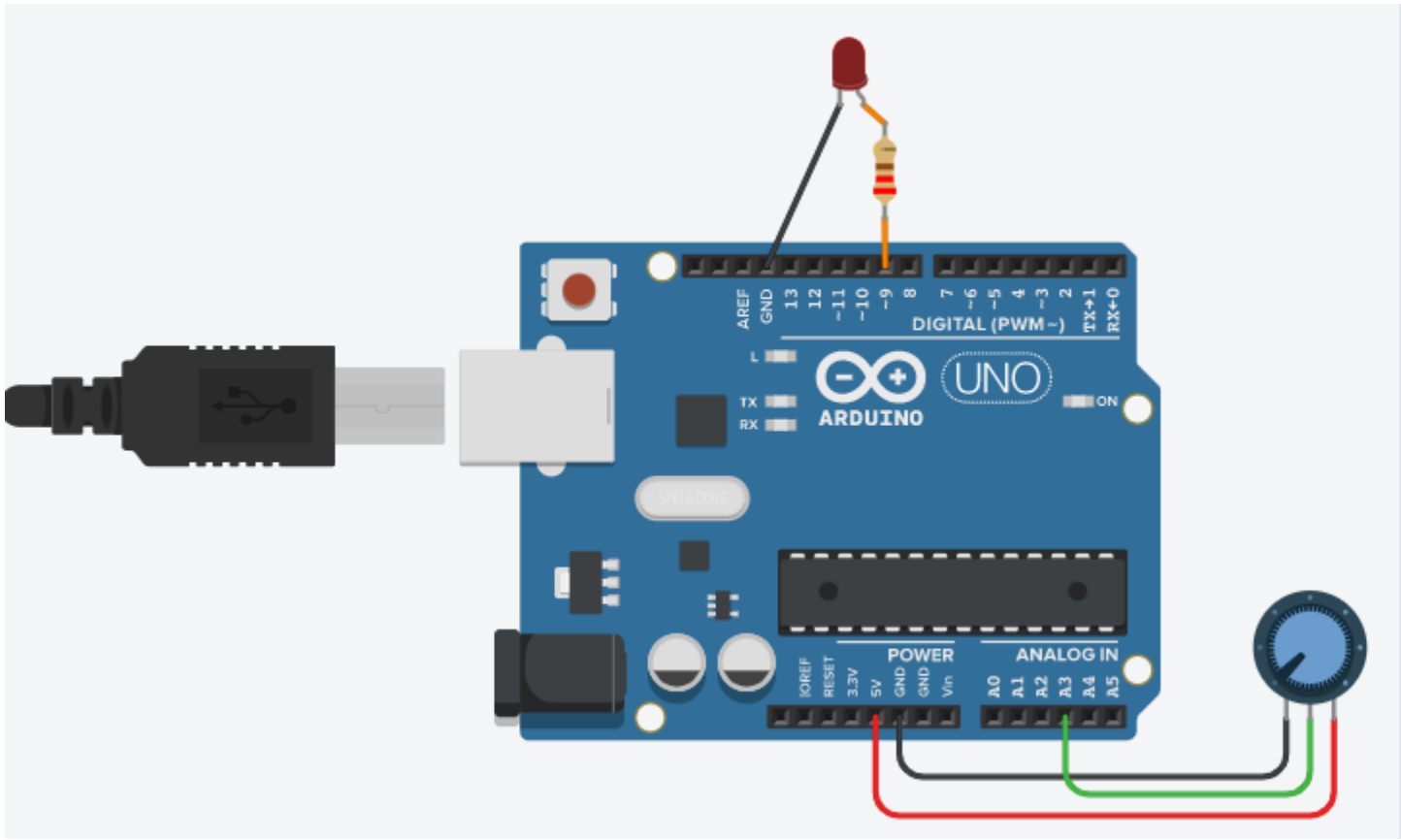
وحدة ADC

- تحتوي لوحة اردوينو على وحدة ADC لتحويل الإشارة التماثلية الى رقمية بدقة 10 خانات ثنائية أي حاصل تقسيم 5 فولت على 1023 (4.9 ميلي فولت لكل عدد عشري)



مثال: مقياس الجهد

- برنامج لقراءة قيمة تماثلية باستخدام وحدة ADC ومقياس الجهد (potentiometer)

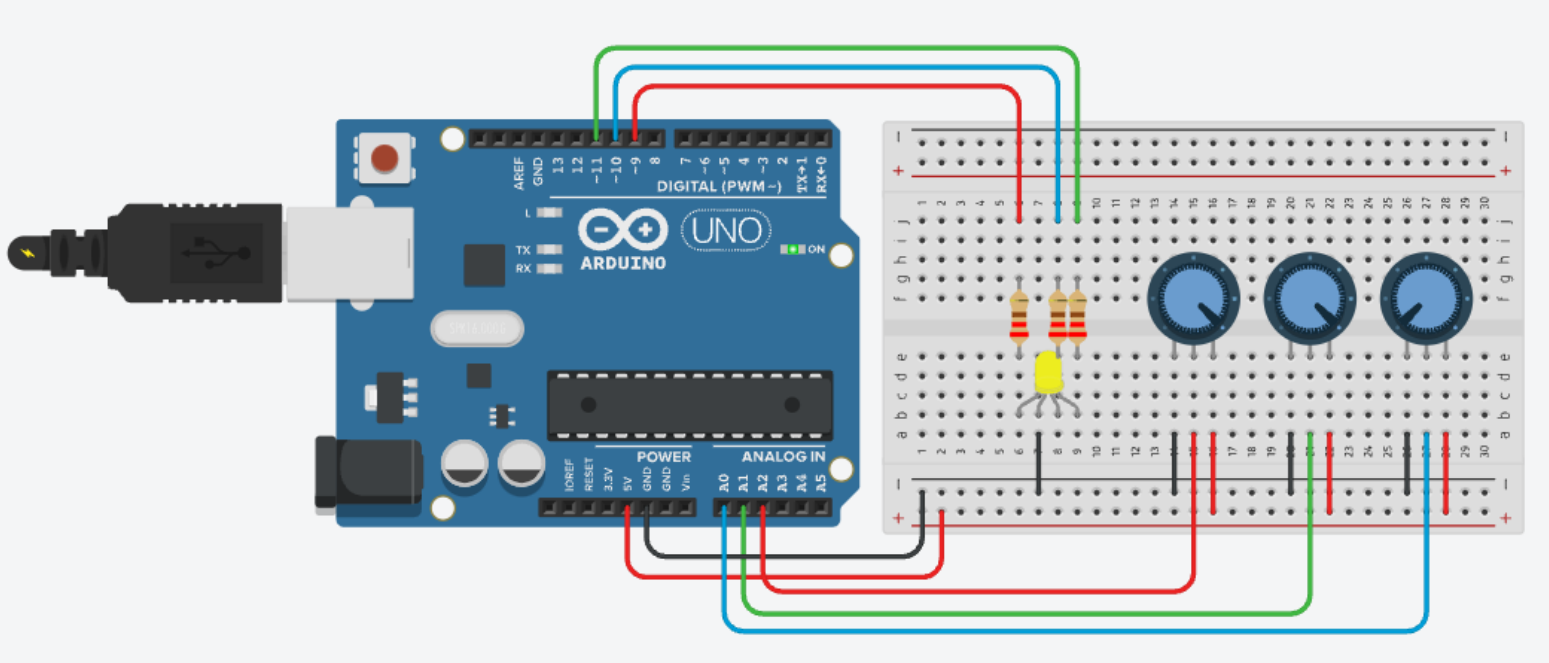



```
1  int x;
2  void setup()
3  {
4    pinMode(9, OUTPUT);
5  }
6
7  void loop()
8  {
9    x = analogRead(3);
10   analogWrite(9, x / 4);
11 }
```

تمرين: التحكم في لون الاضاءة

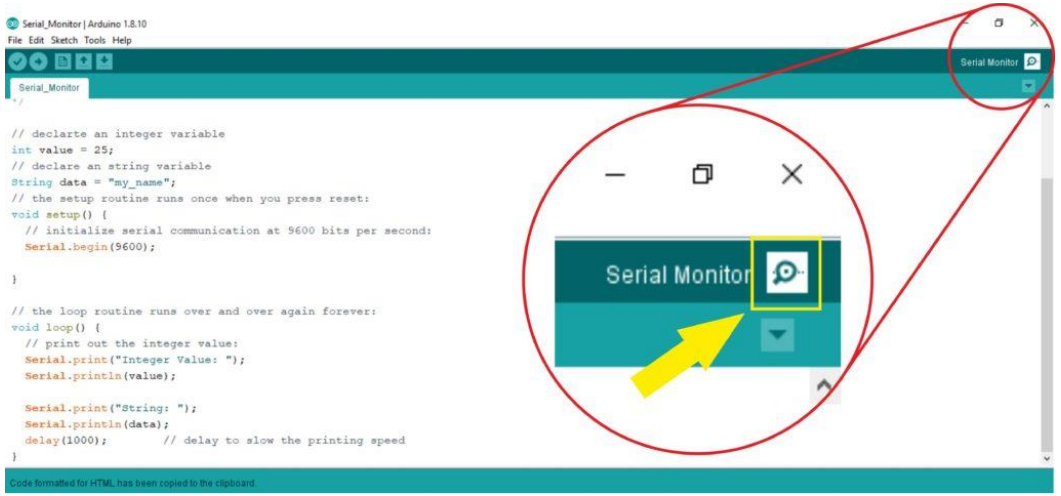
- يتكون الضوء الملون (RGB LED) من ثلاثة الوان هي الاحمر والاخضر والازرق، ومن خلال مزج هذه الالوان مع بعضها يمكن انشاء الوان اخرى، اكتب برنامج للتحكم في لون الاضاءة باستخدام مقياس الجهد (Potentiometer) لكل لون من الالوان الثلاثة؟

اللون	RGB
الاحمر	255, 0, 0
الاخضر	0, 255, 0
الازرق	0, 0, 255
الاصفر	255, 255, 0
الازرق السماوي	0, 255, 255
الارجواني	255, 0, 255
الابيض	255, 255, 255



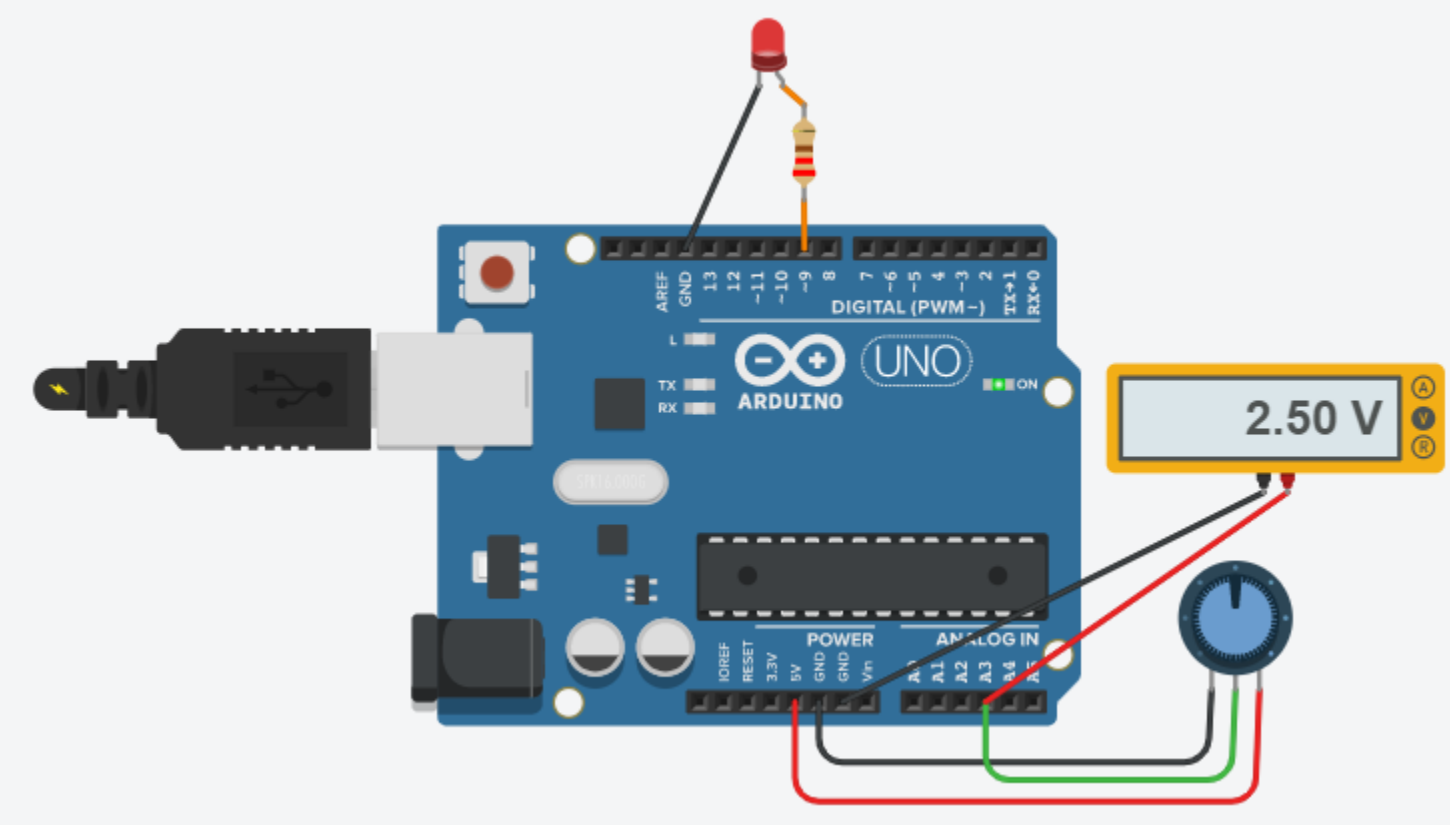
دوال الاتصال

- تحتوي لوحة اردوينو على منفذ تسلسلي (Serial) ويطلق عليه ايضا UART او USART
- يستخدم الدبوس 0 في الاستقبال (Rx) والدبوس 1 في الارسال (Tx)
- تستخدم دالة Serial.begin() لتحديد سرعة المنفذ التسلسلي بوحدة بت لكل ثانية (baud)
- تستخدم دالة Serial.println() لطباعة رسالة نصية على المنفذ التسلسلي
- يمكن استخدام نافذة Serial Monitor في برنامج Arduino IDE لمعاينة المنفذ التسلسلي



مثال: معاينة المنفذ التسلسلي

- برنامج لمعاينة القيمة الرقمية للاشارة التماثلية باستخدام المنفذ التسلسلي




```
1  int x;
2
3  void setup()
4  {
5    pinMode(9, OUTPUT);
6    Serial.begin(9600);
7  }
8
9  void loop()
10 {
11   x = analogRead(3);
12   Serial.println(x);
13   analogWrite(9, x / 4);
14   delay(1000);
15 }
```

Serial Monitor

```
511
511
511
511
511
```

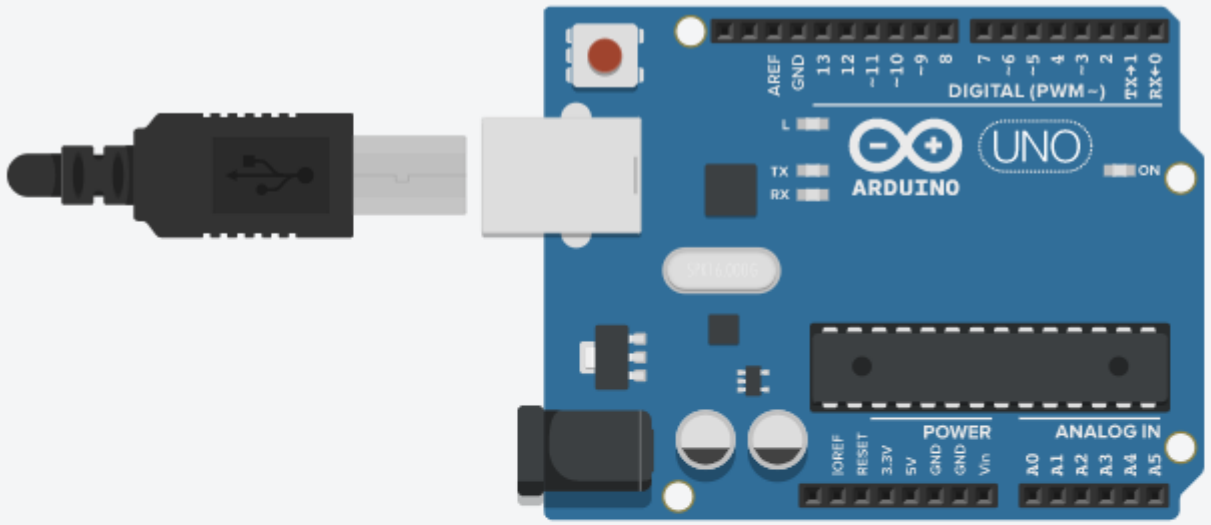
دوال الوقت

- تستخدم دالة delay() لإيقاف تنفيذ البرنامج لمدة من الوقت بوحدة ميلي ثانية (ms)
- تستخدم دالة millis() لمعرفة الوقت الذي مر منذ بداية تنفيذ البرنامج بوحدة ميلي ثانية
- يمكن استخدام دالة millis() بدلا من دالة delay() لتنفيذ مهمة معينة كل فترة من الزمن دون الحاجة الى ايقاف تنفيذ البرنامج بالكامل

```
unsigned long time;

void setup() {
  Serial.begin(9600);
}
void loop() {
  Serial.print("Time: ");
  time = millis();

  Serial.println(time); // prints time since program started
  delay(1000);          // wait a second
}
```

مثال: مؤقت بدء التشغيل

- برنامج لمعرفة الوقت الذي مر منذ بدء تشغيل لوحة اردوينو بوحدة ميلي ثانية (ms)



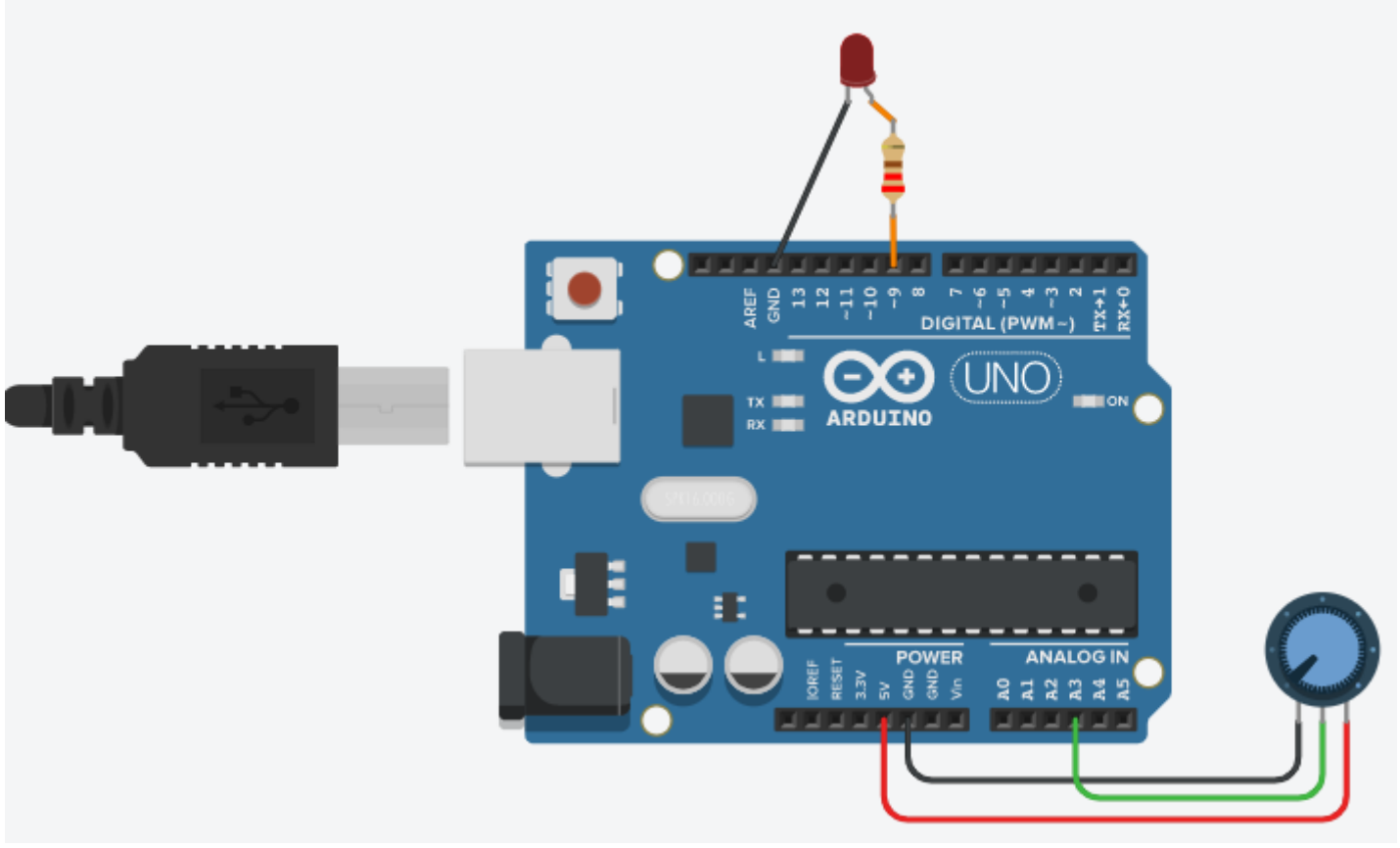

```
1  long time;
2
3  void setup()
4  {
5    Serial.begin(9600);
6  }
7
8  void loop()
9  {
10   Serial.print("Time: ");
11   time = millis();
12   Serial.println(time);
13   delay(1000);
14 }
```

Serial Monitor

```
Time: 1
Time: 1001
Time: 2002
Time: 3003
Time: 4004
Time: 5005
Time: 6006
Time: 7007
```

تمرين: التحكم بمدة الغمّاز

- اكتب برنامج للتحكم بمدة الغمّاز (blinking time) عن طريق قراءة القيمة التماثلية على المدخل وتشغيل LED لمدة من الوقت بما يعادل هذه القيمة رقميا بوحدة ميلي ثانية؟



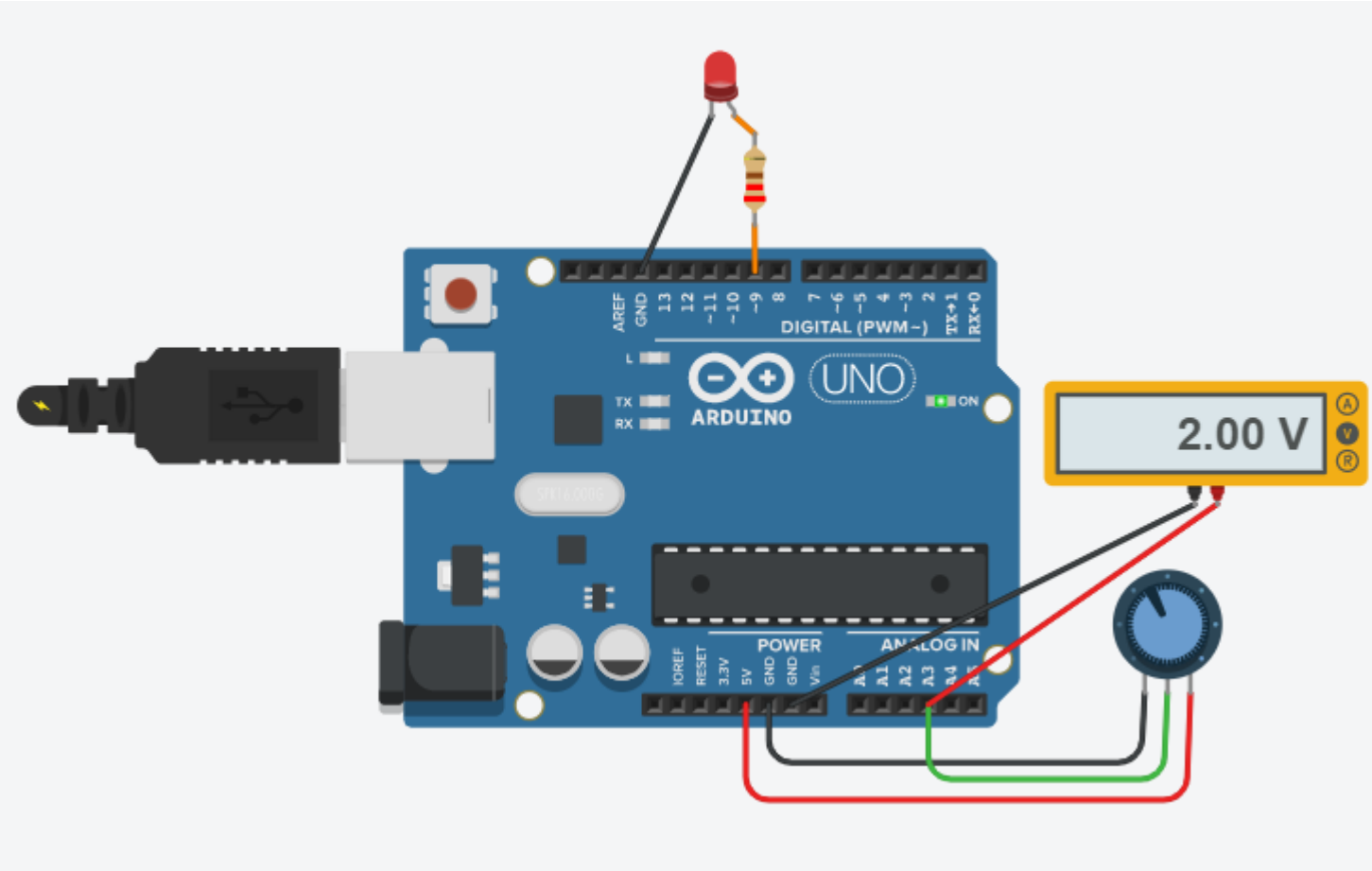
دوال الرياضيات

- تستخدم دالة max() في احتساب اعلى قيمة بين عددين
- تستخدم دالة min() في احتساب اقل قيمة بين عددين
- تستخدم دالة pow() في احتساب القوة للعدد او الأساس والأس
- تستخدم دالة sqrt() في احتساب الجذر التربيعي للعدد
- تستخدم دالة map() في تحويل القيمة من نطاق الى نطاق آخر

```
void setup() {
}

void loop() {
  int val = analogRead(0);
  val = map(val, 0, 1023, 0, 255);
  analogWrite(9, val);
}
```

مثال: دالة map()

- برنامج لقراءة القيمة التماثلية من المدخل وكتابة القيمة المقابلة لها على المخرج




```
1  int inVal;
2  int outVal;
3
4  void setup()
5  {
6    pinMode(9, OUTPUT);
7    Serial.begin(9600);
8  }
9
10 void loop()
11 {
12   inVal = analogRead(3);
13   Serial.print("input value = ");
14   Serial.println(inVal);
15   outVal = map(inVal, 0, 1023, 0, 255);
16   Serial.print("output value = ");
17   Serial.println(outVal);
18   analogWrite(9, outVal);
19   delay(1000);
20 }
```

Serial Monitor

```
input value = 409
output value = 101
input value = 409
output value = 101
input value = 409
output value = 101
```

العوامل

- العوامل الحسابية:
  - + الجمع
  - – الطرح
  - * الضرب
  - / القسمة
  - % باقي القسمة
- عوامل الزيادة والنقصان:
  - ++ الزيادة بمقدار 1
  - -- النقصان بمقدار 1

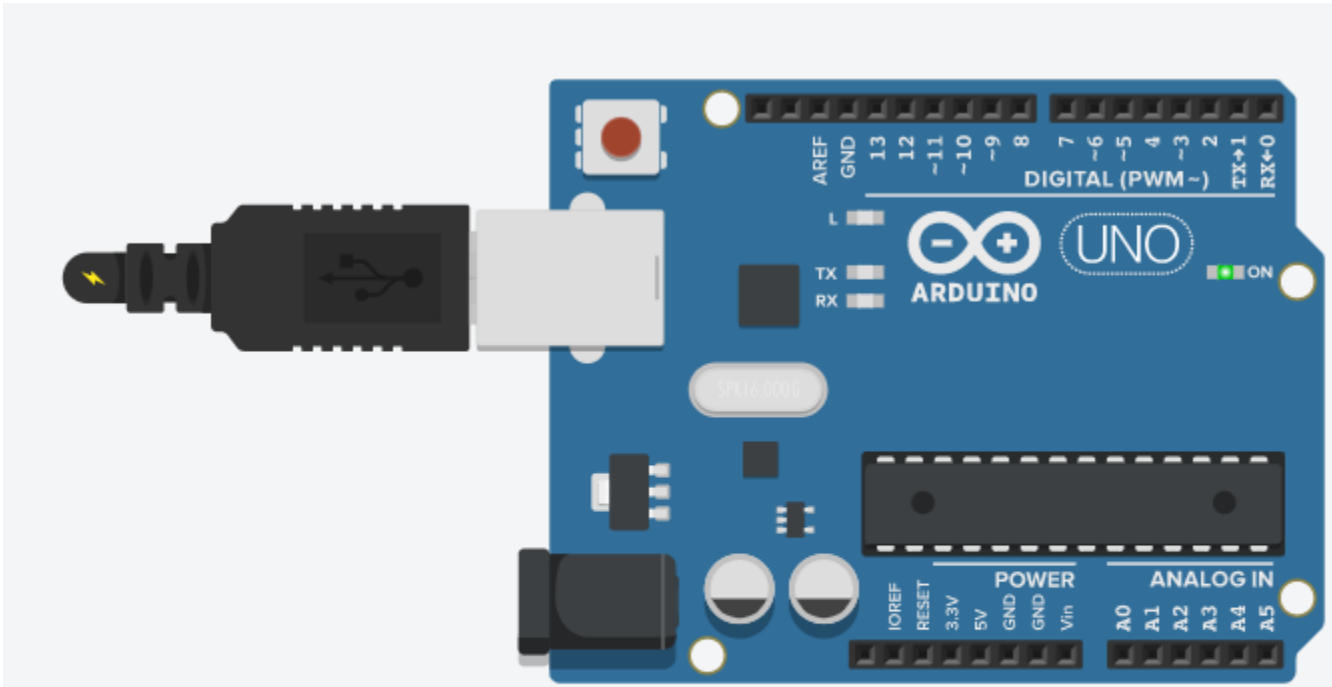
```
int a = 5;
int b = 11;
int c = 0;
c = a * b; // the variable c stores 55
d = b / a; // the variable d stores 2
```

```
int x = 0;
x = 7 % 5;  // x now contains 2
x = 9 % 5;  // x now contains 4
x = 5 % 5;  // x now contains 0
x = 4 % 5;  // x now contains 4
x = -4 % 5; // x now contains -4
x = 4 % -5; // x now contains 4
```

```
x = 2;
y = ++x;  // x now contains 3, y contains 3
y = x++;  // x contains 4, but y still contains 3
```

مثال: عمليات حسابية

- برنامج جمع وضرب وقسمة عددين صحيحين




```
1  int a = 10;
2  int b = 5;
3  int c;
4
5  void setup()
6  {
7    Serial.begin(9600);
8  }
9
10 void loop()
11 {
12   c = a + b;
13   Serial.print("a + b = ");
14   Serial.println(c);
15   c = a * b;
16   Serial.print("a * b = ");
17   Serial.println(c);
18   c = a / b;
19   Serial.print("a / b = ");
20   Serial.println(c);
21   delay(2000);
22 }
```

Serial Monitor

```
a + b = 15
a * b = 50
a / b = 2
```

العوامل

- عوامل المقارنة:
  - == يساوي
  - > اكبر من
  - < اصغر من
  - =! لا يساوي
  - => اكبر من او يساوي
  - =< اصغر من او يساوي
- العوامل المنطقية:
  - && منطق AND
  - || منطق OR
  - ! منطق NOT

```
if (x == y) { // tests if x is equal to y
  // do something only if the comparison result is true
}
```

```
if (x >= y) { // tests if x is greater than or equal to y
  // do something only if the comparison result is true
}
```

```
if (x > 0 && y > 0) { // if both x and y is greater than zero
  // statements
}
```

```
if (x > 0 || y > 0) { // if either x or y is greater than zero
  // statements
}
```

```
if (!x) { // if x is not true
  // statements
}
```

ملخص دوال اردوينو

وصف الدالة	صيغة الدالة
اعداد الدبوس المذكور، اما INPUT او OUTPUT	pinMode(pin, mode)
قراءة قيمة من الدبوس الرقمي، اما HIGH او LOW	value = digitalRead(pin)
كتابة قيمة على الدبوس الرقمي، اما HIGH او LOW	digitalWrite(pin, value)
قراءة قيمة من الدبوس التماثلي	value = analogRead(pin)
كتابة قيمة على الدبوس التماثلي	analogWrite(pin, value)
تحديد سرعة المنفذ التسلسلي بوحدة بت في الثانية	*Serial*.begin(speed)
طباعة قيمة على المنفذ التسلسلي	*Serial*.print(value)
طباعة قيمة على المنفذ التسلسلي مع سطر جديد	*Serial*.println(value)
ايقاف تنفيذ البرنامج مؤقتا بوحدة ميلي ثانية	delay(ms)
استعادة الوقت الذي مر منذ بدأ تشغيل لوحة اردوينو	time = millis()
تحويل القيمة من نطاق الى نطاق آخر	value = map(value, fromLow, fromHigh, toLow, toHigh)

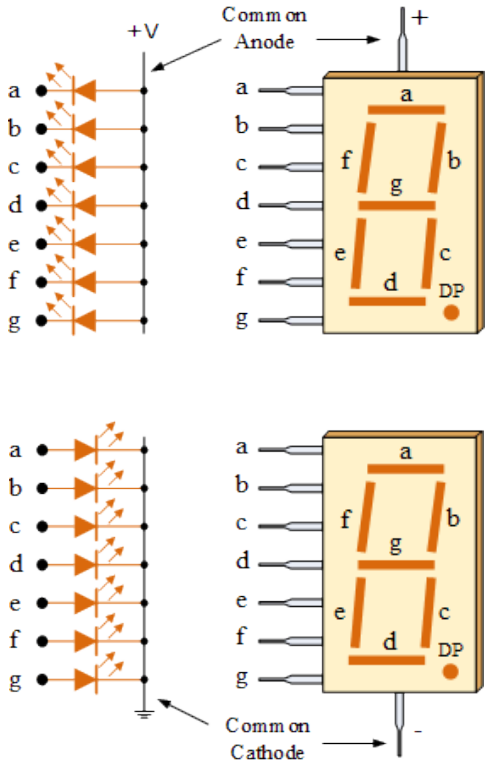
انشاء دالة

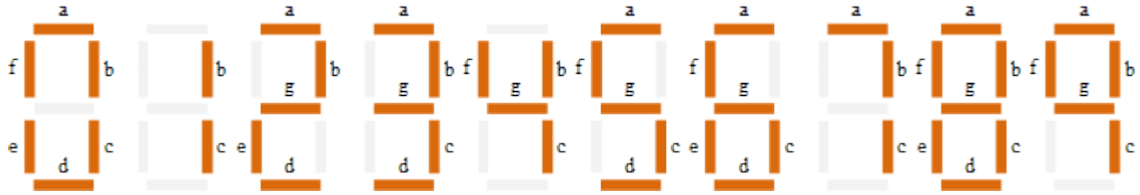
- الدوال نوعان: دوال تم بناؤها مسبقا مثل دوال الادخال والاخراج، ودوال يقوم المستخدم بانشائها عند كتابة البرنامج
- تستخدم الدوال في تقسيم البرنامج الى اجزاء اصغر يمكن فهمها بسهولة واعادة استخدامها
- يمكن انشاء دالة في بداية البرنامج او في نهايته خارج دالة ()setup ودالة ()loop

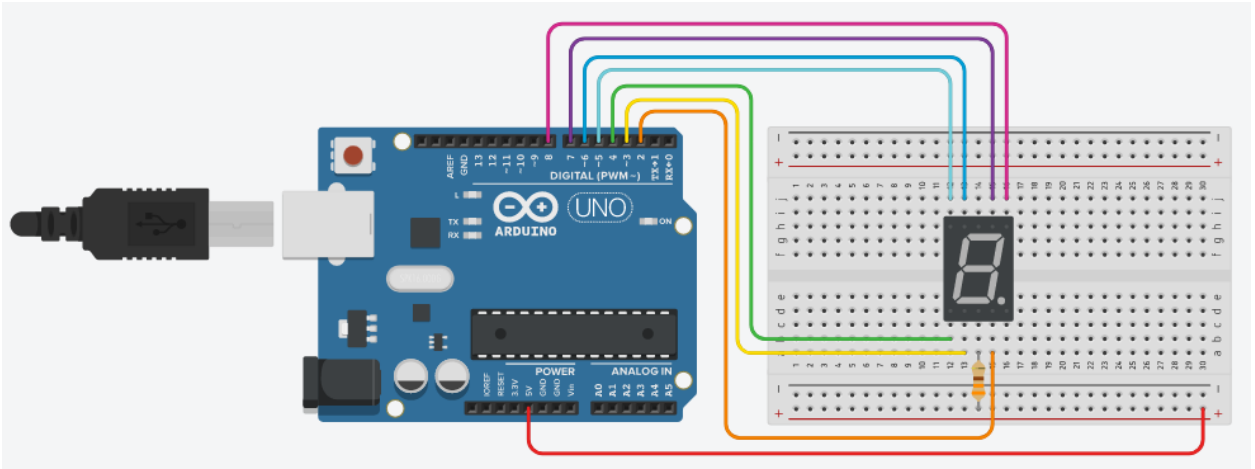
```
void setup() {
  Serial.begin(9600);
}

void loop() {
int x;
  x = multiply(2, 3);
  Serial.println(x);
  delay(500);
}

int multiply(int i, int j) {
  int result;
  result = i * j;
  return result;
}
```

تمرين: عداد رقمي (1)

- تستخدم شاشة الاجزاء السبعة (7 Segment Display) في عرض الارقام العشرية، اكتب برنامج لعرض الارقام من 0 الى 9 مع انشاء دالة منفصلة لعرض كل رقم على الشاشة واضافة فاصل زمني بمقدار ثانية واحدة بين الارقام؟







اتخاذ القرار

- جملة if
- جملة if…else
- جملة switch…case

جملة if

- تستخدم جملة if للتحكم بانسياب البرنامج
- تختبر جملة if الشرط داخل القوسين (...)
  - اذا كان الشرط صائبا (true) يتم تنفيذ الكود داخل جملة if
  - اذا كان الشرط خاطئا (false) لا يتم تنفيذ الكود داخل جملة if

```
if (x > 120) {
  digitalWrite(ledPin1, HIGH);
  digitalWrite(ledPin2, HIGH);
}
```

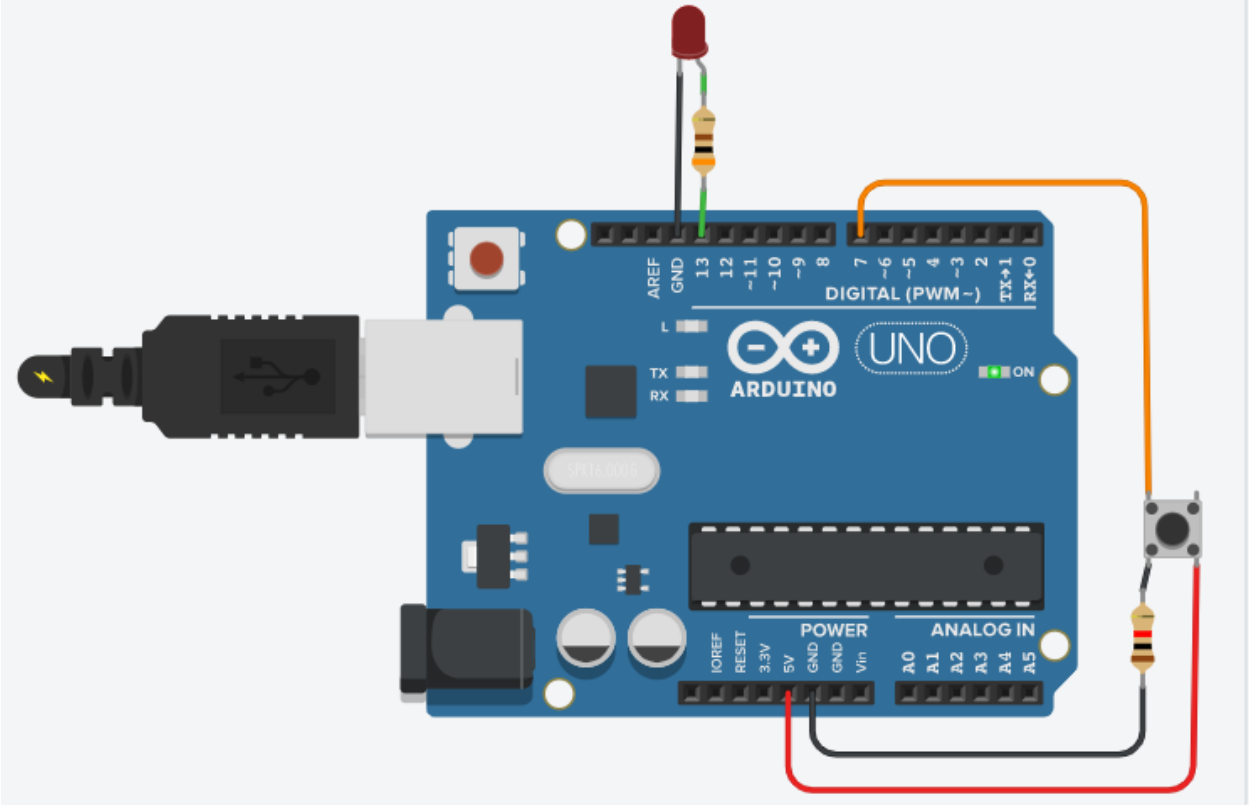
جملة if...else

- تحتوي جملة if على جملة اختيارية هي else
- تختبر جملة if...else الشرط داخل القوسين (...)
  - اذا كان الشرط صائبا (true) يتم تنفيذ الكود داخل جملة if وتجاوز الكود داخل جملة else
  - اذا كان الشرط خاطئا (false) يتم تجاوز الكود داخل جملة if وتنفيذ الكود داخل جملة else
- تستخدم جملة if...else في اختبار اكثر من شرط وفي هذه الحالة تصبح جملة if..else متعددة

```
if (temperature >= 70) {
  // Danger! Shut down the system.
}
else if (temperature >= 60) { // 60 <= temperature < 70
  // Warning! User attention required.
}
else { // temperature < 60
  // Safe! Continue usual tasks.
}
```

مثال: زر تشغيل

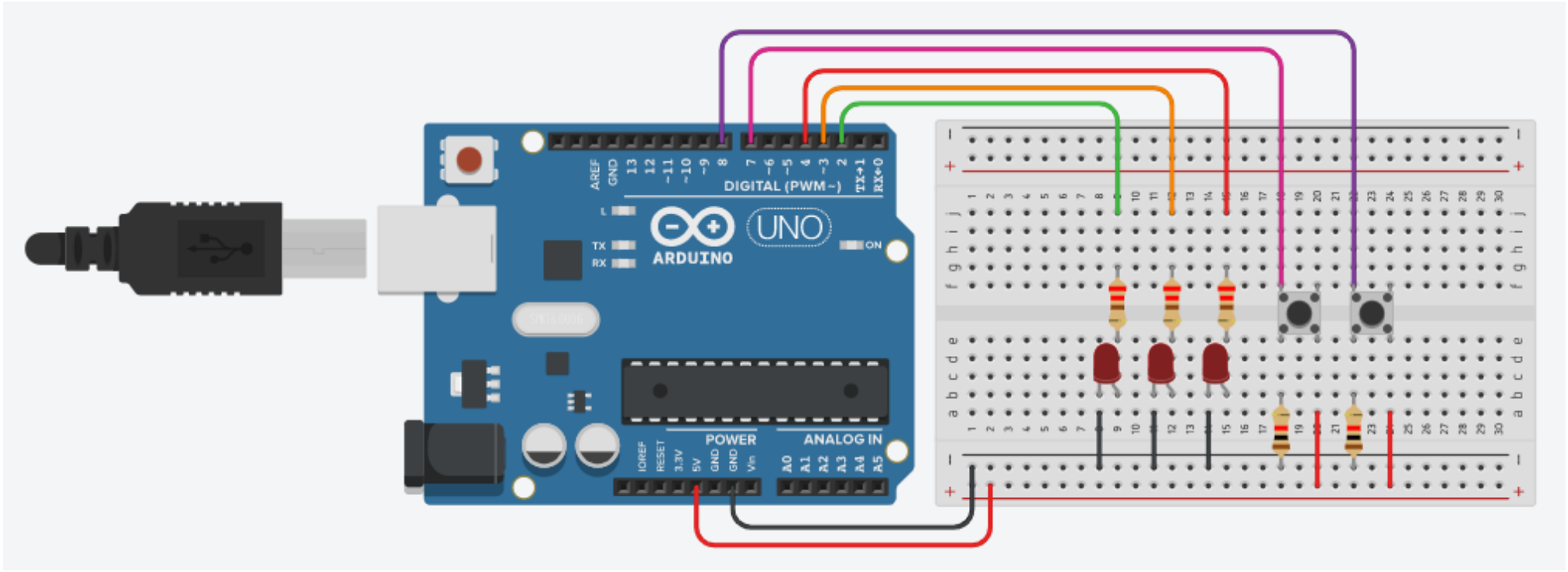
- برنامج لتشغيل واطفاء LED من خلال الضغط على زر pushbutton




```
int x;

void setup()
{
  pinMode(13, OUTPUT);
  pinMode(7, INPUT);
}

void loop()
{
  x = digitalRead(7);
  if (x == HIGH) {
    digitalWrite(13, HIGH);
  }
  else {
    digitalWrite(13, LOW);
  }
}
```

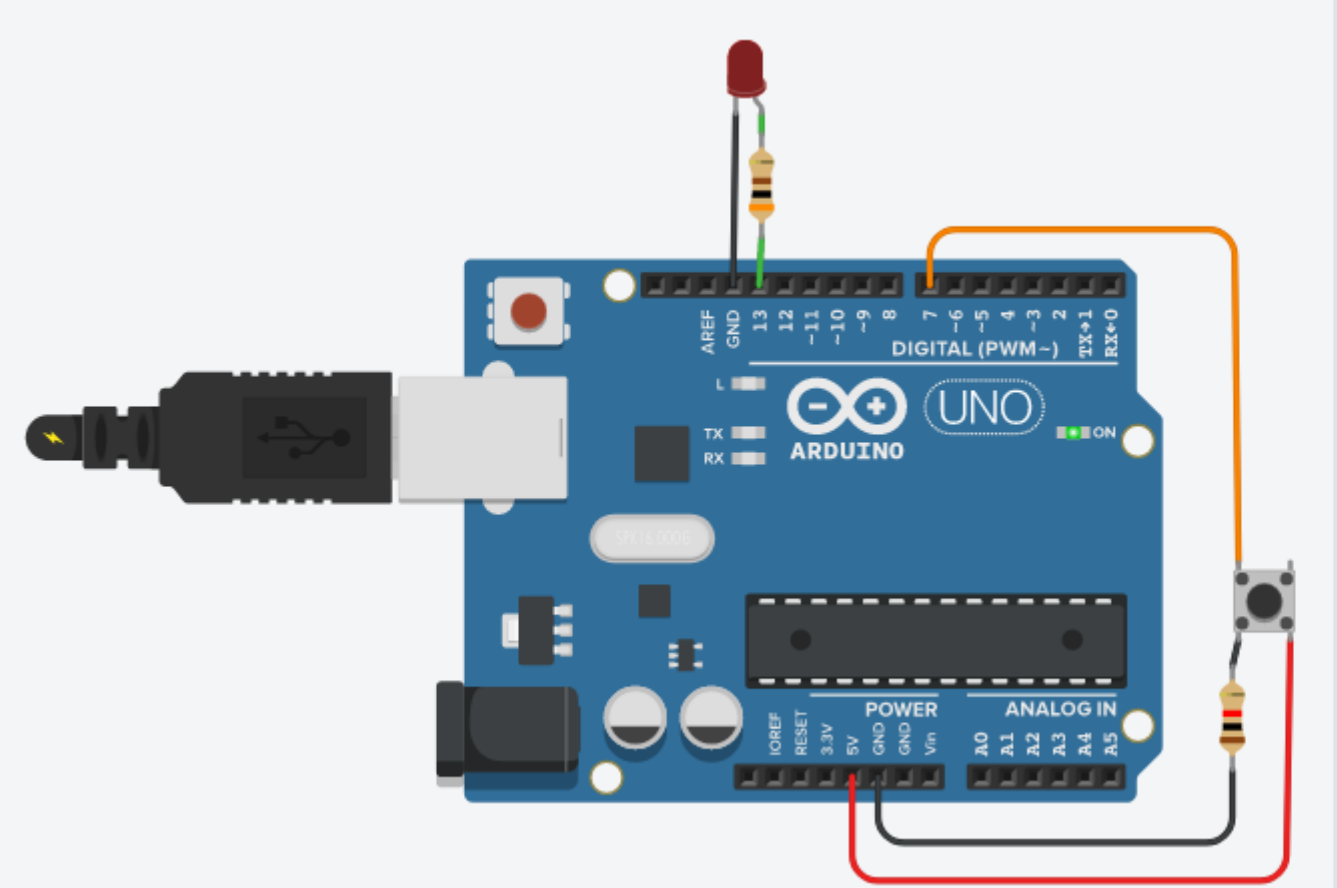
تمرين: تشغيل متتالي

- اكتب برنامج لتشغيل واطفاء مجموعة من LEDs بشكل متتالي وبفاصل زمني 1 ثانية من خلال الضغط على الزر الأول، وتشغيل كافة LEDs من خلال الضغط على الزر الثاني؟



مثال: زر تشغيل او اطفاء

- برنامج لتشغيل او اطفاء LED من خلال الضغط على زر pushbutton مرة واحدة




```
int x;
int btn = LOW;
int led = LOW;

void setup()
{
  pinMode(13, OUTPUT);
  pinMode(7, INPUT);
}

void loop()
{
  x = digitalRead(7);
  if (x != btn) {
    btn = x;
    if (btn == HIGH) {
      led = !led;
    }
  }
  digitalWrite(13, led);
  delay(100);
}
```

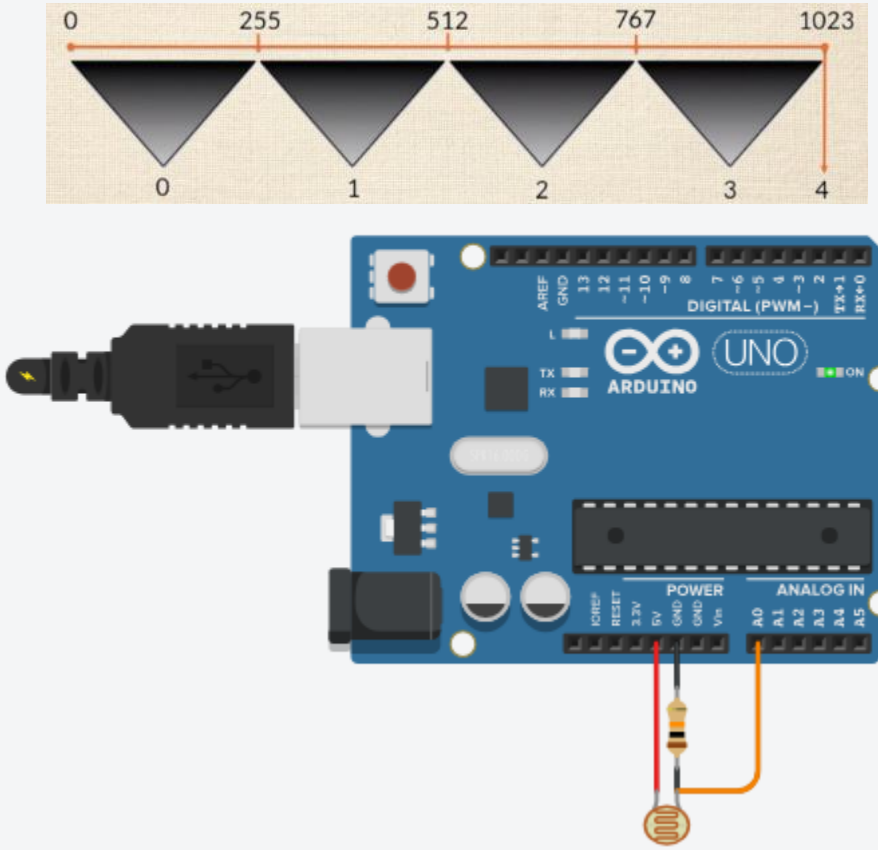
جملة switch...case

- تستخدم جملة switch...case في تقييم العبارة بين القوسين ومقارنتها مع الحالات المذكورة لاختيار احد هذه الحالات وتنفيذ الكود الخاص بها
- في حالة عدم تطابق قيمة العبارة مع الحالات المذكورة يتم تنفيذ الكود الخاص بالحالة الافتراضية وتدعى default
- يجب استخدام كلمة break في نهاية كود كل حالة لإنهاء جملة switch

```
switch (var) {
  case 1:
    //do something when var equals 1
    break;
  case 2:
    //do something when var equals 2
    break;
  default:
    // if nothing else matches, do the default
    // break;
}
```

مثال: مستشعر ضوئي

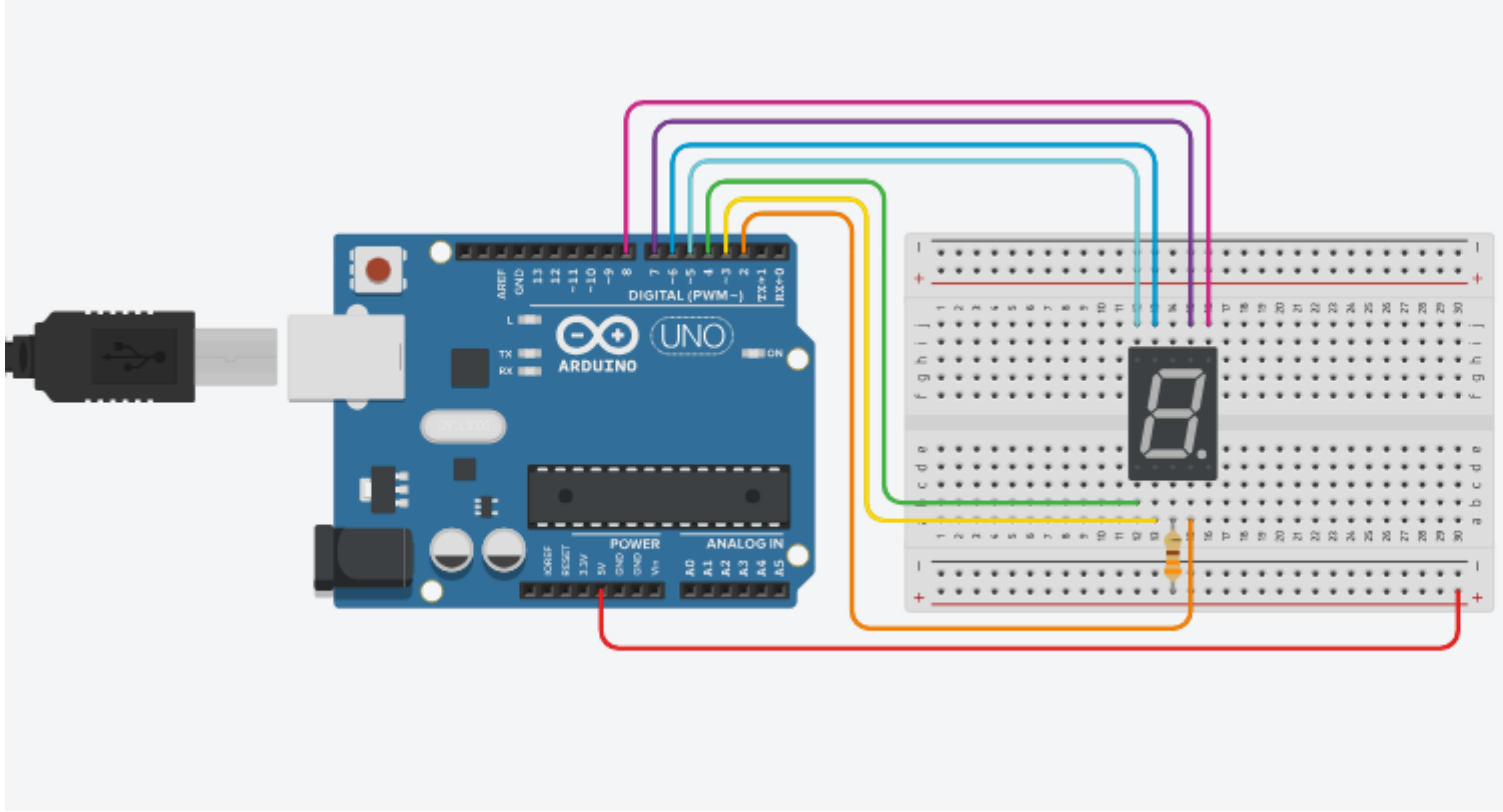
- برنامج لقراءة مستشعر ضوئي (photoresistor) وتحديد أربعة مستويات للإنارة




```
const int ldrPin = A0;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  int ldrVal = analogRead(ldrPin);
  int level = map(ldrVal, 0, 1023, 0, 4);
  switch(level) {
    case 0:
      Serial.println("dark");
      break;
    case 1:
      Serial.println("dim");
      break;
    case 2:
      Serial.println("medium");
      break;
    case 3:
      Serial.println("bright");
      break;
  }
  delay(100);
}
```

تمرين: عداد رقمي (2)

- اكتب برنامج لعرض الارقام من 0 الى 9 على شاشة العرض 7 Segment باستخدام دالة display() لعرض الرقم على الشاشة مع فاصل زمني بمقدار ثانية واحدة بين الارقام؟




```
46  void display(int n)
47  {
48    switch (n) {
49      case 0: zero();
50        break;
51      case 1: one();
52        break;
53      case 2: two();
54        break;
55      case 3: three();
56        break;
57      case 4: four();
58        break;
59      case 5: five();
60        break;
61      case 6: six();
62        break;
63      case 7: seven();
64        break;
65      case 8: eight();
66        break;
67      case 9: nine();
68        break;
69    }
70  }
```

الحلقات

- حلقة for
- حلقة while
- حلقة do…while

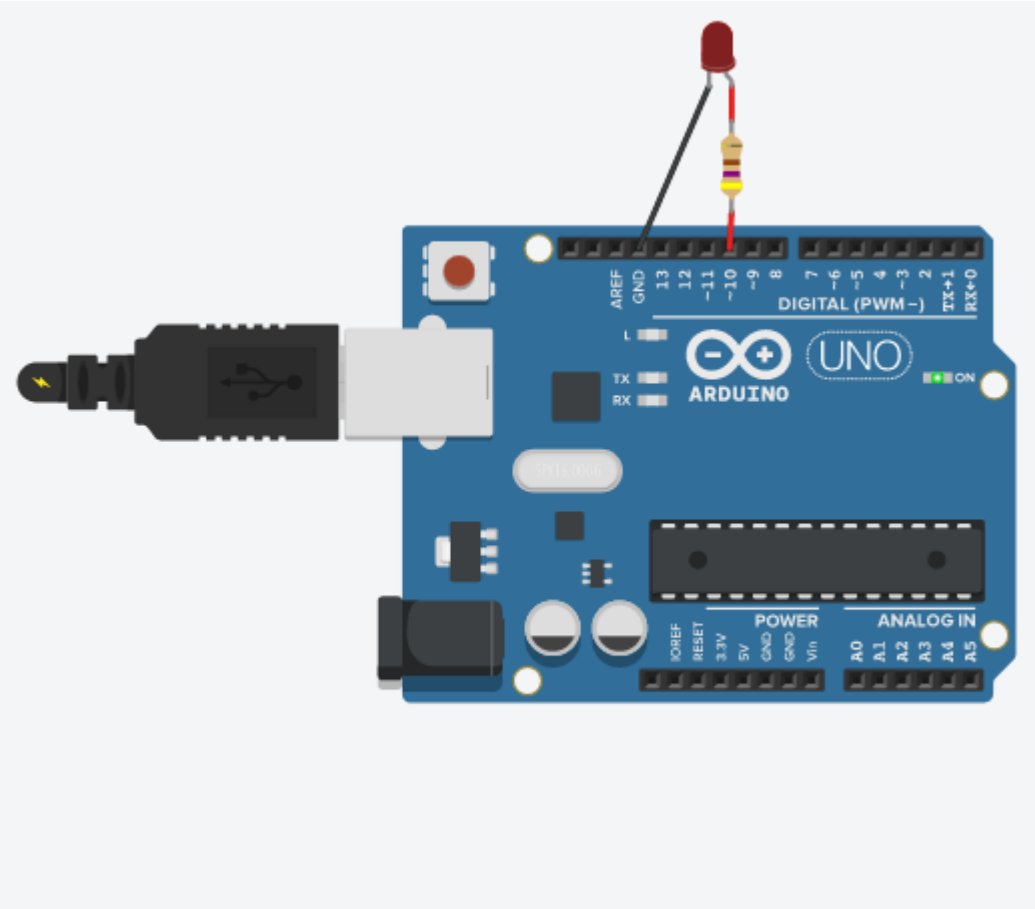
حلقة for

- تستخدم حلقة for لتكرار الكود في الحلقة عدد معين من المرات
- تتكون حلقة for من ثلاثة عبارات (expressions) هي:
  - التهيئة (initialization): تنفذ التهيئة مرة واحدة عند بدء الحلقة
  - الشرط (condition): اذا كان الشرط صائبا ينفذ الكود في الحلقة واذا كان خاطئا يتم انهاء الحلقة
  - التحديث (update): اذا كان الشرط صائبا ينفذ التحديث وعادة يكون التحديث بإضافة او انقاص عدد ثم يقيم الشرط مرة أخرى قبل تكرار الحلقة

```
for (int i = 0; i <= 200; i++) {
     // do something repetitive 200 times
}
```

مثال: تشغيل تدريجي (1)

- برنامج لتشغيل LED تدريجيا باستخدام تقنية PWM



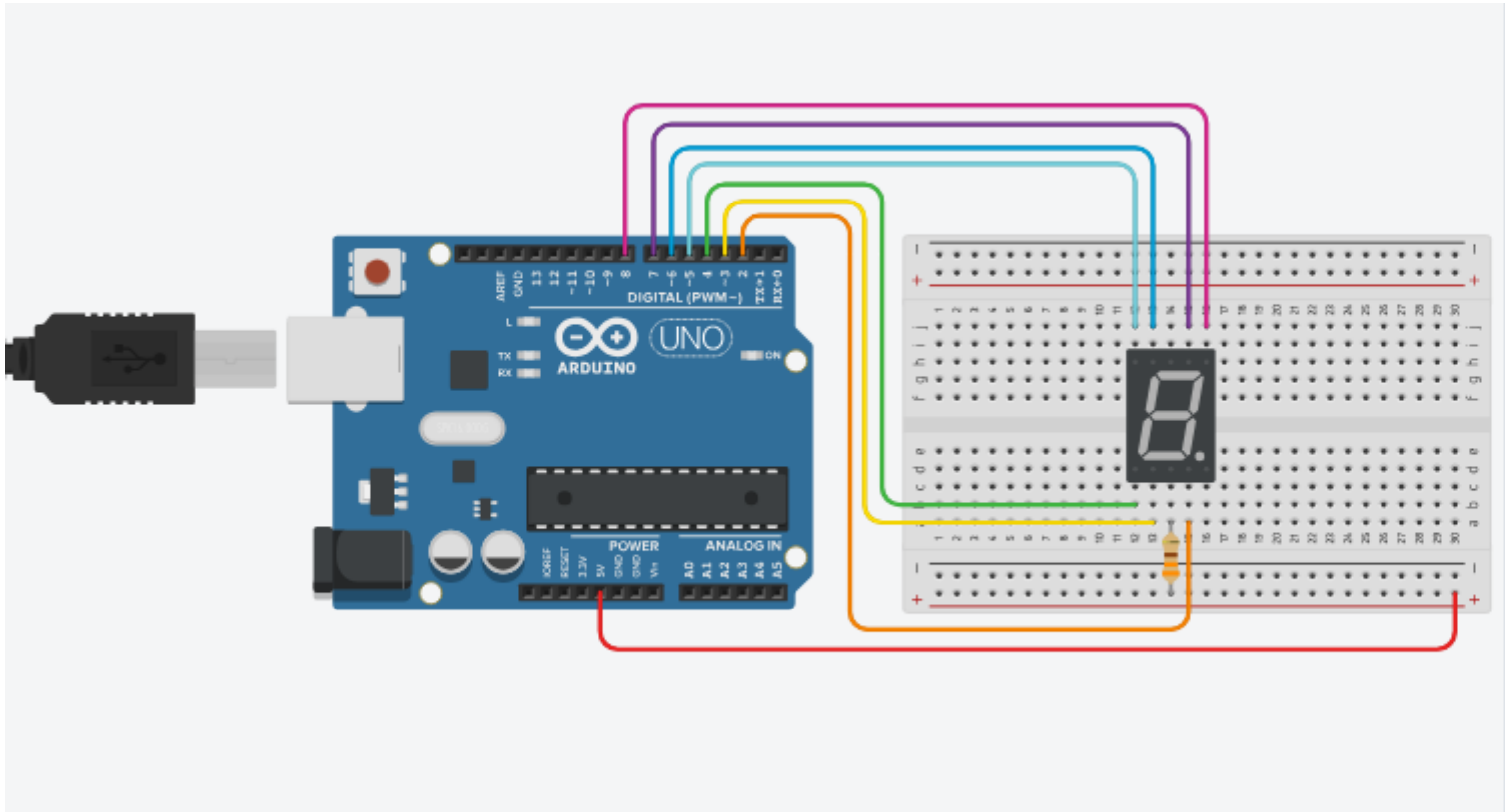

```
1  int led = 10;
2
3  void setup()
4  {
5    pinMode(led, OUTPUT);
6    Serial.begin(9600);
7  }
8
9  void loop()
10 {
11   for (int i = 0; i <= 255; i++) {
12     Serial.println(i);
13     analogWrite(led, i);
14     delay(10);
15   }
16   analogWrite(led, 0);
17   delay(1000);
18 }
```

Serial Monitor

```
248
249
250
251
252
253
254
255
```

تمرين: عداد رقمي (3)

- اكتب برنامج لعرض الارقام من 0 الى 9 على شاشة العرض 7 Segment باستخدام حلقة for مع دالة display() لعرض الرقم على الشاشة مع فاصل زمني بمقدار ثانية واحدة؟




```
46  void display(int n)
47  {
48    switch (n) {
49      case 0: zero();
50        break;
51      case 1: one();
52        break;
53      case 2: two();
54        break;
55      case 3: three();
56        break;
57      case 4: four();
58        break;
59      case 5: five();
60        break;
61      case 6: six();
62        break;
63      case 7: seven();
64        break;
65      case 8: eight();
66        break;
67      case 9: nine();
68        break;
69    }
70  }
```

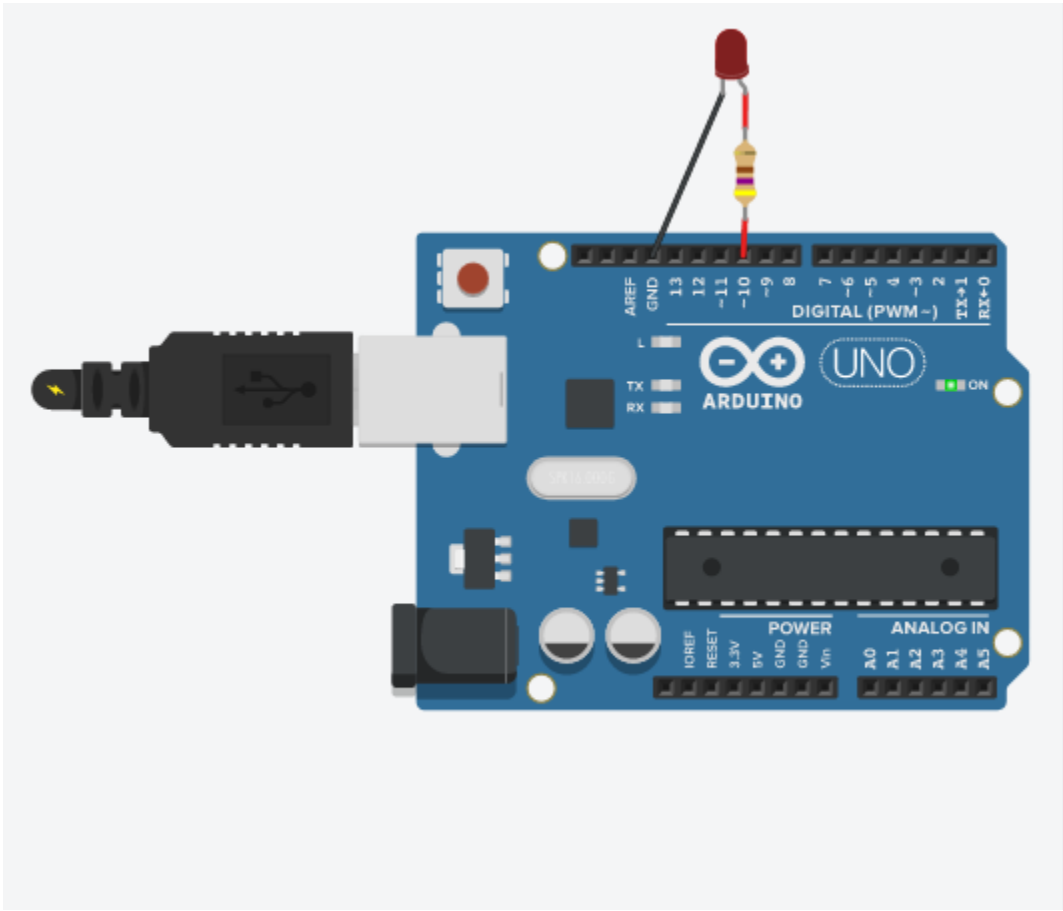
حلقة while

- تستخدم حلقة while لتكرار الكود في الحلقة طالما الشرط يتحقق
- تختبر حلقة while الشرط داخل القوسين (...)
  - اذا كان الشرط صائبا ينفذ الكود في الحلقة ويقيم الشرط مرة أخرى قبل تكرار الحلقة
  - اذا كان الشرط خاطئا يتم انهاء الحلقة
  - تتكرر العملية بشكل مستمر حتى يصبح الشرط خاطئا

```
int var = 0;
while (var <= 200) {
  // do something repetitive 200 times
  var++;
}
```

مثال: تشغيل تدريجي (2)

- برنامج لتشغيل LED تدريجيا باستخدام تقنية PWM




```
1  int led = 10;
2
3  void setup()
4  {
5    pinMode(led, OUTPUT);
6    Serial.begin(9600);
7  }
8
9  void loop()
10 {
11   int i = 0;
12   while (i <= 255) {
13     Serial.println(i);
14     analogWrite(led, i);
15     delay(10);
16     i++;
17   }
18   analogWrite(led, 0);
19   delay(1000);
```

Serial Monitor

```
248
249
250
251
252
253
254
255
```

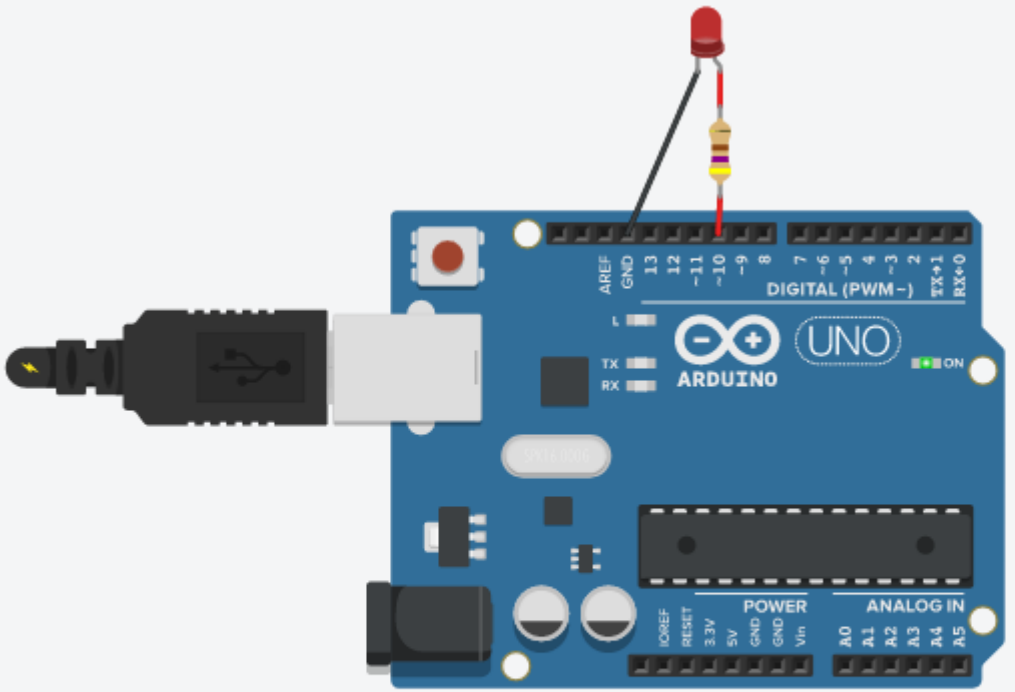
حلقة do...while

- حلقة do...while تشبه حلقة while لكن حلقة do...while تنفذ الكود في الحلقة مرة واحدة ثم تختبر الشرط داخل القوسين (...)
  - اذا كان الشرط صائبا ينفذ الكود في الحلقة ويقيم الشرط مرة أخرى قبل تكرار الحلقة
  - اذا كان الشرط خاطئا يتم انهاء الحلقة
  - تتكرر العملية بشكل مستمر حتى يصبح الشرط خاطئا

```
int var = 0;
do {
  // do something repetitive 200 times
  var++;
} while (var < 200)
```

مثال: تشغيل تدريجي (3)

- برنامج لتشغيل LED تدريجيا باستخدام تقنية PWM



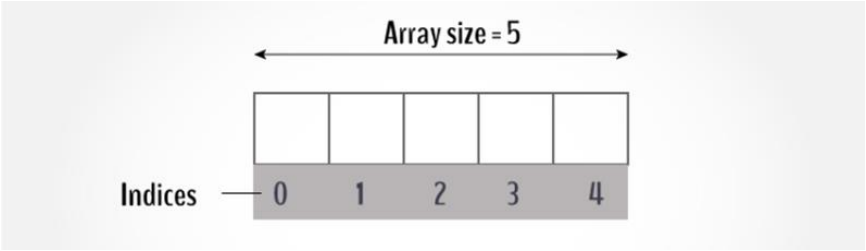

```
1  int led = 10;
2
3  void setup()
4  {
5    pinMode(led, OUTPUT);
6    Serial.begin(9600);
7  }
8
9  void loop()
10 {
11   int i = 0;
12   do {
13     Serial.println(i);
14     analogWrite(led, i);
15     delay(10);
16     i++;
17   } while (i < 255);
18   analogWrite(led, 0);
19   delay(1000);
```

Serial Monitor

```
29
30
31
32
33
34
35
36
```

المصفوفات

- المصفوفة (array) هي متغير لتخزين عدة قيم من نفس نوع البيانات، على سبيل المثال لتخزين 100 رقم يمكن انشاء مصفوفة من الاعداد الصحيحة والوصول لكل عنصر باستخدام الفهرس (index)
- يبدأ العنصر الأول في المصفوفة بالفهرس 0
- اذا كانت المصفوفة حجمها n من العناصر فان الفهرس الأخير هو n-1
- لإنشاء مصفوفة يتم التصريح عنها وتحديد عدد عناصرها ونوع البيانات
- يمكن تعيين عناصر المصفوفة عند التصريح بالمصفوفة



المصفوفات

- انشاء مصفوفة او التصريح عنها

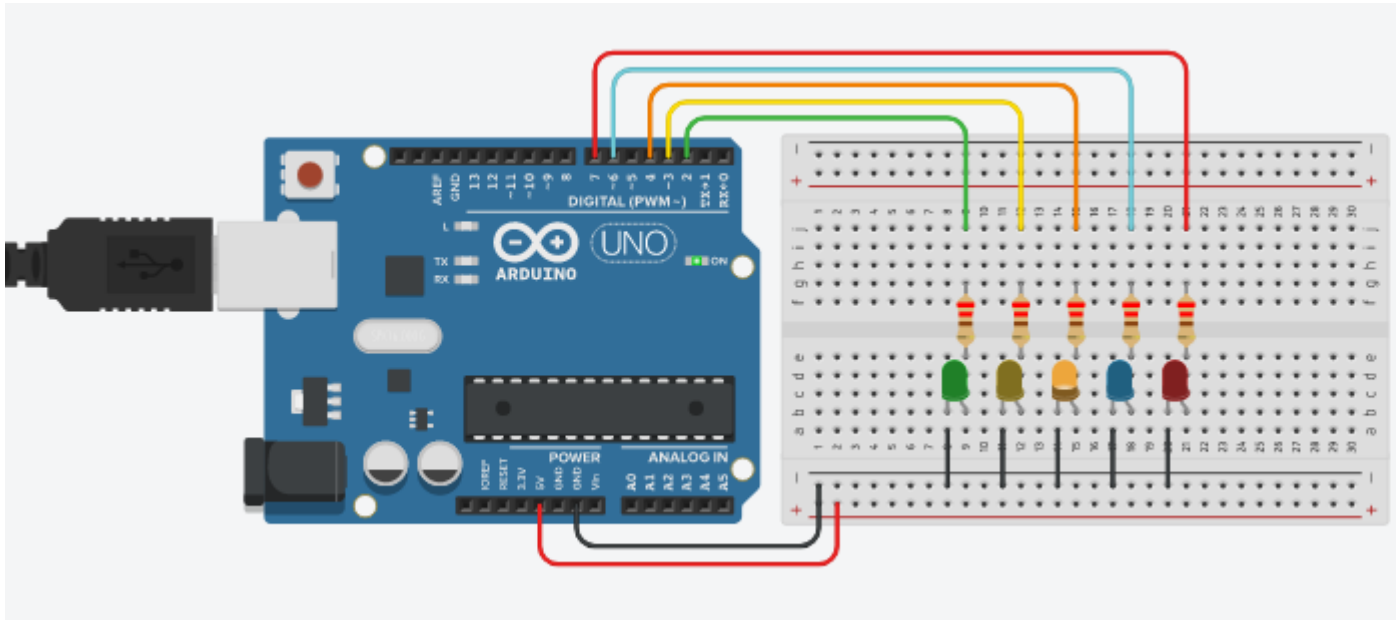
```
int a[6];
int ledPins[] = {2, 4, 8, 3, 6};
int sensorVals[5] = {2, 4, -8, 3, 2};
char message[6] = "hello";
```

- الوصول لعناصر المصفوفة

```
sensorVals[0] = 10;

x = sensorVals[4];

for (int i = 0; i < 5; i = i + 1) {
  Serial.println(ledPins[i]);
}
```

مثال: تشغيل متتالي

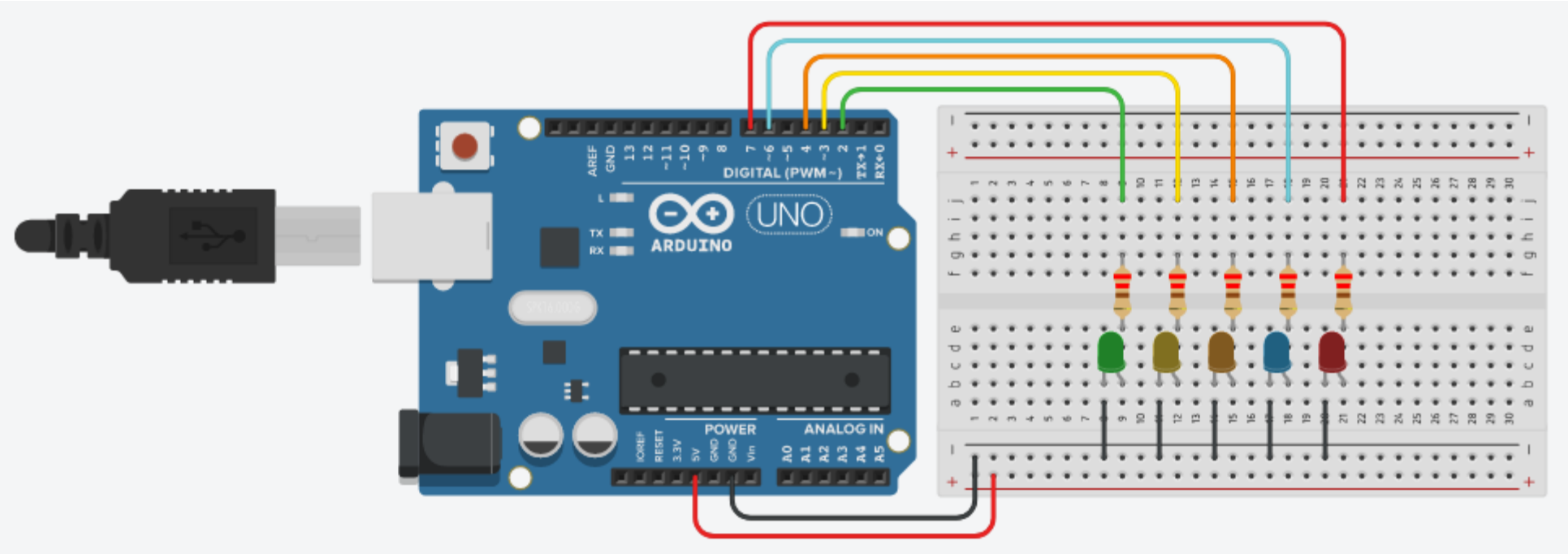
- برنامج لتشغيل واطفاء مجموعة من LEDs بشكل متتالي وبفاصل زمني 1 ثانية




```
int ledPins[5] = {2, 3, 4, 6, 7};

void setup()
{
  for (int i = 0; i < 5; i++) {
    pinMode(ledPins[i], OUTPUT);
  }
}

void loop()
{
  for (int i = 0; i < 5; i++) {
    digitalWrite(ledPins[i], HIGH);
    delay(1000);
    digitalWrite(ledPins[i], LOW);
  }
}
```

تمرين: مصفوفة LED

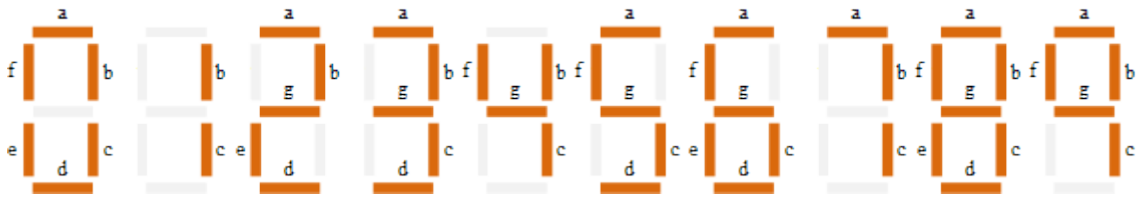
- اكتب برنامج لتشغيل مجموعة من LEDs بشكل متتالي من الاول الى الاخير وبفاصل زمني 1 ثانية ثم اطفاؤها بشكل متتالي من الاخير الى الاول وبفاصل زمني 1 ثانية

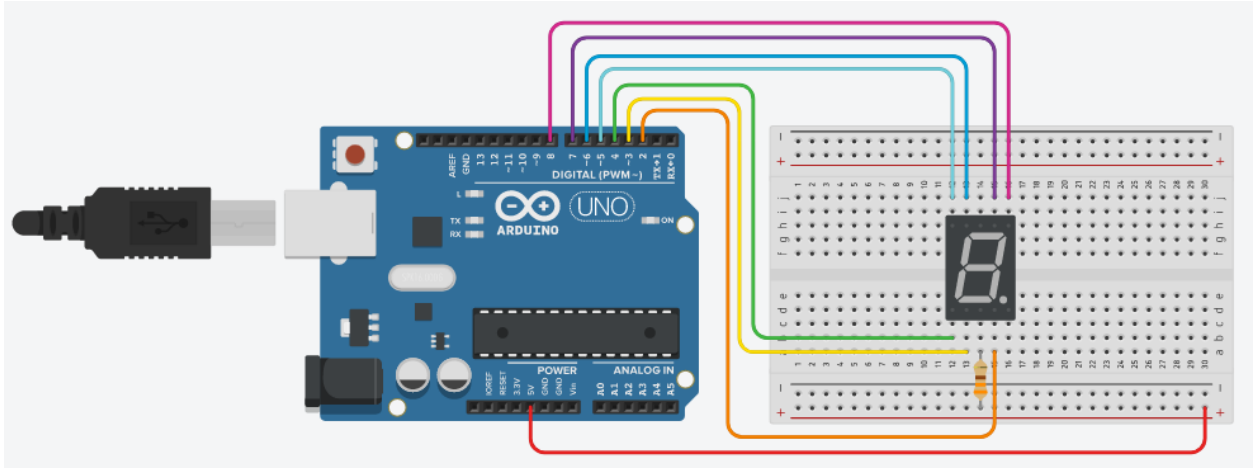


تمرين: عداد رقمي (4)

- اكتب برنامج لعرض الارقام من 0 الى 9 على شاشة العرض 7 Segment باستخدام المصفوفات ودالة لعرض كل رقم على الشاشة مع فاصل زمني مقداره ثانية واحدة؟

	a	b	c	d	e	f	g
0	0	0	0	0	0	0	1
1	1	0	0	1	1	1	1
2	0	0	1	0	0	1	0
3	0	0	0	0	1	1	0
4	1	0	0	1	1	0	0
5	0	1	0	0	1	0	0
6	0	1	0	0	0	0	0
7	0	0	0	1	1	1	1
8	0	0	0	0	0	0	0
9	0	0	0	1	1	0	0



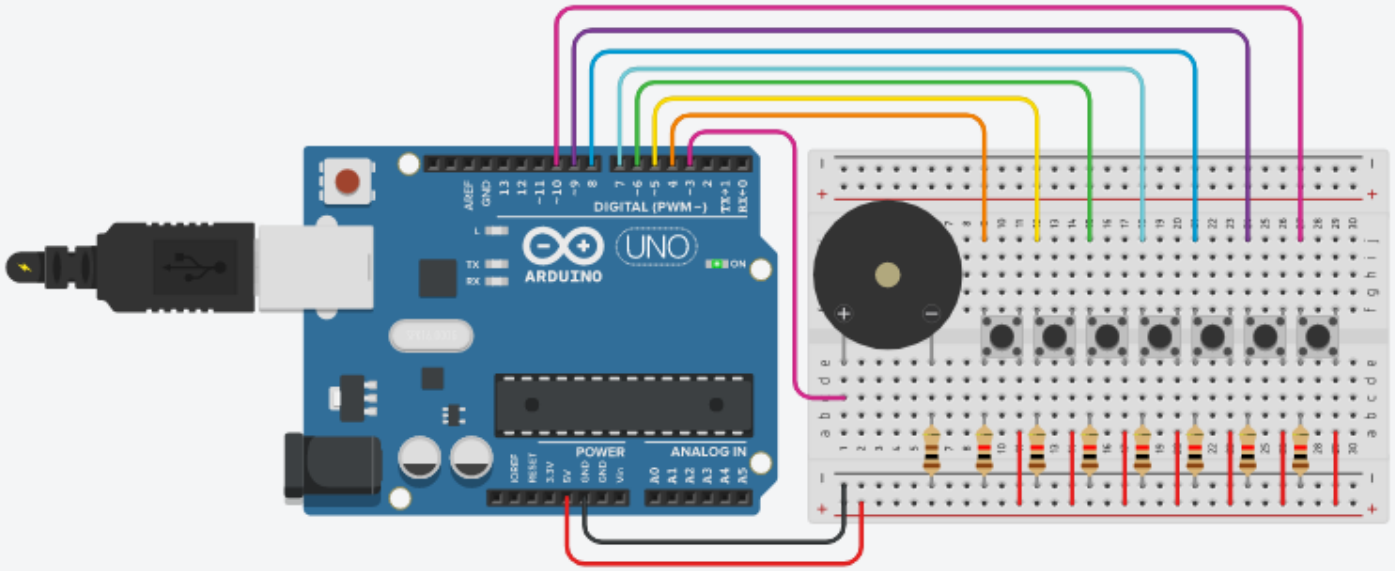


دوال النغمة الصوتية

- يتم وصل طنان كهربائي (piezo buzzer) مع المتحكم الدقيق لاصدار نغمة صوتية
- تستخدم الدالة ()tone في توليد موجة مربعة على المخرج بتردد معين ولمدة زمنية محددة
- تستخدم الدالة ()noTone في ايقاف توليد الموجة المربعة على المخرج
- يمكن اصدار اكثر من نغمة صوتية في نفس الوقت بناء على عدد وحدات التوقيت (timer) على المتحكم الدقيق، على سبيل المثال Arduino UNO يحتوي على 3 وحدات توقيت
- لن يتم اصدار نغمة ترددها اقل من 31 هرتز، واعلى تردد يمكن سماعه هو 20 كيلوهرتز

مثال: بيانو بسيط

- برنامج يقوم بإصدار نغمات مختلفة بالضغط على الازرار من خلال الطنان (Buzzer)




```
void setup()
{
  pinMode(3, OUTPUT);
  for (int i = 4; i <= 10; i++) {
    pinMode(i, INPUT);
  }
  tone(3, 200, 500);
  delay(1000);
}
void loop()
{
  while (digitalRead(4) == HIGH) {
    tone(3, 262); // Do(C)
  }
  while (digitalRead(5) == HIGH) {
    tone(3, 294); // Re(D)
  }
  while (digitalRead(6) == HIGH) {
    tone(3, 330); // Mi(E)
  }
  while (digitalRead(7) == HIGH) {
    tone(3, 349); // Fa(F)
  }
  while (digitalRead(8) == HIGH) {
    tone(3, 392); // Sol(G)
  }
    while (digitalRead(9) == HIGH) {
    tone(3, 440); // La(A)
  }
  while (digitalRead(10) == HIGH) {
    tone(3, 494); // Si(B)
  }
  noTone(3);
}
```