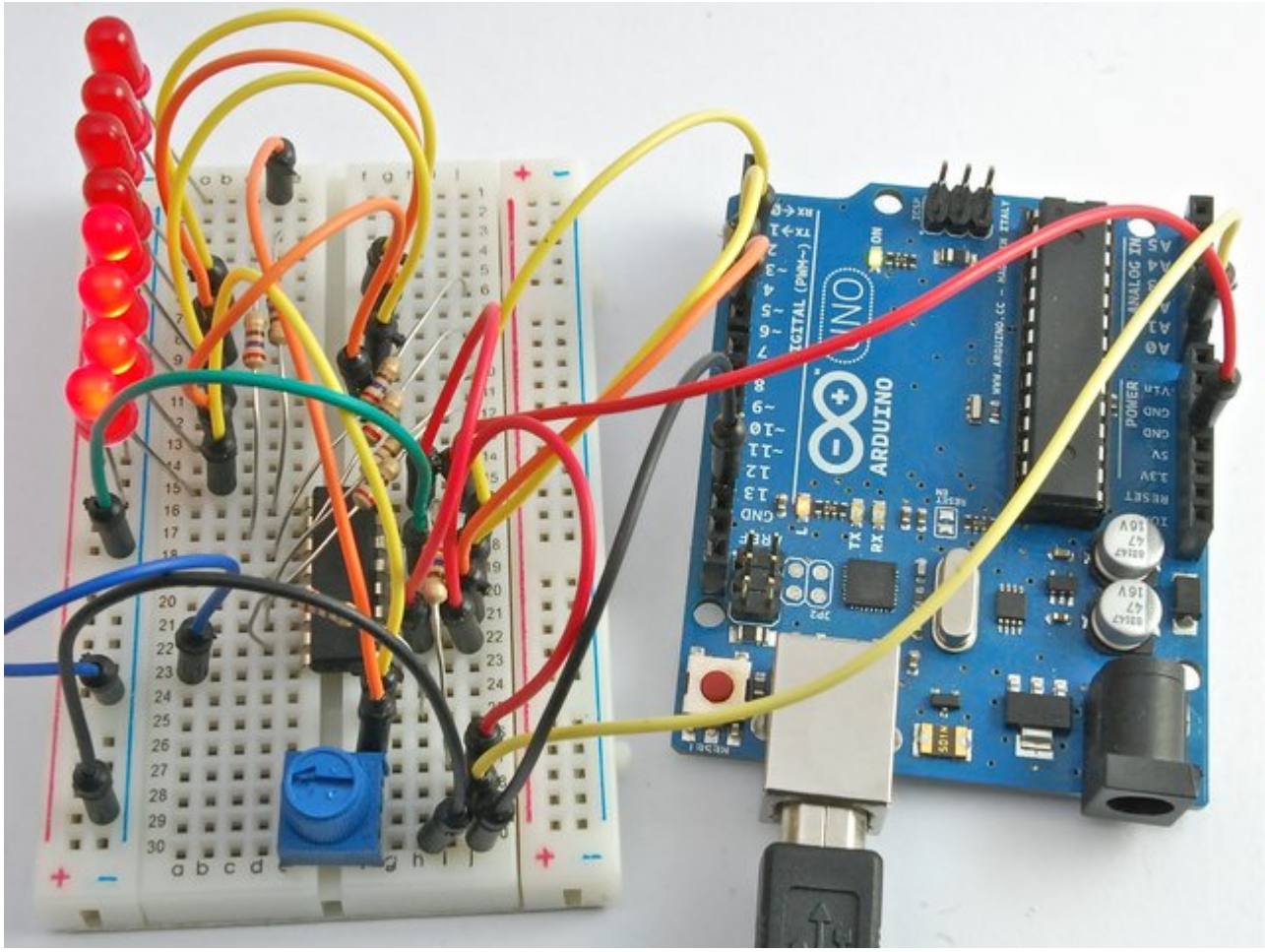
اردوينو – الدرس التاسع – المدخلات التناظرية Analog inputs

مقدمة

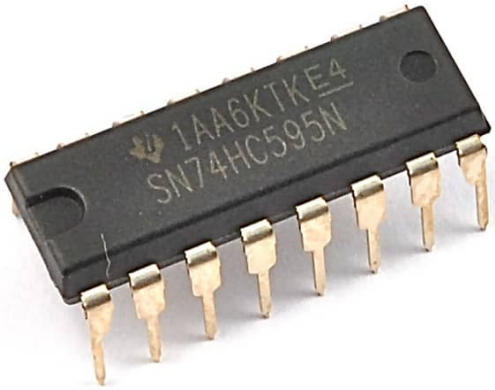
في هذا الدرس، ستقوم بإستخدام الإتصال التسلسلي (Serial Monitor) لعرض قراءات المدخلات التناظرية "Analog Inputs" ومن ثم اضافة 8 مصابح LED (من الدرس الخامس) لتتمكن من التحكم بزيادة وتقليل درجة اضاءة الـLEDS عبر المقاوم المتغير (Variable resistor).



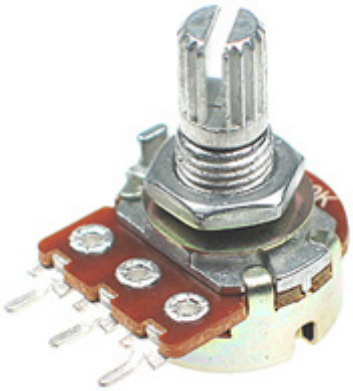
المواد والأدوات

(5mm LED) ×8

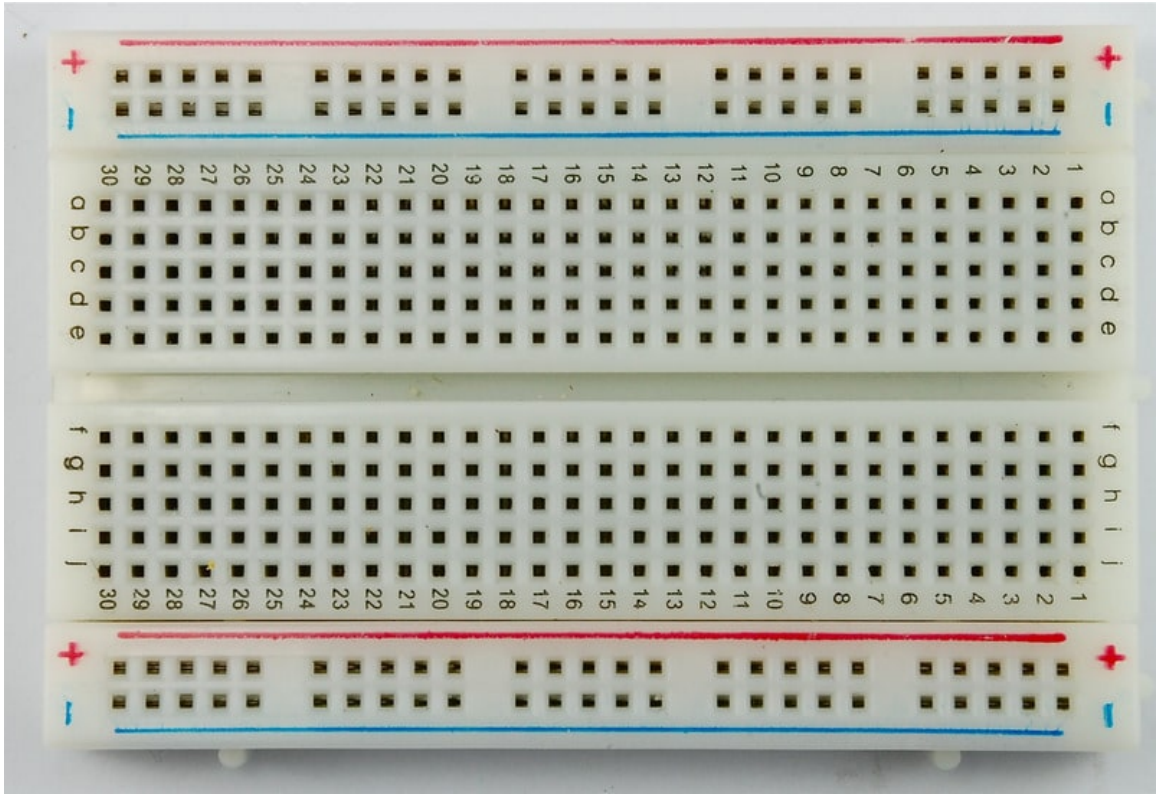
(Ω Resistor 270) ×8



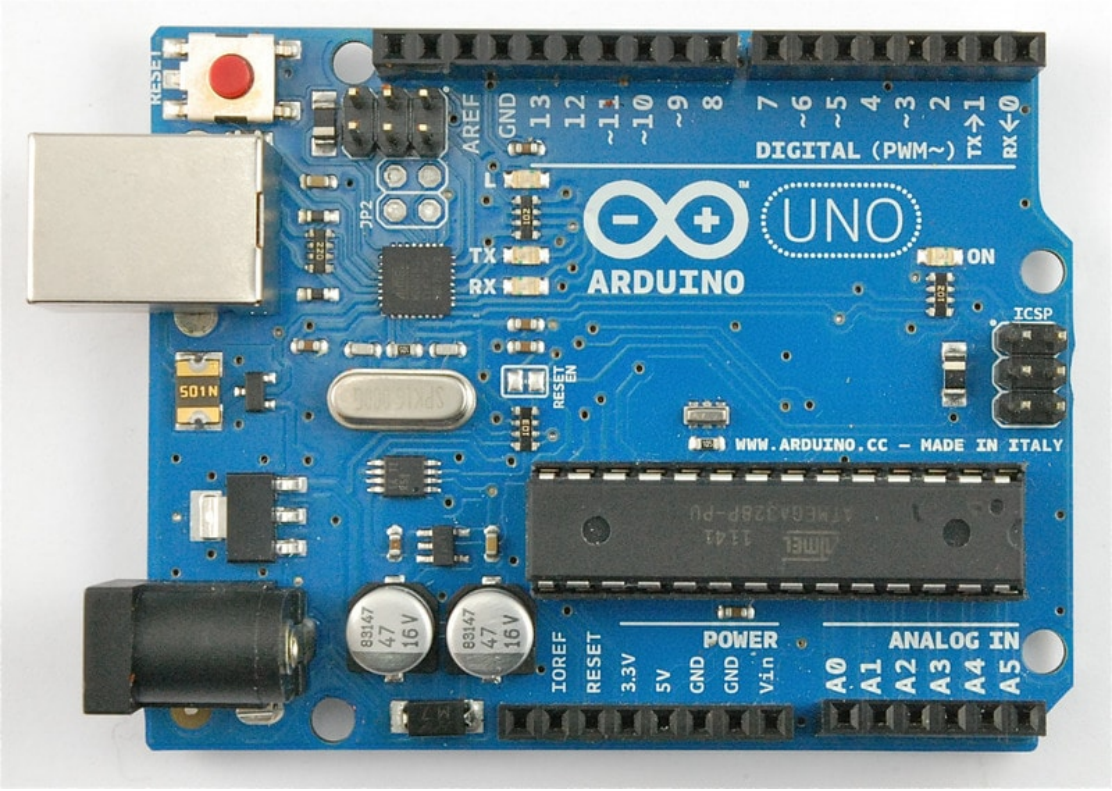
(74HC595 shift register) رقاقة مسجل الإزاحة ×1



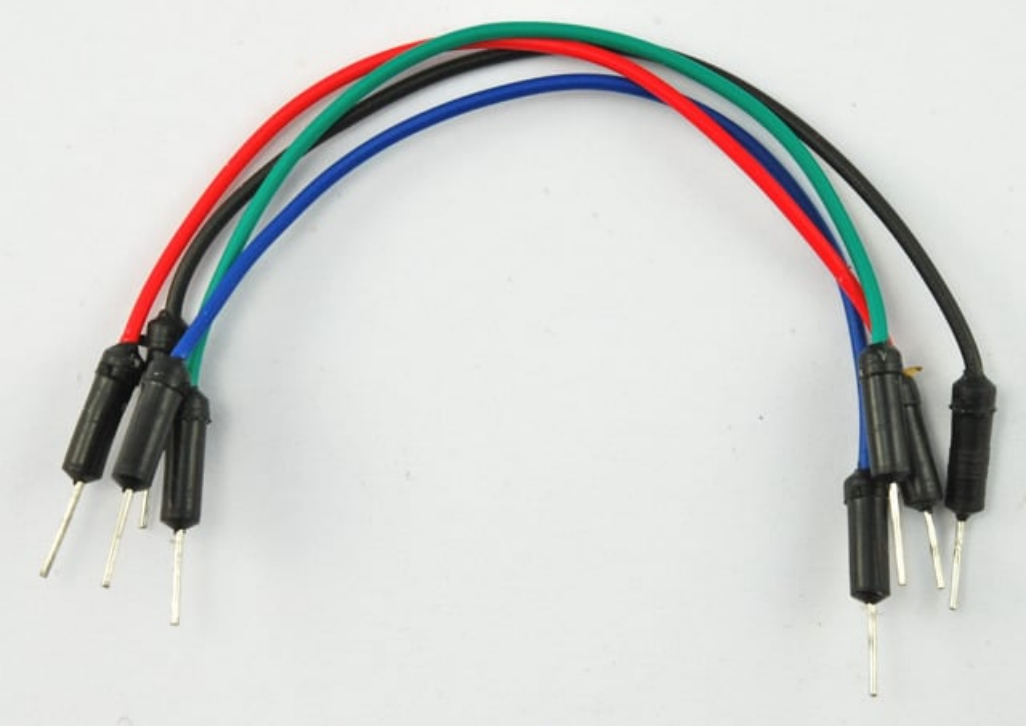
1× مقاوم متغير (kΩ variable resistor 10)



1× لوحة التجارب (Half-size Breadboard)



1× اردوينو اونو



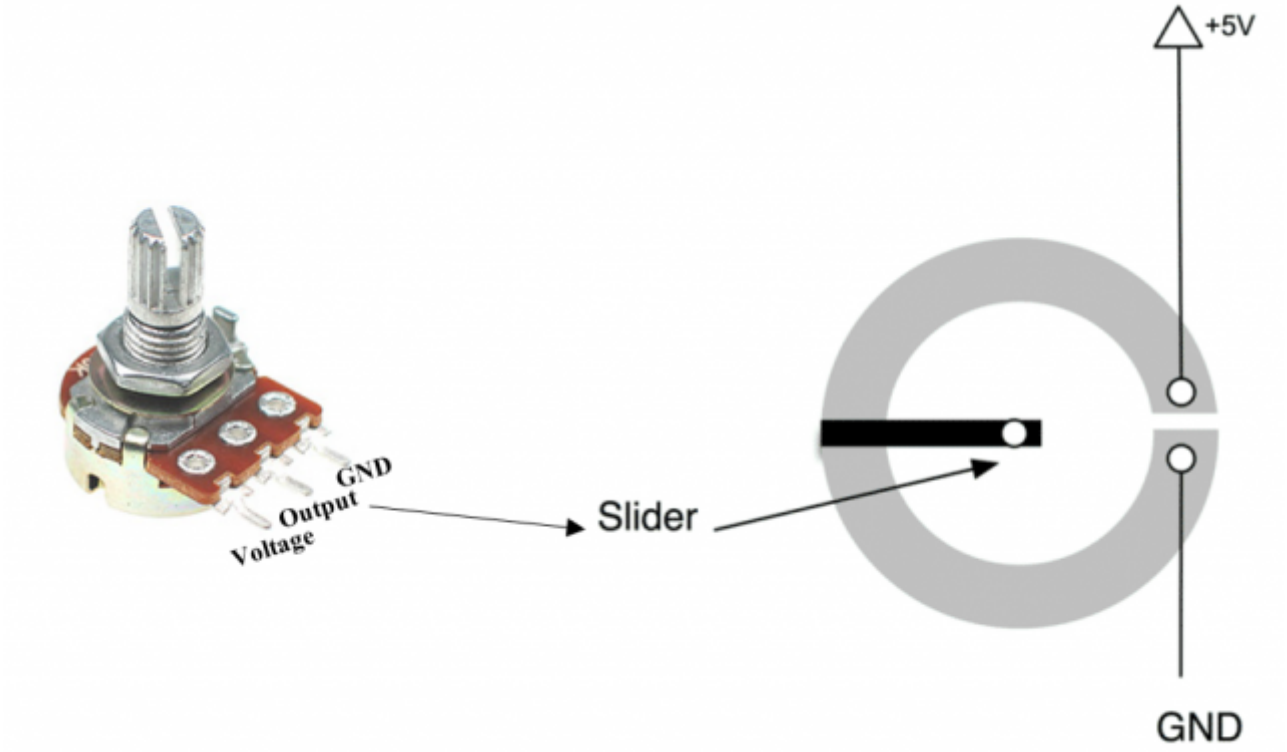
حزمة أسلاك توصيل (ذكر_ذكر)

×1 سلك اردوينو

المقاوم المتغير (Variable Resistors)

المقاومات المتغيرة تدعى "potentiometers" وتختصر بـ pot

في تجربتنا مع شاشة الاتصال التسلسلي (Serial Monitor) يقوم المقاوم بتغيير الجهد على A0 ، كما يقوم الكود البرمجي بتحويل هذا الجهد إلى رقم ما بين 0 و 1023.

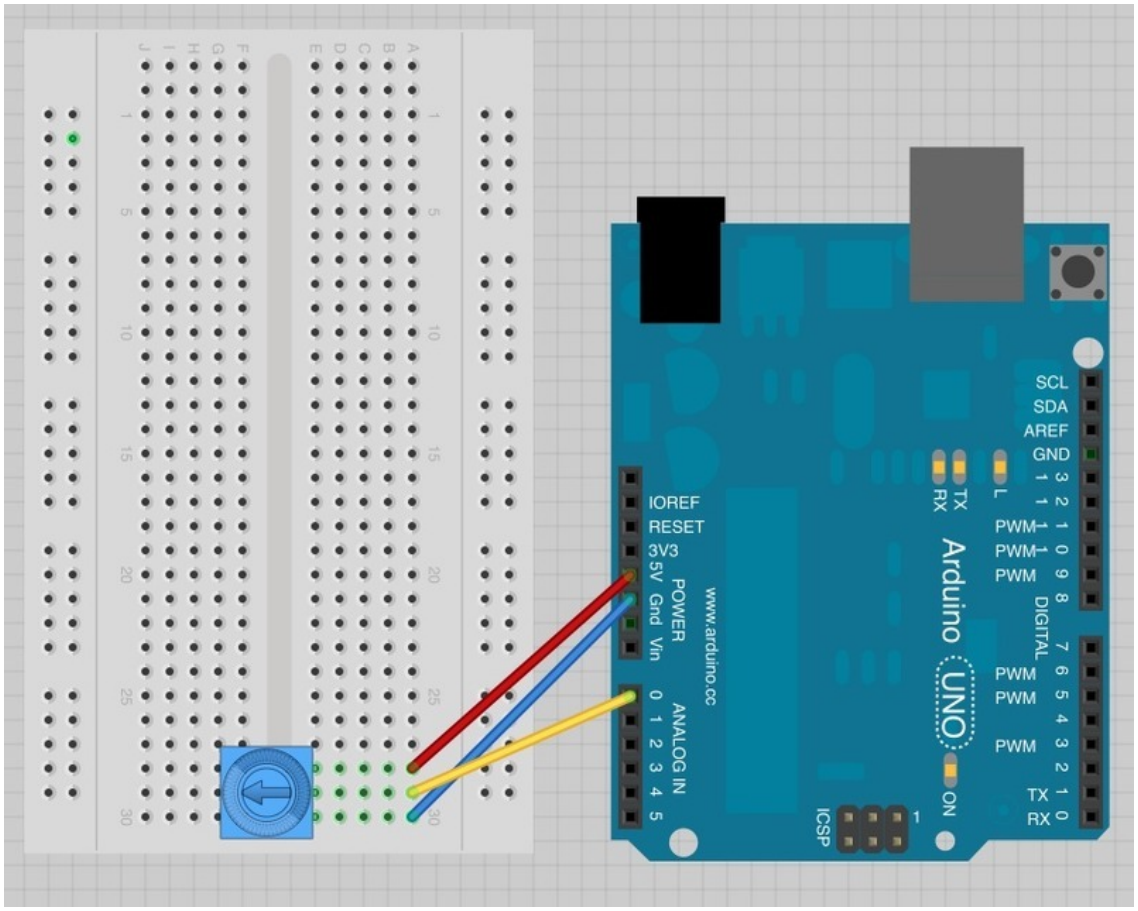


في المقاوم المتغير "pot" يوجد مسار يمثل المقاومة، في هذه الحالة مقاومه مقدارها 10 kΩ . كما يوجد سن متوسط يعتبر الموصل الفاصل يسمى "Slider" وظيفته هي تغيير مقدار المقاومة ما بين 0 وحتى 5V

توصيل الدائرة

قبل البدء بإستخدام مصابيح الـLEDs تستطيع القيام بالتجربة التالية عبر إستخدام المقاوم المتغير (potentiometer)؛ لتجربة المدخلات التناظرية ثم تفعيل خاصية شاشة الاتصال التسلسلي (Serial Monitor) في الاردوينو

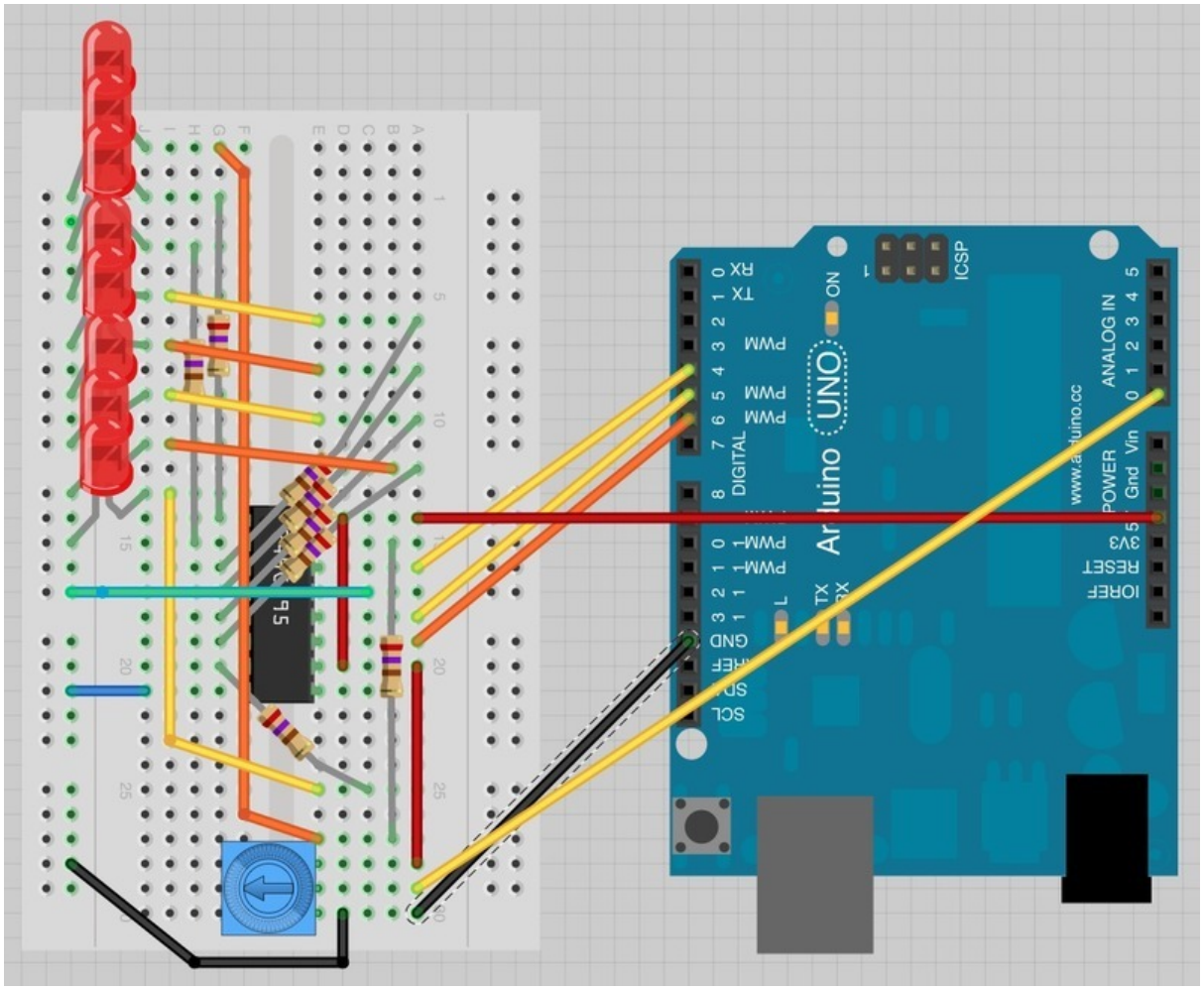
وصل الدائرة رقم (1) لوح التجارب كالتالي:

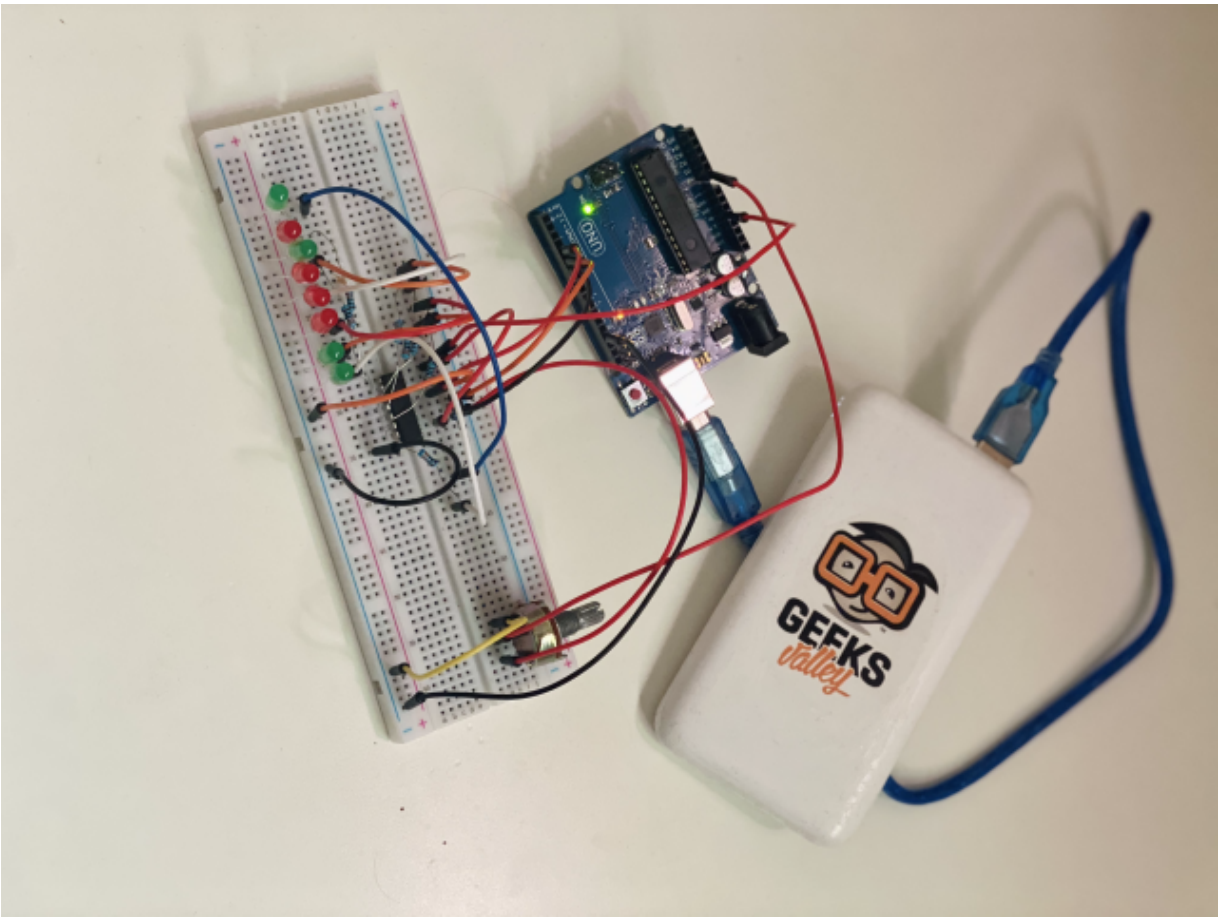


توصيل الدائرة رقم (2):

والآن استخدم المقاوم المتغير "pot" للتحكم في عدد الـLEDs

يعتمد تصميم هذه الدائرة على الدرس الخامس، هناك بعض الاسلاك التي قمت بتحريكها، كما قمت بإضاة المقاوم المتغير 'pot' وبعض الاسلاك له إلى الدائرة.


ICSP
ON
L
TX
RX
Arduino UNO
www.arduino.cc
DIGITAL
PWM
ANALOG IN
POWER
Gnd Vin
3V3
5V
RESET
IOREF
GND
SCL
SDA
AREF
95


UNO
GEEKS
valley

الكود البرمجي

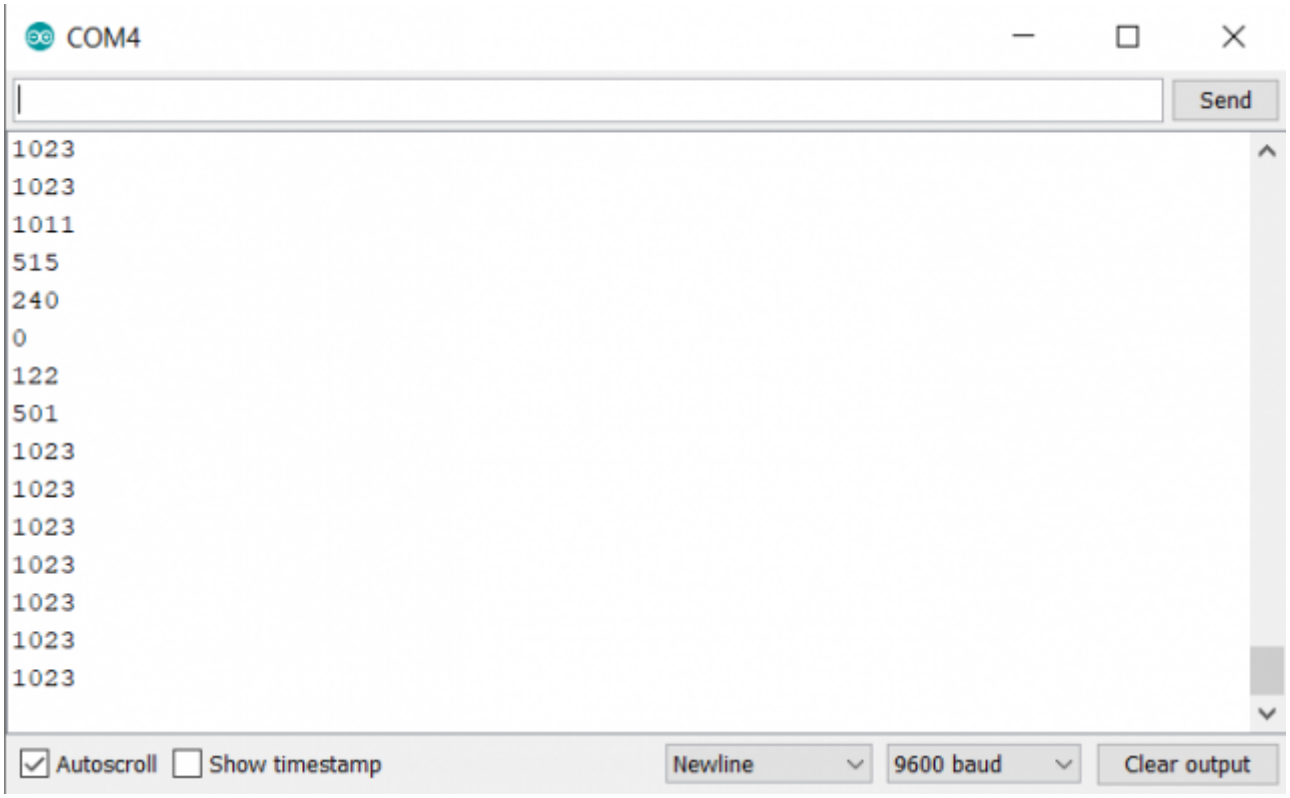
ارفع الكود للدائرة رقم (1) على الاردوينو:

```
int potPin = 0;

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

void loop()
{
  int reading  = analogRead(potPin);
  Serial.println(reading);
  delay(500);
}
```

والآن، افتح شاشة الإتصال التسلسلي (Serial Monitor):



حرك وغير المقاومة وسوف نرى القراءات تتغير مابين 0 و 1023

شاشة الاتصال التسلسلي (Serial Monitor) تقوم بعرض القراءات من A0 بواسطة السطر التالي:

```
int reading  = analogRead(potPin);
```

الجهد (Voltage) في A0 يتم تحويله إلى رقم مابين 0 و 1023.

ارفع الكود التالي للدائرة رقم (2) إلى الأردوينو:

```
int potPin = 0;
int latchPin = 5;
int clockPin = 6;
int dataPin = 4;

int leds = 0;

void setup()
{
  pinMode(latchPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
}

void loop()
{
  int reading  = analogRead(potPin);
  int numLEDSLit = reading / 114;  //1023 / 9
  leds = 0;
  for (int i = 0; i < numLEDSLit; i++)
  {
    bitSet(leds, i);
  }
  updateShiftRegister();
}

void updateShiftRegister()
{
   digitalWrite(latchPin, LOW);
   shiftOut(dataPin, clockPin, LSBFIRST, leds);
   digitalWrite(latchPin, HIGH);
}
```

عليك أن تعلم أن بعض الأسطر في هذا الكود هو من الدرس الخامس، لذا تستطيع العودة للدرس الخامس لمعرفة شرح الكود المتعلق بالتحكم في الـLEDs

اما بالنسبة للجزء الجديد من الكود وهو تعريف منفذ المدخل التناظري "Analog inputs" والتي سنقوم بربط المقاوم المتغير بها

```
int potPin = 0;
```

لاحظ بأنك لم تقم بإضافة أي جديد في دالة 'setup' لتعريف حالة المنفذ للمدخل التناظري.

في دالة 'loop' نقوم بقراءة القيمة التناظرية "Analog value" بالطريقة التالية:

```
int reading  = analogRead(potPin);
```

ولكن عليك تحويل هذه القراءة التي ما بين 0 و 1023 إلى عدد الـLEDs ليتم اضاءتها ، مابين 0 و 8 . الأرقام التي مابين 0 و 8 هي في الحقيقة مجموعها 9 قيم . لذا عليك توسيع نطاق القراءة عبر 1023 مقسومة على 9 أو 114

```
int numLEDSLit = reading / 114;
```

لإضاءة الأعداد الصحيحة للـLEDs ، استخدم for loop للعد من 0 وحتى "numLEDSLit" لوضع البت الصحيح.

```
  leds = 0;
  for (int i = 0; i < numLEDSLit; i++)
  {
    bitSet(leds, i);
  }
```

وأخيرًا نقوم بتحديث رقاقة مسجل الإزاحة 'shift register' عبر السطر التالي:

```
updateShiftRegister();
```

أنشطة أخرى

يمكنك استخدام مصباح LED واحد فقط بهذا المشروع وتتحكم به.