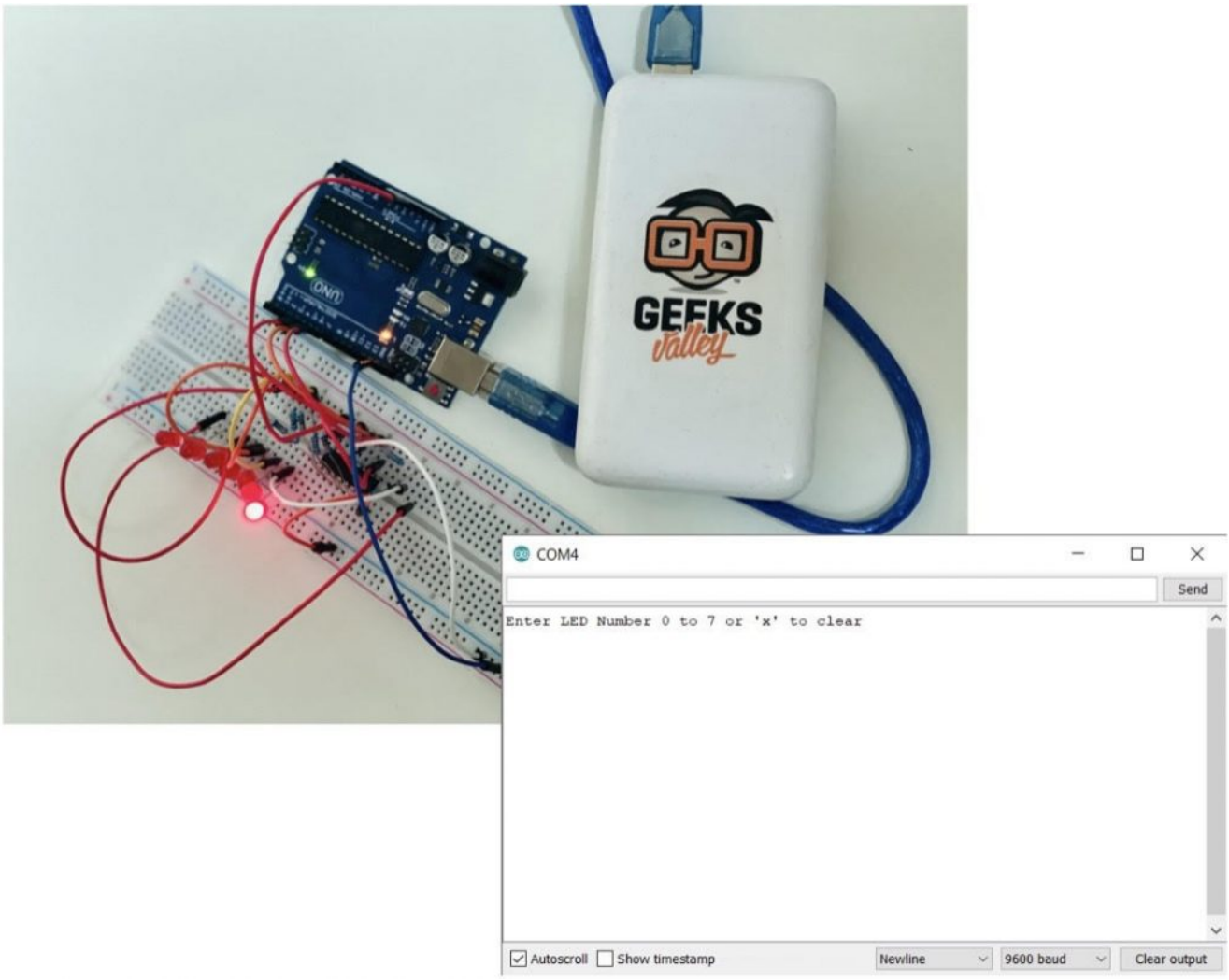
اردوينو – الدرس السادس – شاشة الإتصال التسلسلي – Serial Monitor

مقدمة

في هذا الدرس ستتعلم كيفية التحكم بإضاءة الـ LEDs عبر شاشة الإتصال التسلسلي "serial monitor" وهو يعتبر حلقة الوصل بين جهاز الكمبيوتر والأردوينو حيث يمكنك من ارسال واستقبال الرسائل والتحكم بالأردوينو.

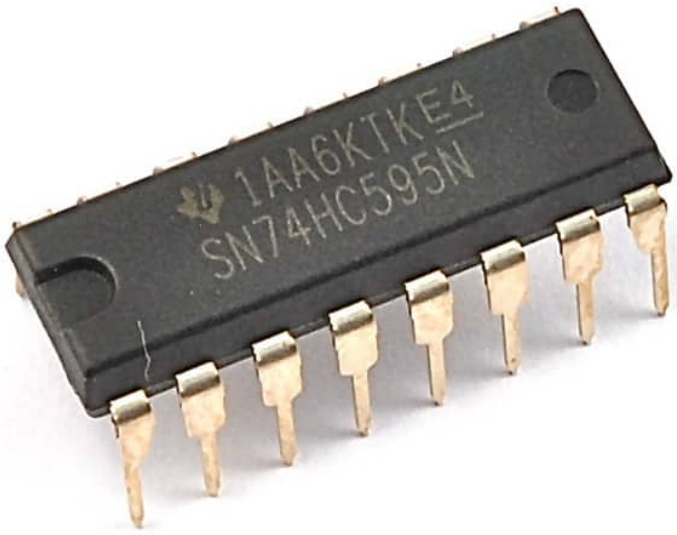
مثال، يمكنك ارسال أوامر من جهاز الكمبيوتر لإضاءة الـLED .



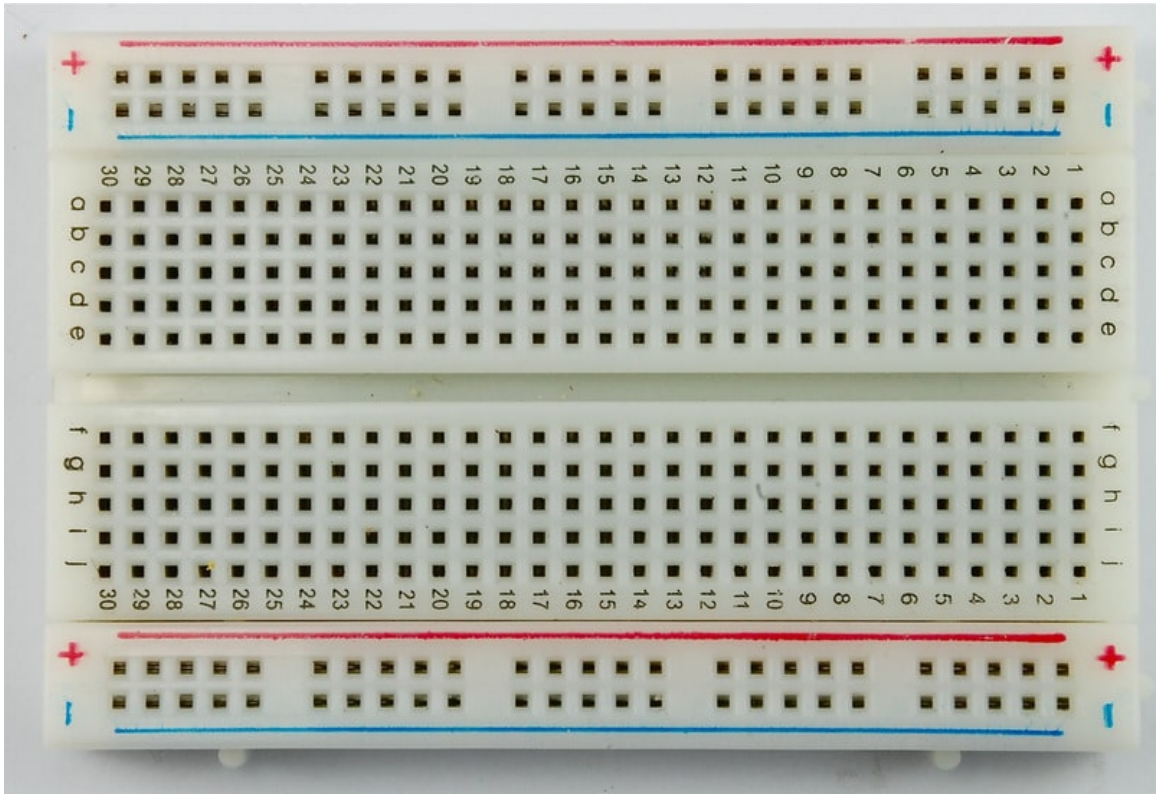
المواد والأدوات

(5mm Red LED) ×8

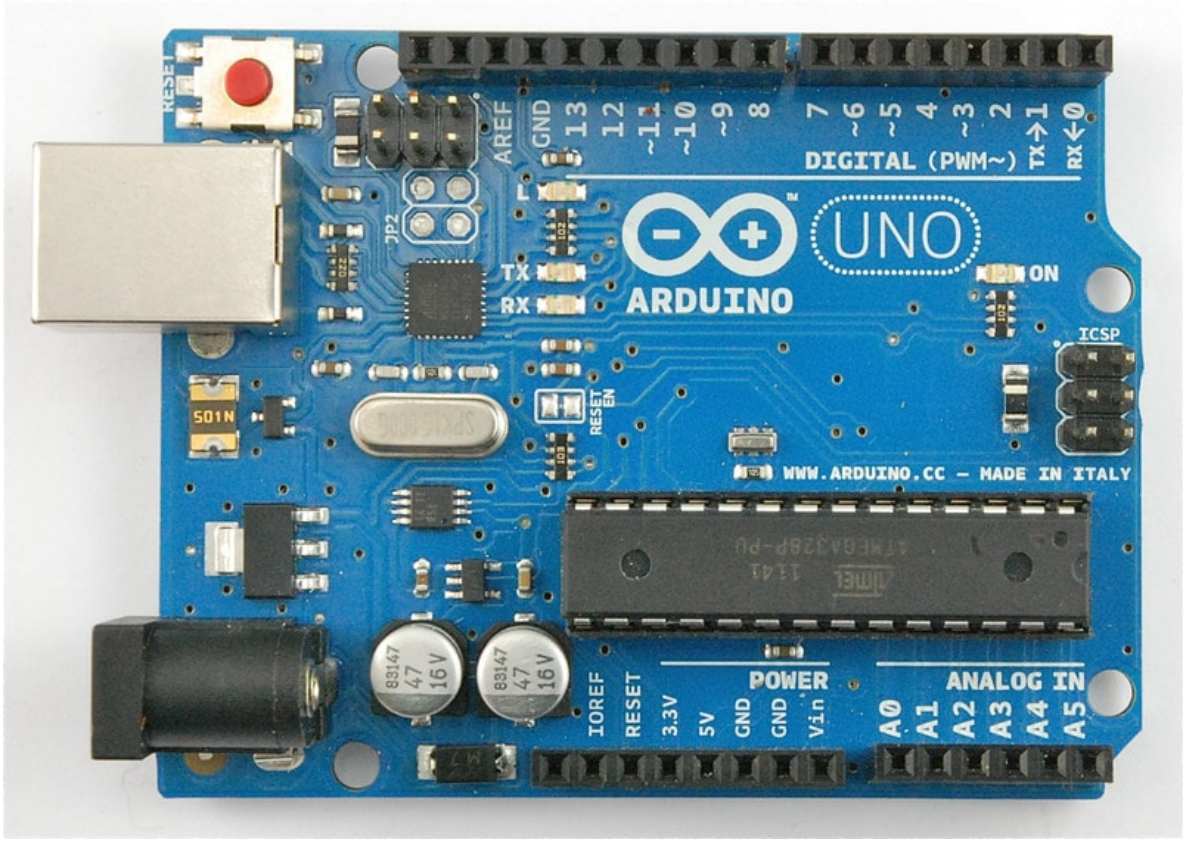
(Ω Resistor 270) ×8



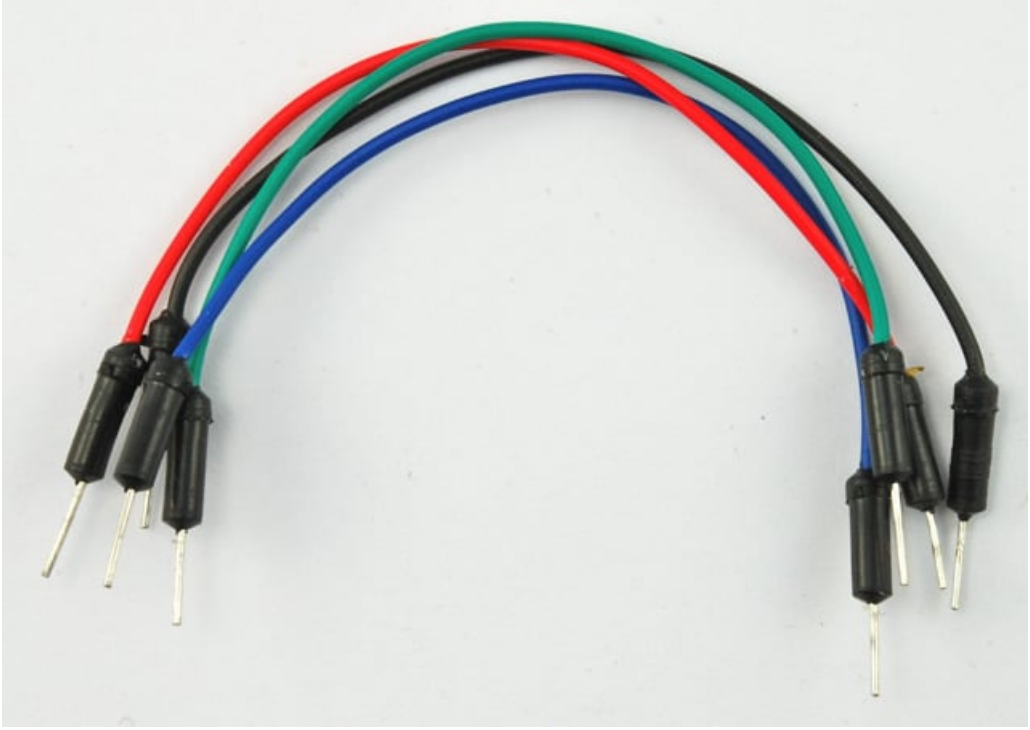
1× رقاقة مسجل الإزاحة (74HC595 Shift Register)



1× لوحة التجارب (Half-size Breadboard)



1× اردوينو اونو

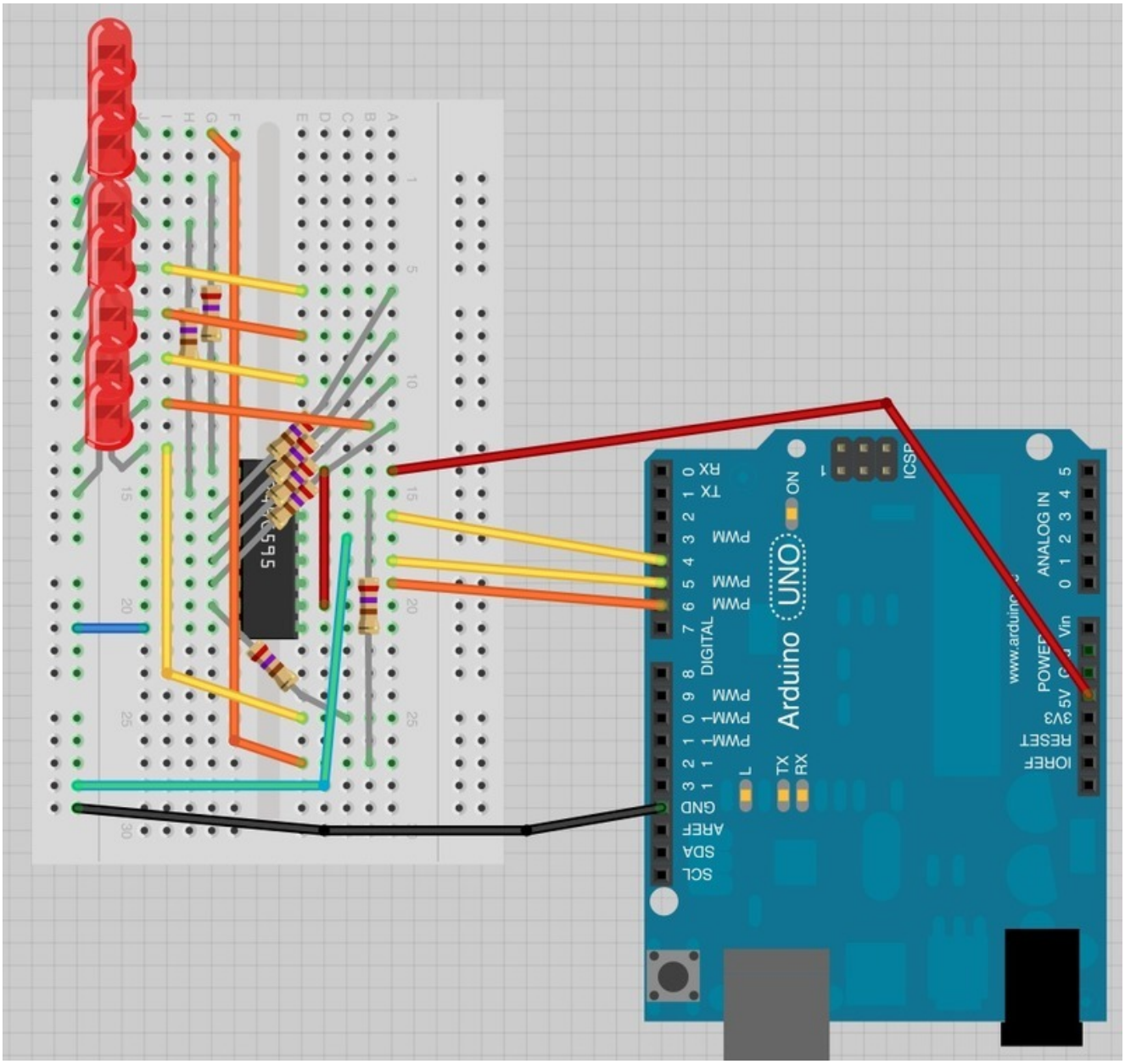


حزمة أسلاك توصيل (ذكر-ذكر)

1× سلك اردوينو

توصيل الدائرة

بما ان لدينا 8LEDs اربط مع 8 مقومات يجب وضع العديد من الأسلاك.



من الأسهل ان تبدأ بوضع رقاقة مسجل الإزاحة 74HC595 في لوح التجارب ،كل شيء مربوط به، ضع الرقاقة حيث يكون السن الذي يشبه شكل U متجه إلى أعلى لوح التجارب.

المنفذ رقم 1 للرقاقة على اليسار.

تربط منفذ الديجيتال 4 من الأردوينو بالمنفذ 14 من مسجل الإزاحة.

تربط منفذ الديجيتال 5 من الأردوينو بالمنفذ 12 من مسجل الإزاحة.

تربط منفذ الديجيتال 6 من الأردوينو بالمنفذ 11 من مسجل الإزاحة.

أغلب منافذ الإستخراج من الرقاقة توجد على اليسار، حتى يسهل ربط الـ LEDs.

بعد وضع رقاقة مسجل الإزاحة أضف المقاومات، يجب ان تتأكد أن ليس هنالك التماس بين مختلف نهايات المقاومات يجب أن تتأكد مجددًا قبل أن تغذي الاردوينو بالطاقة، قد يساعدك التقصير في طول نهاية المقاومات.

ضع الـ LED في لوح التجارب

يجب على المنافذ الطويلة للـ LED أن تواجه الرقاقة مباشرة.

المرحلة الأخيرة هي إضافة الأسلاك كما في الصورة لا تنسى أن نربط المنفذ 8 من الرقاقة إلى خانة الأرض أو GND.

حمل الكود وجربه على كل LED أن يعمل مرة واحدة ثم الكل معًا وأخيرًا ينطفئ الكل معًا قبل أن يعيد الاردوينو الدورة.

الكود البرمجي

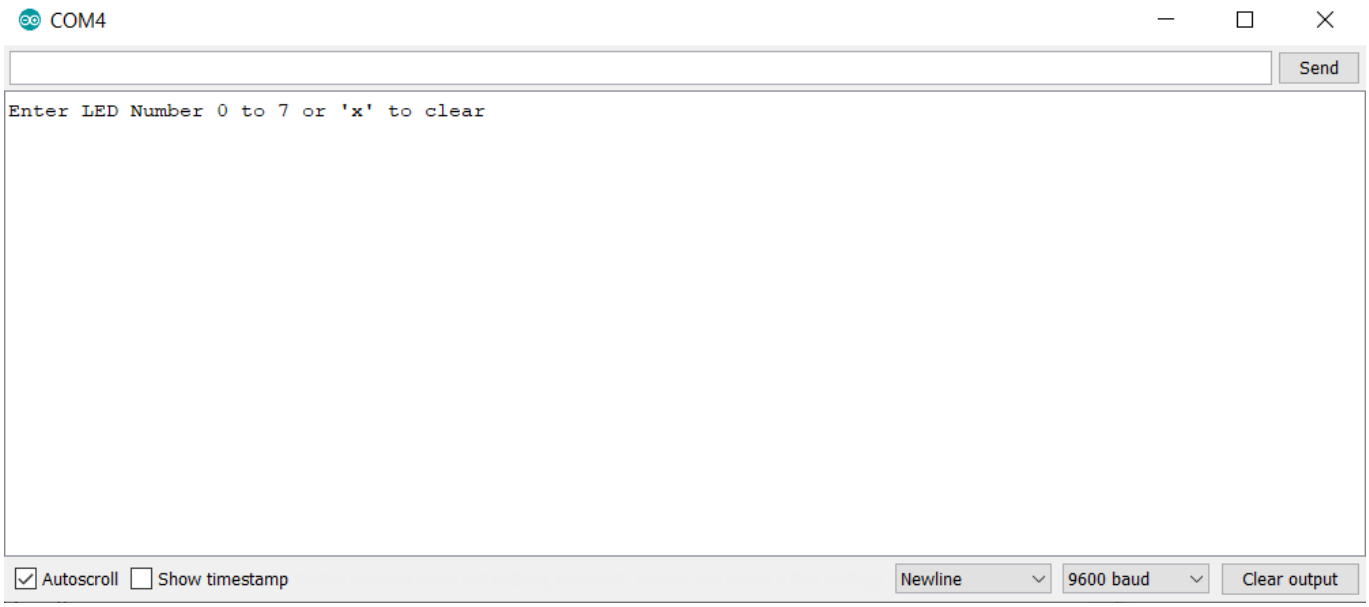
ارفع الكود التالي إلى الاردوينو، وسترى كيف سوف يعمل:

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

byte leds = 0;

void setup()
{
  pinMode(latchPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
  updateShiftRegister();
  Serial.begin(9600);
  while (! Serial);
  Serial.println("Enter LED Number 0 to 7 or 'x' to clear");
}

void loop()
{
  if (Serial.available())
  {
    char ch = Serial.read();
    if (ch >= '0' && ch <= '7')
    {
      int led = ch - '0';
      bitSet(leds, led);
      updateShiftRegister();
      Serial.print("Turned on LED ");
      Serial.println(led);
    }
    if (ch == 'x')
    {
      leds = 0;
      updateShiftRegister();
      Serial.println("Cleared");
    }
  }
}

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

```
    digitalWrite(latchPin, HIGH);
}
```

بعد رفع الكود على الاردوينو بنجاح اضغط على الزر الموجود يمين الشاشة بشريط الأدوات حتى تظر شاشة الإتصال التسلسلي (كما في الصورة):

سوف تظهر لك هذه النافذة التالية:



هذه النافذة تسمى "Serial Monitor" شاشة الإتصال التسلسلي وهو أحد خصائص برنامج الأردوينو. وظيفته هي السماح لنا بإرسال وإستقبال الرسائل من جهاز الكمبيوتر إلى الأردوينو عبر كيبل USB.

الرسالة "Enter LED Number 0 to 9 or 'x' to clear" تم ارسالها عبر الأردوينو، تطلب منا معرفة الأمر الذي سيتم تنفيذه. هل هو "X" (لإطفاء جميع الـ LEDs) أو ادخال رقم الـLED الذي ترغب بإضاءته (من 0 إلى 7)

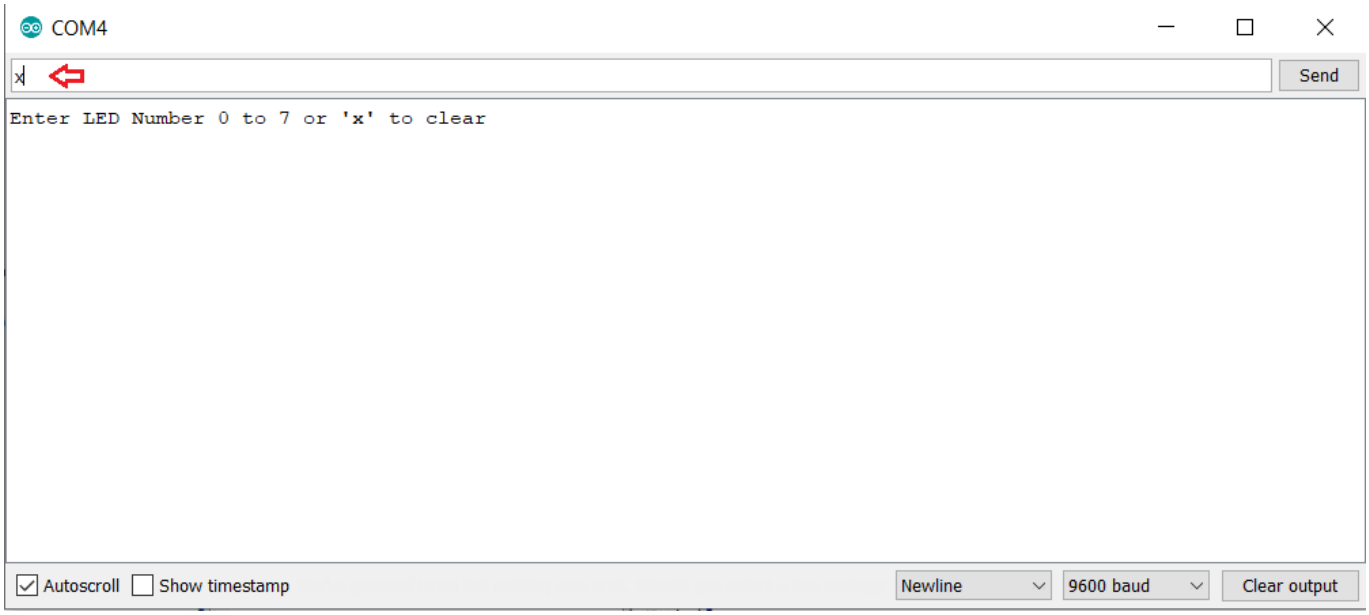
جرب ادخال الأمر التالي:

X

3

5

عملية ادخال X لن تؤثر لأن جميع الـ LEDs مطفأه، ولكن عند ادخال رقم الـLED المراد إضاءته سيقوم الأردوينو بالرد عليك برسالة تؤكد تشغيله كما تشاهد بالصورة التالية:



يمكنك تجربة الأمر X لتشاهد عملية الإطفاء لها جميعًا.

كما ستلاحظ سوف نعتمد على كود الدرس السابق وسوف تقوم فقط بتغطية الأجزاء الجديدة بالكود.

أولاً، دالة "setup" هنالك أربعة أسطر جديدة في نهايته

```
void setup()
    {
      pinMode(latchPin, OUTPUT);
      pinMode(dataPin, OUTPUT);
      pinMode(clockPin, OUTPUT);
      updateShiftRegister();
      Serial.begin(9600);
      while (! Serial);
      Serial.println("Enter LED Number 0 to 7 or 'x' to clear");
    }
```

أولاً لديك الأمر "Serial.begin(9600)" وهو يقوم ببدء عملية الإتصال التسلسلي ليمكن الأردوينو من إرسال الأوامر عبر كيبل الـUSB. القيمة 9600 هو معدل سرعة نقل البيانات، يمكنك تغيير القيمة لقيمة أعلى ولكن يجب عليك تغيير قيمة شاشة الإتصال التسلسلي لنفس القيمة.

السطر الثاني يبدأ بـ"While" الشرطية للتأكد من أن هنالك اتصال عبر الـUSB للأردوينو قبل بدء ارسال الرسائل.

السطر الثالث لطباعة السؤال بالشاشة.

دالة "loop

```
void loop()
    {
      if (Serial.available())
      {
        char ch = Serial.read();
        if (ch >= '0' && ch <= '7')
        {
          int led = ch - '0';
          bitSet(leds, led);
          updateShiftRegister();
          Serial.print("Turned on LED ");
          Serial.println(led);
        }
        if (ch == 'x')
        {
          leds = 0;
          updateShiftRegister();
          Serial.println("Cleared");
        }
      }
    }
```

كل شي يحدث داخل دورة الـ loop يحدث داخل الجملة الشرطية IF (اذا كان)،
"Serial.available()" للتأكد من انه ما زال هناك اتصال تسلسلي نشط (وذلك عبر استرجاع قيمة True – نعم) وعندها فقط يتم تنفيذ سلسلة الأوامر التي بداخلها.
إذا تم استقبال رسالة فسوف ينتقل لسطر الكود التالي:

```
char ch = Serial.read();
```

هذا الأمر سيقوم بقراءة الحرف الذي تم استلامه ويقوم بتخزينه للمتغير. ch ،المتغير ch تم تعريفه لتخزين القيم التي هي عبارة عن حروف char وهي اختصار لـcharacter "حرف " يتم تخزن حرف واحد فقط كقيمة للمتغير

السطر الذي يليله بالكود هي جملة IF الشرطية، وذلك للتأكد من أن القيمة المدخلة أعلى من 0 وأقل من 7. قد تبدو غريبة علينا الآن ولكن عبر الممارسة سنفهمها جيدًا.

كل حرف مدخل يحمل قيمة خاصة بالنظام، يدعى قيمة ASCII .وهذا يعني انه عند استخدام <= و => فنحن نقوم بالمقارنة بقيم ASCII

اذا نجح الاختبار فسوف ينتقل للسطر الذي يليه

```
int led = ch – '0';
```

يقوم باحتساب قيمة الـ LED = وذلك عبر عملية طرح الصفر من قيمة (ASCII) للحرف المدخل

لأنك تعلم برقم الـLED الذي ترغب بإضاءته كل ماعليك فعله هو وضع تلك القيمة لـleads والقيام بتحديث مسجل الإزاحة shift register.

```
 bitSet(leds, led);
    updateShiftRegister();
```

السطرين التاليين سيقومون بإرسال رسالة التأكيد على شاشة الإتصال التسلسلي serial monitor:

```
          Serial.print("Turned on LED ");
          Serial.println(led);
```

كمان تلاحظ لقد قمت بإستخدام Serial.print عوضًا عن Serial.println.الفرق بين الإثنين هو أن Serial.print لايقوم بوضع سطر جديد في الشاشة بعد طباعة الرسالة.
ولأنك نود أن نطبع في الشاشة رسالة التأكيد اضافة إلى رقم الـLED الذي تم اضاءته فيجب عليك ان تستمر بالطباعة على الشاشة في نفس السطر لذلك السطر الذي يليه قمت بإستخدام Serial.println لأنك ترغب بوضع سطر جديد بعد طباعة الرسالة ورقم الـLED

قيمة الـLED هو عبارة عن integer رقم صحيح "int"
بعد جملة IF الشرطية الأولى هنالك جملة IF شرطية ثانية تقوم بالتأكد ما اذا كان قيمة "ch" يحمل الحرف x اولا.

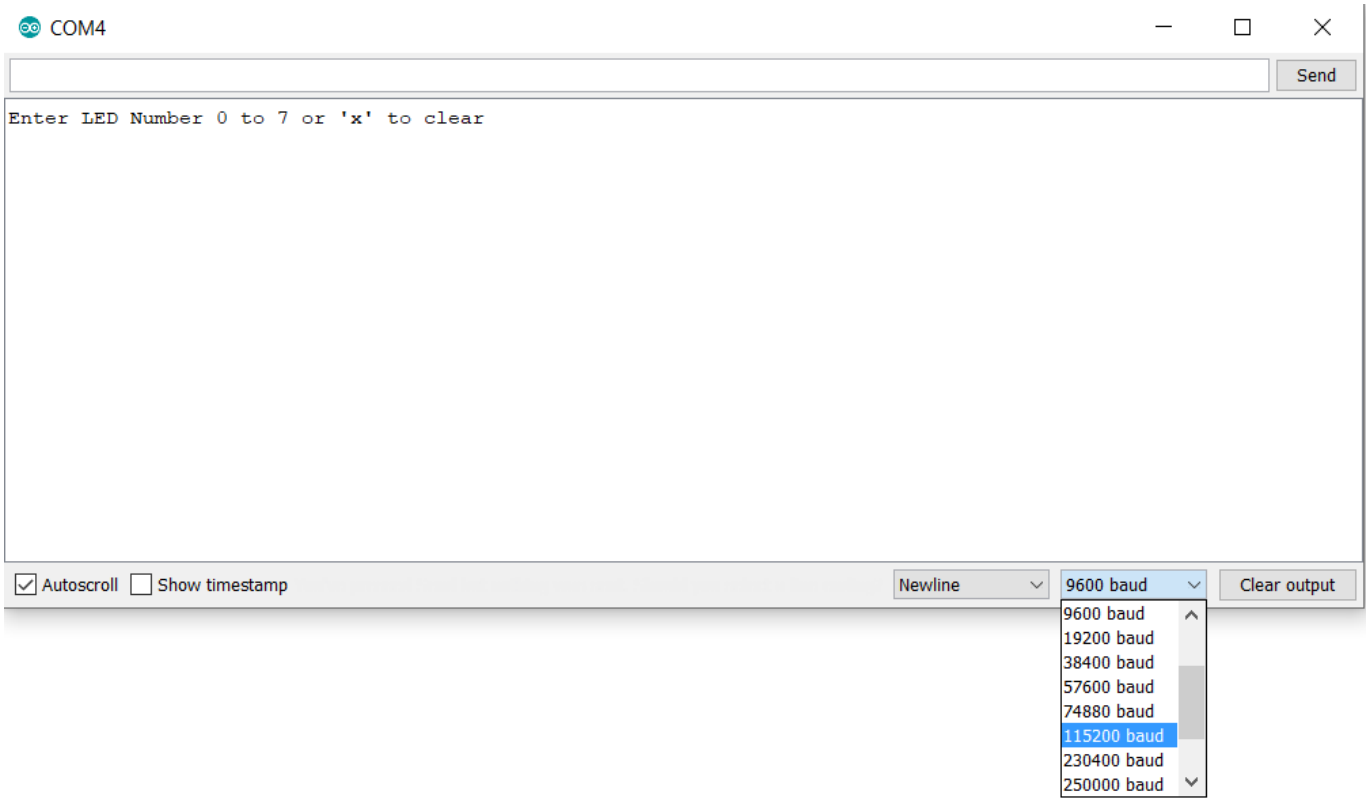
```
        if (ch == 'x')
        {
          leds = 0;
          updateShiftRegister();
          Serial.println("Cleared");
        }
```

أنشطة أخرى

ارسل الحروف منفصله وذلك عبر الضغط على إنتر Enter بعد كل حرف لارسال الأمر، ولكن يمكنك ارسالها جميعًا مرة واحدة في سطر واحد .. جرب ادخال التالي في شاشة الاتصال التسلسلي وارسلها:

لنرى الآن لأي مدى سرعة اتصال الأردوينو . غير معدل سرعة نقل البيانات في الكود من 9600 إلى 115200 وارفعه لمتحكم الأردوينو مره أخرى.

بعد ذلك افتح شاشة الاتصال التسلسلي وغير معدل سرعة نقل البيانات إلى 115200 (مثل الصورة)



ستلاحظ أنه لا يزال كل شي يعمل بشكل صحيح، معدل سرعات نقل البيانات العالية جدًا غير ضرورية لذلك 9600 هي الشائع استخدامًا والعديد من وحدات الـGPS تستخدم هذه السرعة.

كما يمكنك عدم توحيد معدل نقل البيانات بين شاشة الاتصال التسلسلي وكود الأردوينو، لتشاهد ماذا يحدث ولنعرف كيف يظهر الخطأ وما سببه.

شاشة الاتصال التسلسلي "Serial Monitor" هي طريقة جيدة أيضًا لتحليل أخطاء الكود وتصحيحها أيضًا.