

DEPARTMENT OF EDUCATION AND TRAINING OF HO CHI MINH CITY

FPT UNIVERSITY

FACULTY OF INFORMATION ASSURANCE



FPT UNIVERSITY

IOT102 – INTERNET OF THINGS

Project report

BLUETOOTH CONTROL LED MATRIX

Advisor: Mrs. Tran Cong Doi

HO CHI MINH CITY, MARCH 2023

MEMBER LIST & WORKLOAD

No.	Full name	Student ID	Percentage of work
01	Nguyen Minh Tri	SE172250	100
02	Pham Trung Hau	SE171764	100
03	Pham Tri Duc	SE171459	100
04	Dang Hoang Nguyen	SE171946	100

Abstract

A Bluetooth project involves the use of Bluetooth technology to enable wireless communication between two or more electronic devices. The project typically includes designing, building, and testing a system that incorporates Bluetooth modules, microcontrollers, and other electronic components to establish and maintain a stable connection between devices. The project may involve developing custom software to control and monitor the devices, as well as integrating the system with other hardware or software platforms. Bluetooth projects are often used in a wide range of applications, including home automation, healthcare, automotive, and industrial settings, among others.

Introduction

Arduino definition

The open source Arduino platform is used to create electronic projects. The Arduino's control circuit and the computer's software, or IDE, are its components. The code is created on the computer and uploaded to the control circuit. Pushbuttons, Lights, motors, speakers, GPS units, cameras, etc. may all be controlled by Arduino. When the internet is used to create simple circuits like motor control circuits, light sensor circuits that switch on and off lights, etc., you may create items like 3D printers, robots, hot air balloons, drones, etc.

Bluetooth led matrix controller installation

Circuit

The circuit used in this section will be as follows. Keep in mind that these wires we plug in directly into the Arduino board to make the blueprint easier to see. In fact, when doing this, please do not plug in too many wires into one hole. Use breadboard instead

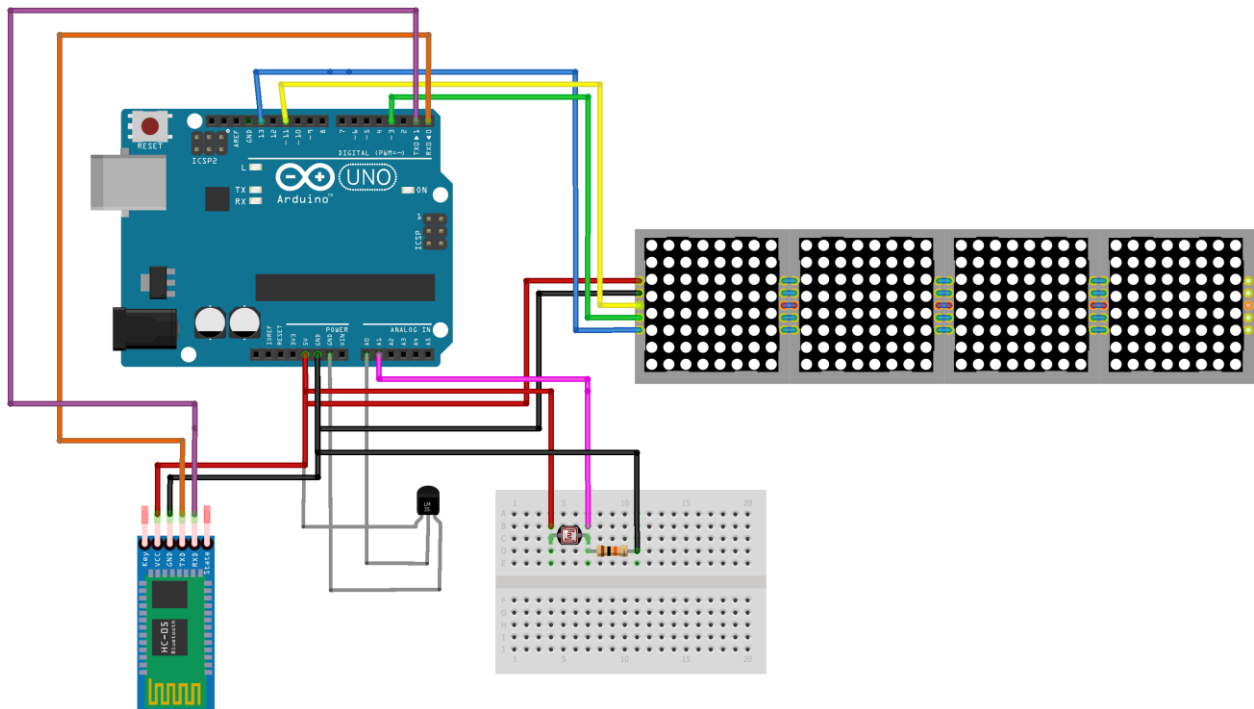


Figure 1. The picture shows the circuit of Bluetooth led matrix controller

Schema

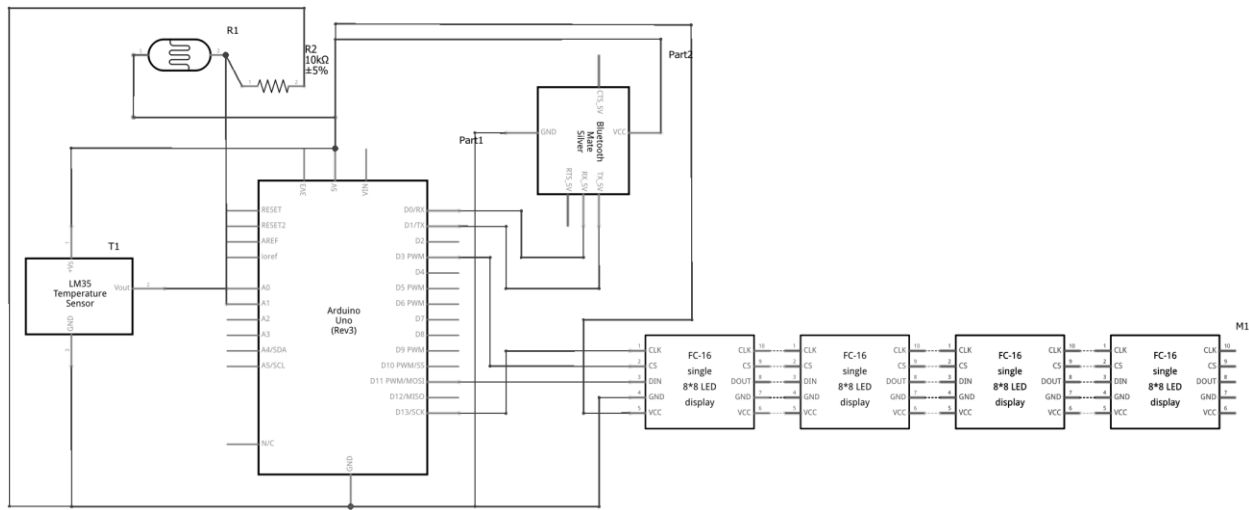


Figure 2. The picture shows the schema of Bluetooth led matrix controller

Hardware required

- Bluetooth module HC-05
- Module LED matrix 8x32 MAX7219
- Arduino UNO Components
- Hook-up wires
- 10K Ohm resistor
- Photoresistor
- Temperature Sensor – LM35

Software requirement:

- Fritzing
 - Can be downloaded form: <https://fritzing.org/>
- Arduino IDE
 - Can be downloaded form: <https://www.arduino.cc/en/software>
- ArduTooth (Only for Android)
 - Can be downloaded form: <https://shorturl.at/bijrH>

Testing phases:

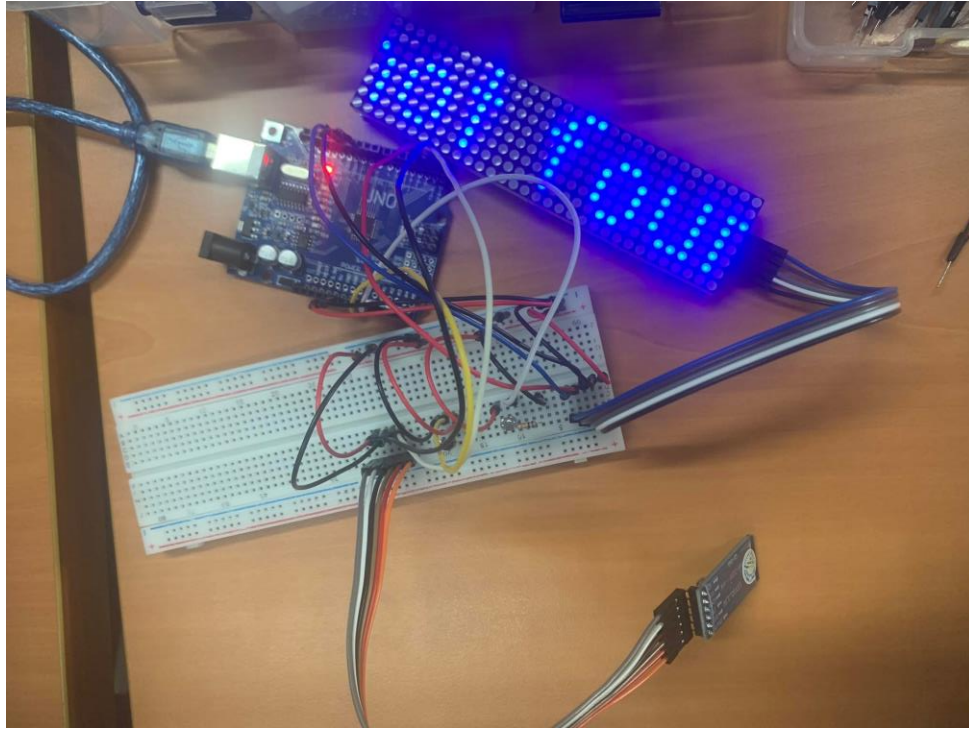


Figure 3. The picture shows the test in print mode

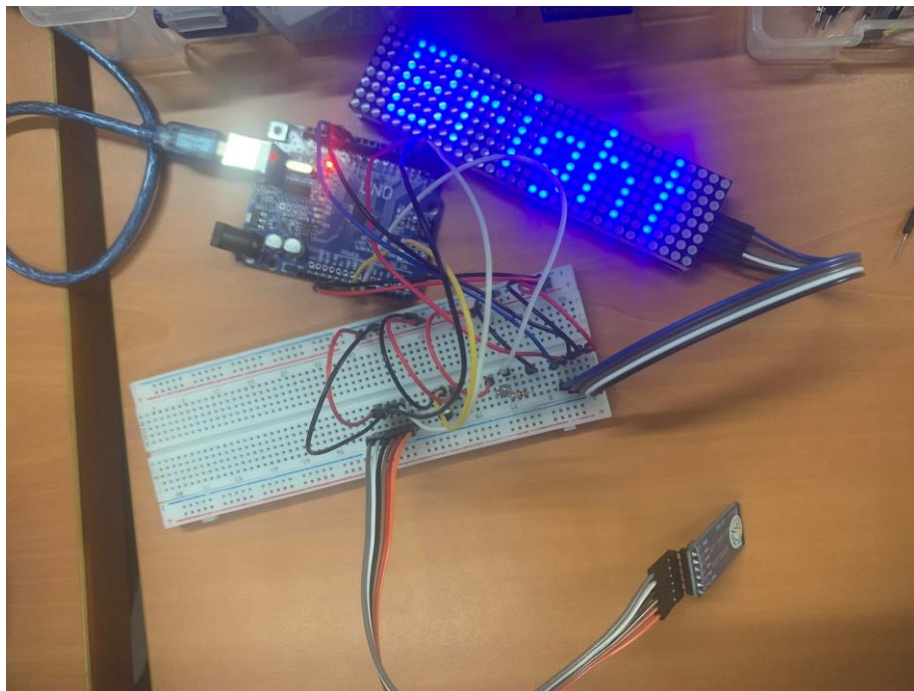


Figure 4. The picture shows the test in photoresistor mode

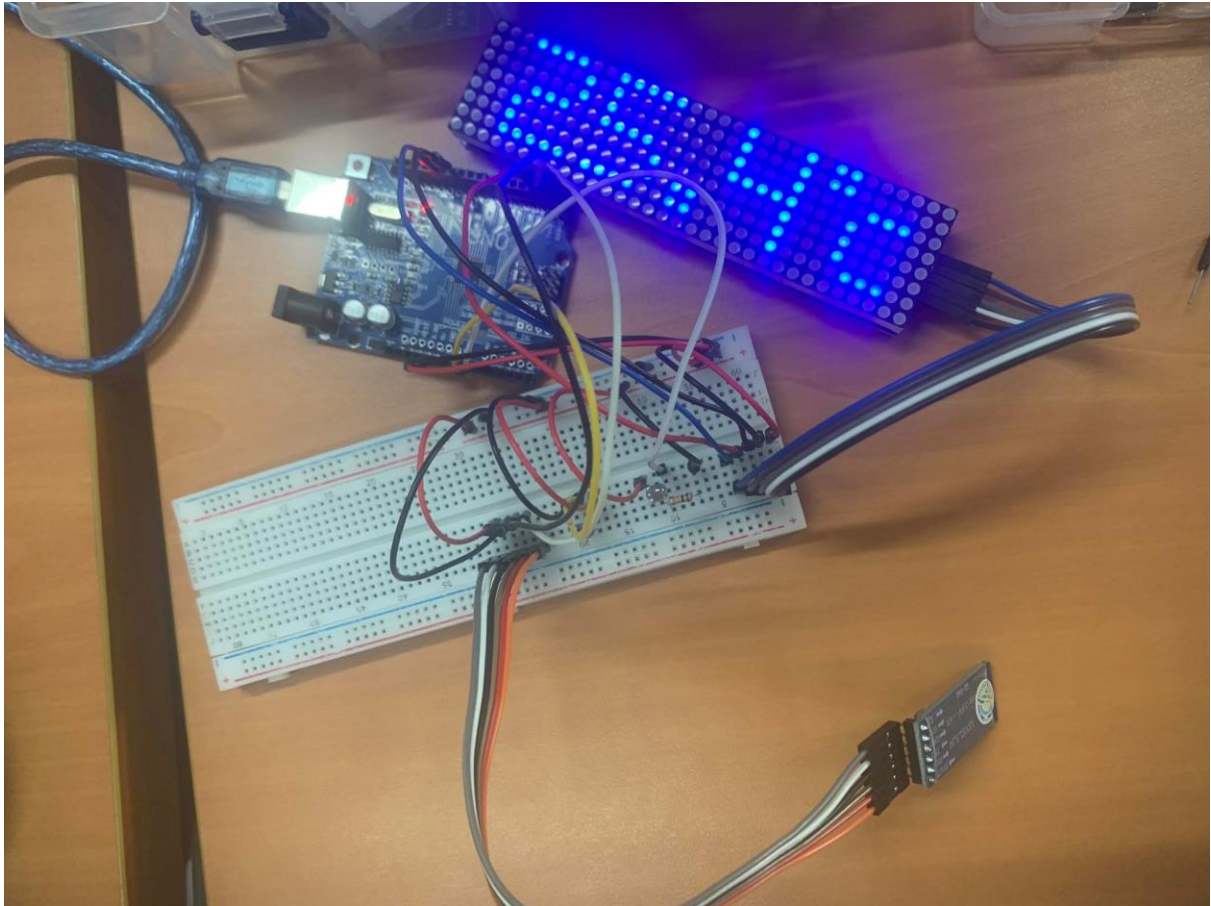


Figure 5. The picture shows the test in temperature mode

Code guides for Bluetooth led control matrix:

```
#include <MD_Parola.h>

#include <MD_MAX72xx.h>

#define HARDWARE_TYPE MD_MAX72XX::FC16_HW
#define MAX_DEVICES 4
#define CS_PIN 3
#define PHOTORESISTOR_PIN A0
#define TEMP_CENSOR_PIN A1
#define BUFFER_SIZE 64
```

IOT102 – INTERNET OF THINGS

```
MD_Parola myDisplay = MD_Parola(HARDWARE_TYPE, CS_PIN,
MAX_DEVICES);

uint8_t degC[] = { 6, 3, 3, 56, 68, 68, 68 }; // Deg C

char buffer[BUFFER_SIZE];

int mode = 0;

bool bScrolling = false;

class Smoothing {

    int Index = 0;

    int Total = 0;

    int SmoothFactor;

    int* Values;

public void begin(int smoothFactor = 10) {

    SmoothFactor = smoothFactor;

    Values = new int[smoothFactor];

    for (int num = 0; num < smoothFactor; num++) {

        Values[num] = 0;

    }

}

void add(int value) {
```


IOT102 – INTERNET OF THINGS

```
Total = Total - Values[Index];

Values[Index] = value;

Total = Total + Values[Index];

Index++;

if (Index >= SmoothFactor) {

    // ...wrap around to the beginning:

    Index = 0;

}

}

int get() {

    return Total / SmoothFactor;

}

Smoothing::~~Smoothing() {

    delete[] Values;

}

};

Smoothing myTempSensor;

Smoothing myPhotoresistor;

void setup() {

    Serial.begin(9600);

    // Intialize the display:
```

IOT102 – INTERNET OF THINGS

```
myDisplay.begin();

myDisplay.setIntensity(0); //brightness (0-15)

myDisplay.displayClear();


myDisplay.addChar('$', degC);
//myDisplay.addChar('&', degF);


myTempSensor.begin(50);
myPhotoresistor.begin(10);
}


void printScroll(String text) {
    bScrolling = true;
    strcpy(buffer, text.c_str());

    myDisplay.displayReset();
    myDisplay.displayScroll(buffer, PA_CENTER, PA_SCROLL_LEFT,
100);
}


void printAlign(String text, textPosition_t align) {
    bScrolling = false;
    strcpy(buffer, text.c_str());

    myDisplay.displayReset();
```

IOT102 – INTERNET OF THINGS

```
    myDisplay.setTextAlignment(align);

    myDisplay.print(buffer);
}

void tempCommand() {

    myTempSensor.add(analogRead(TEMP_CENSOR_PIN));

    float mv = (myTempSensor.get() / 1023.0) * 5000;

    float cel = mv / 10;

    String strcel = String(cel, 1);

    printAlign(strcel + '$', PA_CENTER);

    delay(100);
}

int max = 0;

void lightCommand() {

    myPhotoresistor.add(analogRead(PHOTORESISTOR_PIN));

    int range = map(myPhotoresistor.get(), 0, 600, 0, 3);

    switch (range) {

    case 0:

        printAlign("Dark", PA_CENTER);
```

```
        break;

    case 1:

        printAlign("Dim", PA_CENTER);

        break;

    case 2:

        printAlign("Medium", PA_CENTER);

        break;

    case 3:

        printAlign("Bright", PA_CENTER);

        break;

    }

    delay(10);

}

void loop() {

    if (Serial.available() > 0) {

        String line = Serial.readString();

        if (line[0] == '!') {

            String cmd = line.substring(1, 6);

            if (cmd == "print") {

                mode = 0;
            }
        }
    }
}
```

```

        String print = line.substring(7, line.length());

        if (print[0] == '-')
        {
            String text = print.substring(3,
print.length());

            Serial.println(text);

            if (print[1] == 's')
                printScroll(text);
            else if (print[1] == 'c')
                printAlign(text, PA_CENTER);
            else if (print[1] == 'l')
                printAlign(text, PA_LEFT);
            else if (print[1] == 'r')
                printAlign(text, PA_RIGHT);
            else
                printScroll("Subcommand not found!");
        }

        else
            printScroll(print);
    }

    else if (cmd == "tempe") {
        mode = 1; //temp mode
    }

    else if (cmd == "light") {

```

```
        mode = 2; //light mode
    }

    else {

        printScroll("Command not found!");

    }

}

else {

    mode = 0; //printing mode

    printScroll(line);

}

}

if (bScrolling && myDisplay.displayAnimate()) {

    myDisplay.displayReset();

}

if (mode == 1)

    tempCommand();

else if (mode == 2)

    lightCommand();

}
```

References

- Mpa, V. a. P. B. (2015, September 18). *Programming the Bluetooth module HC-05 using AT commands*. KYchem. <https://kychem.wordpress.com/2015/09/18/programming-the-bluetooth-module-hc-05-using-at-commands/>
- Serial - Arduino Reference*. (n.d.).
<https://www.arduino.cc/reference/en/language/functions/communication/serial/>
- B. (2022, March 2). *MAX7219 LED dot matrix display Arduino tutorial*. Makerguides.com.
<https://www.makerguides.com/max7219-led-dot-matrix-display-arduino-tutorial/>
- Arduino - Bluetooth LED Matrix | Arduino Getting Started*. (n.d.). Arduino Getting Started.
<https://arduinogetstarted.com/tutorials/arduino-bluetooth-led-matrix>