



29/05/2016

Compte-rendu Projet Arduino

Lampe colorée



Florian Quinart & Jean Durand
ELECTRONIQUE ARDUINO

Table des matières

Introduction.....	2
Développement du projet	2
Partie électronique	3
Partie informatique.....	6
Les plus et les moins de notre projet	6
Les problèmes rencontrés au cours du projet	6
Sources	7

Introduction

D'où nous est venue cette idée de lampe colorée ? Au départ, nous voulions faire un projet réalisable, que nous pourrions finir. C'est pourquoi nous sommes partis sur l'idée de travailler autour de la lumière et des couleurs. Le but de notre projet était de créer une lampe reproduisant la couleur passée devant un capteur lui étant dédié, avec un support adapté. Nous aurions aussi aimé rajouter à cette lampe quelques mélodies.

Développement du projet

Au bout d'une séance de cours nous avons établis qu'il serait nécessaire de se répartir des tâches, trois aspects principaux constituaient notre projet :

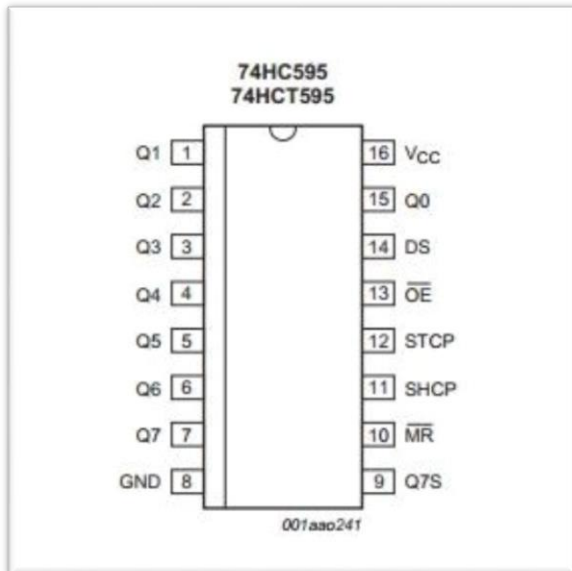
La détection des couleurs : Nous nous sommes procurés un capteur de couleur de référence TCS 230D. Ce dernier nous était indispensable dans la création de notre projet étant donné que l'on souhaitait moduler la couleur de plusieurs LEDS selon une couleur de carte (Pantone) souhaitée. Nous avons donc eu besoin de la datasheet de ce composant, une datasheet très dense par ailleurs. Après plusieurs tests nous avons établis un équilibre de reconnaissance des couleurs pour que les couleurs secondaires ressortent au mieux avec nos LEDS.

Le contrôle des LEDS : Cette partie fut la plus technique à mettre en œuvre. En effet, nous avons eu besoin de deux registres à décalage pour pouvoir contrôler 4 LEDS à la fois. Il nous a fallu comprendre ce que le registre pouvait faire exactement. Cette partie est très intéressante car ici elle nous sert à contrôler plusieurs LEDS en même temps à l'aide de seulement trois sorties de l'arduino. Mais bon nombre d'applications, ayant besoin d'un registre, existent. Nous pourrions en effet contrôler d'autres composants en même temps que les LEDS, comme des servomoteurs ou des haut-parleurs. Nous avons aussi eu l'idée d'intégrer à notre lampe un VU-mètre mais par manque de temps il n'a pas été réalisé.

La communication avec l'objet : Un des objectifs que nous nous étions fixés était de pouvoir communiquer à distance avec l'objet. C'est pourquoi un aspect de communication à distance a été mis en place avec le module bluetooth. Nous voulions absolument intégrer ce module bluetooth car de nos jours l'ensemble des objets que nous possédons est doté d'un module de communication homme-machine.

Partie électronique

Le 74HC595, un registre à décalage dédié à l'arduino :



L'alimentation du 74HC595 se fait sur la pin 16 sous une tension allant de -0.5 à +7V (d'après la datasheet). Pour ce qui est des fonctions principales du registre à décalage, de l'enregistrement du registre et des sorties à 3 états, on utilisera les 8 pins de sortie parallèles (1 à 7 + 15), plus les 3 à 5 pins de contrôle.

Voici en gros la séquence effectuée pour l'envoi d'une donnée sur 8 bits en sortie du 74HC595 : on passe d'abord ST_CP (la latch) à 0, puis on présente un par un les 8 bits de la donnée sur DS (entrée série) en envoyant en même temps un 1 logique sur SH_CP (l'horloge de décalage) à chaque fois, on repasse ensuite ST_CP (la latch) à 1 pour recopier la nouvelle valeur dans le second registre.

La pin 10 (/MR) permet de remettre tout le premier registre à zéro et est active au niveau bas, il convient donc (pour éviter une remise à zéro permanente du registre) de placer cette pin au niveau haut (C'est pourqu'on la relie à +VCC).

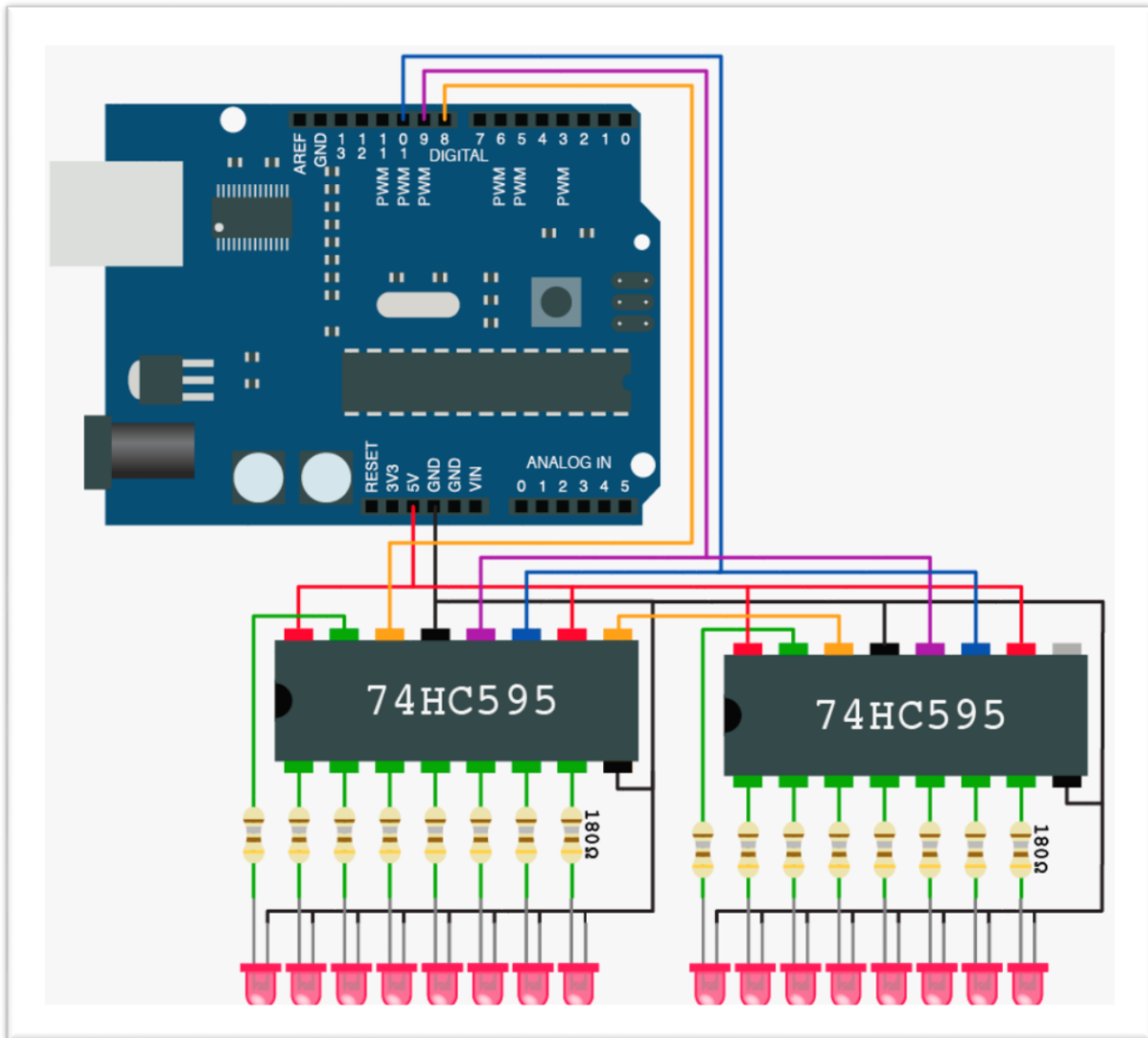
La pin 13 (/OE) active ou non la sortie (elle est elle aussi active au niveau bas). On placera cette fois cette pin sur la masse.

La pin /OE contrôle quant à elle les sorties à 3 états. Le fonctionnement des sorties à 3 états est simple. Pour deux des états, il s'agit des classiques 0 et 1 logiques. Dans le cas où /OE est au niveau bas, cette sortie est un simple fil.

Par contre, dans le cas où /OE est au niveau logique haut (quand la sortie est désactivée), cet étage atteint un état de haute impédance qui n'est ni zéro, ni un mais qui a le bénéfice d'isoler

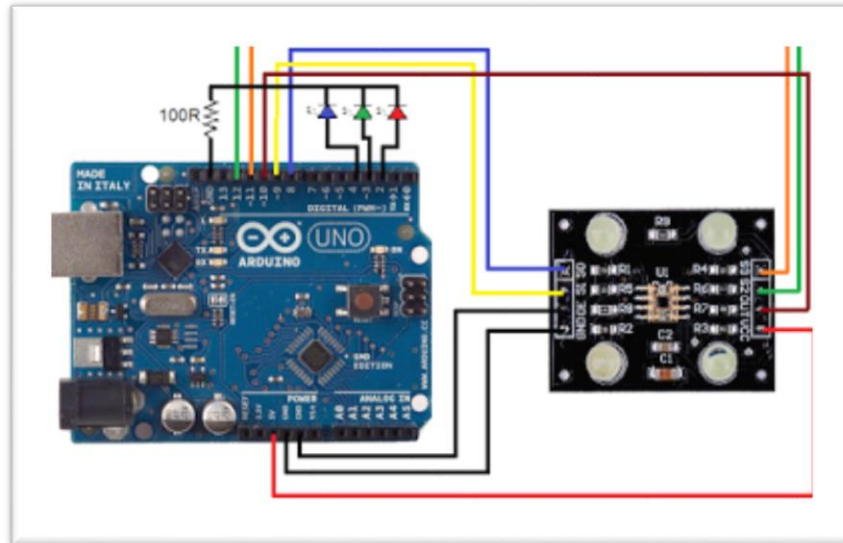
efficacement le 74HC595 du reste du circuit. La pin 9 du 74HC595 correspond à sa sortie série. En utilisant cette sortie série, on peut chaîner les 74HC595 et avoir ainsi des registres à décalage sur 16, 24, 32 etc... bits.

Il suffit alors de câbler les registres comme tel :



D'ailleurs pourquoi avoir utilisé 2 registres à décalage pour faire une chose assez simple de prime abord : contrôler des LEDS ? Utiliser des registres à décalage ne fut pas intuitif mais c'est en réfléchissant que nous est venue l'idée car nous voulions allumer des LEDS RGB et pour cela il nous fallait des sorties analogiques pour recevoir une valeur numérique entre 0 et 255 sur chaque diode rouge, verte et bleue contenue dans une seule LED. En plus de cela il fallait gérer la persistance rétinienne afin que les couleurs apparaissent proprement. Les registres à décalage furent donc une obligation pour contrôler plusieurs LEDS sachant que de plus il n'y avait pas assez de sorties analogiques aux ports de l'arduino pour contrôler plus d'une LED.

Le TCS230D, un capteur de couleur simple d'utilisation avec Arduino :



Le capteur est constitué de plusieurs broches : So, S1, S2, S3, OUT.

Les branches So et S1 servent à augmenter ou diminuer une échelle de puissance du capteur.

Les broches S2 et S3 servent à imposer au capteur la détection d'une couleur, ce dernier nous renverra le taux de cette couleur dans la couleur qui lui est présentée. C'est pourquoi nous changeons, dans la partie code, les états de S2 et S3 afin de récupérer la valeur de chaque teinte comme expliqué ci-dessous :

S2	S3	Color
L	L	Red
L	H	Blue
H	L	Clear
H	H	Green

Ce que nous renvoie ce capteur n'est autre qu'une courbe que nous analysons dans la partie informatique pour ainsi la traiter.

Partie informatique

D'une part, pour ce qui est du registre à décalage, une librairie a été utilisée afin de rendre la programmation de ce dernier beaucoup plus simple et compréhensible. Grâce à cette librairie, tout l'aspect de la gestion de la persistance rétinienne était beaucoup plus simple et clair (la librairie est expliquée en commentaires dans le code arduino). De plus de nombreuses fonctions de cette librairie furent simples d'utilisation et le code fut par conséquent plus simple à mettre en place même si le principe du registre à décalage en lui-même restait le même.

D'autre part on récupère la valeur renvoyée par le capteur de couleur à l'aide de la fonction « pulsein » qui nous renvoie une valeur pour chaque couleur en microseconde.

On passe ensuite cette valeur en KHz comme expliqué dans la datasheet. Or en comparant les valeurs données par la datasheet et celle trouvées expérimentalement pour les maxima et les minima de chaque longueur d'onde, on remarque qu'elles sont différentes on a alors dû élaborer une fonction très spécifique pour obtenir les couleurs les mieux approximées et ainsi les retransmettre aux diodes rouges, vertes et bleues de la LED. Cette partie de code fût assez laborieuse, elle nous a pris beaucoup de temps notamment pour tous les tests effectués sur les différentes couleurs afin de savoir exactement quoi changer dans la fonction pour obtenir un meilleur rendu de couleur.

Les plus et les moins de notre projet

Il est vrai que notre projet peut paraître simple de prime abord, or il est très flexible dans le sens ou beaucoup d'améliorations peuvent être apportées. En effet il y a matière à améliorer ce prototype ou mieux encore, changer les composants contrôlés par les registres à décalage. En effet nous pourrions contrôler n'importe quel composants à la place des LEDS et ainsi leur accorder à chacun des sorties PWM. Ce projet nous a permis de nous rendre compte qu'avec de simples composants, énormément de projets sont réalisables.

Pour ce qui est du côté négatif ; nous aurions aimé avoir une carte Arduino un peu plus performante que celle que nous avions à disposition. Nous avons en effet été restreints par cette dernière qui ne supportait pas la mise en place de librairies n'utilisant pas les même timer de l'Arduino (explication plus de détails dans la partie problèmes).

Les problèmes rencontrés au cours du projet

De prime abord, il semblerait que notre projet soit simple. Or nous avons rencontrés certains problèmes, notamment au niveau du capteur de couleur. Ce dernier nous a posé certains problèmes dans le fait où son manque de précision nous a fait défaut. Nous avons donc dû y remédier par nous-même et faire en sorte d'obtenir un rendu final satisfaisant.

Le deuxième problème majeur fut la mise en corrélation des parties Bluetooth + registre et capteur de couleur. Nous nous sommes finalement rendu compte que nous ne pouvions pas rassembler les 2 parties car les programmes utilisés pour chacun de nos parties utilisaient des temps de l'arduino différents (dû aux librairies).

Sources

[capteur de couleur](#)

[Registre à décalage](#)