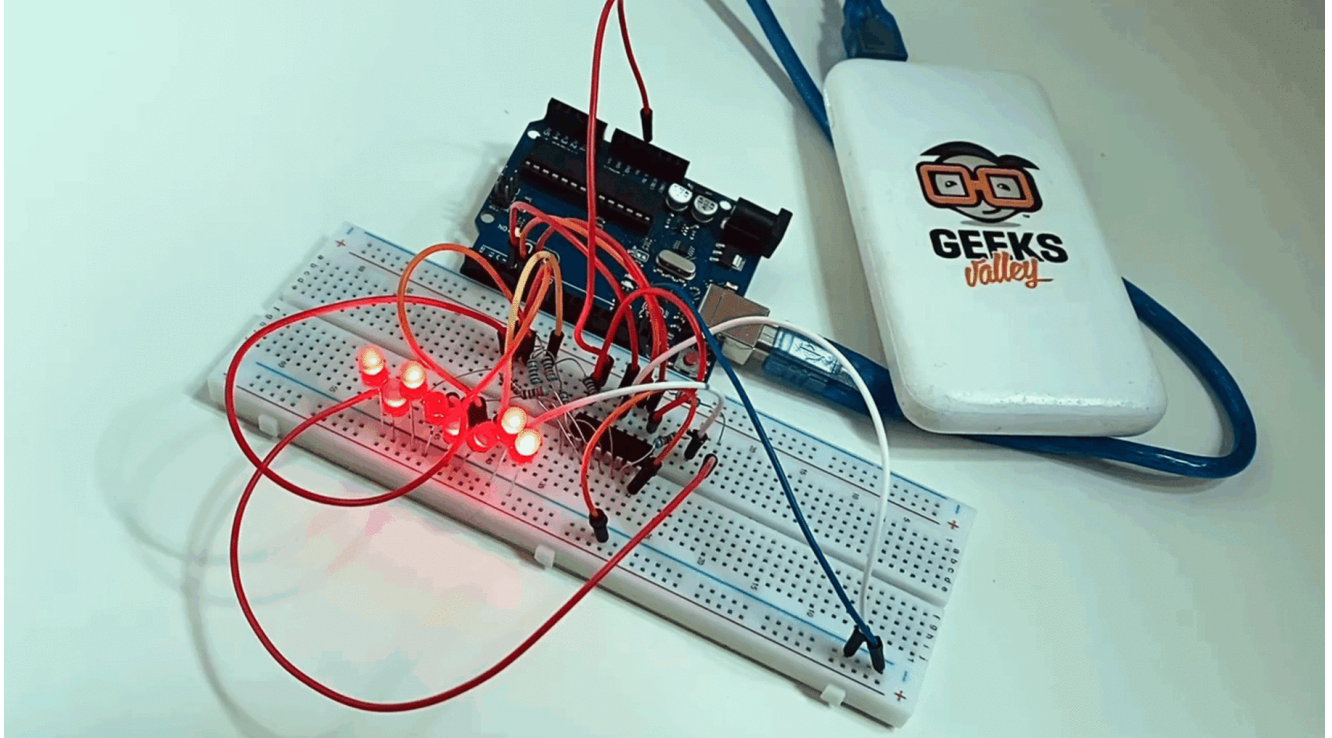


أردوينو – الدرس الخامس – إضاءة 8LEDs باستخدام رقاقة shift register

مقدمة

في هذا الدرس سنتعلم طريقة تشغيل 8LEDs حمراء بدون إستعمال 8 منافذ مخارج رقمية باستخدام رقاقة shift register. رغم أنه يمكنك ربط LED بمنفذ منفرد بالأردوينو ولكن في هذه الحالة لن يبقى لديك الكثير من المنافذ لاستعمالات أخرى. قد يكون الأمر ممكن في حالة عدم وجود الكثير من الأجهزة مرتبطة بالأردوينو، ولكنك في أغلب الأحيان تود إضافة أزرار، محركات ومستشعرات إلخ.. تمتلك رقاقة HC595 shift register ثمانية منافذ إستخراج وثلاثة منافذ إدخال.

ستقوم هذه الرقاقة في إبطاء سرعة الـ LEDs:
(من 8000000 إلى 500000 تحويله في الثانية ولكنها تبقى سريعة جداً ولن نلاحظ الفرق)



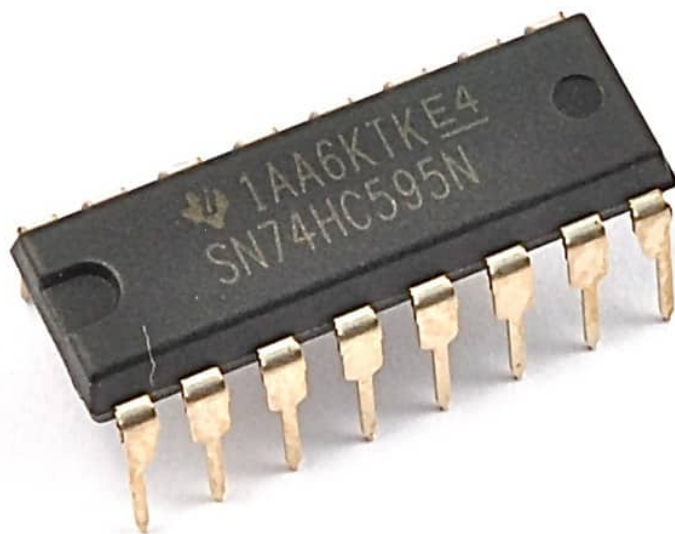
المواد والأدوات



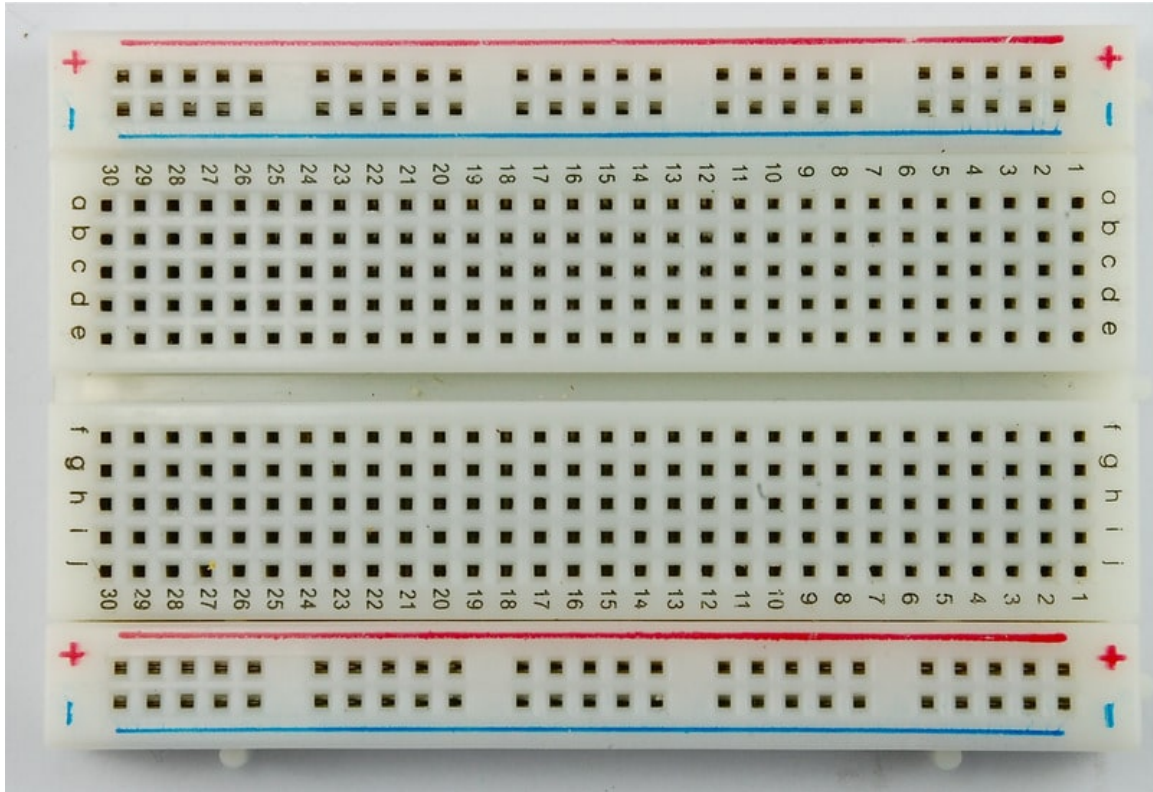
(5mm Red LED) ×8



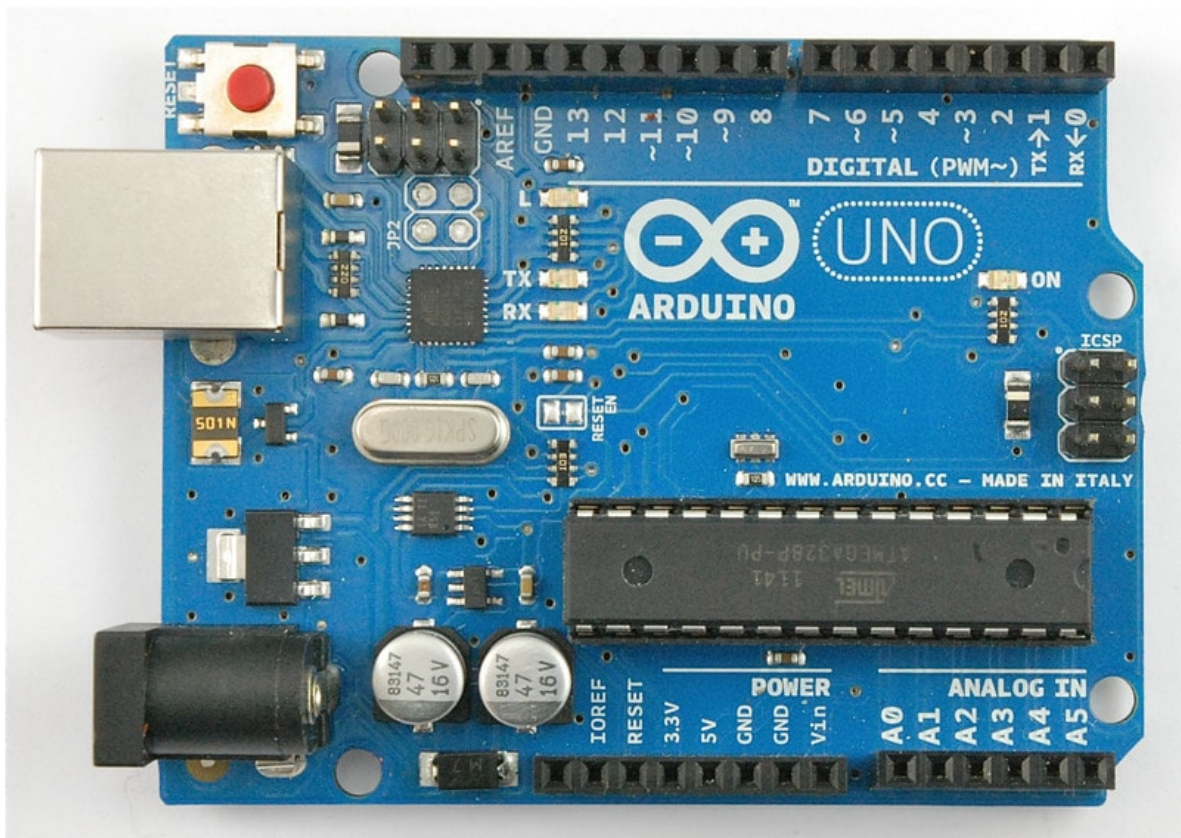
×Ω Resistor) 8 270)



×1 رقاقة مسجل الإزاحة (74HC595 shift register)



1x لوحة التجارب (Half-size Breadboard)



1x اردوينو اونو



حزمة أسلاك توصيل (ذكر-ذكر)

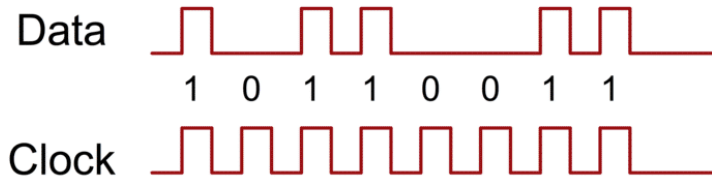
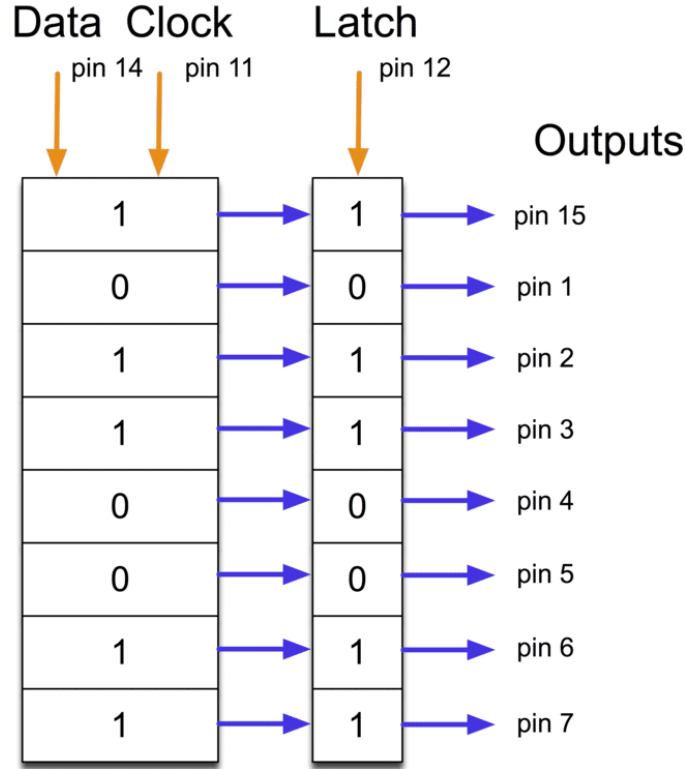


1 × سلك اردوينو

74HC595 shift register

من الافضل فهم طريقة عمل رقاقة مسجل الإزاحة shift register حتى تفهم الكود جيداً.

الرقاقة Shift register: هي عبارة عن مسجل للإزاحة.



تحتل 8 أماكن تعتبر مثل الذاكرة الصغيرة، يمكن لكل واحدة فيها أن تسجل 1 أو 0 كقيم m

لننقل كل من هذه الخانات بالقيمة المراد وضعها، ندخل كل من المعلومات عبر منفذ "Data" ومنفذ "Clock".

على منفذ الـ "Clock" أن يستقبل 8 نبضات، إذا كان منفذ "Data" عالي فسيتم إدخال القيمة "1" إلى داخل مسجل الإزاحة.

بعد إستقبال النبضات الثمانية تعطى الإشارة لمنفذ "Latch" لتخرج الإشارة لجميع الـ LEDs الثمانية دفعة واحدة.

تمتلك الرقاقة منفذ "OE" (إختصار لـ output enable) وهي التي تتحكم في كل المخرجات معاً. يمكن لنا التحكم في البريق الكلي للـ LED إذا قمنا بربط منفذ "OE" بمنفذ إستخراج "MWP" من الاردوينو وقمنا بإستخدام الدالة "analogWrite" بالكود فسوف يعمل الـ LED على مستوى منخفض، لذلك سنربطها بالأرضي.

توصيل الدائرة

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

المرحلة الأخيرة هي إضافة الأسلاك كما في الصورة لا تنسى أن تربط المنفذ 8 من الرقاقة إلى خانة الأرض أو GND. حمل الكود وجربه على كل LED أن يعمل مرة واحدة ثم الكل معاً وأخيراً ينطفئ الكل معاً قبل أن يعيد الـ Arduino الدورة.

الكود البرمجي

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

byte leds = 0;

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

void loop()
{
  leds = 0;
  updateShiftRegister();
  delay(500);
  for (int i = 0; i < 8; i++)
  {
    bitSet(leds, i);
    updateShiftRegister();
    delay(500);
  }
}

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

تمتلك مكتبة الـ Arduino دالة "مميزة تسمى "ShiftOut" وهي مصممة خصيصاً لإرسال البيانات لرقاقة مسجل الإزاحة shift register.

أول ما عليك فعله هو التعريف بالمنافذ الثلاث التي ستستعملها. هذه هي منافذ الـ Arduino التي ستستعملها للتحكم في "Latch"، "Clock" ومنفذ "Data" من الرقاقة 74HC595.

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

ثم يأتي التعريف بالمتغير المسمى Leds ستستعمل هذا المتغير لإحتواء نمط تغيير (تشغيل أو اطفاء) الـ LED. تمثل البيانات من نوع " - Byte بايت" أرقام تكتب بإستعمال 8 أجزاء. يمكن أن يحمل كل جزء قيمة 1 أو 0

وهذا سيمكنك من تتبع قيمة كل LED.

```
byte leds = 0;
```

تقوم دالة التنصيب "Setup" هذه بإعداد المنافذ الثلاث التي ستستخدمها كمخرجات للـ LEDs.

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

تقوم دالة "Loop" بإطفاء كل LED وذلك بإعطاء قيمة '0' كقيمة للـ LEDs ثم تقوم بمناداة دالة "updateShiftRegister" التي سترسل نمط Leds إلى رقاقة مسجل الإزاحة حتى تنطفئ كل LED. سوف تتعلم كيفية عمل "updateShiftRegister" لاحقاً.

تقف دالة "Loop" لنصف ثانية ثم تبدأ بالعد من 0 إلى 7 عبر دورة "For" والمتغير "i" وفي كل مرة تستعمل وظيفة "bitSet" لتثبيت الجزء المتحكم في الـ LED داخل متغير Leds. ثم تنادي دالة "updateShiftRegister" حتى يتم تحديث حالة الـ LED التي تعكس قيمة المتغير "leds".

هناك فترة إنتظار نصف ثانية في كل تكرار للدورة "For".

```
void loop()
{
  leds = 0;
  updateShiftRegister();
  delay(500);
  for (int i = 0; i < 8; i++)
  {
    bitSet(leds, i);
    updateShiftRegister();
    delay(500);
  }
}
```

تقوم دالة UpdateShiftRegister بوضع منفذ "Latch" إلى الوضع المنخفض ثم تنادي دالة shiftOut قبل أن تعيد منفذ "Latch" إلى الوضع العالي مره أخرى، تأخذ هذه الدالة أربع متغيرات، تحدد المتغيرتان الأولى المنافذ المستعملة لنقل البيانات عبر "Data" و "Clock".

تحدد المتغيرة الثالثة جهة البدء بقراءة البايت سوف نقرأ البايت من اليمين إلى اليسار.

المتغير الأخير يمثل القيمة الفعلية التي سترسل إلى رقاقة مسجل الإزاحة shift register.

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

إذا كنت تود تغيير قيم أحد الـ LEDs، كل ما عليك هو أن تنادي دالة مشابهة اسمها (bitClear) لتغيير قيمتها إلى 0 هكذا ستطفئ واحدة من الـ LEDs وكل ما يتبقى عليك فعله هو مناداة دالة "updateShiftRegister" لتحديث حالة الـ LED.

التحكم في البريق:

لم نتحدث كثيرًا في رقاقة الـ 74HC595 عن منفذ "OE" أو "OutputEnable"، رقم هذا الـ LED هو 13، وفي لوح التجارب مربوط دومًا بالأرض. يعمل هذه المنفذ كالمحول، هو الذي يمكن له أن يطفئ ويشغل المخرجات الـ (LEDs) ولكن عليك أن تحذر لأنها تعمل في الوضع المنخفض. ذلك يعني، إن كانت كلها مربوطة بطاقة 5V ستطفئ جميعها. وإن كانت مربوطة بالمنفذ الأرضي ستعمل جميعها حسب ما يأتيها من بيانات.

يمكن لك أن تستعمل هذه المنفذ والإستعانة بدالة "analogWrite" للتحكم في بريق إضاءة الـ LED معًا.

كل ما عليك فعله هو تغيير المنفذ 13 من الأرض إلى المنفذ 3 في الأردوينو.

سيجعل الكود التالي كل الـ LEDs تعمل ببطء ثم تنطفئ ببطء.

```
int latchPin = 5;
int clockPin = 6;
int dataPin = 4;
int outputEnablePin = 3;

byte leds = 0;

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

void loop()
{
  setBrightness(255);
  leds = 0;
  updateShiftRegister();
  delay(500);
  for (int i = 0; i < 8; i++)
  {
    bitSet(leds, i);
    updateShiftRegister();
    delay(500);
  }
  for (byte b = 255; b > 0; b--)
  {
    setBrightness(b);
    delay(50);
  }
}

void updateShiftRegister()
{

```

```
digitalWrite(latchPin, LOW);  
shiftOut(dataPin, clockPin, LSBFIRST, leds);  
digitalWrite(latchPin, HIGH);  
}  
  
void setBrightness(byte brightness) // 0 to 255  
{  
    analogWrite(outputEnablePin, 255-brightness);  
}
```

أنشطة أخرى

عندما تتمكن من فهم رقاقة مسجل الإزاحة shift register ستجد العديد من التطبيقات التي يمكن لك فعلها بإستعمال عدد كبير من الـLEDs.

واحدة من الأشياء الرائعة في رقاقة 74HC595 أنه يمكن وضعها بجانب بعضها والتحكم في عدد أكبر من الـLEDs بإستعمال نفس منافذ المخرج من الأردوينو.