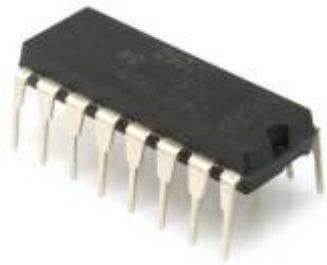
CIRC-05

.:8 LEDs de Plus:.

.:Registre à Décalage 74HC595:.



Ce que nous allons faire:

Il est temps de s'amuser avec les puces, ou circuits intégrés (CI) comme on les appelle. Le boîtier externe peut être très trompeur. Par exemple regardez la puce de la carte Arduino (un microcontrôleur) et celle que nous allons utiliser dans ce circuit (un registre à décalage) qui lui ressemble beaucoup mais qui en est très differente. Le prix de l'ATMega sur l'Arduino coûte quelques euros alors que le 74HC595 coûte quelques dizaines de centimes. C'est une bonne puce pour commencer et lorsque vous serez à l'aise avec le composant ainsi que sa datasheet (disponible sur http://ardx.org/74HC595), le monde des puces vous sera ouvert. Le registre à décalage vous donnera 8 sorties additionnelles (pour contrôler des LEDs par exemple) en utilisant seulement 3 pattes de votre Arduino. Ils peuvent aussi être connectés ensemble pour vous permettre d'avoir un nombre de sorties proche de l'infini. Pour l'utiliser vous envoyez les données et les stockez dans la puce. Pour ce faire vous changez l'état de la patte de données (HIGH et LOW), vous lui donnez une horloge jusqu'à ce que les 8 bits soient envoyés. Ensuite vous envoyez une impulsion sur la patte de verrouillage (latch) et les 8 bits sont stockés. Cela parait compliqué mais c'est relativement simple.

Le Circuit:

Composants :

CIRC-05
Feuille de Connexions
x1

Connecteur 2 patte
x4

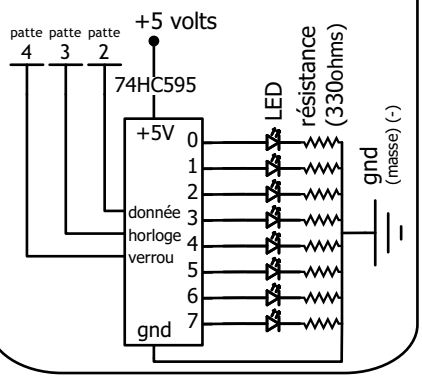
Registre à Décalage
74HC595
x1

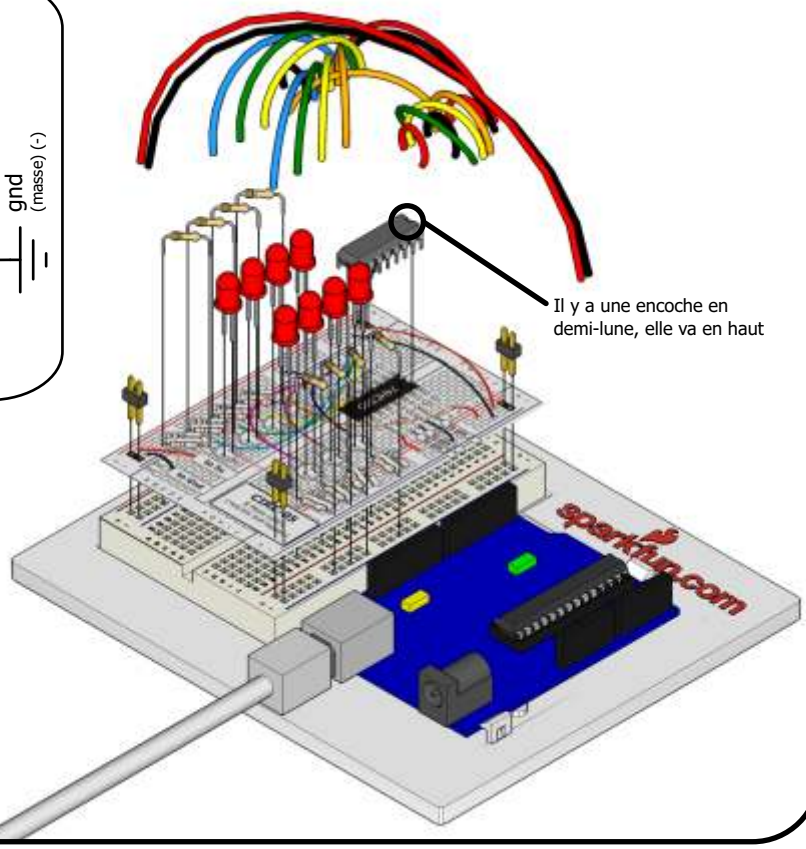
Fil

LED Rouge
x8

Résistance 330 Ohms
Orange-Orange-Marron
x8

Schéma





Internet

.:Télécharger:.
Feuille de Connexions
http://ardx.org/BBLS05S
.:Voir:.
Vidéo de Montage
http://ardx.org/VIDE05

CODE (Il n'est pas nécessaire de taper quoi que ce soit)

Téléchargez le code (http://ardx.org/CODE05)

(et ensuite copiez le texte dans un nouveau sketch Arduino)

```
//Sélection des pattes
//le 74HC595 utilise une liaison
série//connexion qui utilise 3 pattes
int data = 2;
int clock = 3;
int latch = 4;

void setup()  //s'exécute une fois< br>{
  pinMode(data, OUTPUT);
  pinMode(clock, OUTPUT);
  pinMode(latch, OUTPUT);  }

void loop()   //s'exécute à l'infini< br>{
  int delayTime = 100;
     //attente de la mise à jour des LED<
br>  for(int i = 0; i < 256; i++){
   updateLEDs(i);
   delay(delayTime);   }
}

/*
 * updateLEDs() - envoie les données des
LEDs< br> *en sequence au 74HC595< br> */
void updateLEDs(int value){
 digitalWrite(latch, LOW);

//met le verrou (latch) au niveau bas
  shiftOut(data, clock, MSBFIRST, value);
     //envoie 8 bits au registre à décalage< br>

digitalWrite(latch, HIGH);
     //met le verrou(latch) au niveau haut pour
démarrer l'affichage des donnees< br>}

  ----------Plus de Code en Ligne ----------
```

CELA FONCTIONNE PAS ? (3 choses à essayer)

La LED de l'Arduino s'éteint

C'est arrivé une ou deux fois, cela arrive quand la puce est branchée à l'envers. Si vous le changez rapidement, rien ne sera abimé.

Cela fonctionne toujours pas ?

Désolé pour la répétition mais c'est surement dû à de mauvaises connexions

Frustré?

Envoyez-nous un mail, ce circuit est à la fois simple et complexe en même temps. Nous voulons avoir des informations sur vos problèmes pour en tenir compte dans les prochaines versions.

AMÉLIORER LE MONTAGE

La manière forte :

Une Arduino permet de simplifier quelque peu les actions complexes. Le déplacement de données est l'une d'entre elles. Cependant l'un des attraits d'Arduino est sa capacité à rendre les choses faciles ou difficiles selon vos besoins. Essayons sur un exemple. Dans la boucle de switch, la ligne :

```
updateLEDs(i) -> updateLEDsLong(i);
```

Télécharge le programme et vous signal que rien n'a changé. Si vous regardez le code vous pouvez voir comment nous communiquons avec la puce, bit par bit. (pour plus de détails http://ardx.org/SPI).

Contrôle de LEDs individuelles:

Il est temps de commencer à controler des LEDs avec une méthode similaire que la méthode utilisée dans CIRC02. Comme l'état des huit LED est stocké dans un octet (une valeur 8 bit) pour plus de détails sur comment ça marche esssayez http://ardx.org/BINA. Une Arduino est parfaitement adaptée pour manipuler des bits et il existe une panoplie complète d'opérateurs pour nous aider. Plus de details sur les operations sur les bits (http://ardx.org/BITW).

Notre implémentation.
Remplacez le code `loop()` avec

```
int delayTime = 100; //le nombre de millisecondes
                     //de délais
                     //entre les mises à jour LED
 for(int i = 0; i < 8; i++){
   changeLED(i,ON);
   delay(delayTime);
 }
 for(int i = 0; i < 8; i++){
   changeLED(i,OFF);
   delay(delayTime);
 }
```

Télécharger ce code entraînera l'allumage de la lumière sur l'une puis l'autre et l'extinction de la même manière. Étudiez le code et wikipedia pour voir comment ça marche, ou envoyez nous un e-mail si vous avez des questions.

Plus d'animations :

Maintenant les choses deviennent intéressantes. Si vous regardez en arrière le code du (8 LED Fun) vous verrez que l'on change les LEDs en utilisant digitalWrite(led, state), elle est composée des mêmes éléments que la fonction changeLED(led, state) que nous avions écrit. Vous pouvez utiliser l'animation que vous aviez écrit pour le CIRC02 en copiant le code dans ce sketch et en changeant tous les digitalWrite() en changeLED()'s. Puissant ? Très ! (vous devrez également changer quelques autres choses mais en suivant les erreurs du compilateur ça se fait tout seul).

PLUS, PLUS, PLUS :

Plus de détails, où acheter des composants, où poser plus de questions :

http://ardx.org/CIRC05