

Micro-Contrôleurs et Open-Source Hardware

Ce rapport est réalisé dans le cadre des TP de MOSH dont le but est de faire un bilan sur la découverte de la programmation d'Arduino et les possibilités quasi infinies qu'offre l'Arduino.

Un microcontrôleur est un circuit intégré sur lequel figurent le processeur, les mémoires et les unités périphériques sans oublier les interfaces d'entrées-sorties. Les microcontrôleur participe à la miniaturisation des appareils ainsi que la démocratisation de l'informatique.

Arduino est une plateforme d'open-source de conception de prototypes électroniques qui permet aux utilisateurs de créer des objets électroniques interactifs.



Mon premier programme : LED Clignotant

Ce premier programme permet d'appréhender les premiers principes de la programmation d'Arduino.

```
int ledPin = 13;    // LED connected to digital pin 13
// The setup() method runs once, when the sketch starts
void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}

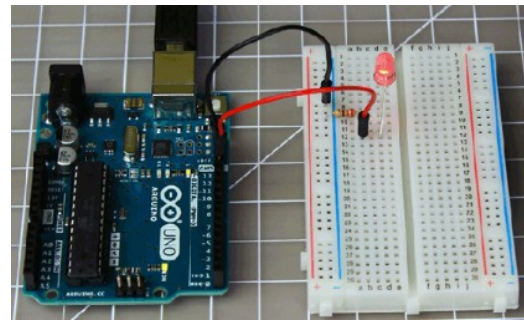
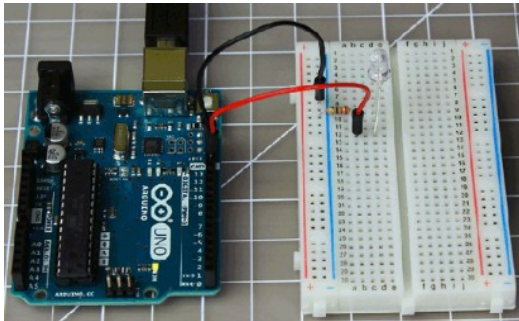
// the loop() method runs over and over again,
// as long as the Arduino has power

void loop()
{
  digitalWrite(ledPin, HIGH);  // set the LED on
  delay(1000);                 // wait for a second
  digitalWrite(ledPin, LOW);   // set the LED off
  delay(1000);                 // wait for a second
}
```

La syntaxe est celle du langage C, le programme est séparé en 2 fonctions setup() et loop() qui sont en réalité exécutés dans une fonction main() et la fonction loop() est dans une boucle while(true).

La fonction setup comme son nom l'indique permet de mettre en place les pins qui sont utilisées pour envoyer ou recevoir de l'information.

Par exemple : ici on souhaite brancher la LED sur la pin 13 et on programme cette pin de telle sorte qu'elle soit alternativement à l'état haut puis à l'état bas toutes les secondes.



Buzzer Melody :

The calculation of the tones is made following the mathematical operation:

```
*  
*  
*   timeHigh = period / 2 = 1 / (2 * toneFrequency)  
*  
* where the different tones are described as in the table:  
*  
* note   frequency   period   timeHigh  
* c       261 Hz     3830     1915  
* d       294 Hz     3400     1700  
* e       329 Hz     3038     1519  
* f       349 Hz     2864     1432  
* g       392 Hz     2550     1275  
* a       440 Hz     2272     1136  
* b       493 Hz     2028     1014  
* C       523 Hz     1912     956
```

Ce programme permet de jouer avec les fréquences que peut produire le capteur piézo d'Arduino. On établit une liste avec l'ensemble des notes utilisées pour une mélodie définie puis utiliser une boucle for pour lire la liste avec un temps de pause entre chaque note jouée.

Par exemple voici un extrait de la mélodie de téttris :

```
const byte PIN_BUZZER = 9;  
  
void setup() {  
  pinMode(PIN_BUZZER, OUTPUT);  
}  
  
void loop() {  
  tone(PIN_BUZZER, 2637, 200);
```

```

delay(400);
tone(PIN_BUZZER, 1975, 200);
delay(200);
tone(PIN_BUZZER, 2093, 200);
delay(200);
tone(PIN_BUZZER, 2349, 200);
delay(400);
tone(PIN_BUZZER, 2093, 200);
delay(200);
tone(PIN_BUZZER, 1975, 200);
delay(200);
tone(PIN_BUZZER, 1760, 200);

```

Pour des raison de temps de pause variable, la boucle for n'est pas appropriée pour cette mélodie. Vous retrouverez le code source sur mon dépôt github.

Utiliser l'écran LCD

```

#include <LiquidCrystal.h>

LiquidCrystal lcd(12,11,5,4,3,2);
void setup() {
  lcd.begin(16,2); //mettre en place le nombre de colonnes et de rangées
  lcd.setCursor(0,0);
  lcd.print("Hello World ! _ MOSH"); //texte à afficher
}

void loop() {
  lcd.scrollDisplayLeft(); //on fait défiler le texte de droite à gauche
  delay(1000); //le défilement dure 1s par caractère
}

```

Pour utiliser l'écran LCD on doit importer la librairie LiquidCrystal.
A partir de ce code au lieu d'afficher une chaîne de caractères on peut afficher la valeur d'un capteur de température :

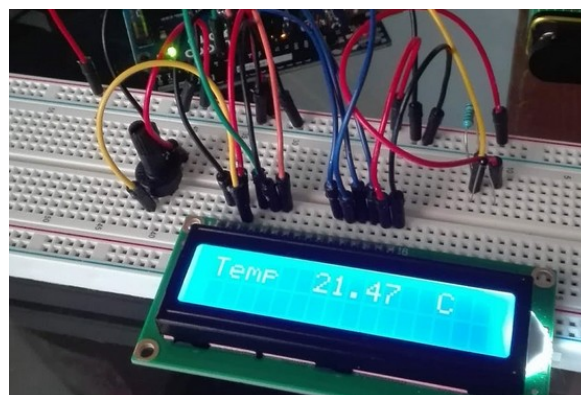
```

float lectureTemp()
{
  // Lecture de la valeur sur l'entrée analogique // Retourne une valeur entre 0->1024 pour 0->5v
  int valeur = analogRead(tempSensorPin); // Converti la lecture en tension
  float tension = valeur * 5.0; tension /= 1024.0; // Convertir la tension (mv) en temperature
  float temperature = ((tension * 1000) - 500) / 10;
  return temperature;
}

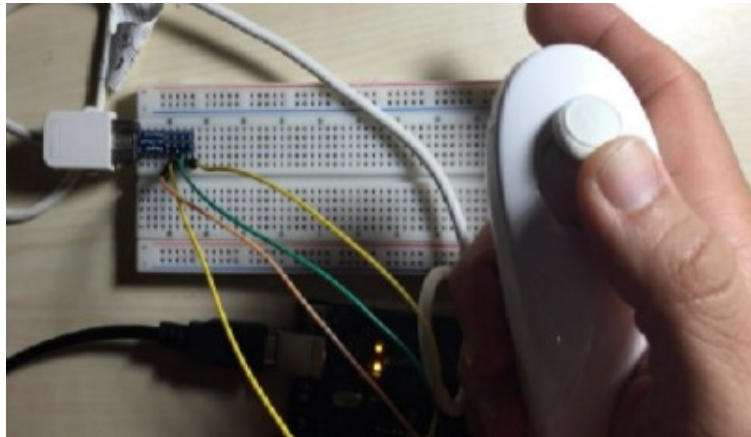
```

On récupère la tension retournée par le capteur de température puis convertir par rapport à la précision du capteur.

On est donc capable désormais d'afficher cette valeur sur l'écran LCD



Pilotage du Nunchuk Wii avec Processing:



Processing est un logiciel complémentaire à l'IDE d'Arduino qui permet à l'utilisateur d'avoir une interface graphique pour les besoins visuels comme la 3D par exemple. La syntaxe du processing est celle du java.

Dans ce TP tout d'abord, j'ai commencé à appréhender la récupération des données du Nunchuk Wii via l'IDE d'Arduino pour ensuite les afficher dans la console de l'IDE

Line	X	Y	Ax	Ay	Az	Buttons
48	174	700	506	728	0	0
49	177	363	706	199	0	0
48	171	706	523	619	0	0
49	175	558	719	447	0	0
49	174	644	499	672	1	1
74	133	355	495	258	1	1
128	131	549	619	1013	0	1

Les 2 premières colonnes montrent les positions du joystick selon 2 axes (X et Y) et les coordonnées de l'accéléromètre tri-axial sont affichées dans les 3 colonnes suivantes. Enfin les 2 dernières colonnes affichent l'état des 2 boutons intégrés au nunchuk (1 si le bouton est appuyé 0 sinon).

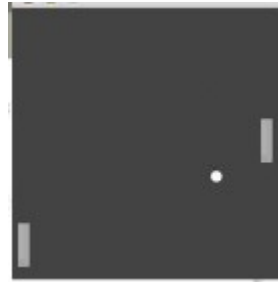
Ensuite, j'ai commencé à coder sur le jeu « Pong » avec Processing :

```
Echier Modifier Sketch Dépanner Outils Aide

PotentiometrePong_Processing

77 }
78 }
79 if(ball_x == 256 - py
80   && ball_y > paddle_y - paddle_height - ball_size
81   && ball_y < paddle_y + paddle_height + ball_size) {
82   ball_dir *= -1;
83   if(paddle_pos != 255-paddle_ppos) {
84     dy = (paddle_pos - paddle_ppos)/2.0;
85     if(dy > 5) { dy = 5; }
86     if(dy < -5) { dy = -5; }
87   }
88 }
89 // If ball hits paddle or back wall, reverse direction
90 if(ball_x < ball_size && ball_dir == -1) {
91   ball_dir *= -1;
92 }
93 // If the ball is touching top or bottom edge, reverse direction
94 if(ball_y > height-ball_size) {
95   dy = dy * -1;
96 }
97 if(ball_y < ball_size) {
98   dy = dy * -1;
99 }
100 // Draw ball
101 fill(255);
102 ellipse(ball_x, ball_y, ball_size, ball_size);
103 // Draw the paddle
104 rect(width-dist_wall, paddle_y, paddle_width, paddle_height);
105 rect(255-(width-dist_wall), paddle_y, paddle_width, paddle_height);
106 }
```

Le jeu de base consiste à réceptionner avec un bilboquet la balle rebondissant contre un mur. Pour développer davantage le code j'ai décidé de rajouter un deuxième joueur pour passer de « Pong » à « Ping Pong ».



Pour cela, on duplique tout d'abord le bilboquet avec les coordonnées inversées par rapport au 1^{er} pour le deuxième joueur, ensuite on récupère, d'un côté, la sortie du potentiomètre, de l'autre côté, la sortie du Nunchuk Wii et le tout est joué. Le premier joueur contrôlera le potentiomètre et le 2^e joueur le joystick du Nunchuk.

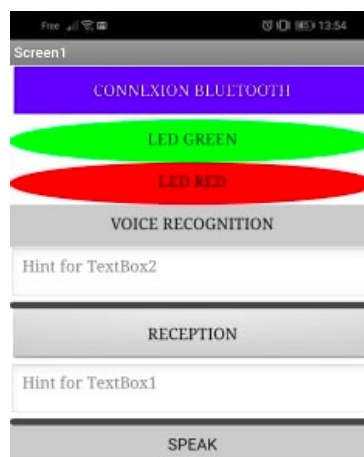
Pour arriver à ce résultat, il y a eu beaucoup de problèmes :



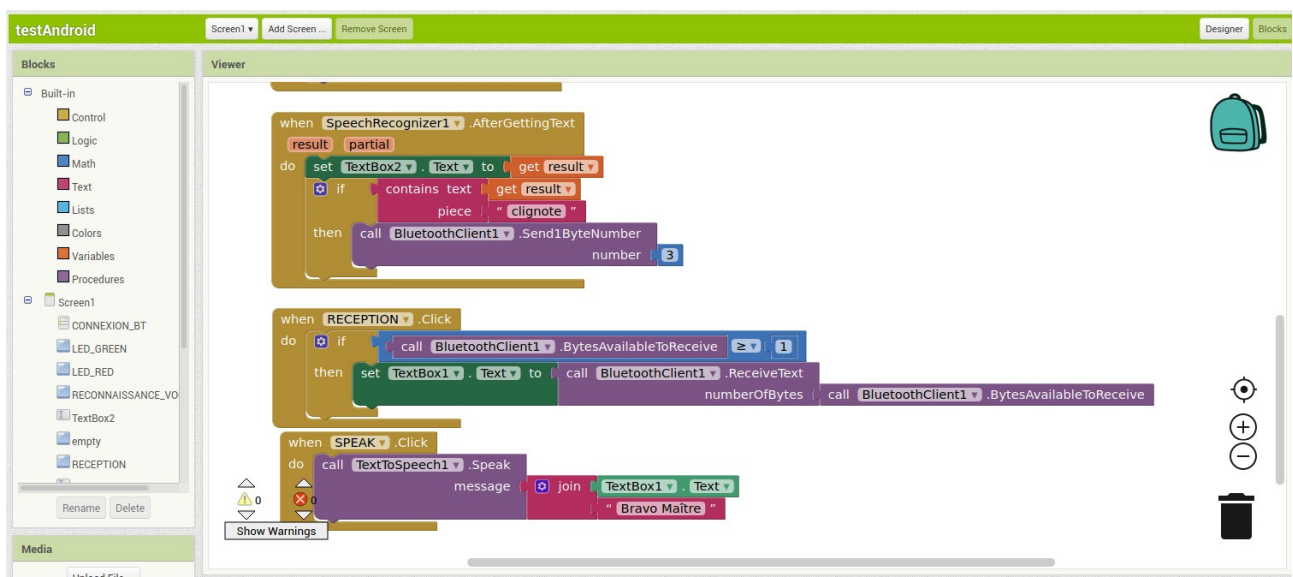
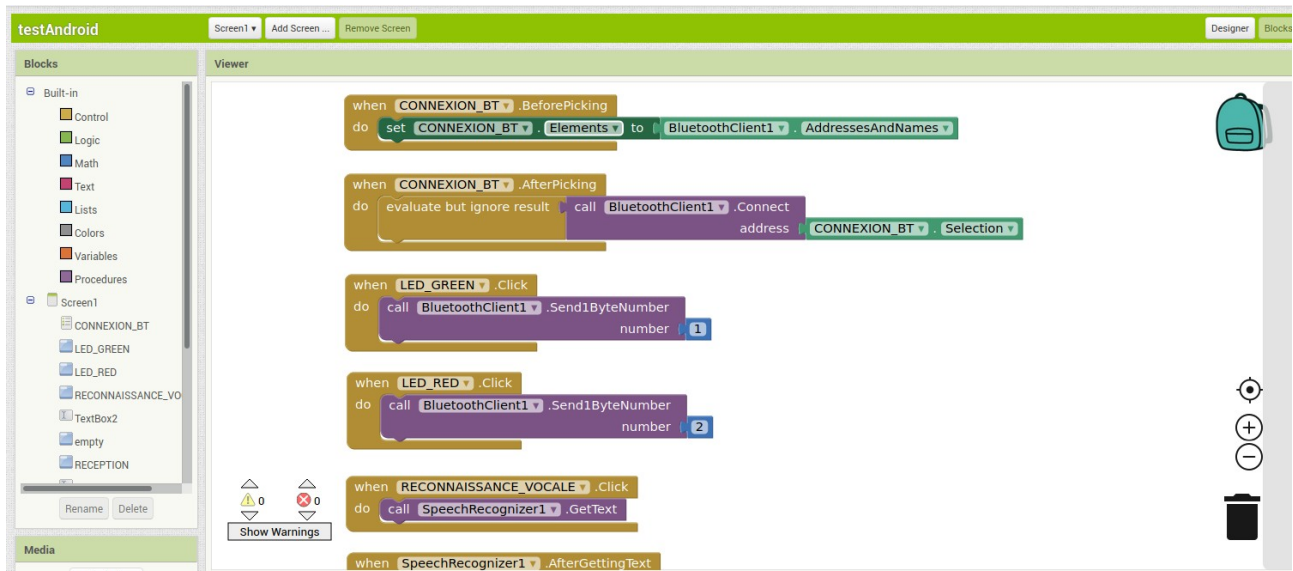
Par exemple, j'ai eu un problème d'assigner les données fournies par le potentiomètre et le Nunchuk Wii aux bilboquets, je n'arrivais pas séparer les deux flux de données ce qui fait que lorsque je bougeais le Nunchuk les 2 bilboquets se déplaçaient au même moment.

Vers ANDROID :

Pour étendre la portée et les possibilités de la programmation d'Arduino vers les applications sur téléphones, j'ai utilisé <http://ai2.appinventor.mit.edu> afin de designer une application Android :



Après la partie interface, on construit ensuite la partie fonctionnelle :



L'application peut désormais se connecter à l'appareil Arduino via Bluetooth. Et en utilisant les boutons et la reconnaissance vocale du téléphone on est capable d'allumer et éteindre les LED. Projet constitue une excellente base pour faire interagir les deux plateformes.

Conclusion :

Dans l'ensemble et personnellement, cette UF a été très formatrice pour les étudiants ne venant pas d'AE ou de GP. Ce fut une belle découverte.

Finalement, j'aimerais remercier M. Grisolia pour son accompagnement et ses conseils avisés.