|  |
| --- |
|  |
| Jeux de lumières |
| *Découverte de la programmation sur microcontrôleur* |
|  |
| **BARBESANGE Benjamin, GARCON Benoît** |
| **avril 2014** |

|  |
| --- |
|  |

Table des matières

[Introduction 3](#_Toc384826419)

[Présentation du projet 3](#_Toc384826420)

[Carte EasyPIC 3](#_Toc384826421)

[LED RGB 4](#_Toc384826422)

[Présentation du programme 4](#_Toc384826423)

[Fichier main.c 4](#_Toc384826424)

[Fichier chiffre.c 4](#_Toc384826425)

[Fichier glcd.c 4](#_Toc384826426)

[Fichier ledRGB.c 5](#_Toc384826427)

[Partie 1 - La matrice de LED 6](#_Toc384826428)

[Partie 2 - L'écran GLCD 7](#_Toc384826429)

[Partie 3 - La LED RGB 8](#_Toc384826430)

[Port PWM "classique" 8](#_Toc384826431)

[Création d'un port PWM 8](#_Toc384826432)

[Partie 4 - Essais divers 9](#_Toc384826433)

Remerciements

Introduction

# Présentation du projet

Dans le cadre de la deuxième année de Prep' ISIMA, il est demandé d'effectuer un projet de groupe occupant 24h de travail. Le sujet que nous avons ainsi choisi est le sujet "Jeux de lumière".

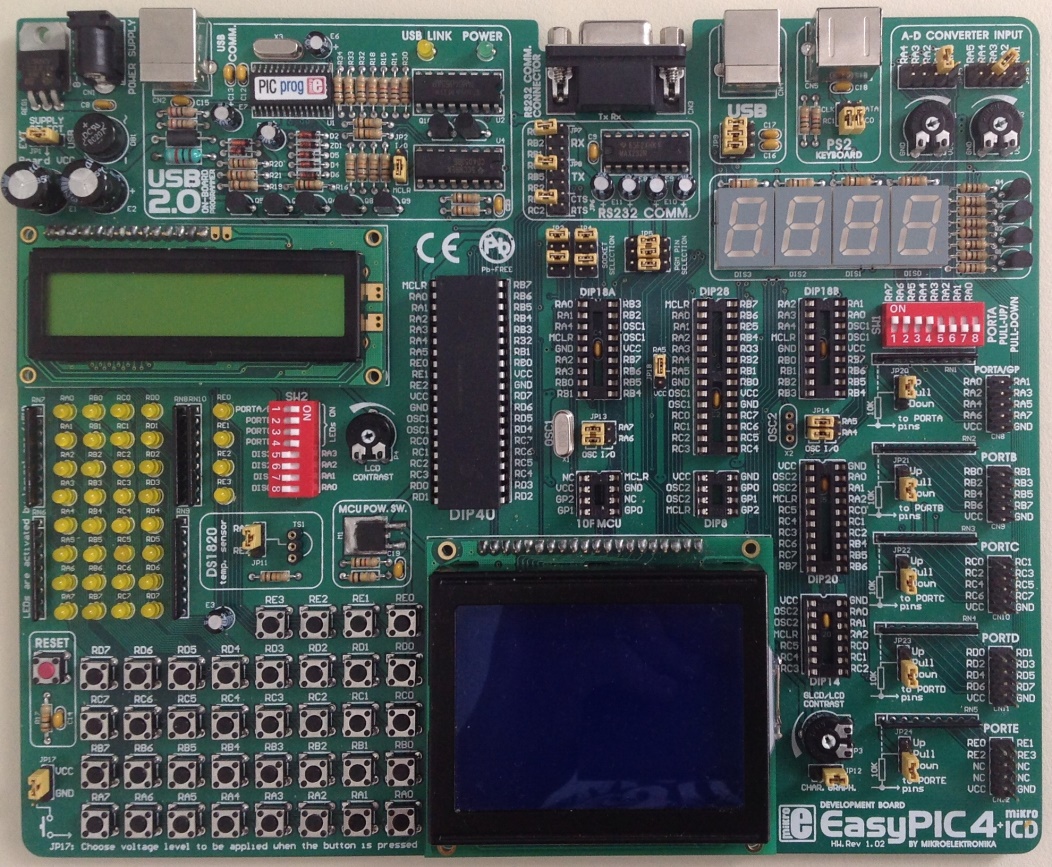
Afin de pouvoir réaliser ce projet, le matériel suivant a été mis a disposition :

* Une carte EasyPIC
* Un microcontrôleur (réf. P18F452)
* 6 résistances de 1 kΩ
* Une LED RGB
* Une plaquette de branchement ainsi que des fils de connexions

## Carte EasyPIC

La carte EasyPIC fournie est le modèle 4. C'est la base du projet, qui fait le lien entre les différents composants. La carte est composée d'une matrice de LED dite de "contrôle", ainsi que d'un écran GLCD sur lequel il est possible d'afficher des images. Ce sont les deux principaux éléments sur lesquels nous intervenons.

La partie la plus importante de la carte est le microcontrôleur, qui va ainsi permettre d'interagir avec les différents composants grâce à un programme qu'il faut préalablement charger.



Matrice de LED

Ecran GLCD

Figure 1 - Carte EasyPIC 4

## LED RGB

En plus de la carte PIC, une LED RGB est mise à disposition. Celle-ci vient accompagnée de plusieurs résistances de 1 kΩ (afin d'éviter de griller la LED). De plus, une plaquette de branchement ainsi que des fils viennent s'ajouter afin de pouvoir relier cette LED à la carte PIC, qui va ainsi déterminer de fonctionnement de cette LED.

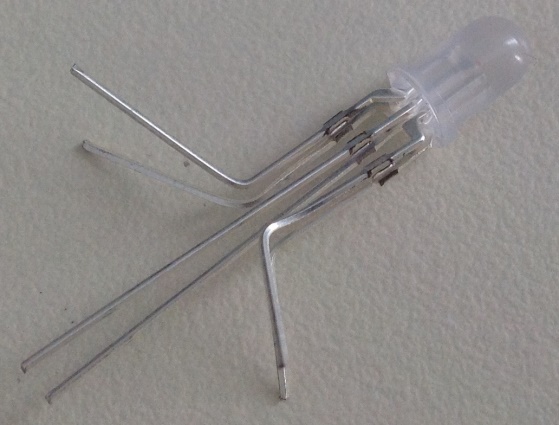


Figure 2 - LED RGB

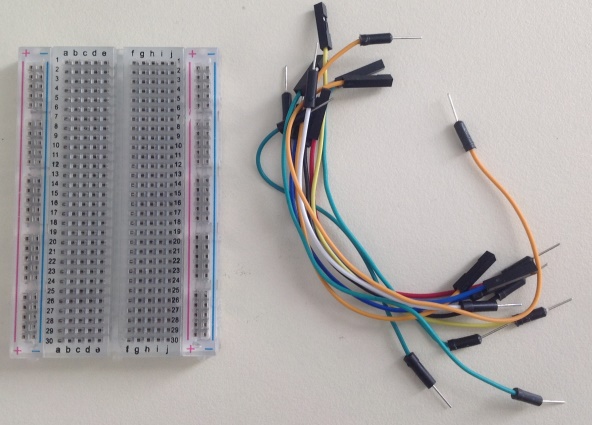


Figure 3 - Plaquette de branchement et fils

# 

# Présentation du programme

Plusieurs langages de programmation sont disponibles pour le microcontrôleur. Etant plus à l'aise avec le langage C, il est donc naturel que celui-ci ait été utilisé.

L'intégralité du code est consultable sur github à l'adresse suivante : <https://github.com/BBS007/Jeux_lumiere>.

Afin d'améliorer la lisibilité du code, celui-ci est commenté et fractionné en plusieurs fichiers, chacun regroupant les fonctions propres à la manipulation d'un composant.

## Fichier main.c

Ce fichier est le cœur du programme. Il contient la boucle principale servant à appeler les différentes fonctions de manipulation des composants. il est possible de lancer les différentes fonctions du programme en appuyant sur les boutons de la carte.

Voici la liste des fonctions associées à leurs boutons :

* RB0 : Affiche le logo de l'ISIMA sur l'écran GLCD
* RB1 : Affiche un décompte de 9 à 0 sur la matrice de LED
* RB2 : Fait varier l'intensité de la LED RGB sur le port PWM natif
* RB3 : Fait varier l'intensité de la LED RGB sur le port PWM recrée

IL est recommandé d'appuyer sur le bouton reset entre différents appels de fonction.

## Fichier chiffre.c

Dans ce fichier se trouvent les fonctions permettant les affichages de chiffre sur la matrice de LED.

## Fichier glcd.c

Ce fichier répertorie les fonctions propres à l'affichage du logo de l'ISIMA sur l'écran GLCD.

## Fichier ledRGB.c

Ce fichier permet de manipuler la LED RGB avec le port PWM natif et celui recrée.

1. La matrice de LED

Le projet a débuté le Mardi 21 Janvier. Pour assurer la compatibilité des machines utilisées avec la carte, il a fallu installer les drivers ainsi que les programmes associés : mikroProg Suite For PIC (interface avec la carte) et mikroC PRO for PIC (l'IDE).

Ensuite nous avons ouvert le projet LedBlinking, puis changé la montre (20Mhz) et changé le modèle (P18F452). Au branchement de la carte les voyants POWER et USB se sont allumés et une faible lueur a jaillit d'une des LED de la matrice. Lorsque nous avons chargé le programme sur le MCU cette LED "défectueuse" clignote une fois très faiblement sans qu'il ne se passe quoi que ce soit.

Pour essayer de trouver la source du problème nous avons essayé de changer le type de MCU mais le logiciel nous dit bien que le bon est le P18F452, nous avons changé de PC (OS plutôt) pensant que c'était dû à un problème de driver, nous avons essayé avec tout un tas d'autres programme, etc.

Lorsque l'on s’est documenté sur internet, nous avons bien vu qu'une simple "compilation écriture" ou pression sur Ctrl+F11 devait suffire à faire tout le travail mais chez nous la partie logicielle ne renvoie aucune erreur et la partie matérielle ne répond pas (hormis un sursaut faible de la LED faiblement allumée).

Nous nous sommes donc rendu dans le local d’ISIBOT et les conclusions ont été rapides : le EasyPIC 4 était défectueux, nous en avons donc reçu une nouvelle ainsi qu’un nouveau quartz de 8Mhz par la même occasion. Nous nous sommes ainsi lancés dans la programmation le jeudi 23 janvier.

Nous nous sommes ainsi lancés dans l’élaboration de notre alphabet numérique et avons ainsi pu découvrir le fonctionnement pratique de la matrice de LED. Après avoir déterminé un style adapté à la matrice 4\*8 pour chaque chiffre nous les avons codés et testés. Tout fonctionnait parfaitement excepté les LED RA4, RA6 et RA7. Effectivement RA6 et RA7 sont utilisées pour l’oscillateur et non pour les entrées/sorties. On nous a donc conseillé de considérer ces LED comme des "pixel mort" et de faire comme s’ils fonctionnaient mais nous avons continué à chercher un moyen d’utiliser ces LED sans y parvenir. Ainsi nous avons pu mettre en place un programme de compte et de décompte de 0 à 9 secondes.

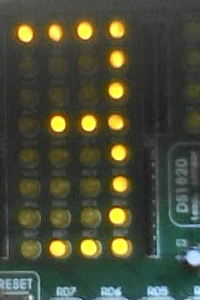


Figure 4 - Affichage du chiffre 3 sur la matrice de LED

1. L'écran GLCD

Une fois le décompte crée sur la matrice de LED, nous nous sommes ensuite intéressés au reste du matériel et avons aussi réussi à faire afficher le logo de l’ISIMA sur l’écran GLCD.



Figure 5 - Affichage du logo ISIMA sur l'écran GLCD

1. La LED RGB

Nous disposons à présent d’une LED RGB à quatre pattes, de six résistances identiques et de deux LED dites classiques. Après avoir reçu la LED RGB nous l'avons branchée en série avec les résistances associées nous avons réussi à faire clignoter chaque couleur allant de l'intensité la plus faible à l'intensité la plus forte.

# Port PWM "classique"

Les deux ports PWM (Pulse With Modulation) correspondent aux ports RC1 et RC2. Ces ports permettent de faire varier l'intensité du courant transmis, notamment en modifiant le rapport cyclique, ce qui a pour effet de créer des intensités de lumières plus ou moins importantes sur une LED.

N'ayant que deux port PWM nous ne pouvons donc utiliser que deux couleur à la fois nous avons choisi dans un premier temps d'utiliser la LED rouge et la LED bleue afin de mieux les distinguer.

Afin d'utiliser le maximum de possibilités de la LED RGB nous avons décidé de faire une fondue de la couleur verte à la couleur rouge puis inversement en passant par toutes les gammes d'intensité disponibles. La prise en main de la librairie dite PWM a été assez rapide. En effet une fonction très simple permet de régler le rapport cyclique : paramétrable de 0 à 255 cela règle la largeur d’impulsion de RC1 et RC2 respectivement de 0 à 100%. Il a alors été assez aisé d’établir des mélanges de couleur.

# Création d'un port PWM

Une fois notre travail terminé avec le port PWM intégré sur la carte, il nous a été demandé de recréer un tel port sur une patte n'ayant pas la possibilité de créer de base un signal de type "carré". Pour cela, nous avons effectuées des mesures avec l'oscilloscope afin de déterminer la demi-période à recréer.

La période de base du PWM étant de 5 kHz

* delay µs
* temps de calcul
* improvements avec registres

1. Essais divers

# Jeu de la vie sur GLCD

# Texte sur écran LCD