

COMPTE RENDU – SÉANCE 8

Lors de cette séance, nous avons changé notre ancienne carte esp32 Heltec Wifi Lora 32 par une esp32 WROOM. J'ai, avec cette nouvelle carte, réussi à faire communiquer l'esp32 WROOM et la arduino UNO. Nous arrivons donc maintenant à faire transiter deux données entre les cartes : une pour la validation de l'ouverture (variable "boite") et l'autre pour la réinitialisation du code, le nouveau code saisi par le propriétaire est alors envoyé (variable "code").

J'ai donc finalement rassemblé tous les codes ensemble pour n'en avoir plus que deux (un pour chaque carte) et refait marcher tous les modules ensemble. L'écran LCD ne fonctionne parfois pas mais ce "bug" doit être dû à un mauvais contact dans un fil : lorsqu'on bouge légèrement l'écran il remarche parfaitement.

Pour l'instant, je ne parviens pas à remplacer l'ancien code par le nouveau, l'ancien étant de type char : `char codebon[] = "1234"`, et le nouveau de type String ou int, on obtient une erreur lors de la compilation (liée aux deux variables de différent type). Il faudrait alors parvenir à "écraser" l'ancien code et remplacer celui-ci par le nouveau directement dans "codebon".

Le code pour la carte maitre (esp32) est le suivant :

```
#include <SoftwareSerial.h>
#include "SPIFFS.h"
#include "WiFi.h"
#include "ESPAsyncWebServer.h"

SoftwareSerial ArduinoSlave(16,17);

// variables
String etat;
String answer;
String msg;
int code_int=1234;
int boite=0;
int intVal1=0,oldIntVal1=0;
int intVal2=1234,oldIntVal2=1234;

void setup(){
  Serial.begin(9600);
  ArduinoSlave.begin(9600);
```

```

void loop(){

  intVal1 = boite;
  intVal2 = code_int;

  //Send data to slave
  if(oldIntVal1!=intVal1 || oldIntVal2!=intVal2 ){
    Serial.print("Master sent : ");
    Serial.print(intVal1);
    Serial.print("x");
    Serial.println(intVal2);

    ArduinoSlave.print(intVal1);
    ArduinoSlave.print("x");
    ArduinoSlave.print(intVal2);
    oldIntVal1=intVal1;
    oldIntVal2=intVal2;
  }
  delay(1000);
  //Read answer from slave
  readSlavePort();

  //Send answer to monitor
  if(answer!=""){
    Serial.println("Slave received : ");
    Serial.println(answer);
    answer="";
  }
}

void readSerialPort(){
  while (Serial.available()) {
    delay(10);
    if (Serial.available() >0) {
      char c = Serial.read(); //gets one byte from serial buffer
      msg += c; //makes the string readString
    }
  }
  Serial.flush();
}

void readSlavePort(){
  while (ArduinoSlave.available()) {
    delay(10);
    if (ArduinoSlave.available() >0) {
      char c = ArduinoSlave.read(); //gets one byte from serial buffer
      answer += c; //makes the string readString
    }
  }
}

```

Le code pour la carte esclave (UNO) est le suivant :

```

#include <SoftwareSerial.h>
#include <Keypad.h>
#include <Servo.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

SoftwareSerial ArduinoMaster(0,1);

String msg="",boite="",code="";
int num1=-1,num2=-1;
int sep;
void setup(){
  Serial.begin(9600);
  ArduinoMaster.begin(9600);
void loop(){

  readMasterPort();
  convertMsgToMultiCmd();
  lecturechiffre();

  // Send answer to master
  if(num1!=-1 && num2!=-1){
    Serial.print("boite : " );
    Serial.println(num1);

    Serial.print("Code : " );
    Serial.println(num2);

    ArduinoMaster.print("Sensor 1 : " );
    ArduinoMaster.println(num1);

    ArduinoMaster.print("Sensor 2 : " );
    ArduinoMaster.println(num2);
    //nouveau_code = num2;
    Serial.println(boite);
    Serial.println(code);

    num1=-1;
    num2=-1;
  }
}

```

```

void readMasterPort(){
  while (ArduinoMaster.available()) {
    delay(10);
    if (ArduinoMaster.available() >0) {
      char c = ArduinoMaster.read();

      msg += c;
    }
  }
  ArduinoMaster.flush();
}

void convertMsgToMultiCmd(){
  if (msg.length() >0) {
    sep = msg.indexOf('x');
    boite = msg.substring(0, sep);
    code = msg.substring(sep+1, msg.length()+1);

    char carray1[6]; //magic needed to convert string to a number
    boite.toCharArray(carray1, sizeof(carray1));
    num1 = atoi(carray1);

    |
    char carray2[5];
    code.toCharArray(carray2, sizeof(carray2));
    num2 = atoi(carray2);
    char nouveau_code = (char) num2;
    Serial.print(nouveau_code);
    for(int j=0; j<code.length(); j++){
      codebon[j] += code.charAt(j);
    }

    msg="";
  }
}

```

Les codes ci-dessus ne montrent que les parties servant à la communication entre les deux cartes.

Pour la présentation finale, une authentification pour la page html sera sûrement ajoutée pour que seul le propriétaire puisse s'y connecter avec des identifiants.