



Boulila Elyssa

[elyssa.boulila@imt-atlantique.net](mailto:elyssa.boulila@imt-atlantique.net)

Bouzaien Mokhles

[mokhles.bouzaien@imt-atlantique.net](mailto:mokhles.bouzaien@imt-atlantique.net)

Thabet Achraf

[achraf.thabet@imt-atlantique.net](mailto:achraf.thabet@imt-atlantique.net)

# LAMPE COMMANDÉE PAR RFID

## FICHE TECHNIQUE DU PROJET

Version 0.1 - 20/11/2018

Formation d'ingénieur

Année scolaire 2018-2019

# I. ARCHITECTURE GLOBALE

La lampe commandée par RFID est un dispositif permettant de lire des données présentes sur un tag RFID et effectuer par suite des tâches selon le besoin de l'utilisateur, à savoir allumer la lampe imprimée en 3D ou afficher des messages.

Le produit est constitué des composants suivants :

- Une carte Arduino UNO contenant un microprocesseur.
- Un module RFID RC522.
- Un anneau Neopixel 12 LEDs.
- Un afficheur LCD 16x2 I2C.
- Une lampe imprimée en 3D (voir Annexe 1).
- 2 tags RFID.

Le produit va permettre d'effectuer les différentes tâches en utilisant deux tags RFID numérotés : le tag bleu n°1 et le tag blanc n°2. Il suffit de rapprocher le tag au lecteur (à une distance de 1 à 2 cm).

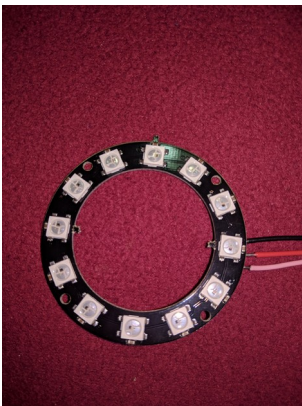


Illustration 2: Anneau Neopixel 12 LEDs

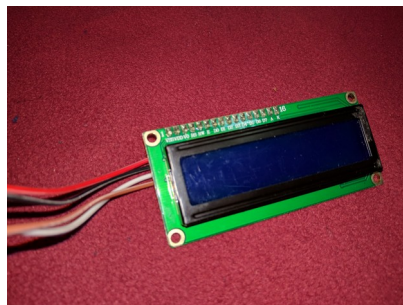


Illustration 3: Afficheur LCD 16x2 I2C



Illustration 1: Lampe imprimée en 3D



Illustration 5: Tag RFID n°1



Illustration 4: Tag RFID n°2

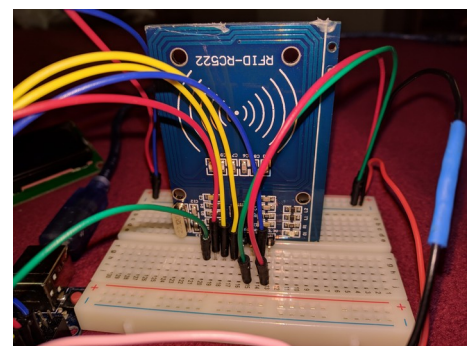


Illustration 6: Le module RFID RC522

Les différents composants électroniques peuvent être contrôlés en utilisant une carte Arduino contenant un microprocesseur. La programmation de la carte se fait à l'aide de l'environnement de développement Arduino (Illustration 7: Interface Arduino IDE).

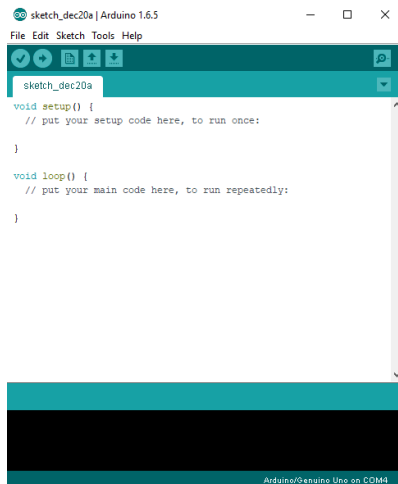


Illustration 7: Interface Arduino IDE



Illustration 8: Carte Arduino

## II. DESCRIPTION TECHNIQUE

Dans cette partie, on détaille le fonctionnement des principaux composants utilisés.

### II.1 LA CARTE ARDUINO

La carte Arduino est principalement constituée d'un microcontrôleur. Ce dernier fonctionne comme un mini ordinateur: il analyse les données récoltées à partir des différents pins d'entrée et envoie des commandes via les pins de sortie. Les pins de 1 à 13 fonctionnent comme un analogique ce qui veut dire qu'elles ne peuvent prendre que les valeurs 0V ou 5V mais certaines bornes (3,5,6,9,10,11) peuvent utiliser des tensions intermédiaires grâce au PWM.

### II.2 LE LECTEUR RFID

A notre disposition un lecteur RFID et un tag RFID passif. Pour envoyer une certaine requête, le lecteur émet de l'énergie électromagnétique avec une fréquence faible (de l'ordre de 20kHz). Celle-ci est reçue par le tag quand il est suffisamment proche (distance de lecture de quelques centimètres). Un transfert d'énergie électromagnétique aura donc lieu et permettra l'échange d'informations entre le lecteur et le tag. De cette manière, on aura accès à l'identifiant de l'étiquette RFID et aux données enregistrées sur celle-ci.

### II.3 LE NEOPIXEL

Le Neopixel utilisé est la concaténation de 12 circuits WS2812 numérotés de 0 à 11. La communication entre le Neopixel et le microcontrôleur est unifilaire. L'information envoyée vers le circuit est codée sur 24bits: 8bits pour chaque couleur qui permet de coder 256 nuance de la couleur en question. Par exemple pour coder la couleur rouge, une séquence de type "000000001111111000000000" est transmise. Sauf que le microcontrôleur doit transmettre l'information concernant tout le circuit, c'est à dire les 12 LEDs du Neopixel, donc l'information est 12x24bits reçue par le premier circuit qui s'intéresse au premier bloc de 24bits et renvoie le reste du code vers le circuit suivant, et ainsi de suite. Le schéma de codage utilisé par le circuit WS2812 pour représenter le 0 et le 1 logiques est un peu spécifique : il dépend de la période du signal on et off comme représenté dans la figure.

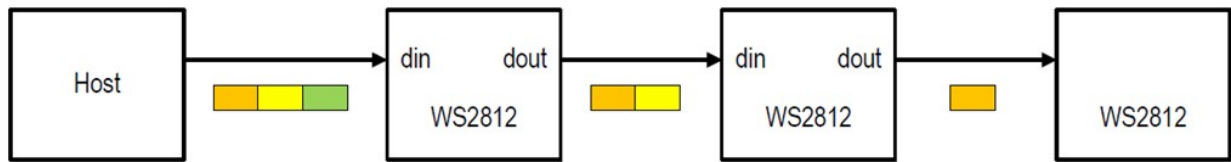


Illustration 9: Le fonctionnement du Neopixel

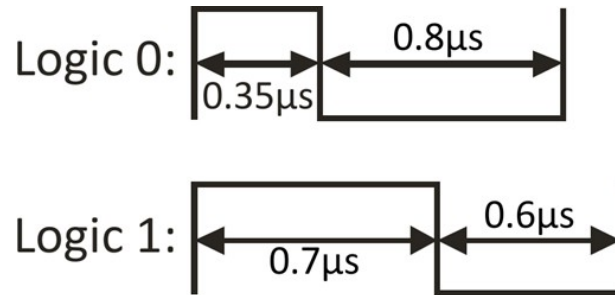


Illustration 10: Le codage du circuit WS2812

### III. MODE D'EMPLOI

Dans cette partie, vous allez être capable de brancher et exécuter différentes tâches.

#### III.1 BRANCHEMENT

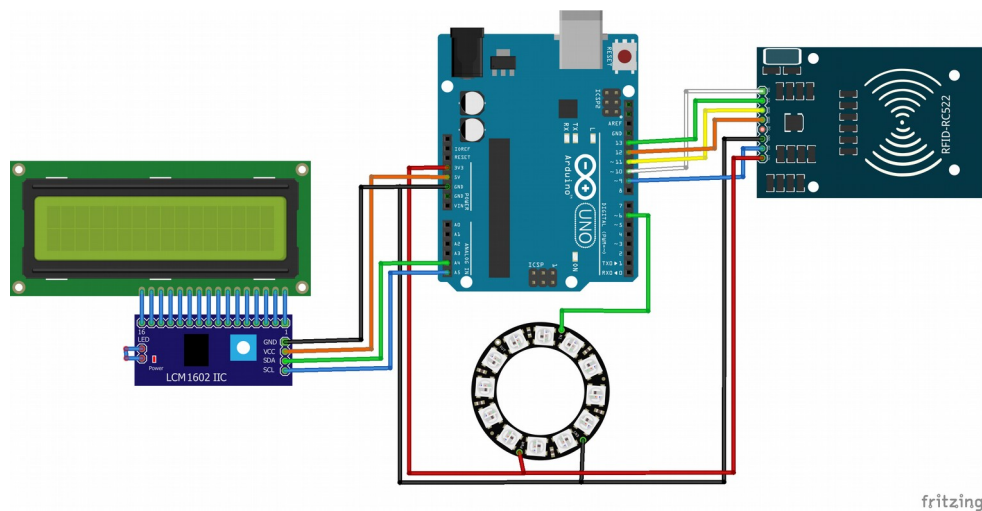
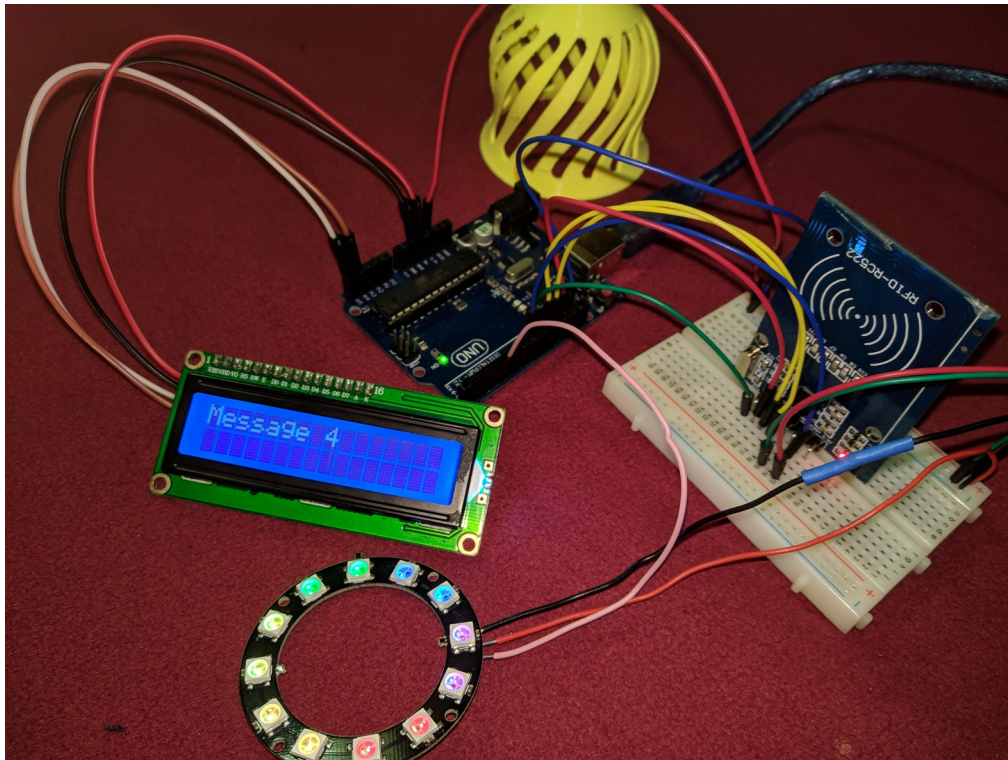


Illustration 11: Le branchement des composants



*Illustration 12: Un cas d'utilisation du prototype*

Après avoir fait le branchement comme décrit à l'illustration précédente, il est maintenant possible d'exécuter le code afin d'observer les fonctionnalités du prototype.

## III.2 LES FONCTIONS (FO ET FF)

Ci-dessous le code de toutes les fonctions obligatoires et facultatives. Pour vérifier le fonctionnement de chaque fonction, il faut décommenter la partie correspondante.

Les étapes à suivre sont dans la description de chaque fonction.

L'intégrité du code utilisé pour programmer les tâches est disponible sur [GitHub](#).

### III.2.1 IMPORTATION DES LIBRAIRIE

```
#include <Wire.h> // for communication with I2C
#include <LiquidCrystal_I2C.h> // I2C library import
#include <MFRC522.h> // RFID library import
#include <SPI.h> //Serial Peripheral Interface
#include <Adafruit_NeoPixel.h> // NeoPixel library import
#ifdef __AVR__
  #include <avr/power.h>
#endif
```

Le code ci-dessus présent dans les entête des différentes fonctions sert à importer les bibliothèques qui contiennent des fonction prédéfinies essentielles pour le fonctionnement des composantes.

### III.2.2 LES FONCTIONS INTERMÉDIAIRES

**Lire l'ID du tag RFID :**

```
String getID() {
    String strID = "";
    for (byte i = 0; i < 4; i++) {
        strID +=
            (rfid.uid.uidByte[i] < 0x10 ? "0" : "") +
            String(rfid.uid.uidByte[i], HEX) +
            (i != 3 ? ":" : "");
    }
    strID.toUpperCase();
    return strID;
}
```

**Afficher un message selon le tag lu :**

```
void printText(String usercard, int i) {
    if (usercard == "B0:D4:4E:80") {
        if (i%2 == 0){
            lcd.setCursor(0,0);
            lcd.print("Bonjour!");
            lcd.setCursor(0,1);
            lcd.print("Badge 1");
            delay(2000);
        } else {
            lcd.setCursor(0,0);
            lcd.print("Rebonjour!");
            lcd.setCursor(0,1);
            lcd.print("Badge 1");
            delay(2000);
        }
    } else if (usercard = "80:11:F7:73") {
        if (i%2 == 0){
            lcd.setCursor(0,0);
            lcd.print("Au revoir!");
            lcd.setCursor(0,1);
            lcd.print("Badge 2");
            delay(2000);
        } else {
            lcd.setCursor(0,0);
            lcd.print("A bientôt!");
            lcd.setCursor(0,1);
            lcd.print("Badge 2");
            delay(2000);
        }
    }
}
```

```
}  
}
```

### Allumer le Neopixel en monochromatique

```
void setAll(uint32_t color) {  
    for(int i=0;i<NUMPIXELS;i++){  
        pixels.setPixelColor(i, color);  
        pixels.show();  
    }  
}
```

### III.2.3 LES FO ET LES FF

```
/******
```

#### FONCTION OBLIGATOIRE 1

```
* Passer le badge n°1 devant le lecteur RFID: le message "Bonjour Badge1"  
s'affiche sur l'afficheur LCD *  
* Repasser le badge n°1 devant le lecteur RFID: Vérifier que le message  
"Rebonjour Badge1" s'affiche sur l'afficheur LCD *  
* Passer le badge n°2 devant le lecteur RFID: Vérifier que le message "Au  
revoir! Badge2" s'affiche sur l'afficheur LCD *  
* Passer le badge n°2 devant le lecteur RFID: Vérifier que le message "A  
bientot! Badge2" s'affiche sur l'afficheur LCD *
```

```
*****/
```

```
/******
```

#### FONCTION OBLIGATOIRE 2

```
* Compiler cette partie du programme *  
* Deux LEDs opposées du Néopixel clignotent, l'une bleue et l'autre rouge *  
*****/
```

```
/******
```

#### FONCTION OBLIGATOIRE 3

```
* Les LEDs du Néopixel sont blanches à la mise sous tension de la carte  
Arduino *  
* Passer le badge n°1 devant le lecteur RFID: Vérifier que les LEDs du  
Néopixel émettent une lumière rouge *  
*****/
```

```
/******
```

#### FONCTION OBLIGATOIRE 4

```
* Les LEDs du Néopixel sont blanches à la mise sous tension de la carte  
Arduino *  
* Passer le badge n°2 devant le lecteur RFID: Vérifier que les LEDs du  
Néopixel émettent une lumière multicolore variable avec le temps *  
*****/
```

/\*\*\*\*\*

### **FONCTION OBLIGATOIRE 6**

\* Cette fonction nous a servi pour ecrire le nom des propriétaires sur les deux badges \*

\* Pour pouvoir lire ce qui est écrit sur chaque badge, veuillez exécuter DumpInfo \*

\* Passer les badge n°1 devant le lecteur RFID: Vérifier que le nom du propriétaire "elyssa boulila" est écrit dans le bloc 16 (secteur 4), il faut convertir l'écriture vers ASCII \*

\* Passer les badge n°2 devant le lecteur RFID: Vérifier que le nom du propriétaire "mokhles bouzaïen" est écrit dans le bloc 16 (secteur 4) \*

\*\*\*\*\*/

/\*\*\*\*\*

### **FONCTION FACULTATIVE 1**

\* 3 motifs lumineux sont programmés dans le badge 2 \*

\* Exécuter ce programme et passer à chaque fois le badge 2 devant le lecteur pour obtenir un des 3 motifs aléatoirement \*

\*\*\*\*\*/

/\*\*\*\*\*

### **FONCTION FACULTATIVE 2**

\* Ce programme permet à l'utilisateur de choisir parmi 3 couleurs : rouge, bleu, vert via le port série de l'arduino \*

\* Veuillez compiler le programme et ouvrir le port série de l'arduino \*

\*\*\*\*\*/

/\*\*\*\*\*

### **FONCTION FACULTATIVE 3**

\* Ici sont programmés 4 messages : "Message 1", "Message 2", "Message 3", "Message 4" \*

\* A chaque fois que vous passez le badge 1 ou le badge 2, un des 4 messages s'affiche aléatoirement \*

\*\*\*\*\*/

/\*\*\*\*\*

### **FONCTION FACULTATIVE 4**

\* Ne pas débrancher l'arduino! \*

\* Compiler cette partie du programme \*

\* Ouvrir le port série \*

\* Choisir un temps de réveil comme indiqué \*

\* La lampe se transforme en réveil \*

\*\*\*\*\*/



## **IV. GLOSSAIRE**

**FF** : Fonction Facultative

**FO** : Fonction Obligatoire

**LCD** : Liquid Cristal Display

**LE** : Light Emitting Diode

**PWM** : Pulse Width Modulation

**RFID** : Radio Frequency Identifiacion

# V. ANNEXES

## V.1 ANNEXE 1

