

CFPT Ecole d'informatique - Technicien ES en informatique

Travail de semestre inter-degré 2016-2017

Cube à Led

Documentation technique

Elèves :

M. Kevin AMADO

M. Alan DEVAUD

M. Gregory MENDEZ

Enseignants :

M. Denis CARBONE

M. Nicolas WANNER

Version 1.0 du
13 mars 2017

1 Résumé

Table des matières

1	Résumé	1
2	Introduction	3
3	Cahier des charges	4
3.1	Sujet	4
3.2	But	4
3.3	Spécification	4
3.4	Restriction	4
3.5	Environnement	4
3.6	Livrables	4
3.7	Réddition	5
4	Analyse de l'existant	6
4.1	Représentation du Cube dans l'application	6
4.2	Format ".cube"	8
5	Analyse fonctionnelle	9
6	Analyse organique	10
6.1	Bibliothèque <i>CubeLedCommunicationLibrary</i>	10
7	Conclusion	12

2 Introduction

Notre projet consiste à améliorer le projet Cube de M. AUBERT Jonathan dans le cadre de l'atelier Technicien, qui regroupe les deuxièmes année avec les premières. Nous sommes trois à travailler ensemble. L'objectif est de reprendre le travail réaliser en ajoutant une vue 3D. Notre application doit pouvoir gérer la couleur des Leds. La gestion d'animation est aussi demandée.

3 Cahier des charges

3.1 Sujet

Création d'une interface pour la gestion d'un cube led.

3.2 But

Créer une interface graphique permettant à l'utilisateur de gérer le cube à led. Cette interface est un logiciel C# munie d'un cube 3D. L'utilisateur peut alors sélectionner les leds du cube et les modifier (allumer, éteindre, intensité). Le cube à led - physique - se modifier en fonction des actions de l'utilisateur sur l'application.

3.3 Spécification

Le logiciel sera capable de :

- * afficher un cube virtuelle en 3D
- * communiquer avec le cube
- * sélectionner une face du cube virtuelle
- * modifier une led du cube modifier

3.4 Restriction

Le logiciel sera incapable de :

- générer un fichier .cube
- modifier la couleur des led du cube (virtuelle et physique)
- gerer plusieurs cubes led en même temps
- modifier le programme du micro-controlleur du cube led

3.5 Environnement

- * Système d'exploitation : *Windows 7*
- * Outil de développement : *Visual Studio*
- * Langage de programmation : *C#*

3.6 Livrables

- * Code source
- * Documentation technique
- * Journal de bord

3.7 Réddition

- * 20 mars 2017 : code source
- * 20 mars 2017 : documentation technique
- * 20 mars 2017 : journal de bord
- * 20 mars 2017 : présentation final

4 Analyse de l'existant

4.1 Représentation du Cube dans l'application

Le cube possède huit étages, chaque étage contient huit rangées qui, elles-mêmes contiennent huit Leds. Le Cube à Leds est représenté sous la forme d'un tableau à trois dimensions :

$$\text{CUBELED}[x][y][\text{nbImage}] = \text{value}$$

CubeLED : Le nom du tableau
x : Position d'une Led sur l'axe "x"
y : Position d'une Led sur l'axe "y"
nbImage : Numéro de l'image (animation)
value : Valeur stockée

Explication de la valeur stockée dans "*value*" :

Ce sont les deux premiers paramètres qui sont utiles pour représenter la position d'une Led allumée, c'est donc grâce à la valeur stockée que nous indiquerons au microcontrôleur quelle Led il va devoir allumer. On envoie donc une valeur entre 0 et 255, sous forme binaire. Dans cette chaîne binaire, chaque "1" représente une LED allumée et chaque "0" une LED éteinte.

EXEMPLE 1 : $\text{CUBELED}[0][0][0] = 128_{10} \rightarrow 10000000_2$

-> La dernière Led de la ligne $x = 0$ / $y = 0$ est allumée.

EXEMPLE 2 : $\text{CUBELED}[6][4][0] = 170_{10} \rightarrow 10101010_2$

-> Une Led sur deux est allumée sur la ligne $x = 6$ / $y = 4$.

EXEMPLE 3 : $\text{CUBELED}[2][7][0] = 0_{10} \rightarrow 00000000_2$

-> Aucunes Leds sont allumées sur la ligne $x = 2$ / $y = 7$.

EXEMPLE 4 : $\text{CUBELED}[8][1][0] = 255_{10} \rightarrow 11111111_2$

-> Toutes les Leds sont allumées sur la ligne $x = 8$ / $y = 1$.

Ce format est utilisé pour simplifier la communication avec le microcontrôleur et la création de fichiers ".cube". Il est aussi important de prendre conscience de l'orientation du Cube en utilisant ce format.

Voici donc ci-dessous, un schéma représentant le tableau CubeLED. La face droite est l'équivalent des deux premiers index de notre tableau ($CubeLED[x][y]$) et les valeurs affichées sur cette dernière sont les éléments stockés à ces positions.

Le microcontrôleur analyse la valeur binaire convertie (voir axe Z) et allume une LED à chaque fois qu'elle rencontre un "1".

Note : Le tableau complet contient des animations, ce schéma représente une seule image qui pourrait être présente dans le tableau.

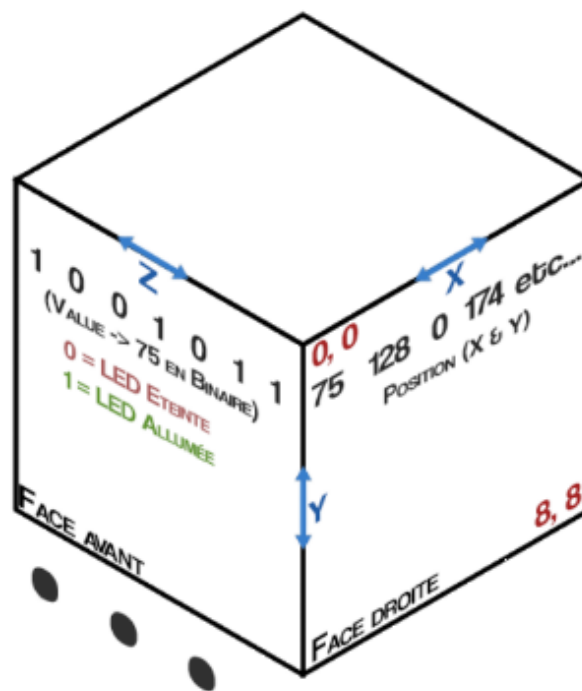


FIGURE 1 – Représentation 3D du tableau *CubeLED*

4.2 Format ".cube"

Le fichier cube est construit en trois catégories principales :

- * En-tête du fichier (ASCII)
- * Images du CubeLED
- * Luminosité

Chaque partie du fichier est séparée par des séparateurs de début et fin. Pour indiquer le début d'une partie, on utilise "#", pour marquer la fin d'une partie on utilise "\$" par convention.

Voici un schéma représentant la structure du fichier ".cube" :



FIGURE 2 – Représentation d'un fichier du ".cube"

En-tête : Contient les informations relatives à l'animation
Images : Contient les informations sur la position des Leds allumées (par étages)
Luminosité : Si le mode est variable, on indique la luminosité pour chaque image, sinon on indique la luminosité générale du cube.

Voici un schéma représentant un étage du cube, un étage contient huit lignes de leds :

Étage n							
Ligne 0	Ligne 1	Ligne 2	Ligne 3	Ligne 4	Ligne 5	Ligne 6	Ligne 7

FIGURE 3 – Représentation d'un étage du *CubeLed*

5 Analyse fonctionnelle

6 Analyse organique

6.1 Bibliothèque *CubeLedCommunicationLibrary*

Pour le processus de communication entre le cube à led et l'application, une bibliothèque C# a été créée. Cette bibliothèque est une sur-couche de *UsbLibraryCfptAdd*. Elle implémente les fonctionnalités qui permettent la communication spécifique entre le cube et l'application.

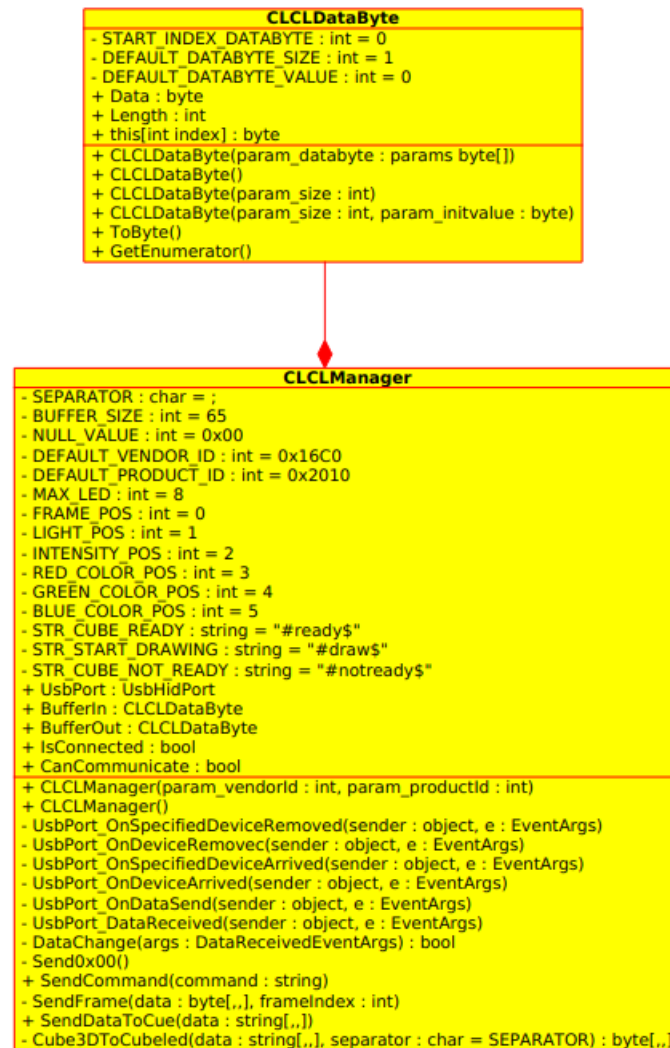


FIGURE 4 – Diagramme *UML* de la bibliothèque *CubeLedCommunicationLibrary*

7 Conclusion

Table des figures

1	Représentation 3D du tableau <i>CubeLED</i>	7
2	Représentation d'un fichier du ".cube"	8
3	Représentation d'un étage du <i>CubeLed</i>	9
4	Diagramme <i>UML</i> de la bibliothèque <i>CubeLedCommunication- Library</i>	11