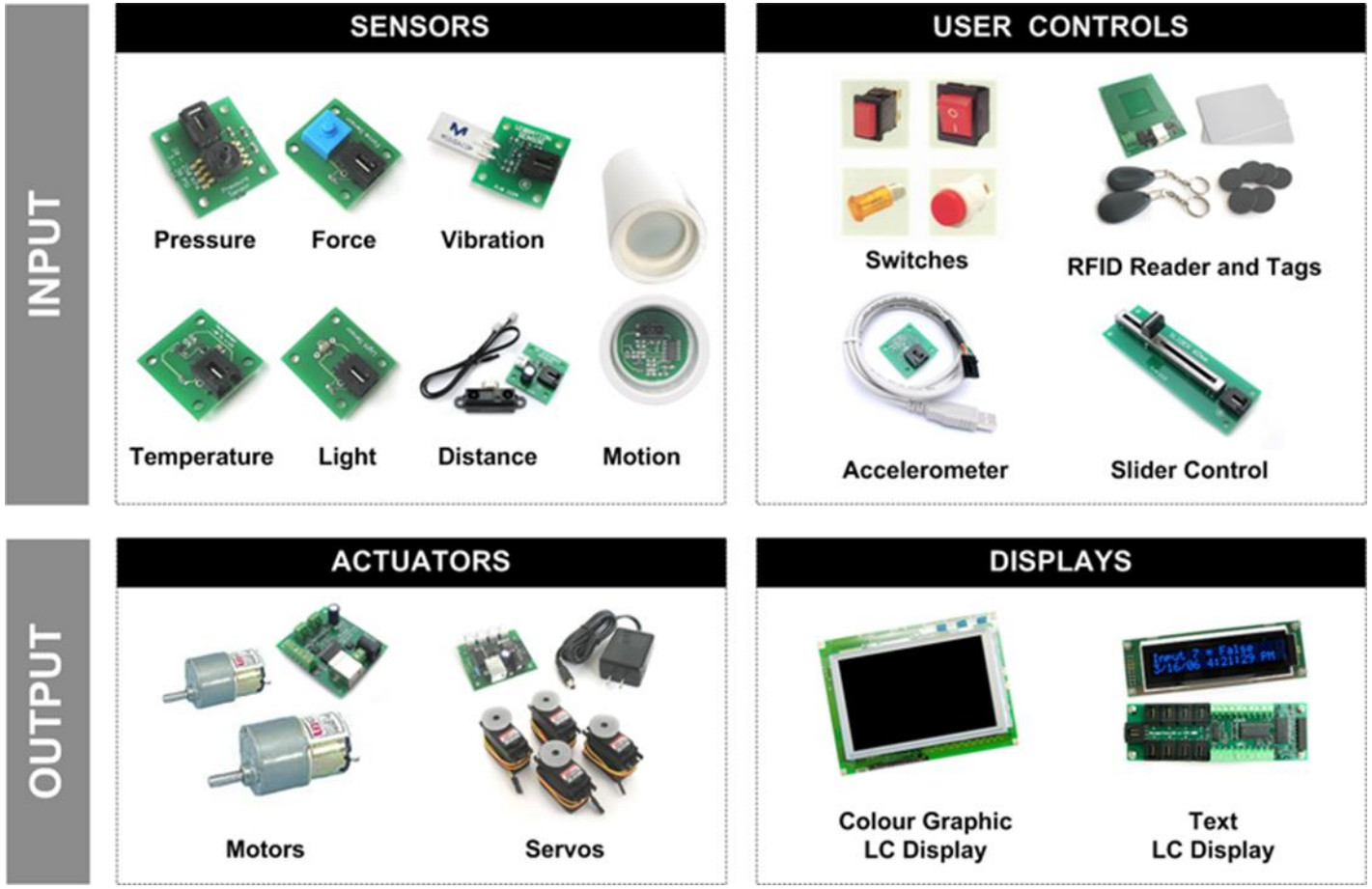
وحدات الادخال والاخراج

المتحكمات الدقيقة

م. غنام الجعبري

ما هي وحدات الادخال والإخراج؟

- وحدات الادخال والإخراج (I/O devices) هي اجهزة يتم وصلها بالمتحكم الدقيق وتضم:
  - المستشعرات (sensors)
  - المشغلات (actuators)
  - وحدات التحكم (controls)
  - وحدات العرض (displays)

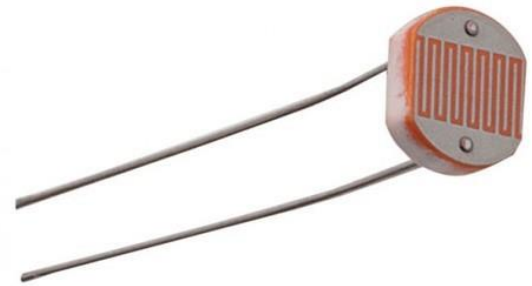


المستشعرات

- تستخدم أجهزة الاستشعار او المستشعرات في رصد الأحداث والتغييرات في البيئة المحيطة مثل:
  - الحرارة (temperature)
  - الضغط (pressure)
  - الضوء (light)
  - الحركة (motion)
  - المسافة (distance)
  - الاهتزاز (vibration)
  - القوة (force)
- تستخدم المستشعرات في ادخال البيانات الى المتحكم الدقيق او لوحة اردوينو

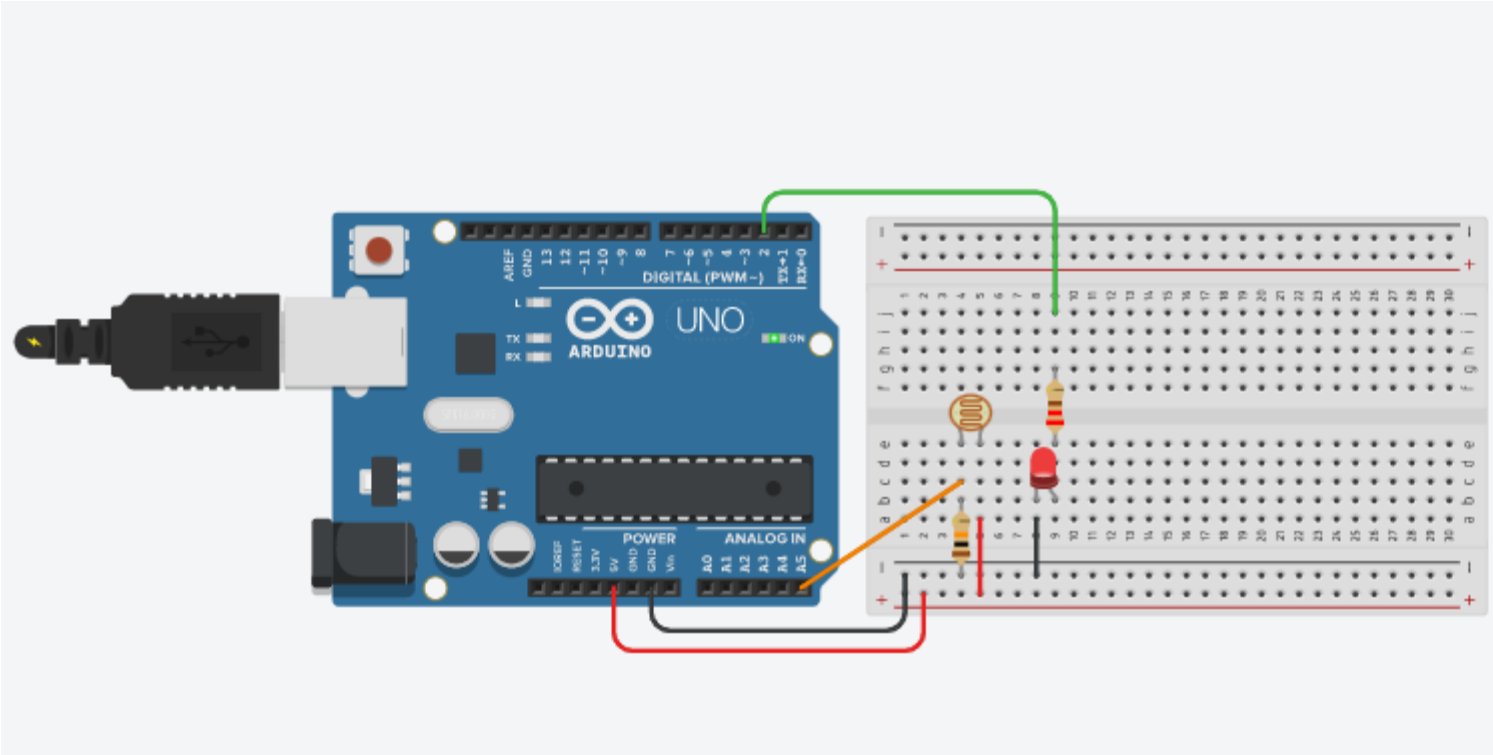
مستشعر الضوء

- مستشعر الضوء (Photoresister) او (LDR) هو مقاومة كهربائية حساسة للضوء تستخدم في قياس مستوى الاضاءة حيث تقل قيمة المقاومة عندما تزداد شدة الضوء



مثال: مستشعر الضوء

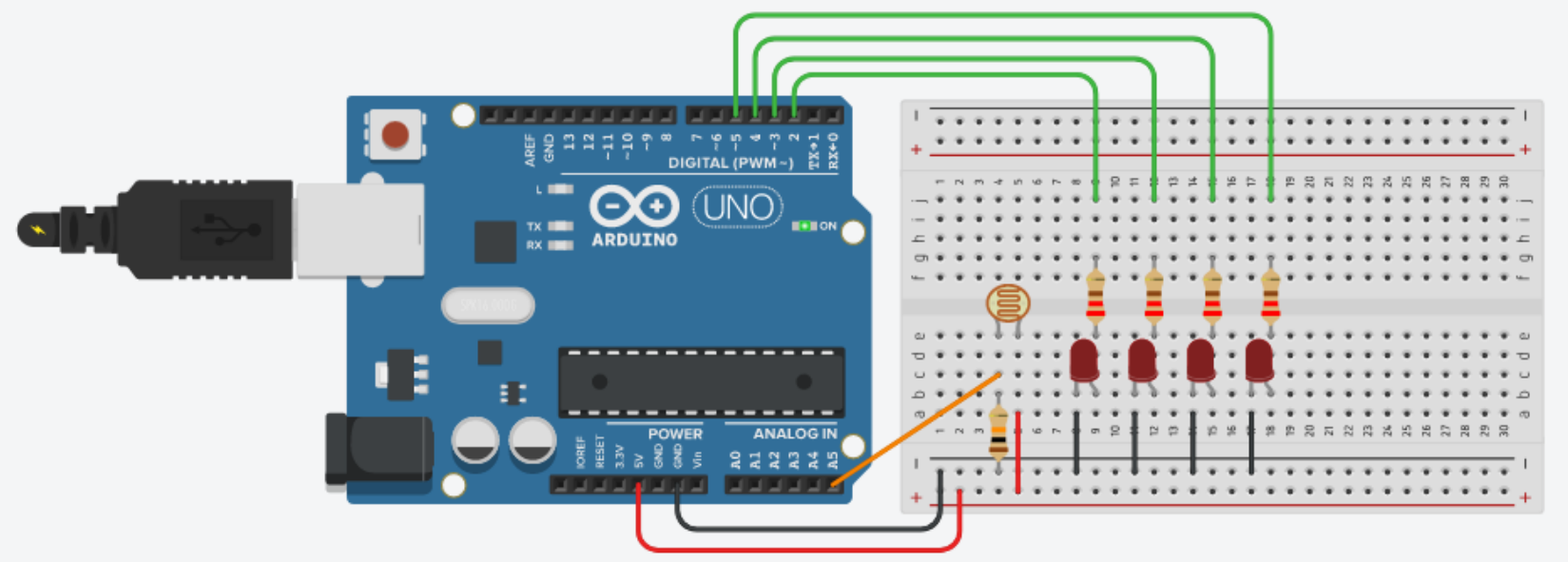
- برنامج لقراءة شدة الإضاءة وتشغيل الضوء الأحمر اذا كانت القيمة اقل من 440




```
int ledPin = 2;
int ldrPin = 5;
void setup()
{
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  int ldrVal;
  ldrVal = analogRead(ldrPin);
  Serial.println(ldrVal);
  if (ldrVal <= 440)
  {
    digitalWrite(ledPin, HIGH);
  }
  else
  {
    digitalWrite(ledPin, LOW);
  }
  delay(1000);
}
```

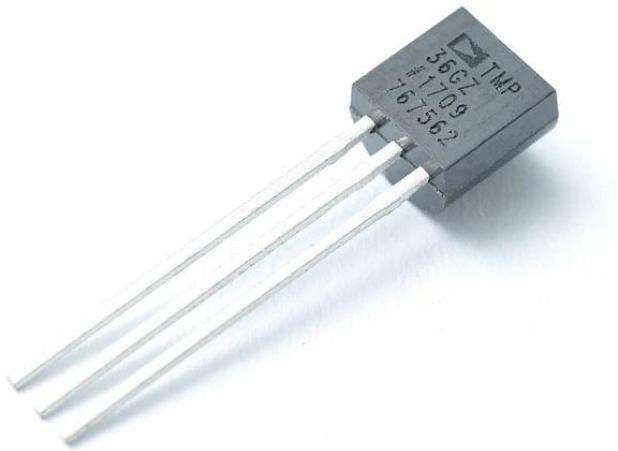
تمرين: مستوى الاضاءة

- اكتب برنامج لقراءة شدة الاضاءة وتشغيل الدايودات (LEDs) وفق الشروط التالية:
  - تشغيل اربعة دايودات اذا كانت شدة الاضاءة اكبر من 900
  - تشغيل ثلاثة دايودات اذا كانت شدة الاضاءة اكبر من 800
  - تشغيل دايودان اذا كانت شدة الاضاءة اكبر من 700
  - تشغيل دايود واحد فقط اذا كانت شدة الاضاءة اكبر من 600



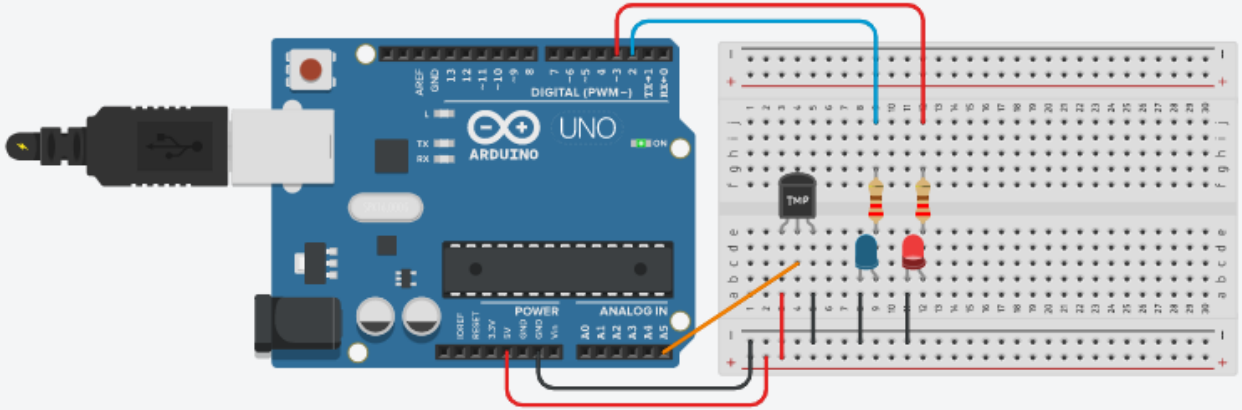
مستشعر الحرارة

- مستشعر الحرارة (TMP36) يعتمد على الدايودات (diodes) في قياس درجة الحرارة بمعدل ثابت أي وفق معادلة خطية، على عكس المقاومة الحساسة للحرارة (thermistor)
- يستخدم مستشعر TMP36 في قياس درجة الحرارة من 50°C- الى 150°C



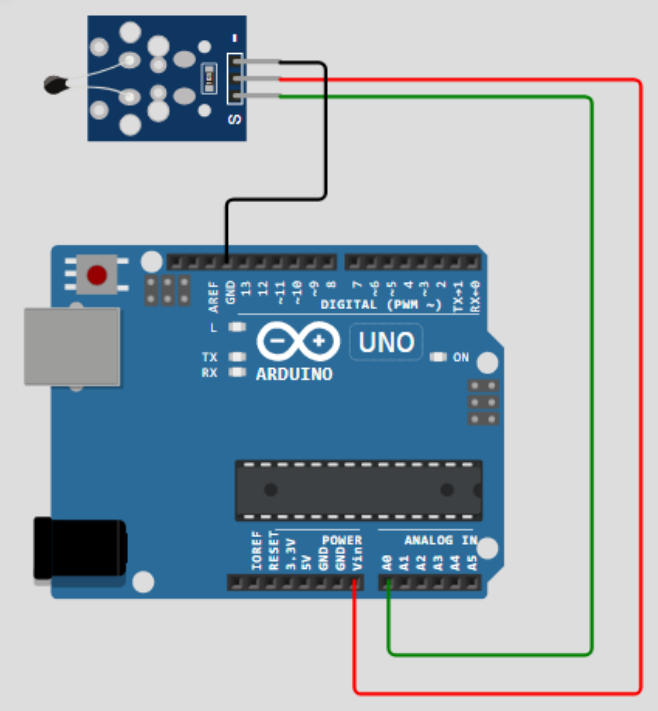
مثال: مستشعر الحرارة

- برنامج لقراءة درجة الحرارة ثم تشغيل الضوء الأزرق اذا كانت درجة الحرارة اقل من 20 درجة مئوية وتشغيل الضوء الاحمر اذا كانت درجة الحرارة اكبر من 20 درجة مئوية




```
const int bluePin = 2;
const int redPin = 3;
const int tmpPin = 5;
int tmpValue;

void setup()
{
  pinMode(bluePin, OUTPUT);
  pinMode(redPin, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  tmpValue = analogRead(tmpPin);
  // convert the reading to volts (10mv per degree)
  float volt = (tmpValue / 1023.0) * 5;
  // convert the voltage to temperature
  float temp = (volt - 0.5) * 100;
  Serial.println(temp);
  if (temp < 20.0) {
    digitalWrite(bluePin, HIGH);
    digitalWrite(redPin, LOW);
  }
  else {
    digitalWrite(bluePin, LOW);
    digitalWrite(redPin, HIGH);
  }
  delay(100);
}
```

مثال: مقاومة حساسة للحرارة

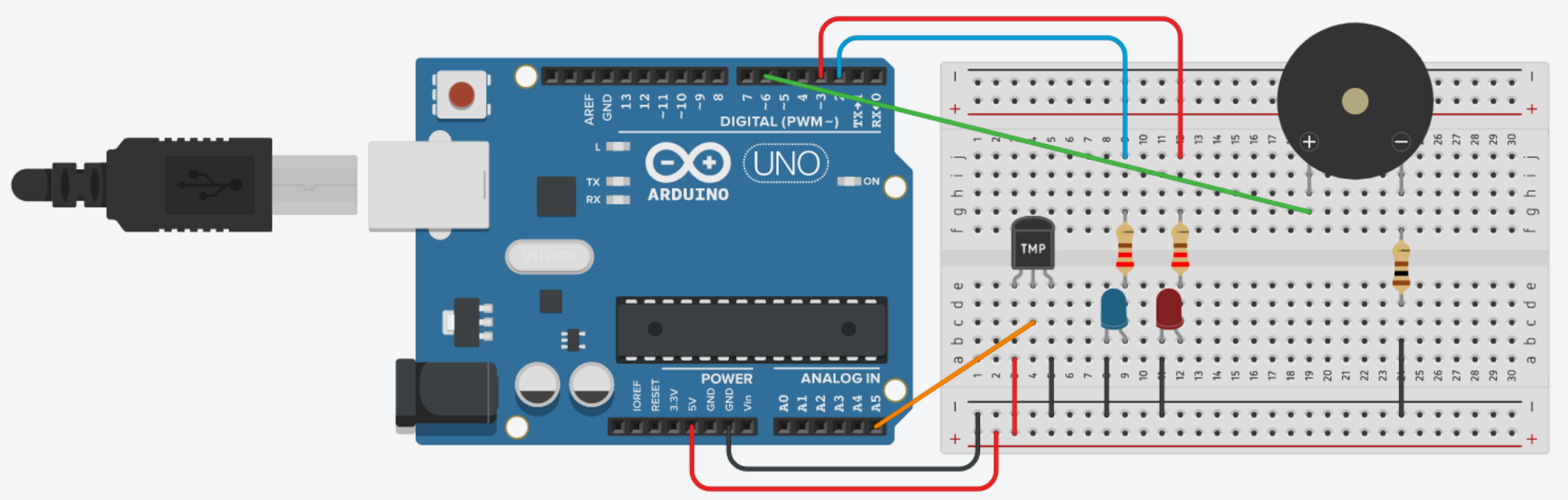



```
1   const float BETA = 3950; // should match the Beta Coefficient of the thermistor
2
3   void setup() {
4     Serial.begin(9600);
5   }
6
7   void loop() {
8     int analogValue = analogRead(A0);
9     float celsius = 1 / (log(1 / (1023. / analogValue - 1)) / BETA + 1.0 / 298.15) - 273.15;
10    Serial.print("Temperature: ");
11    Serial.print(celsius);
12    Serial.println(" °C");
13    delay(1000);
14  }
```

[Wokwi Arduino and ESP32 Simulator]

تمرين: تنبيه الحرارة

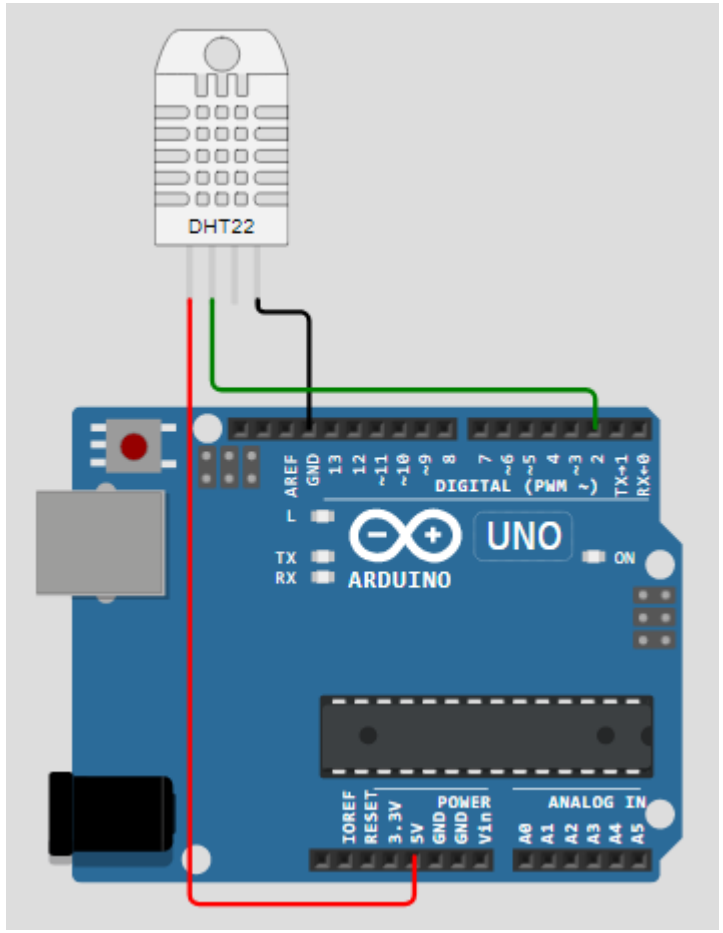
- اكتب برنامج لقراءة درجة الحرارة ثم تشغيل الضوء الأزرق اذا كانت درجة الحرارة اقل من 20 درجة مئوية وتشغيل الضوء الاحمر اذا كانت درجة الحرارة اكبر من 20 درجة مئوية، واصدار تنبيه صوتي اذا كانت درجة الحرارة اكبر من 30 او اقل من 15



مستشعر الحرارة والرطوبة

- يستخدم مستشعر (DHT) في قياس درجة الحرارة والرطوبة النسبية (humidity)
- هنالك نوعان من مستشعر الحرارة والرطوبة النسبية:
  - مستشعر DHT11 لقياس درجة الحرارة من 0-50°C والرطوبة النسبية من 20-80%
  - مستشعر DHT22 لقياس درجة الحرارة من 40-80°C- والرطوبة النسبية من 0-100%
- يتطلب تشغيل مستشعر DHT من خلال لوحة اردوينو تنزيل مكتبة برمجية تدعى DHT.h
- تستخدم دالة dht.readTemperature() في قراءة درجة الحرارة من المستشعر
- تستخدم دالة dht.readHumidity() في قراءة الرطوبة النسبية من المستشعر

مثال: مستشعر الحرارة والرطوبة

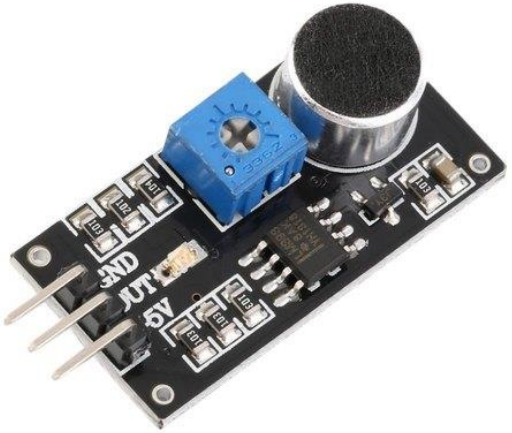


[Wokwi Arduino and ESP32 Simulator]

```
#include "DHT.h"

#define DHTPIN 2
#define DHTTYPE DHT22   // DHT 22  (AM2302), AM2321

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(115200);
  dht.begin();
}

void loop() {
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();
  // Check if any reads failed and exit early (to try again).
  if (isnan(temperature) || isnan(humidity)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }
  Serial.print(F("Humidity: "));
  Serial.print(humidity);
  Serial.print(F("%  Temperature: "));
  Serial.print(temperature);
  Serial.println(F("°C "));
  // Wait a few seconds between measurements.
  delay(2000);
}
```

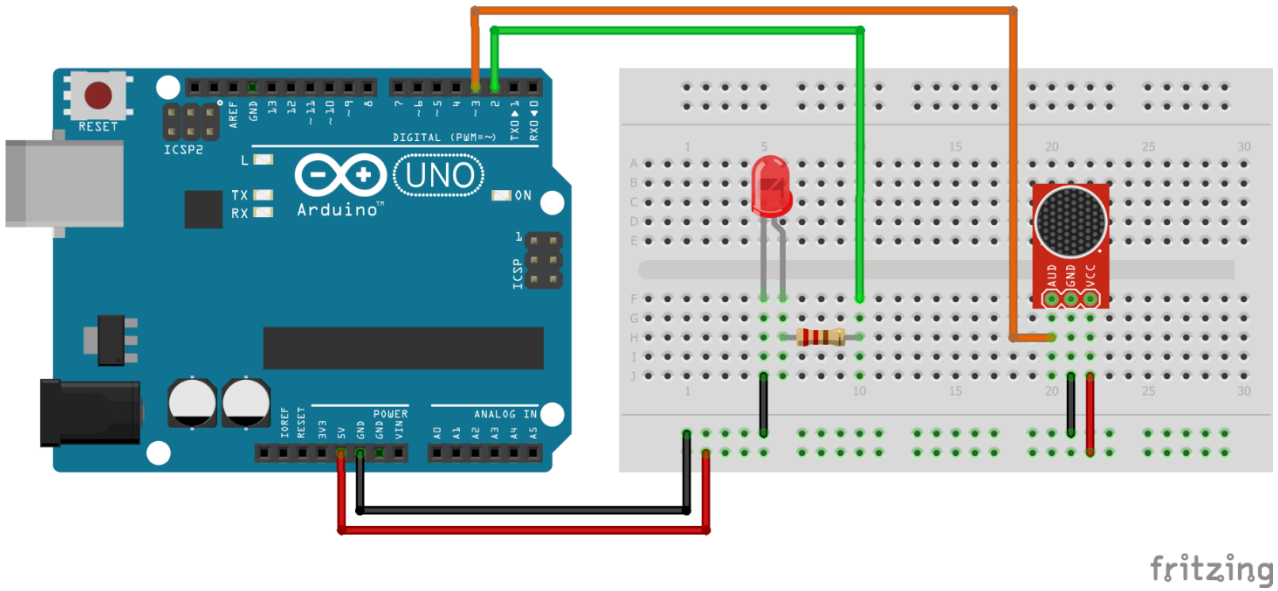
مستشعر الصوت

- مستشعر الصوت (Sound Detector) يعتمد على لاقط صوتي (microphone) في رصد الصوت ويستخدم في قياس شدة الصوت
- يمكن تعديل حساسية اللاقط (sensitivity) باستخدام مقاومة متغيرة (potentiometer)
- عندما يتجاوز الصوت حد معين يضيء LED على المستشعر للدلالة على ان هنالك صوت



مثال: مستشعر الصوت

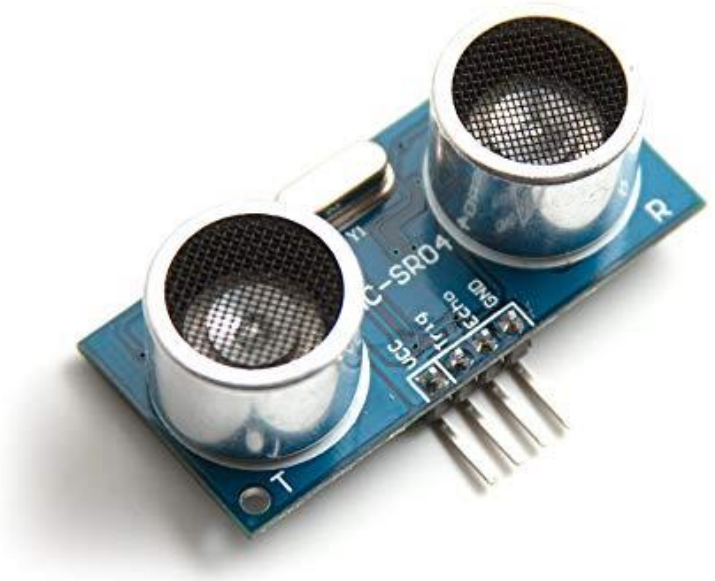
- برنامج لرصد الصوت وتشغيل LED عند التصفيق




```
int ledPin = 2;
int soundPin = 3;
unsigned long lastEvent = 0;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(soundPin, INPUT);
}

void loop() {
  int soundVal = digitalRead(soundPin);
  if (soundVal == LOW) {
    // if 25ms have passed since last LOW state
    // it means sound is detected
    if (millis() - lastEvent > 25) {
      digitalWrite(ledPin, HIGH);
    }
    else {
      digitalWrite(ledPin, LOW);
    }
    lastEvent = millis();
  }
}
```

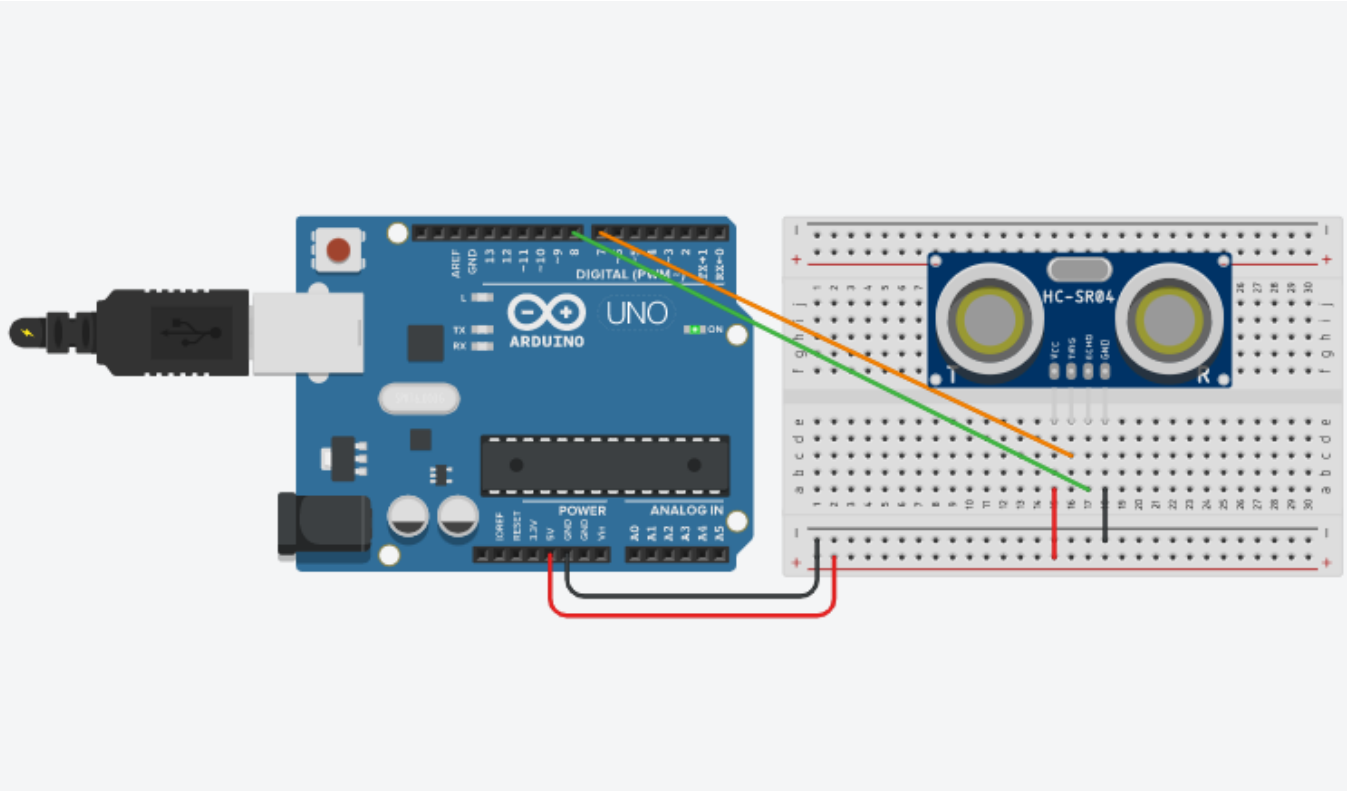
مستشعر المسافة

- مستشعر المسافة (Ultrasonic) يعتمد على الموجات فوق الصوتية وصداها في حساب المسافة ويستخدم في قياس المسافة بين المستشعر والاجسام التي لها سطح صلب
- مستشعر المسافة لا يكتشف الاجسام التي تبعد اكثر من 4 متر او اقل من 2 سم



مثال: مستشعر المسافة

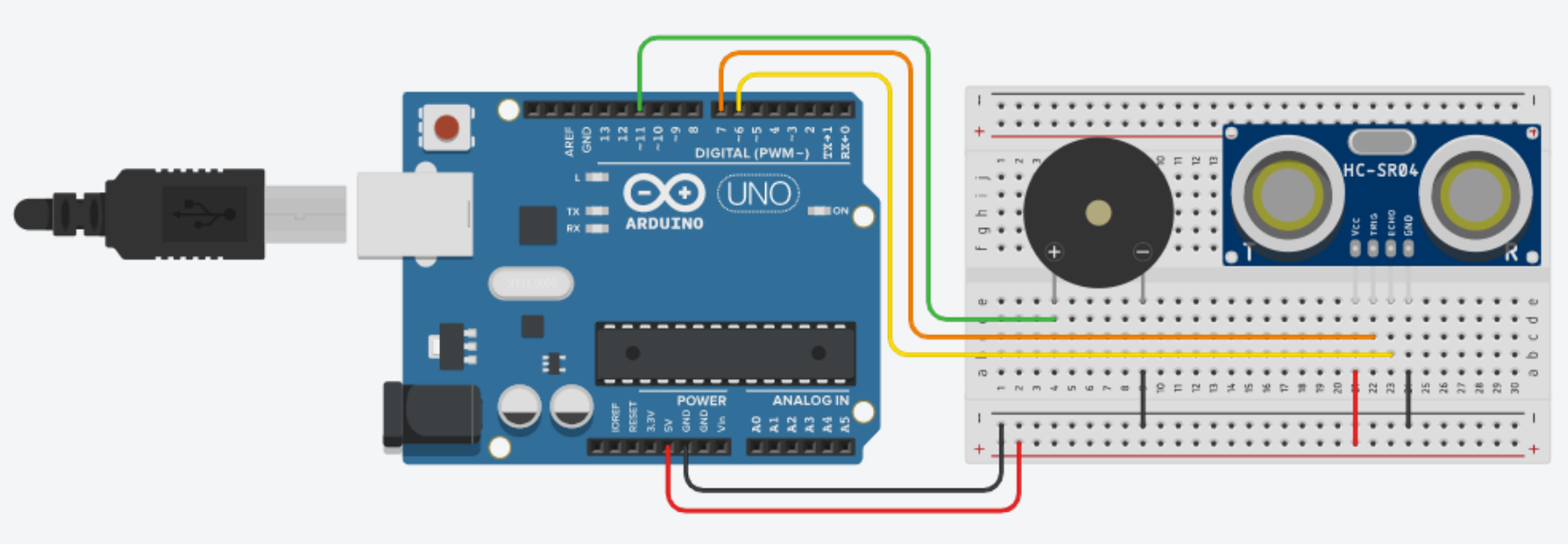
- برنامج لقياس المسافة الى اقرب جسم صلب ثم طباعة النتيجة على المنفذ التسلسلي




```
const int echoPin = 8;
const int trigPin = 7;
long duration, distance;

void setup()
{
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}

void loop()
{
  // clean the pulse
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(5);
  digitalWrite(trigPin, LOW);
  // get the time that it takes for the pulse to return (ms)
  duration = pulseIn(echoPin, HIGH) / 2;
  // convert to centimeters by using this formula
  distance = duration / 29;
  Serial.print(distance);
  Serial.println(" cm");
  delay(500);
}
```

تمرين: تنبيه المسافة

- اكتب برنامج لتحذير السائق عند رجوع السيارة الى الخلف واصدار تنبيه صوتي عند الاقتراب من الاجسام الصلبة اعتمادا على المسافات التالية:
  - اذا كانت المسافة اقل من 80 سم اصدار تنبيه صوتي بتردد 165 هرتز ولمدة نصف ثانية
  - اذا كانت المسافة اقل من 50 سم اصدار تنبيه صوتي بتردد 330 هرتز ولمدة نصف ثانية
  - اذا كانت المسافة اقل من 30 سم اصدار تنبيه صوتي بتردد 659 هرتز ولمدة نصف ثانية

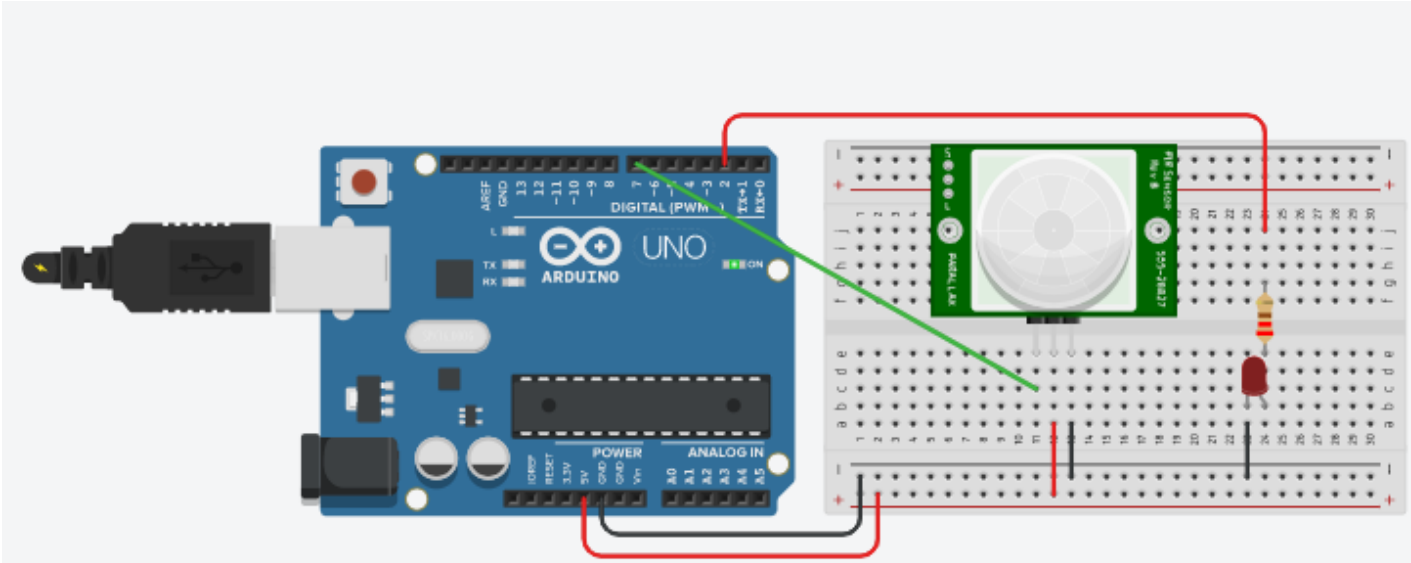


مستشعر الحركة

- مستشعر الحركة (PIR) يعتمد على الاشعة تحت الحمراء (IR) وارتدادها في رصد الحركة ويستخدم في رصد حركة الأشخاص والحيوانات أي الاجسام التي تولد حرارة
- يمكن تعديل حساسية الرصد (sensitivity) باستخدام مقياس الجهد (potentiometer) على اليمين، وتعديل مدة بقاء الاشارة على المخرج (timeout) باستخدام مقياس الجهد على اليسار

مثال: مستشعر الحركة

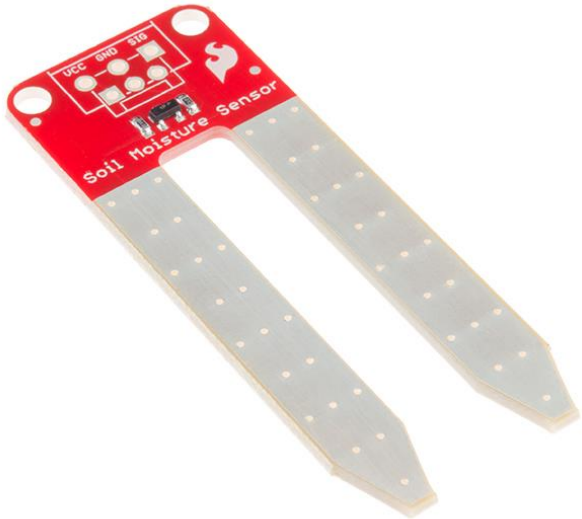
- برنامج لرصد الحركة وتشغيل LED عند رصد حركة شخص ما




```
const int ledPin = 2;
const int detectPin = 7;
int sensor;

void setup()
{
  pinMode(ledPin, OUTPUT);
  pinMode(detectPin, INPUT);
}

void loop()
{
  sensor = digitalRead(detectPin);
  // if sensor detected movement, turn the LED on
  if (sensor == HIGH) {
    digitalWrite(ledPin, HIGH);
    delay(500);
  }
  else {
    digitalWrite(ledPin, LOW);
    delay(500);
  }
  delay(100);
}
```

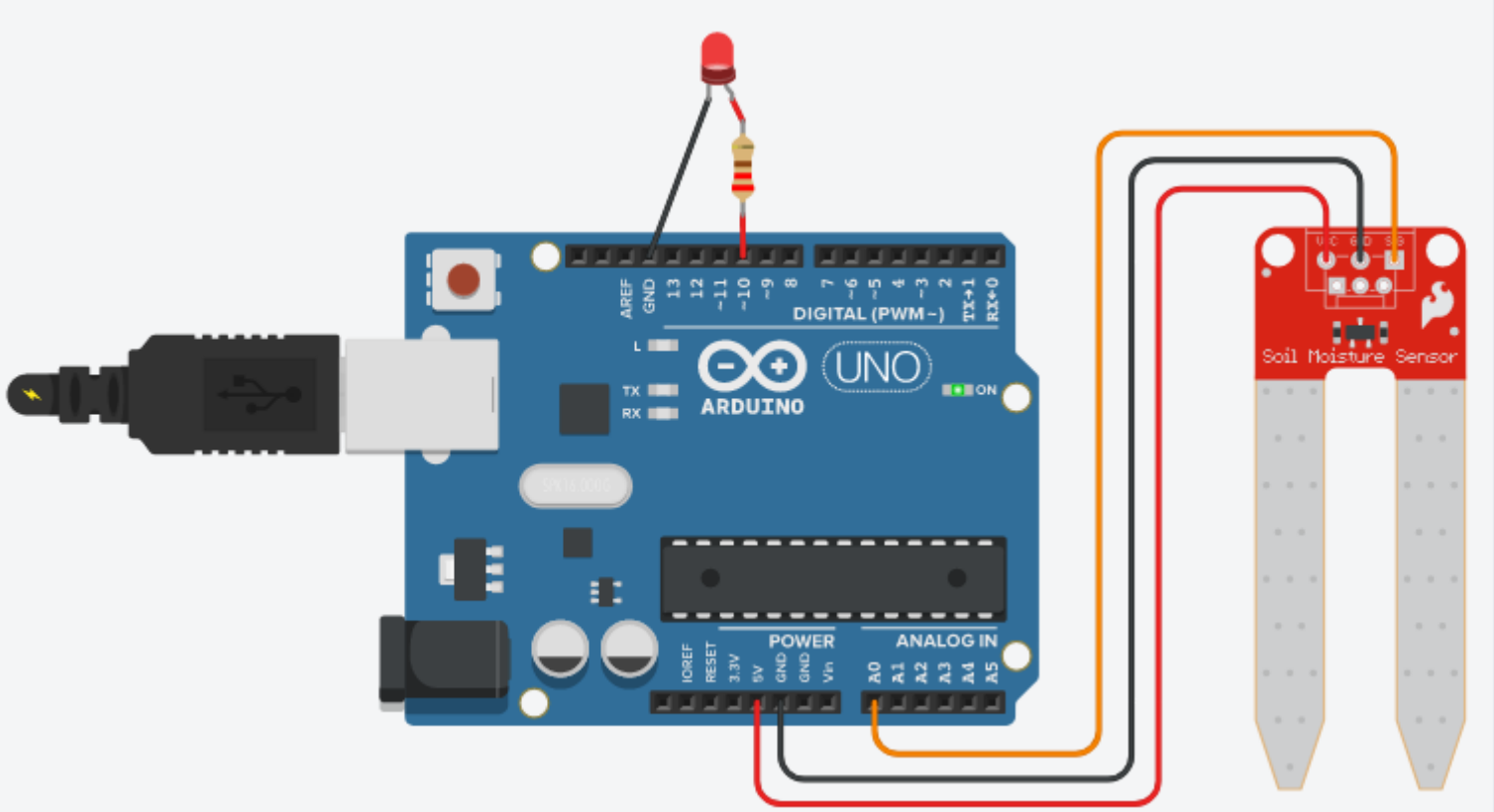
مستشعر رطوبة التربة

- يستخدم مستشعر رطوبة التربة (Soil Moisture) في قياس نسبة الرطوبة في التربة
- يأتي مستشعر رطوبة التربة على شكل شوكة مع موصلين مكشوفين ويعمل مثل مقاومة متغيرة حيث تقل قيمة المقاومة عندما تزداد نسبة الرطوبة في التربة



مثال: مستشعر رطوبة التربة

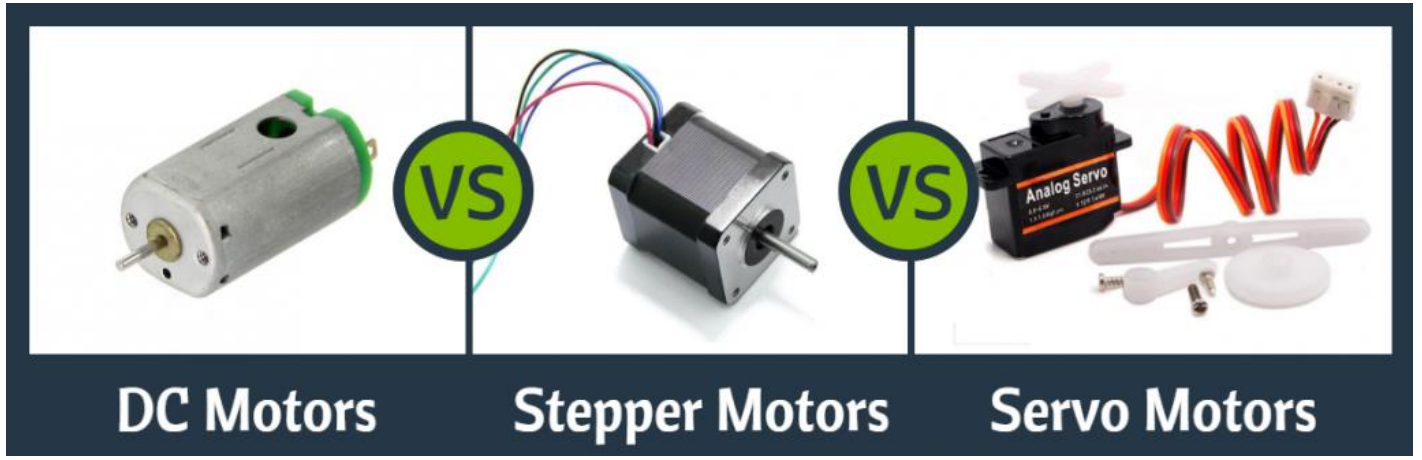
- برنامج لقياس نسبة الرطوبة في التربة وتشغيل LED اذا كانت اقل من 300




```
int ledPin = 10;
int soilPin = 0;

void setup()
{
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  int soilVal = analogRead(soilPin);
  Serial.println(soilVal);
  if (soilVal < 300) {
    digitalWrite(ledPin, HIGH);
  }
  else {
    digitalWrite(ledPin, LOW);
  }
  delay(100);
}
```

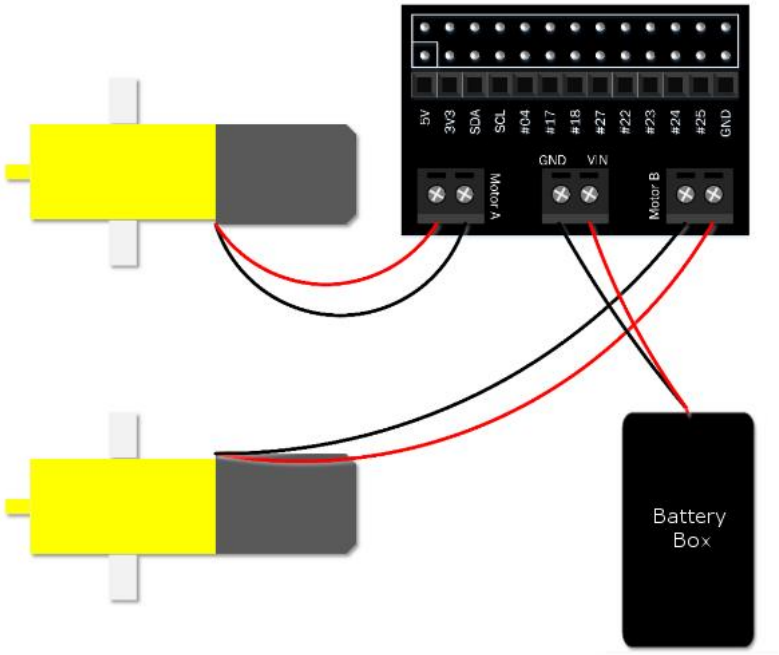
المشغلات

- المشغلات هي وحدات اخراج تستخدم في التحكم بالمحركات الكهربائية غالبا
- هنالك ثلاثة انواع من المحركات الكهربائية:
  - محرك التيار المستمر (DC Motor) يستخدم في التحكم بسرعة الدوران مثل عجلات السيارة
  - محرك الزاوية (Servo Motor) يستخدم في التحكم بزاوية الدوران مثل ذراع الروبوت
  - محرك الخطوة (Stepper Motor) يستخدم في التحكم بدرجات الدوران مثل الماسح الضوئي



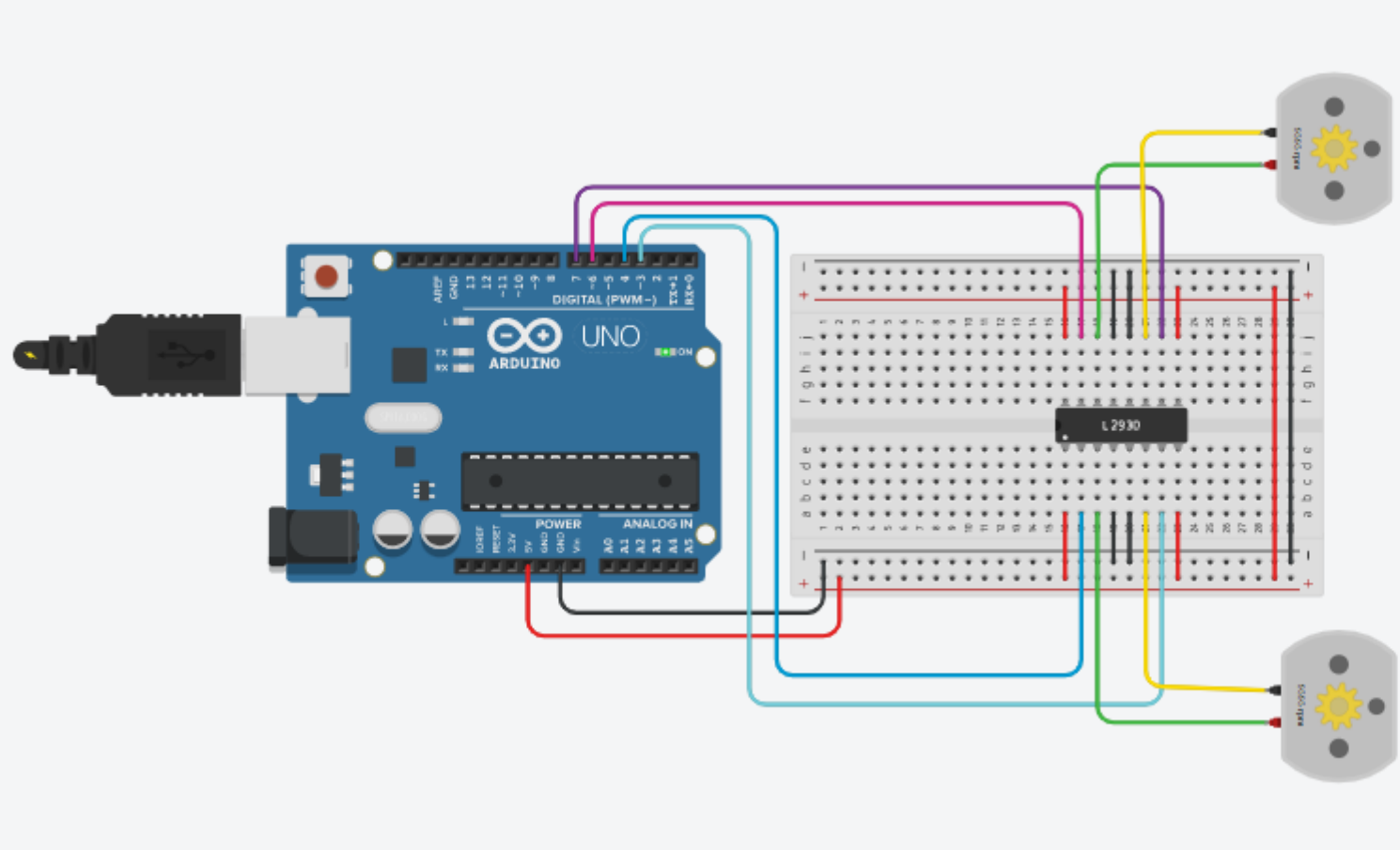
محرك التيار المستمر

- تستخدم محركات التيار المستمر (DC) في التحكم بسرعة الدوران الى الامام او الخلف
- يمكن تشغيل محركات التيار المستمر والتحكم بسرعتها وتغيير اتجاه دورانها باستخدام دائرة الكترونية تدعى H-bridge مثل دائرة L293D ودائرة L298N



مثال: محرك التيار المستمر

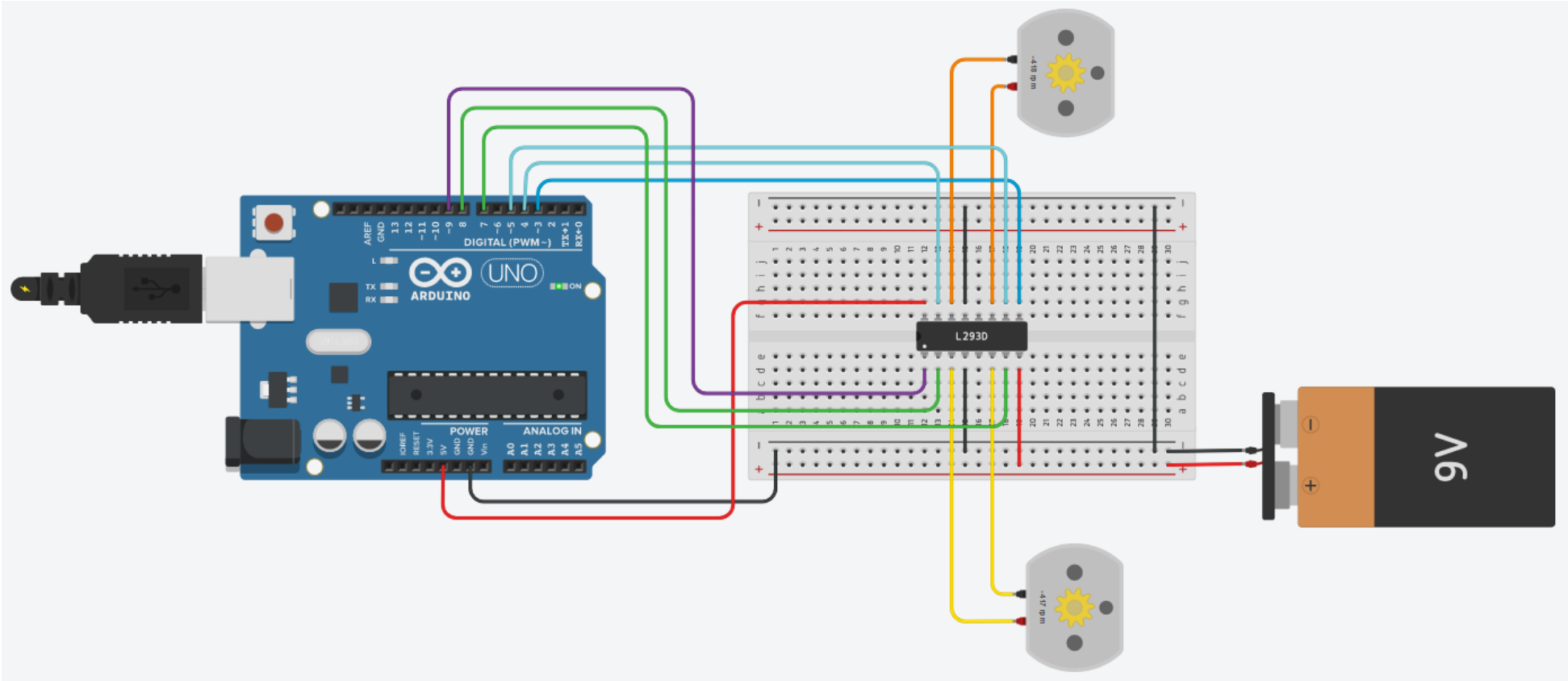
- برنامج للتحكم باتجاه الدوران في المحرك الكهربائي (DC)




```
int in1 = 4, in2 = 3;
int in3 = 7, in4 = 6;

void setup()
{
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
}

void loop()
{
  // motor 1 forward
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  // motor 2 forward
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  delay(5000);
  // motor 1 backward
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  // motor 2 backward
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
  delay(5000);
}
```

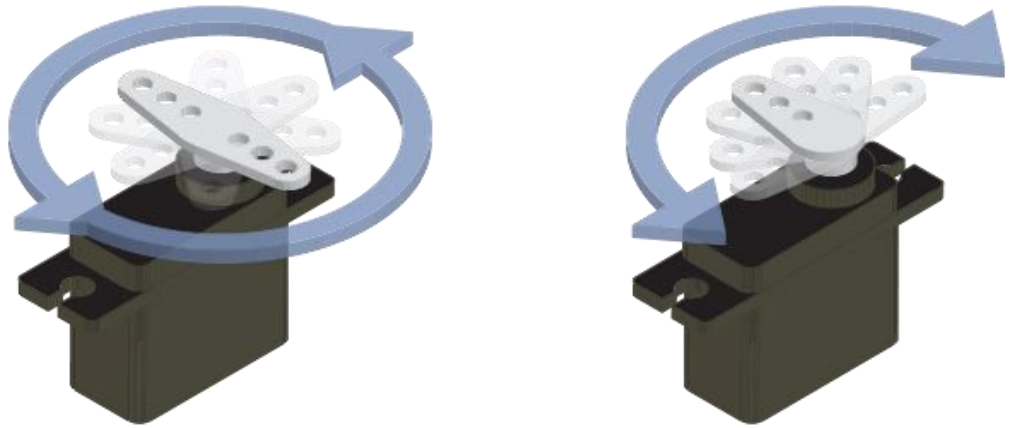
مثال: عجلات السيارة

- اكتب برنامج لتحريك عجلات السيارة الى الامام او الى الخلف وتغيير سرعة السيارة



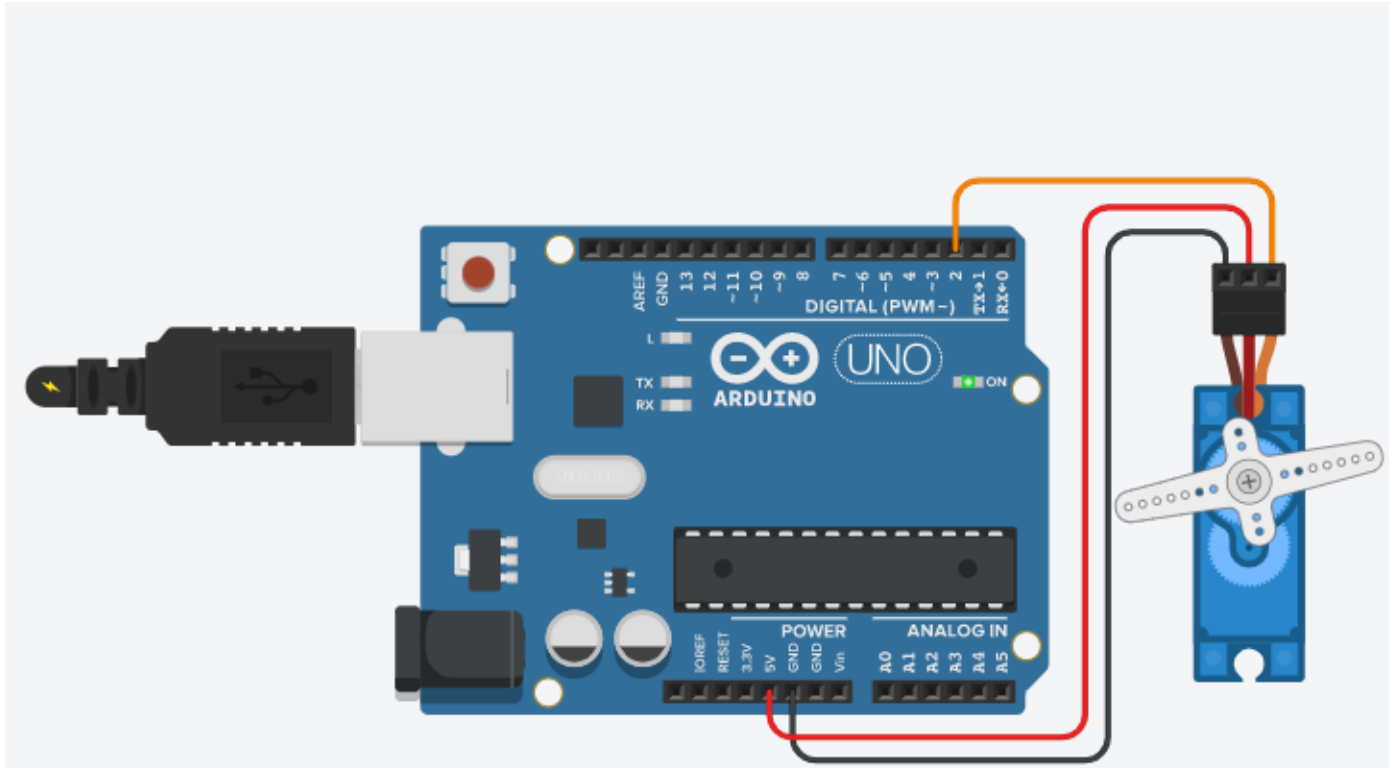
محرك الزاوية

- تستخدم محركات الزاوية (Servo) في التحكم بزاوية الدوران مثل ذراع الروبوت
- محركات الزاوية تتحرك بزاوية دوران 0-180 (نصف دورة) او 0-360 (دورة كاملة)



مثال: محرك الزاوية

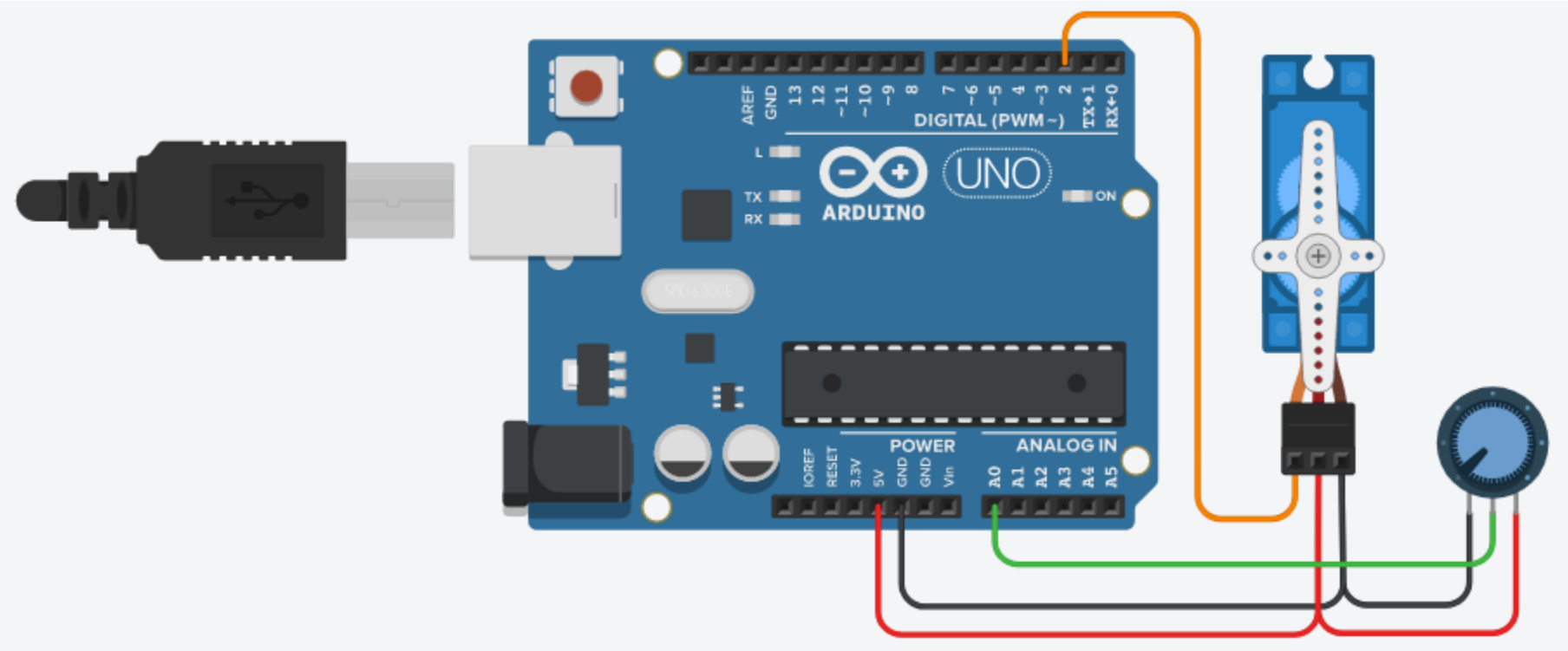
- برنامج للتحكم بزاوية الدوران في المحرك الكهربائي (Servo)




```
#include <Servo.h>
Servo servo;
const int servoPin = 2;
int pos;

void setup()
{
  servo.attach(servoPin);
}

void loop()
{
  // goto 90 degree
  servo.write(90);
  delay(1000);
  // goto 0 degree
  servo.write(0);
  delay(1000);
  // rotate form 0 to 180 degree
  for (pos = 0; pos <= 180; pos++) {
    servo.write(pos);
    delay(15);
  }
  // rotate from 180 to 0 degree
  for (pos = 180; pos >= 0; pos--) {
    servo.write(pos);
    delay(15);
  }
}
```

تمرين: ذراع الروبوت

- اكتب برنامج للتحكم بزاوية الدوران في ذراع الروبوت باستخدام مقاومة متغيرة

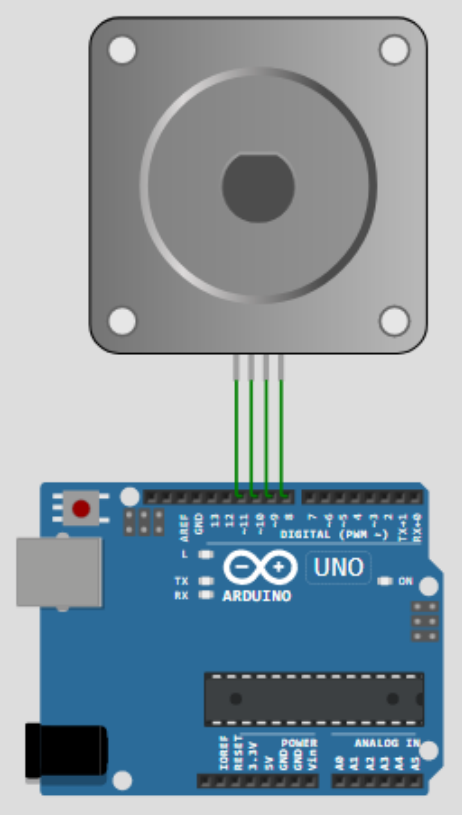


محرك الخطوة

- تستخدم محركات الخطوة (Stepper) في التحكم بدرجات الدوران او عدد الدورات
- هنالك نوعان من محرك الخطوة: احادي القطبية (unipolar) وثنائي القطبية (bipolar) وكلاهما يتيح تحريك المحرك بخطوات ثابتة، مثلا 200 خطوة في الدورة اي ما يعادل 1.8 درجة لكل خطوة
- يمكن تشغيل محركات الخطوة والتحكم باتجاه دورانها باستخدام دائرة الكترونية مثل ULN2003 اذا كانت احادية القطبية او باستخدام A4988 اذا كانت ثنائية القطبية

مثال: محرك الخطوة

- برنامج لتشغيل محرك الخطوة ثنائي القطبية وتغيير اتجاه الدوران



Wokwi Arduino and ESP32 Simulator

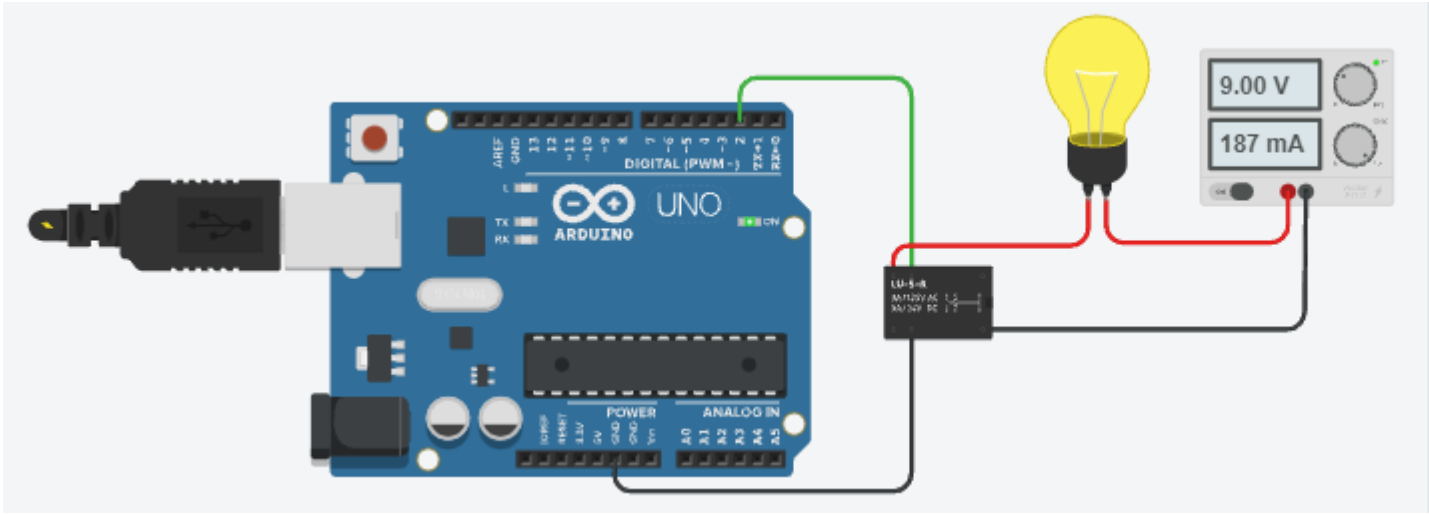
```
#include <Stepper.h>

const int stepsPerRevolution = 200;

Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11);

void setup() {
  // set the speed at 60 rpm:
  myStepper.setSpeed(60);
  Serial.begin(9600);
}

void loop() {
  // step one revolution  in one direction:
  Serial.println("clockwise");
  myStepper.step(stepsPerRevolution);
  delay(500);

  // step one revolution in the other direction:
  Serial.println("counterclockwise");
  myStepper.step(-stepsPerRevolution);
  delay(500);
}
```

المرحل

- المرحل (Relay) هو مفتاح كهربائي يعمل على فتح واغلاق الدائرة الكهربائية
- يستخدم المرحل في التحكم بالاجهزة الكهربائية ذات التيار المرتفع مثل المصباح والمروحة
- يتم وصل المرحل بطريقتين: عندما تكون الدائرة عادة مفتوحة (normally open) او عندما تكون الدائرة عادة مغلقة (normally closed)

مثال: المرحل

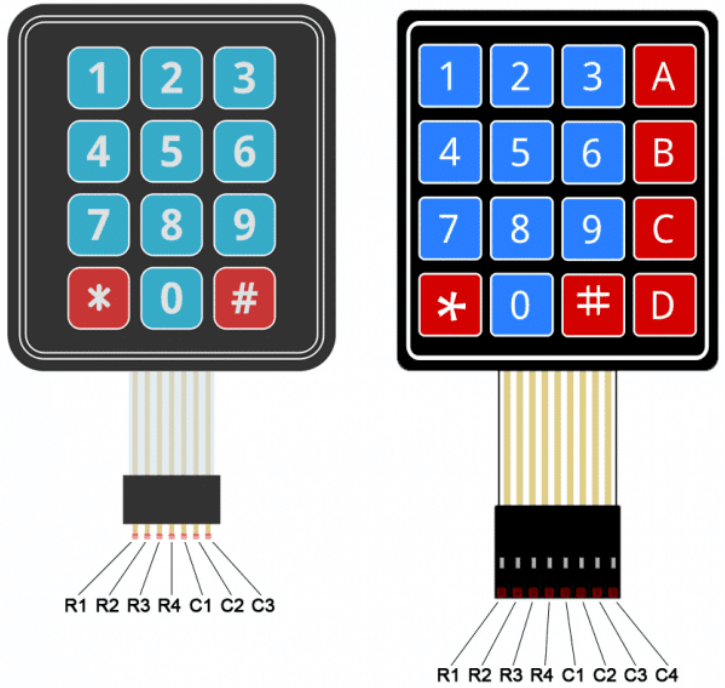
- برنامج للتحكم بالمصباح الكهربائي




```
int bulb = 2;

void setup()
{
  pinMode(bulb, OUTPUT);
}

void loop()
{
  digitalWrite(bulb, HIGH);
  delay(1000);
  digitalWrite(bulb, LOW);
  delay(1000);
}
```

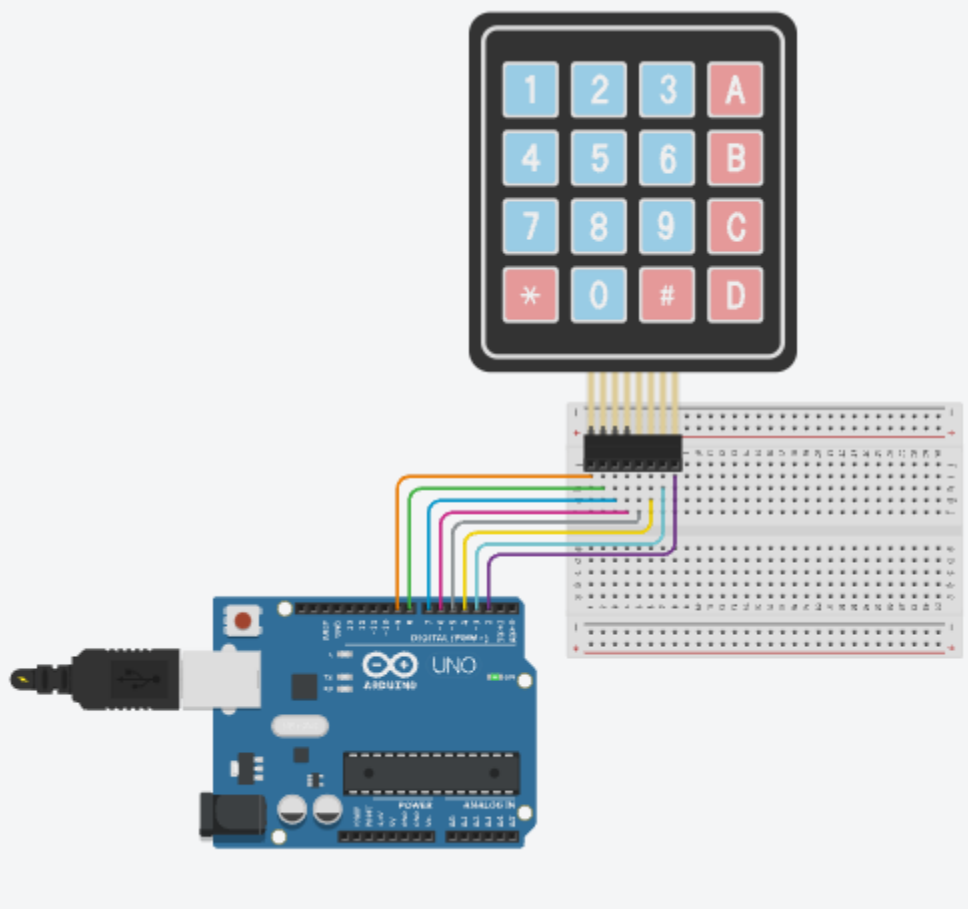
لوحة المفاتيح الرقمية

- تستخدم لوحة المفاتيح الرقمية (Keypad) في ادخال الارقام او كلمات المرور
- تتكون لوحة المفاتيح الرقمية من مصفوفة من الاعمدة والصفوف مثل 3x4 او 4x4
- الضغط على احد المفاتيح يؤدي الى تتبع العمود والصف ومعرفة المفتاح الذي تم ضغطه



مثال: لوحة المفاتيح الرقمية

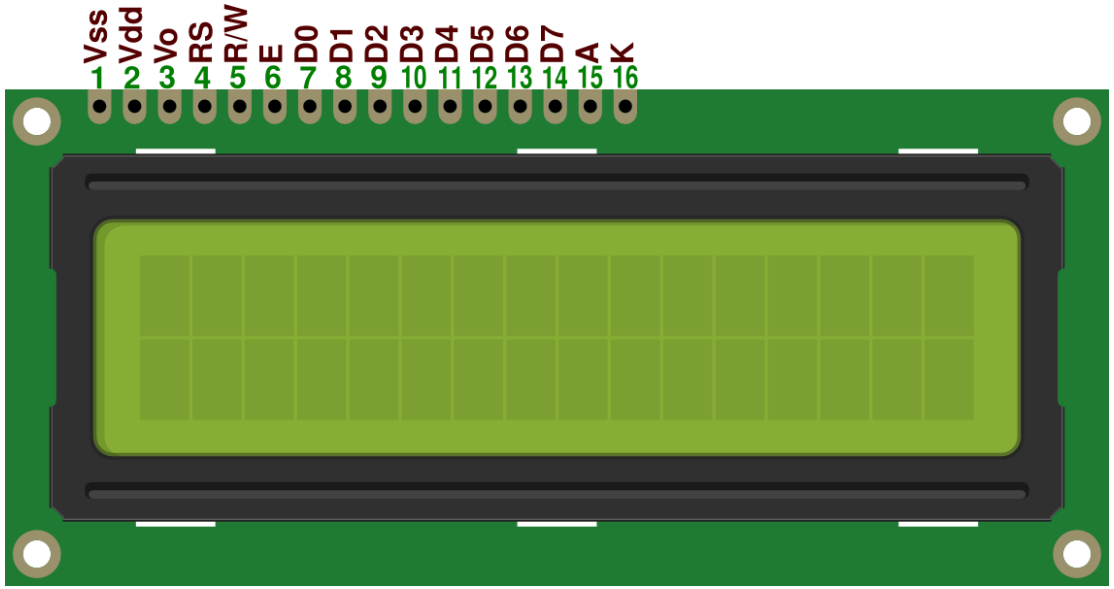
- برنامج لطباعة المفتاح الذي تم ضغطه باستخدام لوحة مفاتيح رقمية 4x4




```arduino
#include <Keypad.h>

char keymap[4][4]=
{
{'1', '2', '3', 'A'},
{'4', '5', '6', 'B'},
{'7', '8', '9', 'C'},
{'*', '0', '#', 'D'}
};

byte row[4] = {9, 8, 7, 6}; // Row Pins
byte col[4] = {5, 4, 3, 2}; // Columns Pins

Keypad keypad = Keypad(makeKeymap(keymap), row, col, 4, 4);

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

void loop()
{
  char key = keypad.getKey();

  if (key){
    Serial.println(key);
  }
}
```

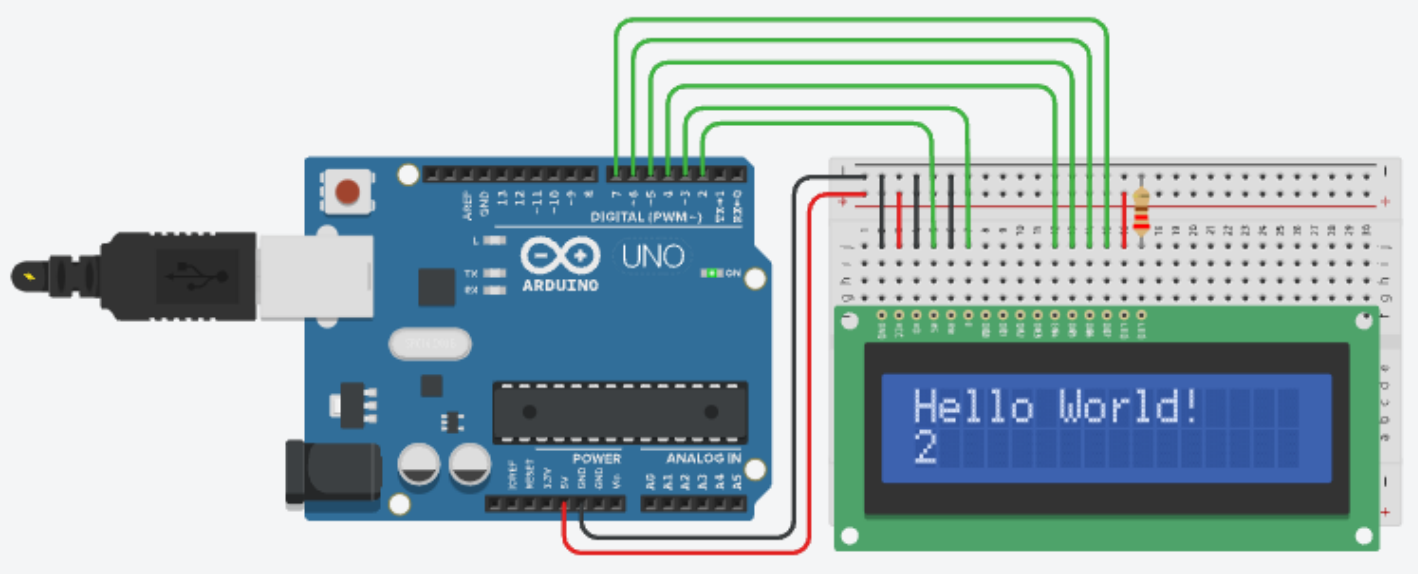
شاشة العرض

- تستخدم شاشة العرض LCD 16x2 في طباعة نص على سطرين و16 حرف في كل سطر
- يمكن انشاء حروف مخصصة وطباعتها على شاشة العرض LCD



مثال: شاشة العرض

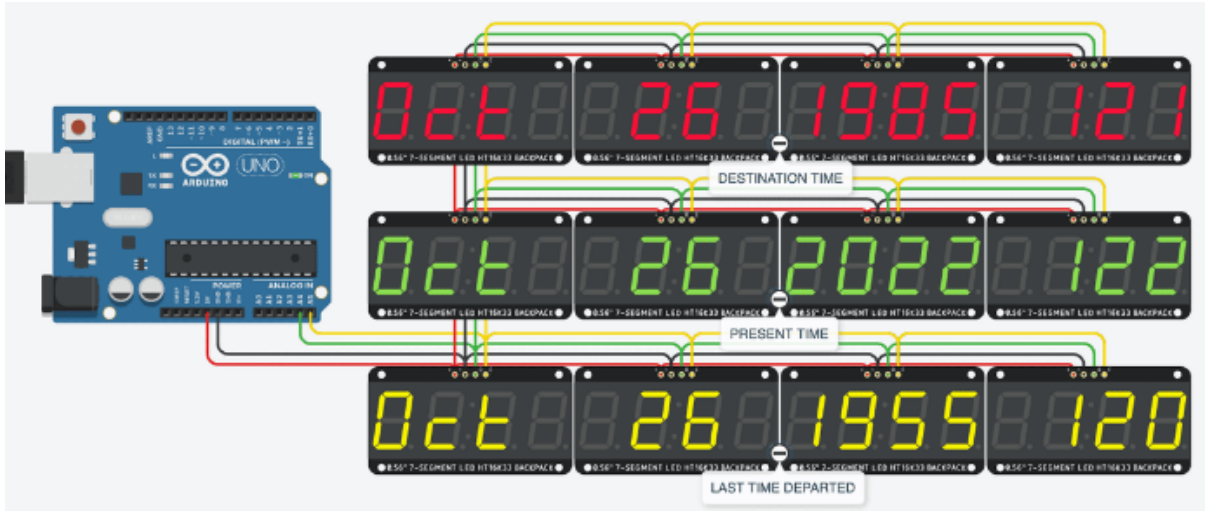
- برنامج لطباعة نص على شاشة العرض LCD وعدد الثواني منذ بدء التشغيل




```
#include<LiquidCrystal.h>
int cs = 2, en = 3, d4 = 4, d5 = 5, d6 = 6, d7 = 7;

LiquidCrystal lcd(cs, en, d4, d5, d6, d7);

void setup(){
  lcd.begin(16, 2);
  // set the cursor to column 0, line 0
  lcd.setCursor(0,0);
  lcd.print("Hello World!");
}

void loop() {
  // set the cursor to column 0, line 1
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis() / 1000);
}
```

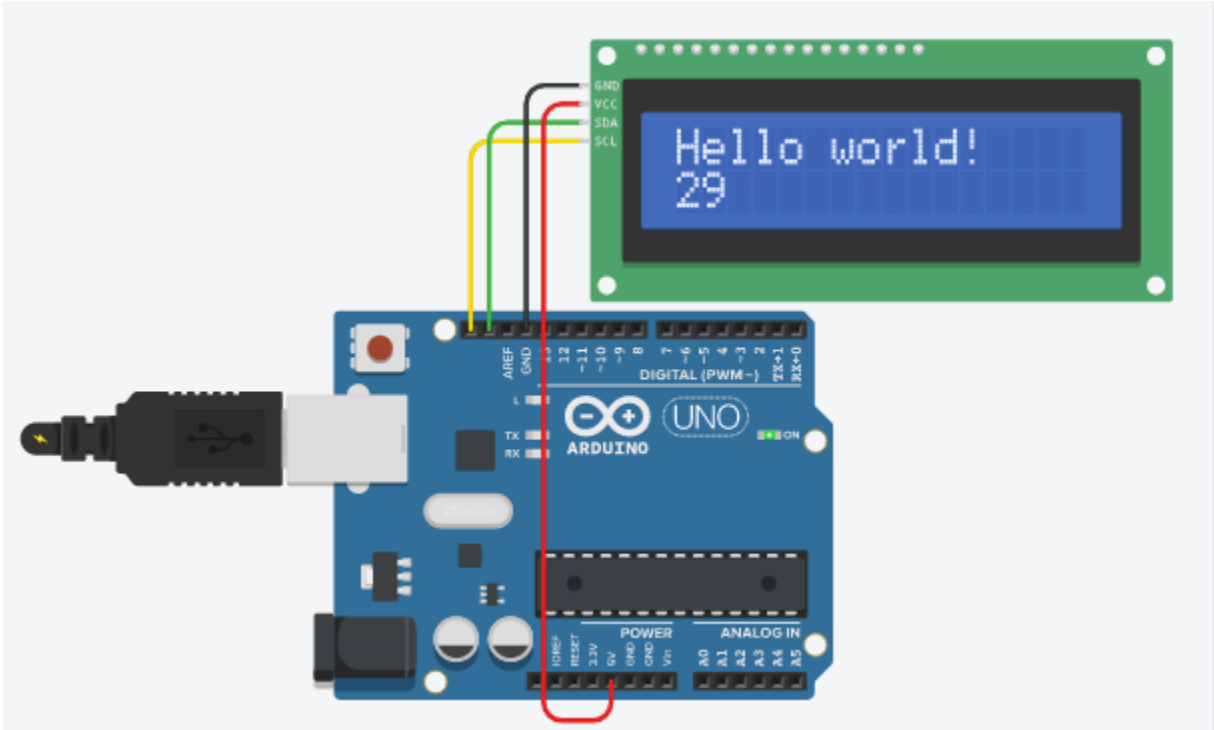
بروتوكول I2C

- بروتوكول I2C هو بروتوكول اتصال تسلسلي يستخدم في نقل البيانات من المتحكم الى الاجهزة الطرفية عبر سلك واحد يدعى SDA او الناقل
- يتطلب بروتوكول I2C سلك آخر لمزامنة البيانات اثناء نقلها عبر الناقل يدعى SCL
- يمكن وصل عدة اجهزة طرفية مع المتحكم الدقيق من خلال نفس السلكين مثل شاشات العرض



مثال: بروتوكول I2C

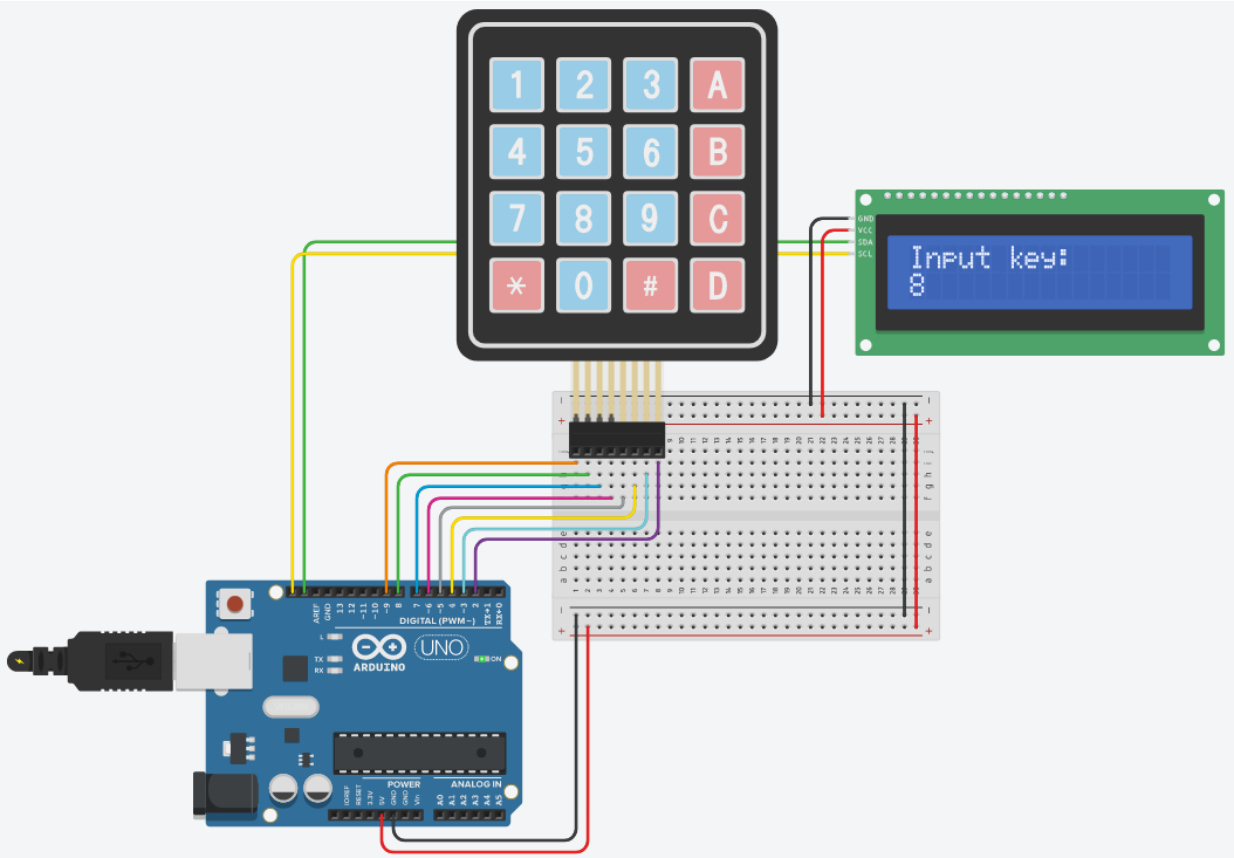
- برنامج لطباعة نص على شاشة العرض LCD وعدد الثواني منذ بدء التشغيل




```
#include <Adafruit_LiquidCrystal.h>

int seconds = 0;

Adafruit_LiquidCrystal lcd(0);

void setup()
{
  lcd.begin(16, 2);
  lcd.print("Hello world!");
}

void loop()
{
  lcd.setCursor(0, 1);
  lcd.print(seconds);
  delay(1000);
  seconds += 1;
}
```

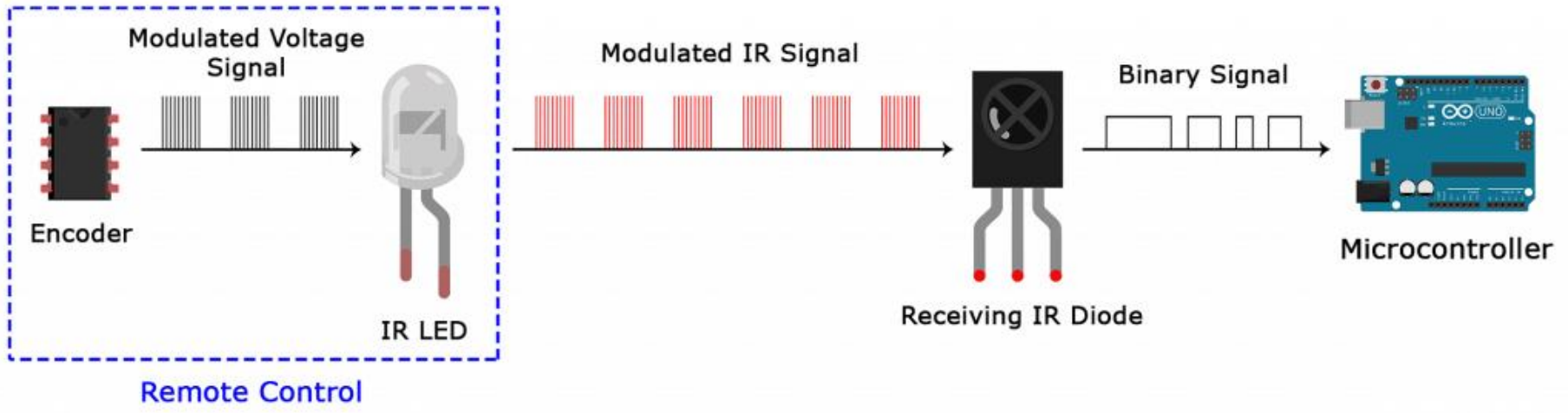
تمرين: شاشة ولوحة مفاتيح

- اكتب برنامج لطباعة المفتاح الذي تم ضغطه من خلال لوحة المفاتيح على شاشة العرض



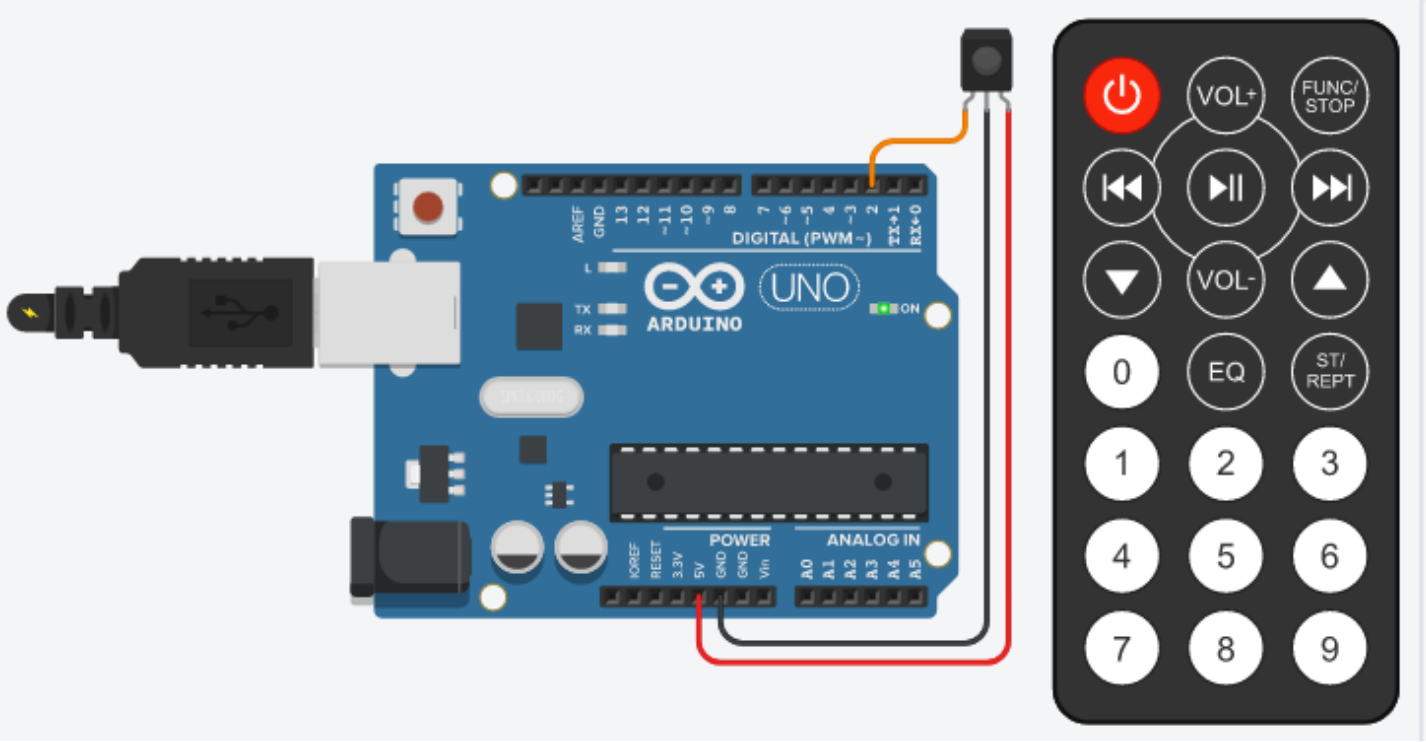
التحكم عن بعد

- يمكن التحكم بالاجهزة عن بعد (remote control) من خلال الاشعة تحت الحمراء (IR)
- يستخدم مستشعر الاشعة تحت الحمراء (IR sensor) في استقبال اوامر التحكم عن بعد من وحدة التحكم عن بعد (IR remote)
- يتم توجيه وحدة التحكم عن بعد نحو مستشعر الاشعة تحت الحمراء لارسال رموز مفاتيح وحدة التحكم عن بعد الى المتحكم الدقيق لاسلكيا



مثال: التحكم عن بعد

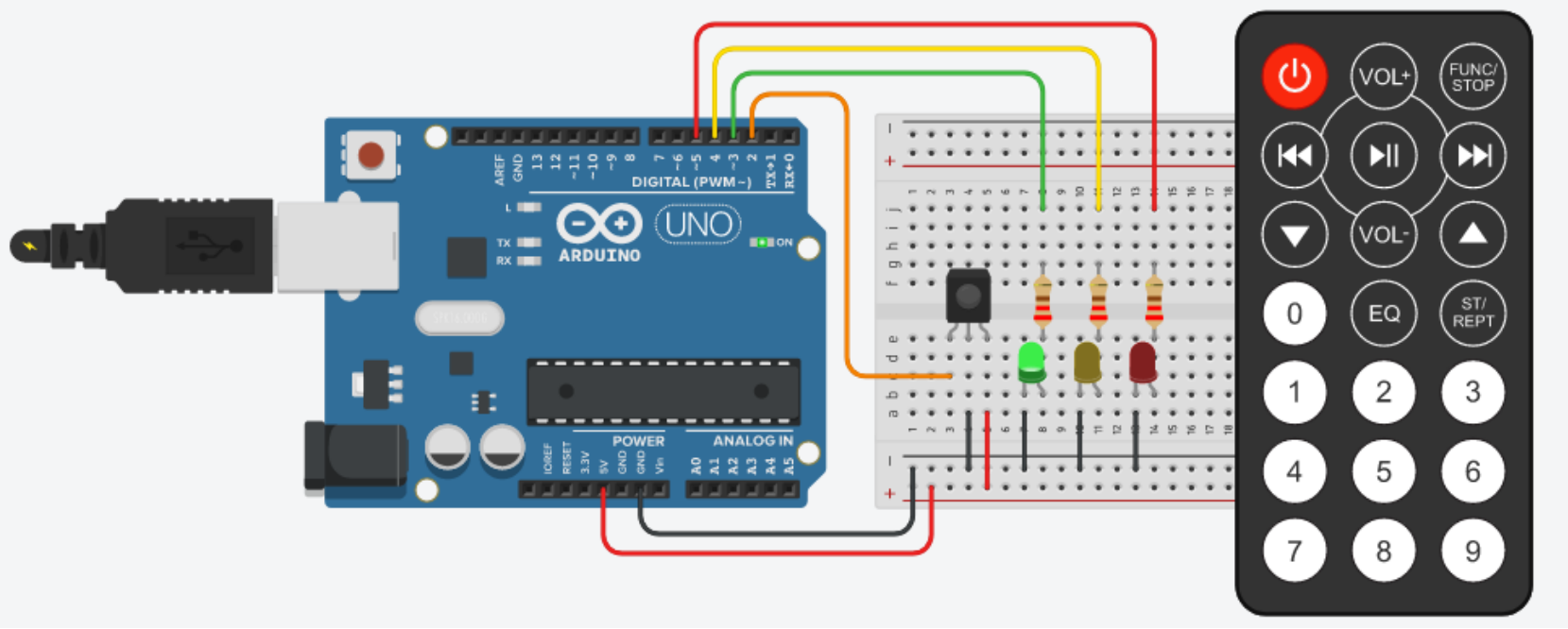
- برنامج لطباعة رمز المفتاح الذي تم ضغطه باستخدام وحدة التحكم عن بعد (IR remote)




```
#include <IRremote.hpp>

int irPin = 2;

void setup()
{
  Serial.begin(9600);
  IrReceiver.begin(irPin, ENABLE_LED_FEEDBACK);
}

void loop()
{
  unsigned long key;
  if (IrReceiver.decode()) {
    key = IrReceiver.decodedIRData.decodedRawData;
    Serial.println(key, HEX);
    IrReceiver.resume();
  }
}
```

تمرين: التحكم بالدايودات عن بعد

- اكتب برنامج للتحكم بالدايودات (LEDs) باستخدام وحدة التحكم عن بعد (IR remote)
  - تشغيل او اطفاء الضوء الاخضر عند الضغط على المفتاح 1
  - تشغيل او اطفاء الضوء الاصفر عند الضغط على المفتاح 2
  - تشغيل او اطفاء الضوء الاحمر عند الضغط على المفتاح 3



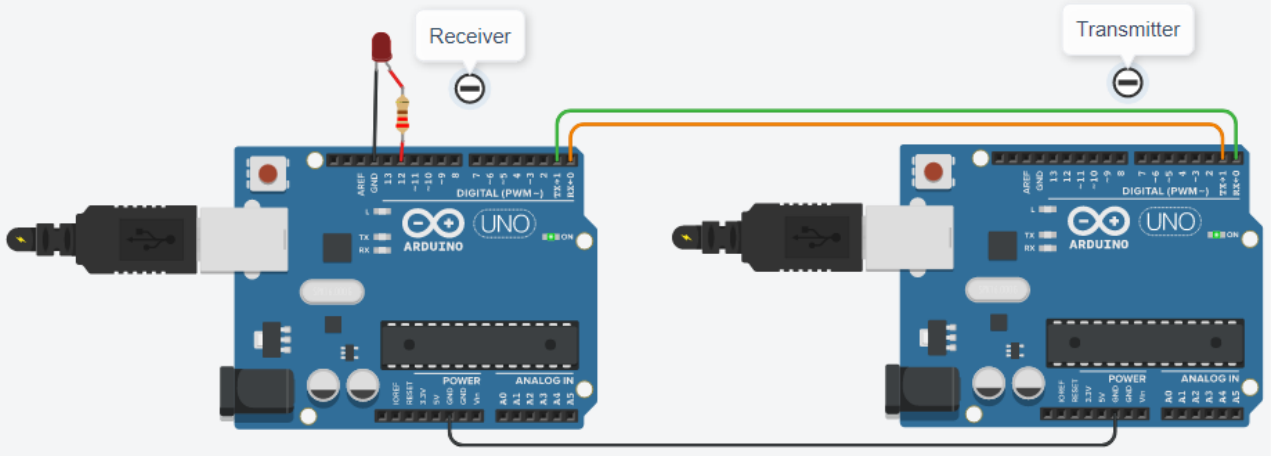
الاتصال التسلسلي

- يستخدم المنفذ التسلسلي (Serial) في الاتصال مع جهاز الحاسوب او الاجهزة الاخرى
- يمكن وصل لوحتي اردوينو عبر المنفذ التسلسلي وارسال الاوامر من اللوحة الاولى الى اللوحة الثانية وتنفيذها على اللوحة الثانية اي التحكم باللوحة الثانية عبر المنفذ التسلسلي
- تستخدم الدالة Serial.begin() في تحديد سرعة المنفذ التسلسلي
- تستخدم الدالة Serial.print() في ارسال الاحرف عبر المنفذ التسلسلي
- تستخدم الدالة Serial.available() في تحديد عدد الاحرف لاستقبالها من المنفذ التسلسلي
- تستخدم الدالة Serial.read() في استقبال الاحرف عبر المنفذ التسلسلي

مثال: الاتصال التسلسلي

- برنامج لارسال اوامر التحكم بالدايود من اللوحة الاولى الى الثانية عبر المنفذ التسلسلي

```
// Receiver
//
int led = 12;
void setup()
{
  pinMode(led, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  if (Serial.available()) {
    char c = Serial.read();
    if (c == '1') {
      digitalWrite(led, HIGH);
    }
    if (c == '0') {
      digitalWrite(led, LOW);
    }
    Serial.println(c);
  }
}
```




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

void loop()
{
  Serial.print(1);
  delay(1000);
  Serial.print(0);
  delay(1000);
}
```