RAPPORT DE SEANCE 4

- Objectif de la séance : Finaliser le code des leds, avancer le code et montage principal puis commencer le bluetooth.

- Problèmes rencontrés : Le montage ne fonctionnait pas pour le code du fichier “chansons_MP3_TEST” donc j’ai essayé de comprendre ce qui n’allait pas sans succès. Finalement, à la fin de la séance, on a appris qu’il y a eu un court-circuit au niveau du lecteur mp3 (la composante responsable du son ne marchait plus.) Cela a été une grande perte de temps.

- Le Code : (avec seulement 4 musiques)

```
#include <SoftwareSerial.h>

/* Inclut la librairie LiquidCrystal pour le LCD */
#include <LiquidCrystal.h>

void updateMenu();
void executeAction();
void action1();
void action2();
void action3();
void action4();

const int PreviousButton = 1;    //button Previous
const int SelectButton   = 2;    //button select
const int NextButton     = 3;    //button next
int menu=1;

/* Objet LCD sur les broches utilisées par la shield LCD DFrobots */
static LiquidCrystal lcd(8, 9, 4, 5, 6, 7);

#define RX 12
```

```
#define TX 13

#define B1 3

#define B2 4

#define B3 5


SoftwareSerial mySerial(RX,TX);


#define LCD lcd



static int8_t Send_buf[8] = {0} ;


#define NEXT_SONG 0X01

#define PREV_SONG 0X02

#define CMD_PLAY_W_INDEX 0X03 //DATA IS REQUIRED (number of song)

#define VOLUME_UP_ONE 0X04

#define VOLUME_DOWN_ONE 0X05

#define CMD_SET_VOLUME 0X06//DATA IS REQUIRED (number of volume from 0 up to 30(0x1E))

#define SET_DAC 0X17

#define CMD_PLAY_WITHVOLUME 0X22 //data is needed  0x7E 06 22 00 xx yy EF;(xx volume)(yy number of song)

#define CMD_SEL_DEV 0X09 //SELECT STORAGE DEVICE, DATA IS REQUIRED

#define DEV_TF 0X02 //HELLO,IM THE DATA REQUIRED

#define SLEEP_MODE_START 0X0A

#define SLEEP_MODE_WAKEUP 0X0B

#define CMD_RESET 0X0C//CHIP RESET

#define CMD_PLAY 0X0D //RESUME PLAYBACK

#define CMD_PAUSE 0X0E //PLAYBACK IS PAUSED

#define CMD_PLAY_WITHFOLDER 0X0F//DATA IS NEEDED, 0x7E 06 0F 00 01 02 EF;(play the song with the directory \01\002xxxxxx.mp3

#define STOP_PLAY 0X16
```

```
#define PLAY_FOLDER 0X17// data is needed 0x7E 06 17 00 01 XX EF;(play the 01 folder)(value xx we dont care)

#define SET_CYCLEPLAY 0X19//data is needed 00 start; 01 close

#define SET_DAC 0X17//data is needed 00 start DAC OUTPUT;01 DAC no output

#define SINGLE_PLAY 0X08//Single play(without folder)



void sendCommand(int8_t command, int16_t dat) {
  delay(20);
  Send_buf[0] = 0x7e; //starting byte
  Send_buf[1] = 0xff; //version
  Send_buf[2] = 0x06; //the number of bytes of the command without starting byte and ending byte
  Send_buf[3] = command; //
  Send_buf[4] = 0x00;//0x00 = no feedback, 0x01 = feedback
  Send_buf[5] = (int8_t)(dat >> 8);//datah
  Send_buf[6] = (int8_t)(dat); //datal
  Send_buf[7] = 0xef; //ending byte
  for(uint8_t i=0; i<8; i++){
    mySerial.write(Send_buf[i]) ;
  }
}



void setup(){
  mySerial.begin(9600);//Start our Serial coms for our serial monitor!
  delay(500);//Wait chip initialization is complete
  sendCommand(CMD_SEL_DEV, DEV_TF);//select the TF card
  delay(200);
  pinMode(PreviousButton,INPUT_PULLUP); //définit le bouton précedent, en entrée numérique sur la broche 1
  pinMode(SelectButton,INPUT_PULLUP); //définit le bouton selectionner, en entrée numérique sur la broche 2
```

```
  pinMode(NextButton,INPUT_PULLUP); //définit le bouton suivant, en entrée numérique sur la broche 3
  lcd.begin(16,2);   //configuration du LCD
  //lcd.backlight();
  updateMenu();
}

void loop() {

    if(!digitalRead(PreviousButton)){
      menu--;
      updateMenu();
      delay(100);
      while(!digitalRead(PreviousButton));
    }
    if(!digitalRead(NextButton)){
      menu++;
      updateMenu();
      delay(100);
      while(!digitalRead(NextButton));
    }
    if(!digitalRead(SelectButton)){
      updateMenu();
      executeAction;
      delay(100);
      while(!digitalRead(SelectButton));
    }
}

void updateMenu(){
  switch(menu){
```

```
    case 0:
      menu=1;
    case 1:
      lcd.clear();
      lcd.print("Ain't no mountain high enough");
      lcd.setCursor(0,1);
      lcd.print("Tammi Terrell");
      break;
    case 2:
      lcd.clear();
      lcd.print( "ABC");
      lcd.setCursor(0,1);
      lcd.print("Jackson Five");
      break;
    case 3:
      lcd.clear();
      lcd.print("Don't stop the music");
      lcd.setCursor(0,1);
      lcd.print("Rihanna");
      break;
    case 4:
      lcd.clear();
      lcd.print("September");
      lcd.setCursor(0,1);
      lcd.print("Earth, Wind & Fire");
      break;
    case 5:
      menu=4;
      break;

}
```

```
void executeAction(){
  switch(menu){
    case 1:
      action1();
      break;
    case 2:
      action2();
      break;
    case 3:
      action3();
      break;
    case 4:
      action4();
      break;
  }
}

void action1(){
  sendCommand(SINGLE_PLAY, 0X000001);
  delay(144000);
}

void action2(){
  sendCommand(SINGLE_PLAY, 0X000002);
  delay(178000);
}

void action3(){
  sendCommand(SINGLE_PLAY, 0X000003);
  delay(307000);
```

```
}

void action4(){
  sendCommand(SINGLE_PLAY, 0X000004);
  delay(215000);
}
```

- Changement au niveau du code :

J'ai modifié le loop du code en changeant les if/else et for en switch/case pour optimiser le code.

- Montage ECRAN LCD + LECTEUR MP3 + HAUT-PARLEURS + BOUTON POUSSOIR