LAPORAN PRAKTIKUM MINGGU KE-8

LCD I2C 1602A

INTERNET OF THINGS

Disusun oleh:

Omar Abdul-Raoof Taha Ghaleb Al-Maktary

1941720237

TI-3H

D4 TEKNIK INFORMATIKA

TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2022

LAPORAN

A. PRAKTIKUM

Practicum 1:

The first practicum is for finding the address on the lCD.

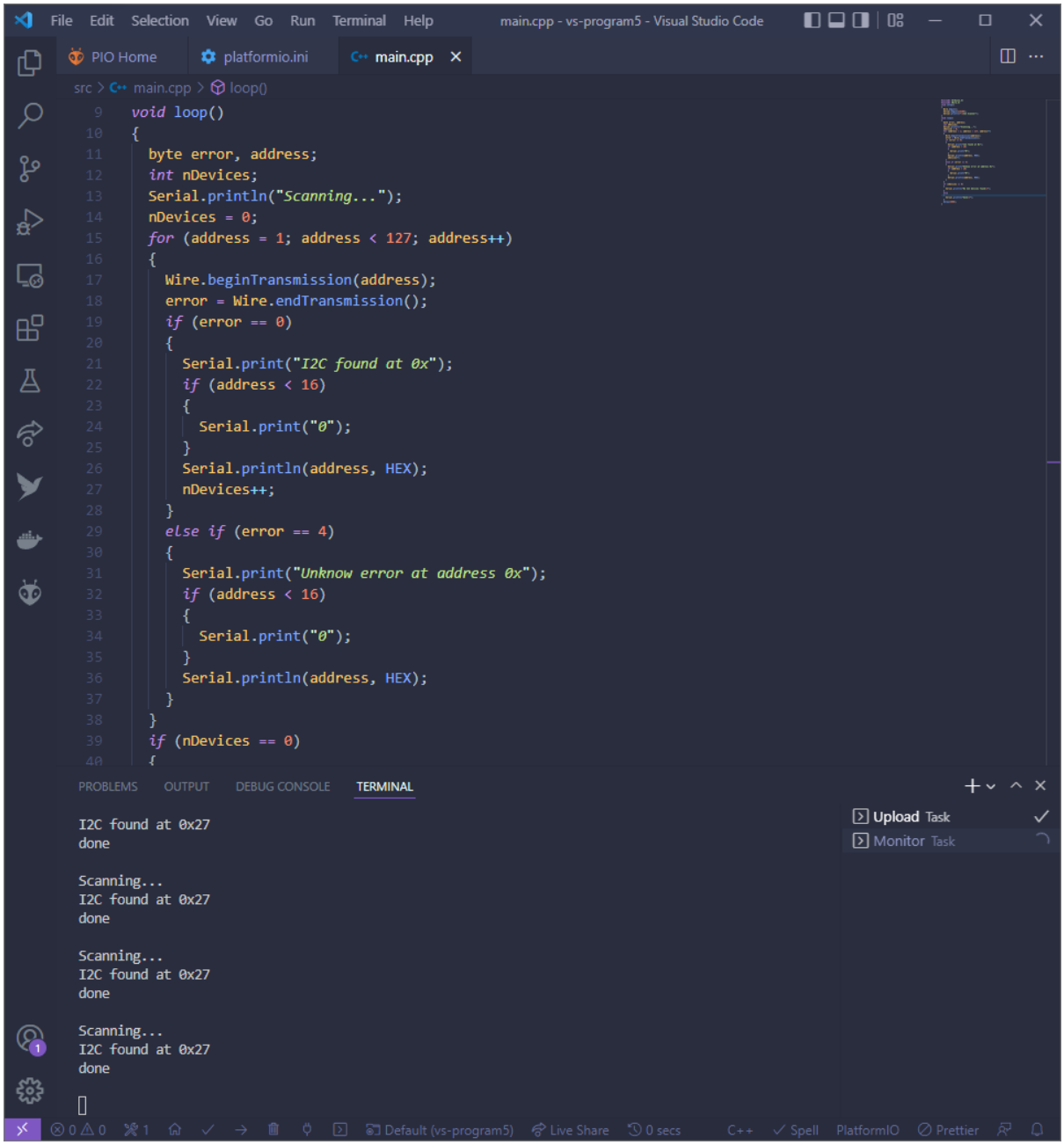
Code:

```
#include <Arduino.h>
#include <Wire.h>
void setup()
{
  Wire.begin();
  Serial.begin(115200);
  Serial.println("\nI2C Scanner");
}
void loop()
{
  byte error, address;
  int nDevices;
  Serial.println("Scanning...");
  nDevices = 0;
  for (address = 1; address < 127; address++)
  {
    Wire.beginTransmission(address);
    error = Wire.endTransmission();
    if (error == 0)
    {
      Serial.print("I2C found at 0x");
      if (address < 16)
      {
        Serial.print("0");
      }
      Serial.println(address, HEX);
      nDevices++;
    }
    else if (error == 4)
    {
      Serial.print("Unknow error at address 0x");
      if (address < 16)
      {
        Serial.print("0");
      }
      Serial.println(address, HEX);
    }
  }
  if (nDevices == 0)
  {
```

```
    Serial.println("No I2C devices found\n");
  }
  else
  {
    Serial.println("done\n");
  }
  delay(3000);
}
```

Results:

The results shows that the LCD has an address of 0x27



Practicum 2:

In this practicum will show some text on the screen. Since the LCD has two rows and 16 columns. The first row will contain the word “Polinema” and the second row will show a sliding text “Omar, Class IoT TI-3H”. for this practicum will be using LiquidCrystal_I2C library.

Code:

```
#include <Arduino.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// set the LCD number of columns and rows
int lcdColumns = 16;
int lcdRows = 2;

// set LCD address, number of columns and rows
// if you don't know your display address, run an I2C scanner sketch
LiquidCrystal_I2C lcd(0x27, lcdColumns, lcdRows);

String messageStatic = "Polinema";
String messageToScroll = "Omar, Class IoT TI-3H";

// Function to scroll text
// The function acepts the following arguments:
// row: row number where the text will be displayed
// message: message to scroll
// delayTime: delay between each character shifting
// lcdColumns: number of columns of your LCD
void scrollText(int row, String message, int delayTime, int lcdColumns)
{
  for (int i=0; i < lcdColumns; i++) {
    message = " " + message;
  }
  message = message + " ";
  for (int pos = 0; pos < message.length(); pos++) {
    lcd.setCursor(0, row);
    lcd.print(message.substring(pos, pos + lcdColumns));
    delay(delayTime);
  }
}

void setup(){
  Serial.begin(9600);
  Wire.begin(D2, D1);
  // initialize LCD
  lcd.init();
  // turn on LCD backlight
```

```
  lcd.backlight();
}

void loop(){
  // set cursor to first column, first row
  lcd.setCursor(0, 0);
  // print static message
  lcd.print(messageStatic);
  // print scrolling message
  scrollText(1, messageToScroll, 250, lcdColumns);
}
```

Result:

B. KESIMPULAN

We can conclude from the practicums above how we can setup an LCD with ESP8266 NodeMCU. I2C LCD has many types, screen colours , and sizes but they all have the same job which is to display characters and shapes from a program.

TUGAS

Question

1. Explain the function of the lcd.backlight() method call?

   The code lcd.backlight() is to turn the back light of the I2C LCD so it can show characters.

2. How do you change the brightness intensity level of your LCD?

   From the back of the LCD there is a scree that can change the contrast of the columns so they can show the characters.

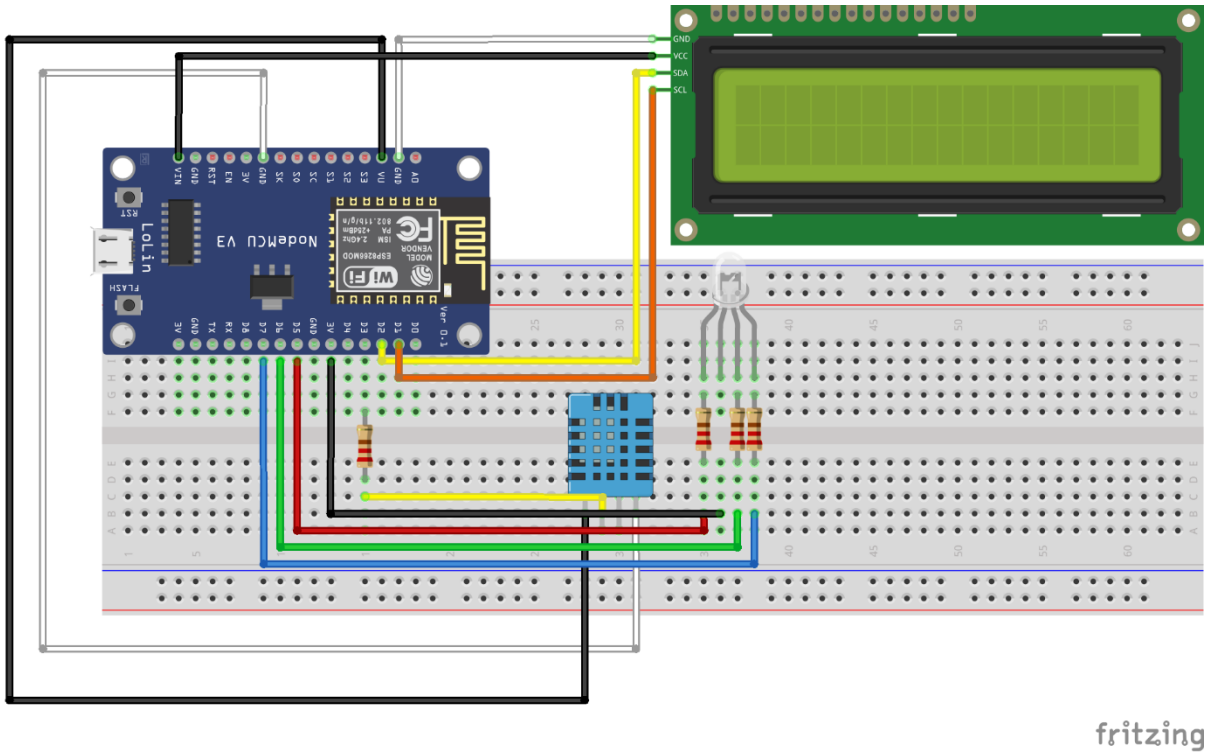
3. Please modify the data displayed on your LCD?

Task

Create a simple application using DHT11, RGB LED, and LCD. The scenario is as follows:
Make these three components in one circuit using fritzing.
Display the temperature in Fahrenheit and Celsius, the temperature displayed is the temperature in the room around you.
When the temperature is normal the blue LED will flash, when the temperature is cold the green LED will flash, and the red LED will flash when the temperature is high.
Display the current time also on the LCD.

In this assignment, I have successfully done the task but there are some issues regarding the sensor. Sometimes DHT sensor is not picking up the right number and sometimes it doesn't work at all. I have checked the wiring and it looks fine; the code works as well but there is an issue that I can't figure out why the sensor behave in such way. Sometime it would work and other times it won't.

Fritzing Design:



Code:

```
#include <Arduino.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <ESP8266WiFi.h>
#include <NTPClient.h>
#include <WiFiUdp.h>

//Network credentials
const char *ssid     = "WI-FI Name Here";
const char *password = "WI-FI Password Here";
// Define NTP Client to get time
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org");
//Week Days
String weekDays[7]={"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",
"Friday", "Saturday"};
//Month names
String months[12]={"January", "February", "March", "April", "May", "June",
"July", "August", "September", "October", "November", "December"};

// RGB LIGHT
#define RED_LED D5
#define GREEN_LED D6
```

```
#define BLUE_LED D7

// DHT SENSOR
#define DHTTYPE DHT11
DHT dht(D3, DHTTYPE);

int lcdColumns = 16;
int lcdRows = 2;

LiquidCrystal_I2C lcd(0x27, lcdColumns, lcdRows);

String tempNow = "temp";
String timeNow = "time";

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

  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  timeClient.begin();
  timeClient.setTimeOffset(25200);


  pinMode(RED_LED, OUTPUT);
  pinMode(GREEN_LED, OUTPUT);
  pinMode(BLUE_LED, OUTPUT);

  Wire.begin(D2, D1);
  lcd.init();
  lcd.backlight();

  delay(2000);
}
void getTemp()
{
  float t = dht.readTemperature();
  float f = dht.readTemperature(true);
  if (isnan(t) || isnan(f))
  {
    lcd.setCursor(0, 0);
    lcd.print("Err Temp");
    Serial.println("Failed to read from DHT sensor!");
    return;
```

```
  }
  tempNow = String(t) + char(0b11011111) + "C " + String(f) + char(0b11011111)
+ "F";
  Serial.println(tempNow);
  lcd.setCursor(0, 0);
  lcd.print(tempNow);
  if (t > 25.0)
  {
    digitalWrite(RED_LED, LOW);
    digitalWrite(GREEN_LED, HIGH);
    digitalWrite(BLUE_LED, HIGH);
  }
  else if (t >= 18.0)
  {
    digitalWrite(RED_LED, HIGH);
    digitalWrite(GREEN_LED, HIGH);
    digitalWrite(BLUE_LED, LOW);
  }
  else if (t < 18.0)
  {
    digitalWrite(RED_LED, HIGH);
    digitalWrite(GREEN_LED, LOW);
    digitalWrite(BLUE_LED, HIGH);
  }
}

void getTime()
{
  timeClient.update();
  time_t epochTime = timeClient.getEpochTime();
  String formattedTime = timeClient.getFormattedTime();
  int currentHour = timeClient.getHours();
  int currentMinute = timeClient.getMinutes();
  int currentSecond = timeClient.getSeconds();
  String weekDay = weekDays[timeClient.getDay()];
  struct tm *ptm = gmtime ((time_t *)&epochTime);
  int monthDay = ptm->tm_mday;
  int currentMonth = ptm->tm_mon+1;
  String currentMonthName = months[currentMonth-1];
  int currentYear = ptm->tm_year+1900;
  String currentDate = String(monthDay) + "-" + String(currentMonth) + "-" +
String(currentYear);

  timeNow = currentDate + " " + String(currentHour) + ":" +
String(currentMinute);
  Serial.println(timeNow);
  lcd.setCursor(0, 1);
  lcd.print(timeNow);
```

```
}
void loop()
{
  digitalWrite(RED_LED, HIGH);
  digitalWrite(GREEN_LED, HIGH);
  digitalWrite(BLUE_LED, HIGH);
  getTemp();
  getTime();
  delay(3000);
}
```

Results: