Université François-Rabelais de Tours
Institut Universitaire de Technologie de Tours
Département Génie Électrique et Informatique Industrielle

Nuage décoratif à LED

Une nouvelle façon de décorer son intérieur

Mathieu PUILLANDRE
Groupe K4B
Promotion 2012/2015

Thierry LEQUEU

Université François-Rabelais de Tours
Institut Universitaire de Technologie de Tours
Département Génie Électrique et Informatique Industrielle

Illustration 1 : "Cloud" Par Richard Clarkson

Nuage décoratif à LED

Une nouvelle façon de décorer son intérieur

Mathieu PUILLANDRE
Groupe K4B
Promotion 2012/2015

Thierry LEQUEU

Sommaire

Introduction

Ce document traite de la réalisation d'une lampe décorative prenant la forme d'un nuage dans le cadre du cours d'Étude et Réalisation du semestre 4. Un projet similaire a déjà été réalisé par Richard CLARKSON. Avant de consulter ce rapport je vous invite à aller voir sa réalisation sur son site internet : *http://www.richardclarkson.com/cloud/*.

Vous aurez l'occasion de constater, tout au long de ce rapport que le nuage n'aura pas les mêmes fonctionnalités que celui de Richard CLARKSON. En effet, la finalité de ce projet est d'avoir une lampe « intelligente » pouvant offrir à l'utilisateur différentes fonctions comme la mise en place d'une ambiance orageuse par les effets de lumière, de fonctions permettant l'imitation de l'ambiance Coucher/Lever de soleil par un effet de lumière à nouveau et la possibilité de piloter la dite lampe à l'aide d'un smartphone.

Ce rapport se divise en quatre parties. La première présente les principaux éléments qui composent le projet. La seconde traite de la programmation à faire pour avoir un effet orageux convenable. La troisième explique la méthode à prendre pour la réalisation de la partie électronique. La quatrième montre comment le nuage à été fabriqué.

1. Présentation

La lampe nuageuse est une lampe intelligente qui se compose principalement d'une carte ARDUINO dont les fonctionnalités vous seront expliquées par la suite et de LED RGB[1]. Nous verrons dans cette partie les principales ressources de la carte ARDUINO qui seront utilisées ainsi que le fonctionnement des LED RGB.

1.1. Carte Arduino

La carte ARDUINO UNO REV 3[2] utilise comme composant principal un microcontrôleur[3] ATMEGA. Celui-ci se compose de différentes pattes que l'on peut associer en entrées ou en sorties du composant. De plus le micro-contrôleur peut employer une fonction MLI[4] qui sera très largement utilisé dans ce projet.

1.1.1. Entrées/Sorties

La carte ARDUINO citée précédemment utilise 14 entrées/sorties et 6 entrées analogiques. Ces entrées/sorties peuvent être programmées de façon à « réagir » ensembles. Elle se compose d'un port de communication USB ainsi que d'une partie alimentation.

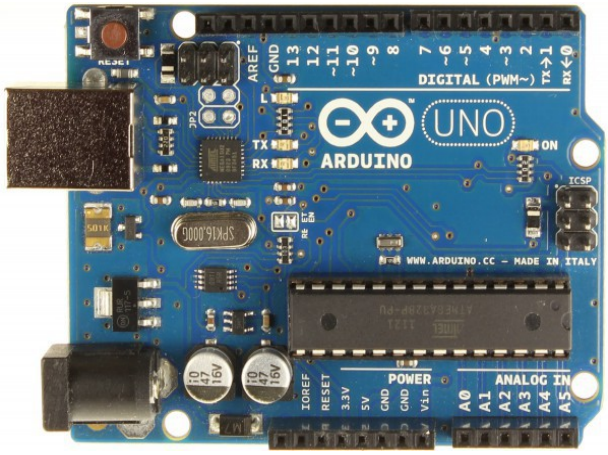


Illustration 2 : Carte ARDUINO UNO REV 3 (Image constucteur)[1]

Le microcontrôleur implanté dans cette carte est un Atmega 328. Sa mémoire flash est de 32 000 Octets, il peut opérer à une fréquence de 20MHz. Pour l'étude du prototype de la lampe nuageuse ce nombre d'entrées/sorties ainsi que sa fréquence d'opération sont amplement suffisant.

Par la suite, la réalisation finale utilisera deux Atmega 8535 afin d'amplifier le nombre d'entrées/sorties et diviser le travail de chaque microcontrôleurs.

---

1 **LED RGB** : Light-Emitting Diode (ou Diode Électroluminescente en français) est un composant électronique capable d'émettre de la lumière rouge, verte et bleue.

2 **ARDUINO UNO REV 3** : Référence constructeur.

3 **Micro-contrôleur** : Selon un arrêté français du 14 septembre 1990 relatif à la terminologie des composants électroniques : " Circuit intégré comprenant essentiellement un microprocesseur, ses mémoires, et des éléments personnalisés selon l'application " .

4 **MLI** : Modulation par Largeur d'Impulsion.

1.1.2. Sortie « analogiques » : MLI

Il faut dans cette partie bien comprendre que la carte ARDUINO, ou plus précisément l'ATmega ne peut fournir une tension de sortie analogique variable. En effet, les pattes du micro-contrôleur ne pouvant fournir que des tensions équivalantes à un 1 logique, soit 5 volt ou un 0 logique, soit 0 volt nous montre qu'il est impossible de fournir une tension intermédiaire. Or pour la réalisation d'un effet orageux ressemblant à un orage naturel il est nécessaire de pouvoir moduler cette tension de sortie.

La carte ARDUINO nous offre donc la possibilité d'utiliser une fonction MLI qui peut remplacer une sortie analogique.

La modulation par largeur d'impulsion consiste à modifier la valeur moyenne d'un signal carré suivant la formule suivante :

$$Vmoy = aVmax$$

Ici **Vmoy** représente la valeur moyenne obtenue ;

a est un coefficient situé entre 0 et 1 , il est aussi appelé le « rapport cyclique »

Vmax est la valeur maximale que peut prendre Vmoy. Généralement elle est égale au 1 logique.

Pour une meilleure visualisation de la valeur moyenne voici une représentation temporelle d'un signal carré avec un rapport cyclique différent.

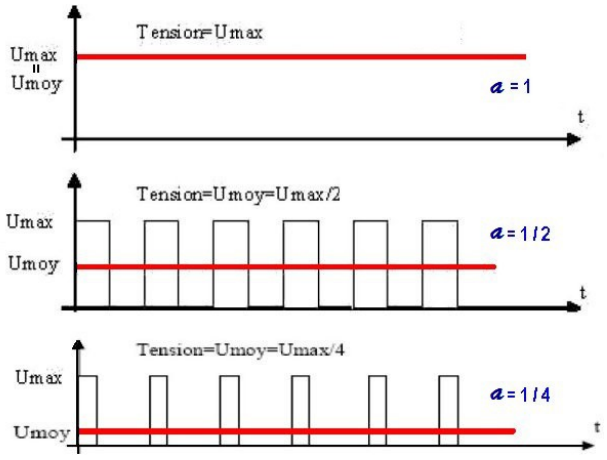


Illustration 3 : Trois signaux MLI pour un rapport cyclique différent

On remarque sur l'image ci-dessus que le signal obtenu est différent suivant le rapport cyclique. De plus, nous pouvons voir que la valeur moyenne change en fonction du rapport cyclique. Ce qui confirme la formule donnée précédemment.

1.2. LED

Afin de réaliser un jeu de lumière convenable nous utiliserons des LED RGB plutôt que d'utiliser des LED blanches. En effet, grâce à l'utilisation de ces LED l'utilisateur pourra choisir lui même la couleur de la lampe ou utiliser des fonctions comme « lever/coucher de soleil » qui emploient des coloris différents.

1.2.1. Code couleur

Chaque ordinateurs, télévisions, ou autres appareils technologiques qui utilisent des écrans couleurs emploie ce que l'on appelle la synthèse additive.

La synthèse additive est un procédé physique qui permet de comprendre la lumiére blanche. En effet, ce fait nous montre que la lumiére blanche consiste l'addition de trois couleurs de lumières de base : le rouge, le vert et le bleu. Ces trois couleurs associées ensembles avec des proportions identiques affichent une lumière blanche. Cependant si ces proportions changent, la couleur vu par nos yeux changent. L'image qui suit nous montre un exemple de ce phénomène.

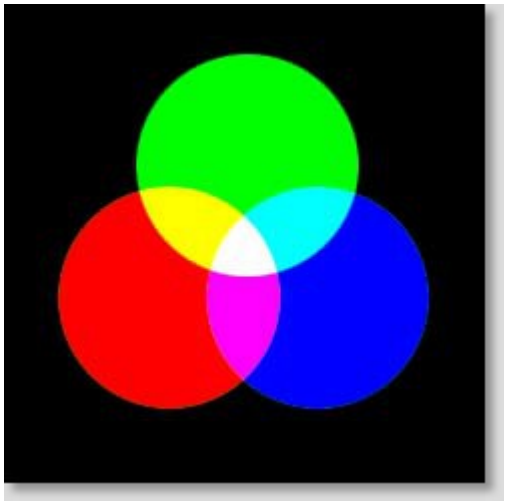


Illustration 4 : Synthèse additive[2]

En revenant sur l'exemple des écrans couleurs, ceux-ci sont composés de pixels[1]. Suivant la technique d'écran, les pixels peuvent êtres fait différemment (Voir D2R2 : *Les écrans plats : Que faut-il pour qu'un écran plat soit tactile?[2]*). Mais ils présentent tous les même caractéristiques, soit trois partie émettrices de lumière : une rouge, une verte et une bleue.

Dans notre projet nous utilisons une fonction MLI pour ajuster la couleur de chaque LED. Chaque couleur sera codée sur un maximum de 255.

1.2.2. Cathodes communes

Il est très difficile d'employer trois LED dont une rouge, une verte et une bleue pour en associer les couleurs comme expliqué en 1.2.1. C'est pourquoi les LED utilisées se composent de trois LED rouge, verte et bleu.On les appelle LED RVB.

1 **Pixel** : *Picture element*, élement qui compose une image ou un écran

Une LED est un composant électrique bipolaire. C'est à dire que c'est un semi-conducteur qui comporte une partie positive et une autre négative. Ce qui joue le rôle de la partie positive d'une LED est l'anode, l'autre partie négative est la cathode.

Technologiquement parlant, il est plus facile de mettre en place un groupement de LED si toutes leur anode ou leur cathode sont reliées ensemble. Suite à cette réflexion, j'ai choisi d'utiliser des LED RVB qui remplissent cette condition. Les LED choisies sont de la marque Kingbright dont le schéma interne est le suivant.

Nous utiliserons principalement ces deux documents dans la suite de ce rapport.

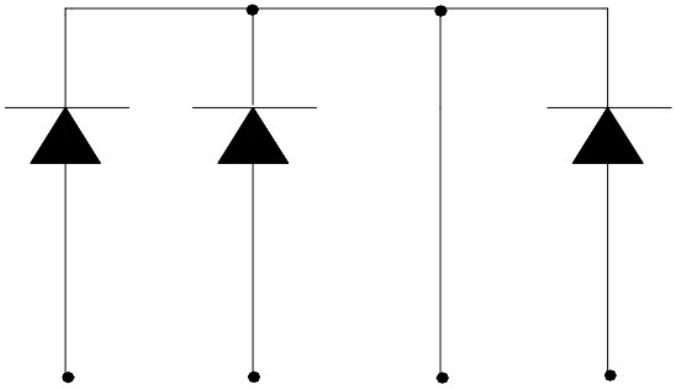


Illustration 5 : Schéma électronique des LED[3]

■Electrical -Optical Characteristics **(Ta=25℃)**

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
DC Forward Voltage	V_F (R)	I_F=20mA	1.8	2.0	2.4	V
	V_F (B/G)	I_F=20mA	2.8	3.4	4.0	V
DC Reverse Current	I_R	V_R=5V	-	-	10	μA
Domi. Wavelength	**λ_D(Red)**	I_F=20mA	620	625	630	nm
	λ_D(Green)	I_F=20mA	520	525	530	nm
	λ_D(Blue)	I_F=20mA	465	470	475	nm
Luminous Intensity	**Iv(Red)**	I_F=20mA	4200	5800	7000	mcd
	Iv(Green)	I_F=20mA	7000	8400	10000	mcd
	Iv(Blue)	I_F=20mA	3000	4200	5800	mcd
50% Power Angle	$2\theta_{1/2}$	I_F=20mA	-	30	-	deg

Illustration 6 : Caractéristiques des LED extraites de la DATAsheet[3]

2. Programmation

Le site ARDUINO nous fourni un IDE[1] gratuitement. Il est impossible de modifier la structure d'un programme implanté dans une carte ARDUINO sans cet IDE car ces cartes ont un BOOTLOADER prédéfini, c'est à dire que sans changer ou enlever cette « porte » l'envoi du programme est impossible.

1 **IDE** : *Integrated Development Environment* ou Environnement de Développement en français.

Dans cette partie nous expliquons en détail chaque fonctions utilisées dans le mode orageux de la lampe.

2.1. Fonction PWM (MLI)

Un des problèmes de la carte ARDUINO et que sur les 14 sorties numériques seulement 6 permettent de réaliser matériellement une MLI. Nous devons alors créer une fonction pour contourner ce problème et ainsi permettre à toutes les sorties disponibles d'avoir une MLI convenable.

Pour réaliser ceci il suffit de réaliser une comparaison entre un nombre qui croît est une valeur quelconque. Pour mieux visualiser voici une représentation d'une MLI numérique.

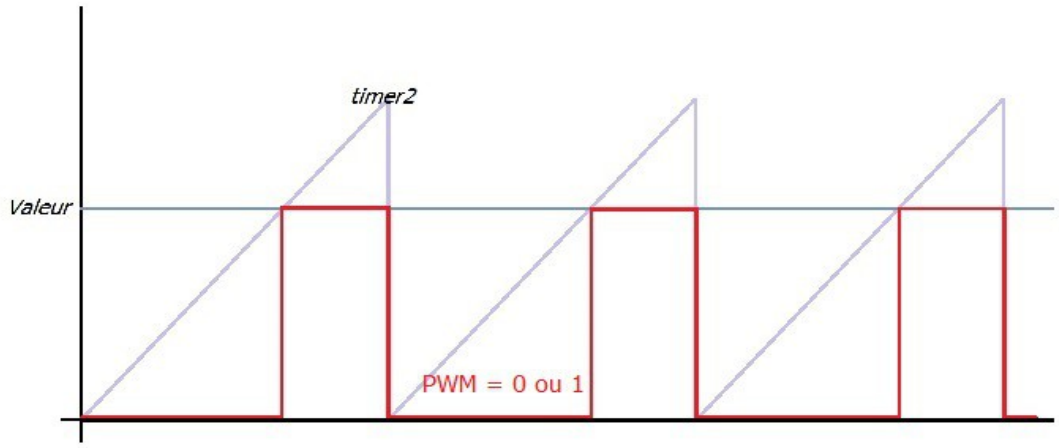


Illustration 7 : Représentation d'une Modulation à Largeur d'Impulsion numérique

On remarque facilement ce qu'il se passe. Une variable entière, *timer2,* croît jusqu'à un maximum puis retombe à zéro et croît à nouveau. Et ainsi de suite. La valeur de *timer2* est comparée à une autre valeur quelconque, c'est cette nouvelle valeur qui va définir le rapport cyclique. En effet, quand celle-ci vaut 0, le rapport cyclique vaut 1. Quand elle prend la valeur max de *timer2* le rapport cyclique vaut 0.

On obtient donc la formule suivante : $a = \frac{timer2}{Valeur}$

Avec : **a** le rapport cyclique

Cette fonction retourne un booléen. Elle s'écrit tel que suit :

```
int FctPWM(int Valeur){
 if(timer2>=Valeur)
 valPWM=0;
 else
 valPWM=1;
  return valPWM;}
```

2.2. Fonction MLI dynamique

Pour réaliser un éclair le plus représentatif qui soit de ce que nous pouvons avoir lors d'un orage en pleine nature nécessite un effet lumineux dynamique. C'est à dire avoir une propagation de la lumière progressive. Pour se faire la fonction MLI doit pouvoir modifier la valeur moyenne du signal carré au cour du temps.

Dans cette partie, nous n'expliquons seulement l'une des trois fonctions MLI dynamique du programme, à savoir : *void* Decr*PWM (int tps)*. En effet, les trois s'écrivent sur le même principe à exception près d'une différence de signe afin d'avoir une qui croît, une autre qui décroît et enfin une dernière qui croit et décroît.

La plupart du temps ARDUINO propose d'utiliser la fonction « **delay**(int **tps**) » où tps est le temps en millisecondes. Le problème de cette fonction est que le microcontrôleur se met en pause pendant une durée définie dans la fonction, ce qui implique qu'il est impossible de communiquer avec lui mis à part avec des interruptions. Or les interruptions sont généralement utilisées en cas de problèmes.

Il est nécessaire de cadencer cette fonction par une variable qui s'incrémente ou se décrémente. *Timer2* joue ce rôle.

Comment rendre cette fonction MLI créée en 2.1.2 dynamique ?

Le prototype de la fonction s'écrit : *int FctPWM(int Valeur)*. On comprend tout de suite que c'est *Valeur* qui doit changer au cour du temps.

Pour changer la variable *valeur* à notre guise deux variables semblables à *timer2* sont nécessaires. L'image qui suit explique le fonctionnement de la fonction.

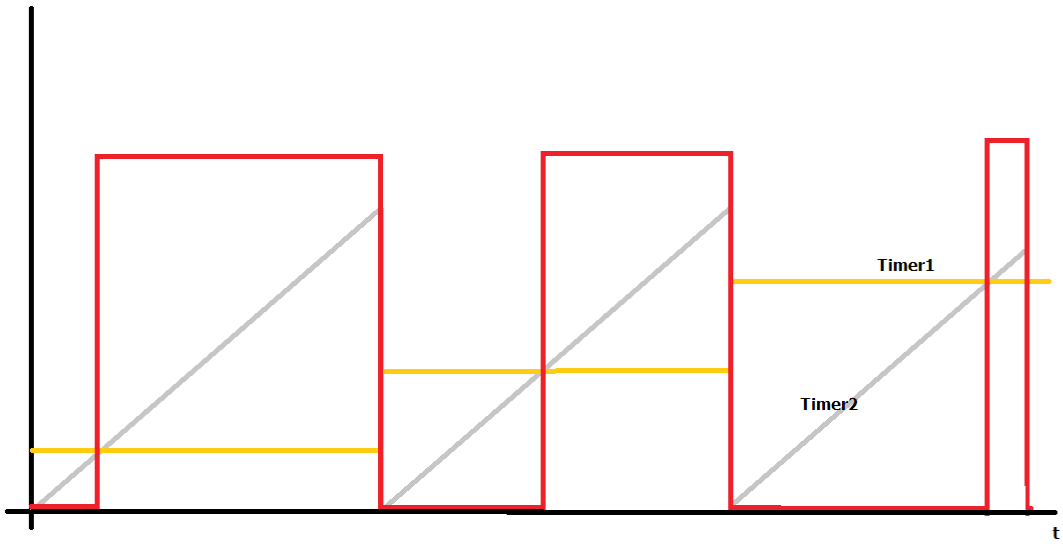


Illustration 8 : Chronogramme des variables timer1 (gris) ; timer2(jaune) et le résultat du signal carré MLI (rouge)

On remarque que la fonction utilise deux variables timer pour fonctionner. De plus *timer1* est utilisé comme référence pour la fonction FctPWM(int Valeur). La variable *timer1* croît jusqu'à un maximum qui, ici est *tps,* implique que la valeur moyenne du signal résultant de cette fonction diminue.

Plus la valeur de *tps* est grande, plus le temps d'exécution total de la fonction est grand.

Voici, ci-dessous la fonction *void* Decr*PWM (int pinRouge, intt pinVert, int pinBleue, int tps).*

```
void DecrPWM (int pinRouge, int pinVert, int pinBleu, int TPS){
 do{
 timer3++;
 timer2=0;
 do {
  timer2++;
  digitalWrite(pinRouge,FctPWM(255-timer1));
  digitalWrite(pinVert,FctPWM(255-timer1));
  digitalWrite(pinBleu,FctPWM(255-timer1));
}while(timer2<=TPS);
if(timer1>=TPS)
   {timer1=0;}
   else
   {timer1++;}}while(timer3<=TPS);
timer3=0;}
```

2.3. Fonction Orage

Les différentes fonctions orage (au nombre de cinq) fonctionnent toutes de la même façon. Elles consistent à réaliser une suite chronologique de plusieurs fonctions MLI dynamique qui sont croissantes, décroissantes et l'association des deux.

La fonction Orage1 consiste à n'allumer qu'une partie du nuage afin de simuler un éclair isolé. Cet éclair s'allume en « clignotant » puis s'éteint progressivement.

Le nuage est divisé en quatre zones. La zone A, B C et D. Voir image ci-dessous.

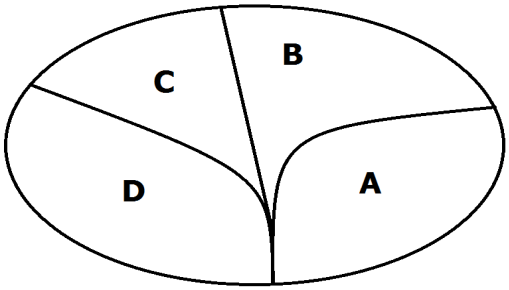


Illustration 9 : Nuage divisé en quatre zones

Voici la fonction Orage1 écrite :

```
void Orage1(void){
  Double(0,1,2,50);
  Double(0,1,2,75);
  Double(0,1,2,100);
  CroiPWM(0,1,2,100);
  DecrPWM(0,1,2,500);
}
```

3. Électronique

Cette troisième partie traite essentiellement de l'étude électronique du projet. L'apparente simplicité de la carte ARDUINO peut s'avérer être un problème. Ce problème vous est expliqué par la suite, ainsi que les solutions apportées.

3.1. Faible courant de sortie Arduino

La carte ARDUINO comprend sur notre projet 14 sorties qui, individuellement, peuvent délivrer un courant maximum de 100mA sous une tension de 5V. Cependant, la carte ne peut fournir au total un courant supérieur à 400mA. Or pour alimenter la totalité des LED, soit 60 LED, il faut un courant égal à $20\,mA * 60 = 1\,200\,mA$. Ce qui pose problème.

3.2. Solution SHIELD de puissance transistor

Pour remédier au problème souligné en partie 3.1, nous réaliserons un SHIELD de puissance transistor. C'est à dire créer et dimensionner une carte électronique basée sur des transistors pour apporter le courant nécessaire au bon fonctionnement du nuage.

Le schéma de base comprend un transistor monté en commutation pour apporter cette puissance. Voici le circuit utilisé.

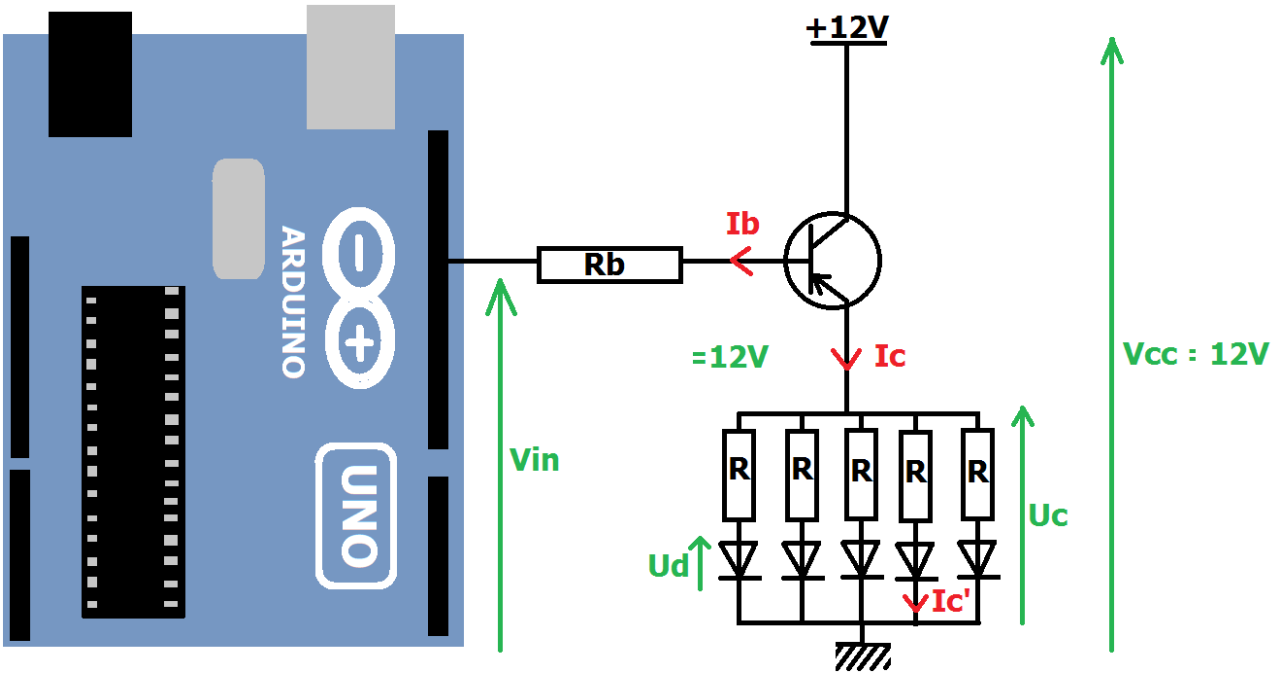


Illustration 10 : Circuit de puissance et de commande pour 5 LED de la même couleur

Pour calculer les valeurs de R et de Rb, il est nécessaire de connaître le courant Ic que va consommer la charge. La charge se compose de 5 circuits LED en série avec une résistance mise en parallèle. Nous savons, d'après la DATAsheet montrée en 1.2.2 qu'une LED, suivant la couleur, consomme un courant Ic' sous une tension Ud. Or d'après la loi des nœuds et le schéma, on sait que : $5 \times Ic' = Ic$

De plus le transistor utilisé est un PNP 2N2907 dont la DATAsheet nous apprends que le β de ce semi-conducteur vaut 100. Or avec l'effet transistor on sait que $Ic = \beta . Ib$

Donc $Ib = \frac{Ic}{100} = 1\,mA$

Avant de commencer les calculs de **Rb** et de **R** on remarque un problème qui vient de la carte ARDUINO. En effet, le 0 logique émis par la carte relie la base du transistor à la masse. La tension de base du transistor est donc toujours positive dû à l'absence d'une résistance de PULL-DOWN de l'Atméga. Après plusieurs tests on constatait que les diodes étaient toujours allumées.

3.3. deuxième transistor de commande

Pour remédier au problème souligné dans la partie précédente nous avons rajouté un deuxième transistor NPN de commande. Son rôle consiste à mettre la base du transistor PNP « volante » afin qu'il joue bien son rôle de commutation.

Voici le nouveau schéma électrique avec un transistor de commande supplémentaire.

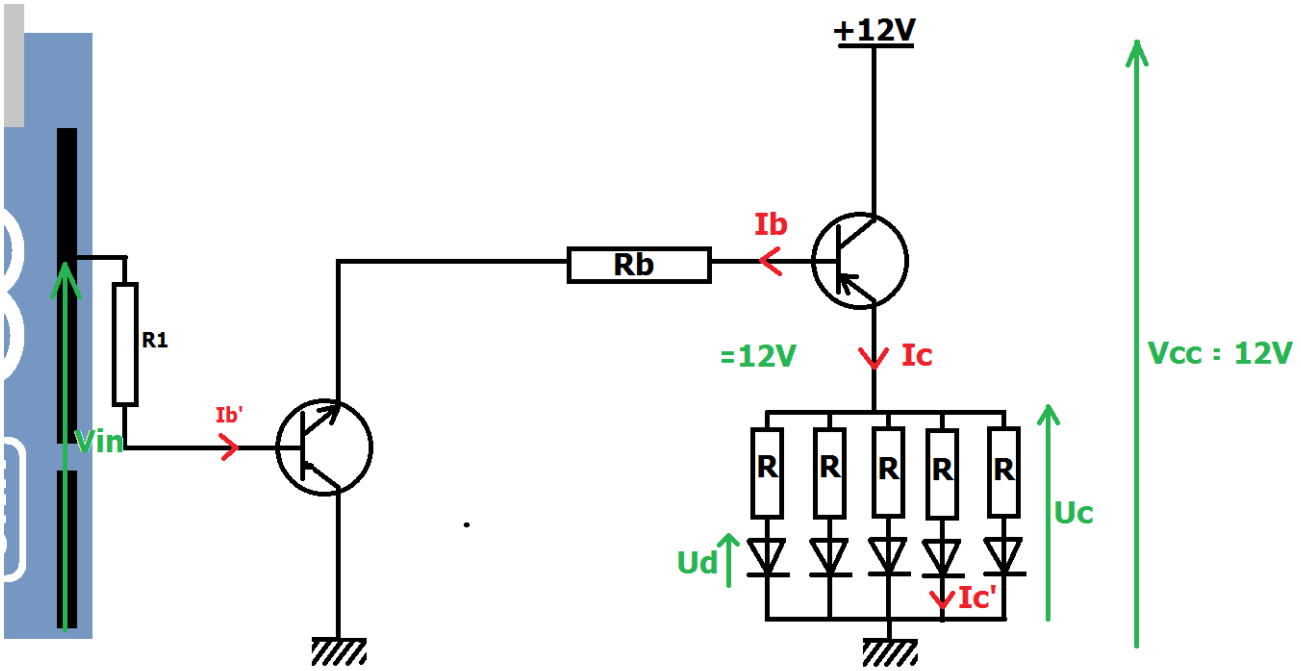


Illustration 11 : Schéma de puissance final

D'après la partie 3.2, on sait que Ib vaut 1mA. Or on utilise comme deuxième transistor un 2N2222 qui a un β égal à 100. Donc $Ib' = \frac{Ib}{\beta} = \frac{1 \cdot 10^{-3}}{100} = 10\,\mu A$

Ensuite, **R1**. $\frac{Vin}{Ib'} = \frac{5}{10 \cdot 10^{-6}} = 500\,k\,Ohms$

Pour garder le transistor en régime bloqué/saturé. Il faut avoir $R1 \leqslant 500\,kOhms$

Nous allons prendre R1 = 430k.

Pour calculer **Rb**. Nous savons que Ib = 1mA. Et la tension aux bornes de Rb vaut 12-0,7=11,3 V. En arrondissant à 11V on obtient : $Rb = \frac{U_{Rb}}{Ib} = \frac{11}{1 \cdot 10^{-3}} = 11\,k\,Ohms$

Pour les même raisons que R1, nous allons prendre Rb = 10k.

Pour les calculs de **R**, nous procéderons par couleurs. En effet, Les LED bleues, rouges et vertes ne nécessitent pas les même tensions de fonctionnement.

$$R_{rouge} = \frac{Vcc - U_t - Vf_{Rouge}}{Ic'} = \frac{11,3 - 1,9}{20 \cdot 10^{-3}} = 470\,Ohms$$

$$R_{Bleue} = \frac{Vcc - U_t - Vf_{Bleue}}{Ic'} = \frac{11,3 - 3,5}{20 \cdot 10^{-3}} = 390\,Ohms$$

Et enfin : $R_{Vert} = \frac{Vcc - U_t - Vf_{Vert}}{Ic'} = \frac{11,3 - 3,6}{20 \cdot 10^{-3}} = 385\,Ohms$

4. Fabrication

Cette partie vous montre la fabrication du nuage. Nous vous montrons comment le prototype de la lampe a été réalisée en suivant les différentes étapes de fabrication ainsi que le coût final de la réalisation.

4.1. Réalisation de la carcasse

Pour réaliser la carcasse du nuage nous utilisons des ballons de baudruche qui serviront de base pour la forme du nuage. Aux nombre de trois, il fournissent un volume suffisant pour la lampe.

Après les avoir attachés ensembles nous les recouvrons de gaze pharmaceutique et de colle afin de consolider cette base pour pouvoir appliquer dessus une couche de résine époxy transparente.

Une fois la résine séchée, on applique sur la totalité de la surface du coton hypoallergénique pour donner un « effet nuage » et éviter que la poussière s'accumule dessus.

Matériel	Prix à l'unité	Nombre de pièces	Total
SHIELD DE PUISSANCE TRANSISTOR			
Transistor 2N2907	1,20 €	12	14,40 €
Transistor 2N2222	1,50 €	12	18,00 €
Résistance 430k	0,05 €	12	0,60 €
Résistance 10k	0,05 €	12	0,60 €
Résistance 100k	0,05 €	12	0,60 €
Connectique femelle x10	0,50 €	4	2,00 €
Connectique Mâle x10	0,50 €	4	2,00 €
Embase Jack alimentation	1,00 €	1	1,00 €
Connectique Jack alimentation	1,00 €	1	1,00 €
Câble connectique Bleu 1m/0,4mm	1,70 €	5	8,50 €
Câble connectique Vert 1m/0,4mm	1,70 €	5	8,50 €
Câble connectique Rouge 1m/0,4mm	1,70 €	5	8,50 €
Gaine thermique diam. 1,8mm/1m	1,00 €	5	5,00 €
LED KINGBRIGHT RGB	1,00 €	20	20,00 €
		TOTAL	**90,70 €**
ARDUINO			
Carte ARDUINO UNO Rev 3	1	24,95	24,95
		TOTAL	24,95 €
NUAGE			
Ballon de baudruche	0,60 €	3	1,80 €
Résine époxy + durcissant 400ml	14,95 €	1	14,95 €
Gaze pharmaceutique	1,37 €	2	2,74 €
Coton hypoallergénique	9,36 €	1	9,36 €
		TOTAL	28,85 €
TOTAL			**144,50 €**

4.2. Dépenses

L'objectif était de fabriquer un objet pas trop cher, nous pouvons constater que l'objectif est rempli.

Conclusion

En conclusion de ce rapport, nous avons pu voir que la fabrication de ce nuage est simple et pas excessivement cher. En effet, la partie électronique n'utilise que des transistors en commutation qui sont simple d'utilisation et la carte ARDUINO offre la possibilité d'avoir une base déjà construite.

La réelle difficulté était de fabriquer la plastique du nuage. En effet, n'ayant pas eu de formation sur la matière il était compliqué de réaliser un ensemble solide sans pour autant supprimer l'esthétique lumineuse .

Le projet n'est pas terminé. Il recevra plus tard des options supplémentaires comme la possibilité d'être piloté via une application smartphone, d'avoir une option « réveil » avec un jeu de lumières, etc.

Index des illustrations

Bibliographie

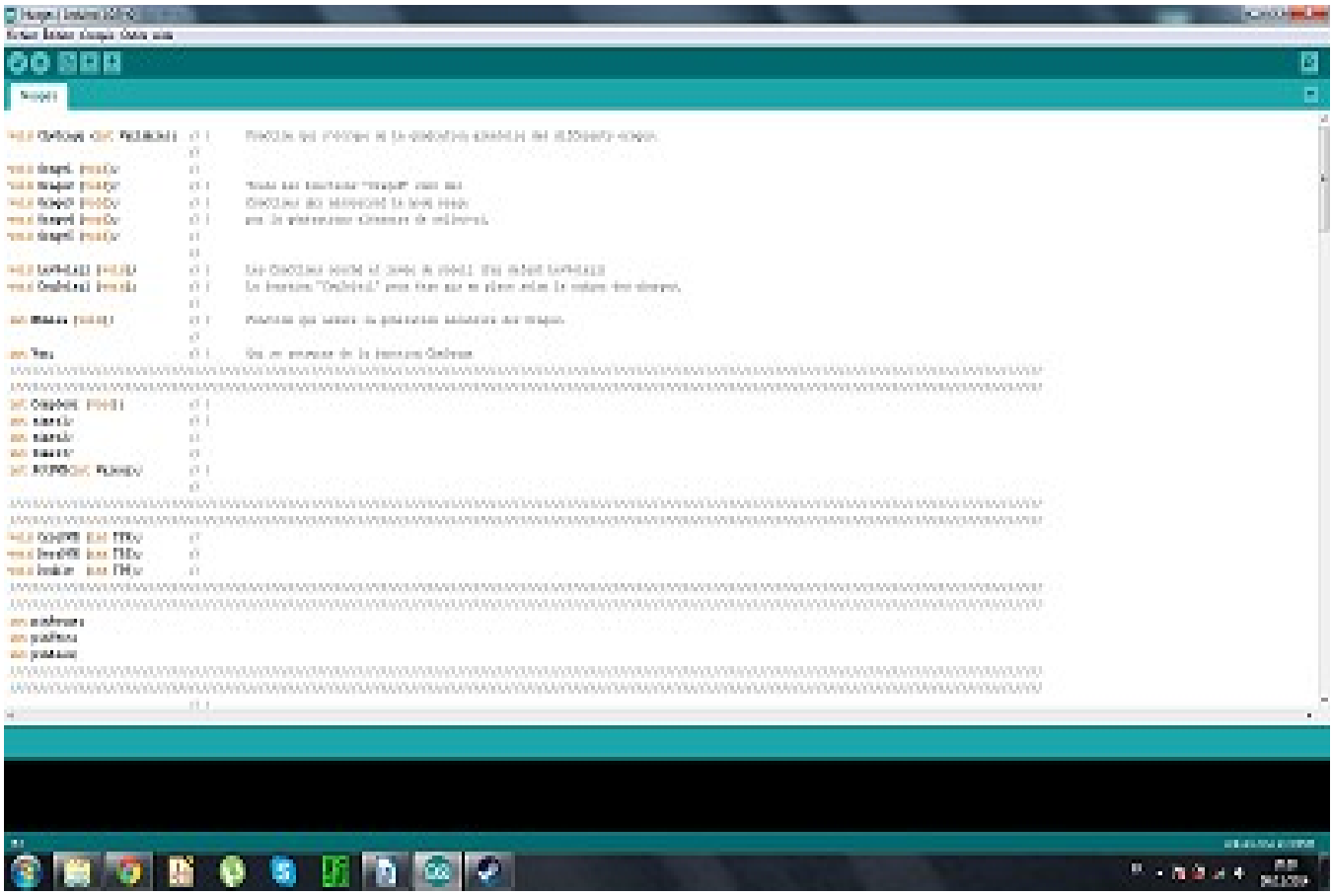
[1] . ARDUINO constructeur, 2014, [En ligne]. (Page consultée le) <www.arduino.cc>

[2] Mathieu PUILLANDRE, "Les écrans plats", , 2013.

[3] Kingbright, "", , .

Annexe 1 :

IDE ARDUINO



Annexe 2

DATASheet LED RGB

Kingbright

T-1 3/4 (5mm) FULL COLOR RGB LAMPS

LF5WAEMBGMBC HIGH EFFICIENCY RED / BLUE / GREEN
LF5WAEMBGMBW HIGH EFFICIENCY RED / BLUE / GREEN

Features

- TWO BLUE, ONE GREEN AND ONE RED CHIPS IN ONE PACKAGE.
- CAN PRODUCE ANY COLOR IN VISIBLE SPECTRUM, INCLUDING WHITE LIGHT.
- WIDE VIEWING ANGLE FOR DIFFSUED LENS AND HIGH INTENSITY FOR WATER CLEAR LENS.

Description

The High Efficiency Red source color devices are made with Gallium Arsenide Phosphide on Gallium Phosphide Orange Light Emitting Diode.
The Blue source color devices are made with GaN on SiC Light Emitting Diode.
Static electricity and surge damage the LEDS. It is recommended to use a wrist band or anti-electrostatic glove when handling the LEDs.
All devices, equipment and machinery must be electrically grounded.
The Green source color devices are made with Gallium Phosphide Green Light Emitting Diode.

Package Dimensions