2023-2 IoT System Design

HW #02

자동 전등제어와

센서 데이터 출력

Due 09.23(Sat) 22:00

전산전자공학부

22100579 이진주

1. **과제 설명**

이 과제에서는 일전에 진행한 Lab4의 회로도를 기초로, 일부 핀을 변경하고 일부 기능을 추가하는 것을 목적으로 한다.

실습 환경은 nodeMCU10과 아두이노 IDE이며, 온습도 출력용 라이브러리는 *DHT sensor library for ESPx v1.17.0*, Display용 라이브러리는 *Adafruit SSD1306* 을 사용한다. 또한 USBLED와 제어를 위한 Relay를 사용한다.

동작

* 1초마다 온습도 데이터를 읽고 그 결과를 OLED에 display하기
* 어두워지는 이벤트가 발생하면 10초동안 USBLED에 불 켜기
* 계속 어두운 상태가 지속될 경우 영향을 주지 않음
* 어두운 상태에서 밝은 상태로 변할 경우 영향을 주지 않음
* 어두운 상태에서 밝은 상태로 바뀐 경우, 이후 다시 어두운 상태가 되면 USBLED에 10초간 불 켜기

출력 형태

|  |
| --- |
| T : 00.00 °C  H : 00.00 % |

핀 연결

|  |  |
| --- | --- |
| Device | Pin Number |
| LED | D0 |
| DHT22 | D3 |
| Relay | D4 |
| I2C. SCL | D1 |
| I2C. SDA | D2 |
| CDS | A0 |

1. **코드 설명**
2. Global scope

|  |
| --- |
| #include <Adafruit\_SSD1306.h>  #include "DHTesp.h"  #define D3 0  #define D0 16  #define D4 2  #define SCREEN\_WIDTH 128 // OLED display width, in pixels  #define SCREEN\_HEIGHT 32 // OLED display height, in pixels  #define OLED\_RESET     -1 // Reset pin # (or -1 if sharing Arduino reset pin)  #define SCREEN\_ADDRESS 0x3C ///< See datasheet for Address; 0x3D for 128x64, 0x3C for 128x32  #define LED\_PIN D0  #define RELAY1\_PIN D4 //active low!!  #define RELAY\_OFF HIGH  #define RELAY\_ON LOW  Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, OLED\_RESET);  DHTesp sensor;  int relay\_state = RELAY\_OFF;  int is\_bright = 0, is\_on = 0;  unsigned long previousMillis = 0, startMillis = 0;  const long interval = 1000;  const long light\_interval = 10000; |

OLED display를 다룰 때 사용할 라이브러리로 Adafruit\_SSD1306.h 를, 온습도 측정용 DHT 센서를 다룰 때 사용할 라이브러리로 DHTesp.h를 include해준다.

이후 사용할 때의 가독성 측면을 위하여 SCREEN\_WIDTH, SCREEN\_HEIGHT, OLED\_RESET, SCREEN\_ADDRESS 등을 define하며, nodeMCU 회로에서 연결해둔 pin 번호를 symbol로 사용하기 위하여 D0, D3, D4 또한 define해준다.

더하여 이후 전역적으로 사용하기 위한 display, sensor를 정의해주고 USBLED의 상태 제어에 사용할 변수로 relay\_state, is\_bright, is\_on를, 시간 측정을 위해 previousMllis, startMillis를, 측정할 시간 간격 상수로 interval, light\_interval 를 선언해 주었다.

1. setup

|  |
| --- |
| Serial.begin(9600);    sensor.setup(D3, DHTesp::DHT22); // Connect DHT sensor to GPIO 17    pinMode(LED\_PIN, OUTPUT); //set LED output    digitalWrite(LED\_PIN, LOW); //init LED    pinMode(A0, INPUT); //set light seneor    pinMode(RELAY1\_PIN, OUTPUT); //set relay signal    // SSD1306\_SWITCHCAPVCC = generate display voltage from 3.3V internally    if(!display.begin(SSD1306\_SWITCHCAPVCC, SCREEN\_ADDRESS)) {      Serial.println(F("SSD1306 allocation failed"));      for(;;); // Don't proceed, loop forever    }    display.setTextSize(1); // Draw 2X-scale text    display.setTextColor(SSD1306\_WHITE);    display.cp437(true);      // Show initial display buffer contents on the screen --    display.display(); |

에러 메시지 출력을 위한 Serial을 셋업 해준다.

온습도 data를 받아오기 위해 sensor에 pin 연결정보와 DHT 정보를 셋업 해준다.

OLED display 또한 미리 선언해둔 정보대로 셋업 해준다.

nodeMCU 회로와 연결될 input, output 또한 미리 정의해둔 정보를 이용해 셋업 하였다.

OLED에 display될 텍스트 정보를 몇 가지 추가로 세팅했다.

텍스트 사이즈는 1x로, 컬러는 흰색으로, 그리고 특수문자를 사용할 수 있도록 설정해 주었다.

마지막으로 display 테스트를 위해 로고가 출력되게 한 후 1초의 pause를 주었다.

1. loop

|  |
| --- |
| float hum = sensor.getHumidity();    float temp = sensor.getTemperature();    int light\_val = analogRead(A0);    unsigned long currentMillis = millis();    display.clearDisplay();    if( is\_on && (currentMillis - startMillis >= light\_interval)){      relay\_state = RELAY\_OFF;      Serial.println(F("USBLED turned off."));      is\_on = 0;    }      if(currentMillis - previousMillis >= interval){      previousMillis = currentMillis;      if(light\_val < 500) {        if(is\_bright){          relay\_state = RELAY\_ON;          is\_on = 1;          Serial.println(F("USBLED turned on."));          startMillis = millis();        }        is\_bright = 0;      }else if(light\_val > 600) {        // relay\_state = RELAY\_OFF;        is\_bright = 1;      }      if( !isnan(hum) && !isnan(temp)){          display.setCursor(10, 5);          display.print(F("T: "));          display.print(temp);          display.print(char(248)) ;          display.println("C");          display.display();          display.setCursor(10, 15);          display.print(F("H: "));          display.print(hum);          display.println("%");        }else  Serial.println(F("Cannot read information from sensor."));        display.display();      }    digitalWrite(RELAY1\_PIN, relay\_state);    digitalWrite(LED\_PIN, !relay\_state); |

Setup 실행 후 무한히 반복되는 loop함수의 내용이다.

우선 온습도 정보와 조도 정보, 현재 시간을 읽어온다.

현재 밝기와 이전 밝기를 비교하여 ‘밝았다가 어두워졌을 경우’의 케이스에 USBLED에 들어갈 신호를 ON으로 맞추어 주고 10초간 신호를 주기 위하여 startMillis를 기록한다.

is\_on 변수가 불이 켜져 있음을 표시할 경우, 매 loop마다 켜진 시점으로부터 10초가 경과되었는지 확인하여 불을 다시 꺼주는 state를 세팅한다.

온습도 값을 잘 읽어왔는지 isnan함수를 통해 확인 한 후, 문제가 없다면 읽어온 내용을 적절한 형식으로 세팅하여 display해준다.

Text가 표시될 위치를 지정해주기 위해 setCursor 함수를 이용했으며, 도씨(°C) 단위를 표기하기 위해 char(248) 기호를 사용해주었다.

그리고 이 과정들에 있어 1초씩 딜레이를 주기 위하여 millis함수를 이용한 if문으로 감싸주었다.

Loop의 마지막에는 state로 세팅 된 값을 digitalWrite를 사용하여 제어해준다.

1. **결과 및 느낀점**
2. 결과

1초 간격으로 업데이트되는 온 습도 정보가 잘 display되며, 어두운 상황에서 USBLED가 점등되고 10초 후 소등됨을 확인하였다. 소등 된 후에 어둡더라도 is\_bright를 통해 이전의 환경 밝기가 밝지 않았다면 아무런 액션이 일어나지 않음을 확인하였다. 즉, 문제에서 지시된 것과 같이 어두움-어두움과 어두움-밝음이 아닌 밝음-어두움 에서만 10초간의 전등을 구현하였다.

1. 배운 점 및 느낀 점

이전 과제에서부터Millis()함수를 이용하여 변수에 시간값을 저장하는 방식으로 1초간의 pause를 구현하였는데, 단순히 기존 변수들만을 이용하려 하니 USBLED가 점등 상태를 유지해야 하는 10초를 구현하기가 까다로웠다. 때문에 새로운 변수 startMillis를 두어 따로 시간을 기록하고 대조하도록 하였다. 배운 내용을 그대로 사용해보는 것을 넘어 조금 나아간 응용을 경험해볼 수 있었던 것 같다.

처음 과제 목표를 확인했을 때는 환경의 변화 상황에 따라 요구하는 동작이 다른 점이 lab에서 단순 현재의 밝기만을 트리거로 삼는 것 보다 복잡하고 불편하다 생각했는데, 교수님께서 현실에서의 현관 조명과 같은 장치를 모티프로 삼았다는 설명을 해 주셔 이해가 쉬었다. 오늘날 우리가 실제로 사용하는 기구들의 동작 원리를 직접 디자인하고 실습해볼 수 있다는 것이 정말 흥미롭고 재미있게 느껴졌다.