

Le 74HC595 : Augmenter le nombre de sorties d'un Arduino

Matériels :

- 1 arduino UNO
- 1 ou 2 CI 74HC595
- 8 ou 16 LEDS
- 1 Breadboard , fils de liaison

Fonctionnement normal

Le but de ce montage est de d'augmenter le nombre de sorties d'un Arduino UNO .

Avec 3 sorties Arduino on peut commander 8 ou 16 sorties , dans notre exemple nous utiliserons 8 ou 16 LEDs avec 1 ou 2 CI [74HC595](#)

voir aussi le : [MCP23017](#) et le [PCF8574](#) comme alternative

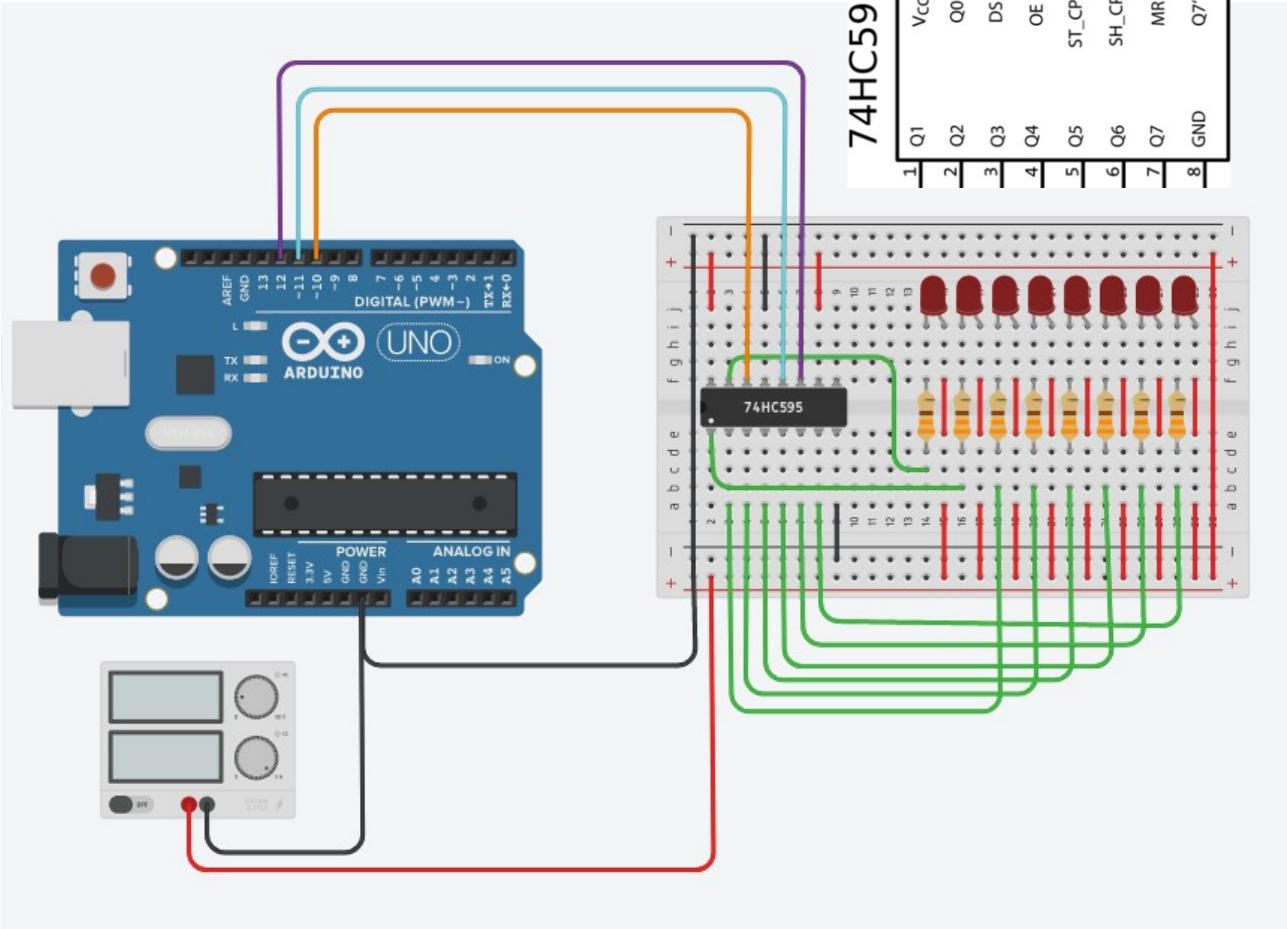
Montage à Réaliser :

74HC595



PIN	SYMBOL	DESCRIPTION
1, 2, 3, 4, 5, 6, 7 and 15	$Q_1, Q_2, Q_3, Q_4, Q_5, Q_6, Q_7$ and Q_0	parallel data output
8	GND	ground (0 V)
9	Q_7'	serial data output
10	\overline{MR}	master reset (active LOW)
11	SH_{CP}	shift register clock input
12	ST_{CP}	storage register clock input
13	\overline{OE}	output enable input (active LOW)
14	D_S	serial data input
16	V_{CC}	DC supply voltage

Pour commander 8 LEDs



Programme 74HC595.ino

```
// Broche connectée au ST_CP du 74HC595
const int verrou = 11;
// Broche connectée au SH_CP du 74HC595
const int horloge = 12;
// Broche connectée au DS du 74HC595
const int data = 10;

void setup()
{
    // On met les broches en sortie
    pinMode(verrou, OUTPUT);
    pinMode(horloge, OUTPUT);
    pinMode(data, OUTPUT);
}

void loop()
{
    // on affiche les nombres de 0 à 255 en binaire
    for (char i = 0; i<256; i++)
    {
        // On active le verrou le temps de transférer les données
        digitalWrite(verrou, LOW);
        // on envoie toutes les données grâce à notre belle fonction
        envoi_ordre(data, horloge, 1, ~i);
        // et enfin on relâche le verrou
        digitalWrite(verrou, HIGH);
        // une petite pause pour constater l'affichage
        delay(1000);
    }
}

void envoi_ordre(int dataPin, int clockPin, boolean sens, char donnee)
{
    // on va parcourir chaque bit de l'octet
    for(int i=0; i<8; i++)
    {
        // on met l'horloge à l'état bas
        digitalWrite(clockPin, LOW);
        // on met le bit de donnée courante en place
        if(sens)
        {
            digitalWrite(dataPin, donnee & 0x01 << i);
        }
        else
        {
            digitalWrite(dataPin, donnee & 0x80 >> i);
        }
        // enfin on remet l'horloge à l'état haut pour
        // faire prendre en compte cette dernière
        digitalWrite(clockPin, HIGH);
    }
}
```

Autre programme avec shiftOut

```
// Broche connectée au ST_CP du 74HC595
const int verrou = 11;
// Broche connectée au SH_CP du 74HC595
const int horloge = 12;
// Broche connectée au DS du 74HC595
const int data = 10;

void setup()
{
    // On met les broches en sortie
    pinMode(verrou, OUTPUT);
    pinMode(horloge, OUTPUT);
    pinMode(data, OUTPUT);
}

void loop()
{
    // on affiche les nombres de 0 à 255 en binaire
    for (int i = 0; i < 256; i++)
    {
        // On active le verrou le temps de transférer les données
        digitalWrite(verrou, LOW);
        // on envoi toutes les données grâce à shiftOut
        // (octet inversée avec '~' pour piloter les LED à l'état bas)
        shiftOut(data, horloge, LSBFIRST, ~i);
        // et enfin on relache le verrou
        digitalWrite(verrou, HIGH);
        // une petite pause pour constater l'affichage
        delay(1000);
    }
}
```