دانشگاه تهران
دانشکده‌ی مهندسی
برق و کامپیوتر

گزارش تمرین اول

سامانه نهفته آبیاری هوشمند

حسین بیاتی – محمد سعید صدیقی – مهدیار هرندی – متین نبی زاده

استاد:

دکتر مهدی مدرسی

بهار ۱۴۰۴

در این پروژه تلاش کردیم یک سامانه آبیاری هوشمند مبتنی بر پروتکل I²C بسازیم که شامل دو نقش اصلی باشد:

- **گره لبه (Edge Node)** روی یک برد Arduino Uno (۲ گره)
- **گره مرکزی (Central Node)** روی یک برد دیگرArduino Uno

هر گره لبه سنسور رطوبت خاک را از پایه A0 می‌خواند، یک LED را به‌عنوان نماینده شیر آب کنترل می‌کند و زاویه‌ی یک سروو موتور را بر اساس رطوبت تنظیم می‌کند. گره مرکزی هر دو ثانیه یک‌بار با I²C از گره لبه درخواست دو بایت داده می‌کند، مقدار رطوبت را دریافت و بر اساس دو آستانه (۸۰۰ برای حالت اشباع و ۵۰۰ برای نیاز به آبیاری) یکی از فرمان‌های 'S'(توقف)، 'W'(آبیاری) یا 'N'(حالت عادی) را به گره لبه می‌فرستد.

۱ .گره لبه (Edge Node)

تعریف و تنظیم اولیه:

```
#include <Wire.h>
#include <Servo.h>

#define SLAVE_ADDRESS    0x08    // آدرس I²C گره لبه
#define SOIL_SENSOR_PIN A0       // پایه سنسور رطوبت
#define LED_PIN          2       // پایه LED نمایش شیر آب
#define SERVO_PIN        8       // پایه سروو موتور

Servo potServo;
int soilMoisture = 0;           // آخرین مقدار رطوبت خوانده‌شده
```

- ابتدا کتابخانه‌های I²C و سروو وارد شدند و آدرس، پین‌های سخت‌افزار و متغیرهایی که مقدار رطوبت را نگه می‌دارند تعریف شدند.

- راه‌اندازی I²C و محرک‌ها:

```
void setup() {
  Wire.begin(SLAVE_ADDRESS);          // تنظیم به عنوان Slave
  Wire.onRequest(onMasterRequest);    // ISR برای ارسال داده
  Wire.onReceive(onMasterReceive);    // ISR برای دریافت فرمان

  pinMode(LED_PIN, OUTPUT);
  potServo.attach(SERVO_PIN);
  Serial.begin(9600);
}
```

- در `setup()`، با `Wire.begin(0x08)` آدرس Slave ثبت شد وISR های `onRequest` و `onReceive` تنظیم گشتند تا هنگام فراخوانی توسط Master اجرا شوند.

- خواندن از سنسور و به‌روزرسانی محرک‌ها:

```
void loop() {
  soilMoisture = analogRead(SOIL_SENSOR_PIN);
  Serial.print("Local Moisture: ");
  Serial.println(soilMoisture);

  updateActuators(soilMoisture);
  delay(500);   // تأخیر برای جلوگیری از بار زیاد حلقه
}
```

- هر ۵۰۰ میلی‌ثانیه مقدار رطوبت خوانده و در سریال‌مانیتور لاگ می‌شود و سپس `updateActuators()` وضعیت LEDو زاویه سروو را بر اساس آن تنظیم می‌کند.

- ISR برای ارسال داده به Master:

```
void onMasterRequest() {
  Wire.write((soilMoisture >> 8) & 0xFF);
  Wire.write(soilMoisture & 0xFF);
}
```

- هنگامی که Master با `requestFrom()` فراخوانی کند، این تابع دو بایت `soilMoisture` را ارسال می‌کند. در عمل گاهیMaster کمتر از دو بایت می‌گرفت که نشان‌دهنده مشکل در زمان‌بندی یا آماده‌نبودن Slave بود.

- ISR برای دریافت فرمان از Master :

```
void onMasterReceive(int bytes) {
  if (bytes < 1) return;
  char cmd = Wire.read();
  Serial.print("Cmd received: ");
  Serial.println(cmd);
}
```

- با دریافت یک بایت فرمان از Master ، این ISR اجرا و فرمان لاگ می‌شود.

- به‌روزرسانی LED و موتور سروو:

```
void updateActuators(int val) {
  if (val < 500) {
    digitalWrite(LED_PIN, HIGH);
    potServo.write(30);
  }
  else if (val > 800) {
    digitalWrite(LED_PIN, LOW);
    potServo.write(90);
  }
  else {
    digitalWrite(LED_PIN, LOW);
    potServo.write(60);
  }
}
```

- **در این تابع بر اساس مقدار رطوبت:**

- زیر ۵۰۰ LED ⇒ روشن و سروو زاویه ۳۰،
- بالای ۸۰۰ LED ⇒ خاموش و سروو زاویه ۹۰،
- در بازه ⇒ زاویه ۶۰ (حالت عادی).

۲ .گره مرکزی (Central Node)

- تعریف ثابت‌ها و راه‌اندازی:

```
#include <Wire.h>

#define EDGE_ADDR        0x08
#define HIGH_THRESHOLD   800
#define LOW_THRESHOLD    500
#define REQUEST_INTERVAL 2000UL

unsigned long lastRequestTime = 0;
```

- آدرس Slave ، آستانه‌ها و بازه زمانی درخواست‌ها تعریف شدند.

```
void setup() {
  Wire.begin();         // پیکربندی به‌عنوان Master
  Serial.begin(9600);
}
```

- حلقه اصلی: درخواست و تصمیم‌گیری:

```
void loop() {
  if (millis() - lastRequestTime < REQUEST_INTERVAL)
 return;
  lastRequestTime = millis();

  int moisture = readSoilMoisture(EDGE_ADDR);
  Serial.print("Moisture from Slave: ");
  Serial.println(moisture);

  char cmd = decideCommand(moisture);
  sendCommand(EDGE_ADDR, cmd);
}
```

- در هر دور، پس از دو ثانیه، تابع `readSoilMoisture()` صدا زده می‌شود، مقدار دریافت‌شده چاپ و بر اساس آن فرمان با `decideCommand()` انتخاب و به Slave ارسال می‌گردد.

- خواندن دو بایت از Slave :

```
int readSoilMoisture(byte addr) {
  Wire.requestFrom(addr, (byte)2);
  delay(10);
  if (Wire.available() < 2) {
    Serial.println("I2C: Insufficient data");
    return -1;
  }
  int highB = Wire.read();
  int lowB  = Wire.read();
  return (highB << 8) | lowB;
}
```

- درخواست دو بایت می‌شود و در صورت دریافت ناکافی، خطا گزارش و `-1` بازگردانده می‌شود. این مسئله در کار ما مکرراً رخ داد.

- تصمیم‌گیری فرمان:

```
char decideCommand(int m) {
  if (m < 0)                return 'N';
  if (m > HIGH_THRESHOLD)   return 'S';
  if (m < LOW_THRESHOLD)    return 'W';
  return 'N';
}
```

- اگر مقدار منفی (خطا) یا بین آستانه بود فرمان `'N'`، بالاتر `'S'` و پایین‌تر `'W'` برگردانده می‌شود.

- ارسال فرمان به Slave :

```
void sendCommand(byte addr, char cmd) {
  Wire.beginTransmission(addr);
  Wire.write(cmd);
  byte status = Wire.endTransmission();
  if (status != 0) {
    Serial.print("I2C TX Error code: ");
    Serial.println(status);
  } else {
    Serial.print(" Cmd sent: ");
    Serial.println(cmd);
  }
}
```

- با این ساختار، Master فرمان را می‌فرستد و وضعیت انتقال (مثلاً NACK) را لاگ می‌کند.

کدهای خواندن آنالوگ، کنترل LED و سروو در گره لبه به درستی کار می‌کنند، اما ارتباط دوطرفه I²C بین Master وSlave ناپایدار است و نیاز به بررسی دقیق زمان‌بندی، مقاومت‌های pull-up و اجزای متصل به باس دارد.

- نسخهٔ فاقد گره مرکزی:

در نسخهٔ مرکزی، تنها گره لبه است که تصمیم‌گیری می‌کند و به منظور رفع نیاز به پروتکل I2C، پیاده‌سازی شده است. اقسام مهم این پیاده‌سازی، در صفحهٔ بعد، شرح داده شده‌اند.

۱. در ابتدا، کتابخانه‌های مورد نیاز را به کد، افزوده و ثوابت را مقداردهی می‌کنیم:

```
#include <LiquidCrystal.h کتابخانه // <LCD

#define NOTE_C4 262  // نت‌ها برای هشدار صوتی

...

#define NOTE_C5 523int T_Sensor = A3;

int M_Sensor = A0;

int W_led = 7;

int P_led = 13;

int Speaker = 9;
```

۲. در ادامه، تابع Setup () را توضیح می‌دهیم. این تابع، علاوه بر فرمان شروع کار دادن به LCD، مقدار آنالوگ سنسور دما را نیز به درجهٔ سلسیوس تبدیل می‌کند. در اینجا، فرض شده است که خروجی میان ۰–۵ ولت است و هر ۱۰ میلی‌ولت معادل ۱ درجه سانتی‌گراد است.

```
lcd.begin(16, 2);

val = analogRead(T_Sensor);

int mv = ( val/1024.0)*5000;

cel = mv/10;
```

۳. در تابع اصلی این کد، یعنی loop ()، هر ثانیه:

- مقدار رطوبت خاک خوانده شده و دما روی LCD نمایش داده می‌شود.

- براساس مقدار رطوبت، یکی از ۳ حالت زیر اعمال می‌شود :

الف) خشک بودن خاک (Moisture > 700)

- نمایش"DRY SOIL" بر صفحهٔ نمایش
- اگر آب موجود باشد : (digitalRead(W_led) == 1)
  - پمپ روشن می‌شود (digitalWrite(13, HIGH))
- اگر آب موجود نباشد :
  - پمپ خاموش می‌شود.
  - هشدار صوتی با نت‌های C4 تاG4

ب) رطوبت متوسط (700 ≥ Moisture ≥ 300)

- نمایش"MOIST SOIL"
- پمپ خاموش

ج) خاک مرطوب (Moisture < 300)

- نمایش"WET SOIL"
- پمپ خاموش

```
lcd.clear();

int Moisture = analogRead(M_Sensor); //Read Moisture Sensor Value
```

```
 lcd.setCursor(0,0);
 lcd.print("TEMP:");
 lcd.setCursor(5,0);
 lcd.print(cel);
 lcd.setCursor(7,0);
 lcd.print("*C");

if (Moisture> 700)   // for dry soil
 {
       lcd.setCursor(11,0);
       lcd.print("DRY");
       lcd.setCursor(11,1);
       lcd.print("SOIL");
      if (digitalRead(W_led)==1) //test the availability of water in storage
      {
        digitalWrite(13, HIGH);
        lcd.setCursor(0,1);
        lcd.print("PUMP:ON");
      }
      else
      {
        digitalWrite(13, LOW);
        lcd.setCursor(0,1);
        lcd.print("PUMP:OFF");

          tone(Speaker, NOTE_C4, 500);
          delay(500);
```

```
          tone(Speaker, NOTE_D4, 500);
          delay(500);
          tone(Speaker, NOTE_E4, 500);
          delay(500);
          tone(Speaker, NOTE_F4, 500);
          delay(500);
          tone(Speaker, NOTE_G4, 500);
          delay(500);
      }
  }
   if (Moisture>= 300 && Moisture<=700) //for Moist Soil
  {
    lcd.setCursor(11,0);
   lcd.print("MOIST");
   lcd.setCursor(11,1);
   lcd.print("SOIL");
   digitalWrite(13,LOW);
   lcd.setCursor(0,1);
   lcd.print("PUMP:OFF");
}


if (Moisture < 300)  // For wet soil
{
  lcd.setCursor(11,0);
   lcd.print("WET");
   lcd.setCursor(11,1);
   lcd.print("SOIL");
```

```
    digitalWrite(13,LOW);
    lcd.setCursor(0,1);
    lcd.print("PUMP:OFF");
  }
delay(1000);
```