IOT시스템 설계 2022년 가을학기

Sensor Node 설계

프로젝트명: 체온 측정기

조 번호: 6조

이름: 이정언, 장선웅

학번: 21800564, 21700626

날짜: 2022/11/3

**목차**

1. **설계 목표 및 필요 기능(정언, 선웅)**

* **설계 목표**
* **필요 기능**

1. **기능 구현 및 실험 결과**

* **IOT device 구현(정언)**
* **IOT web server 구현(선웅)**
* **초음파 센서 동작(정언)**
* **비접촉식 온도 센서 동작(선웅)**
* **체온 측정 구현(정언)**

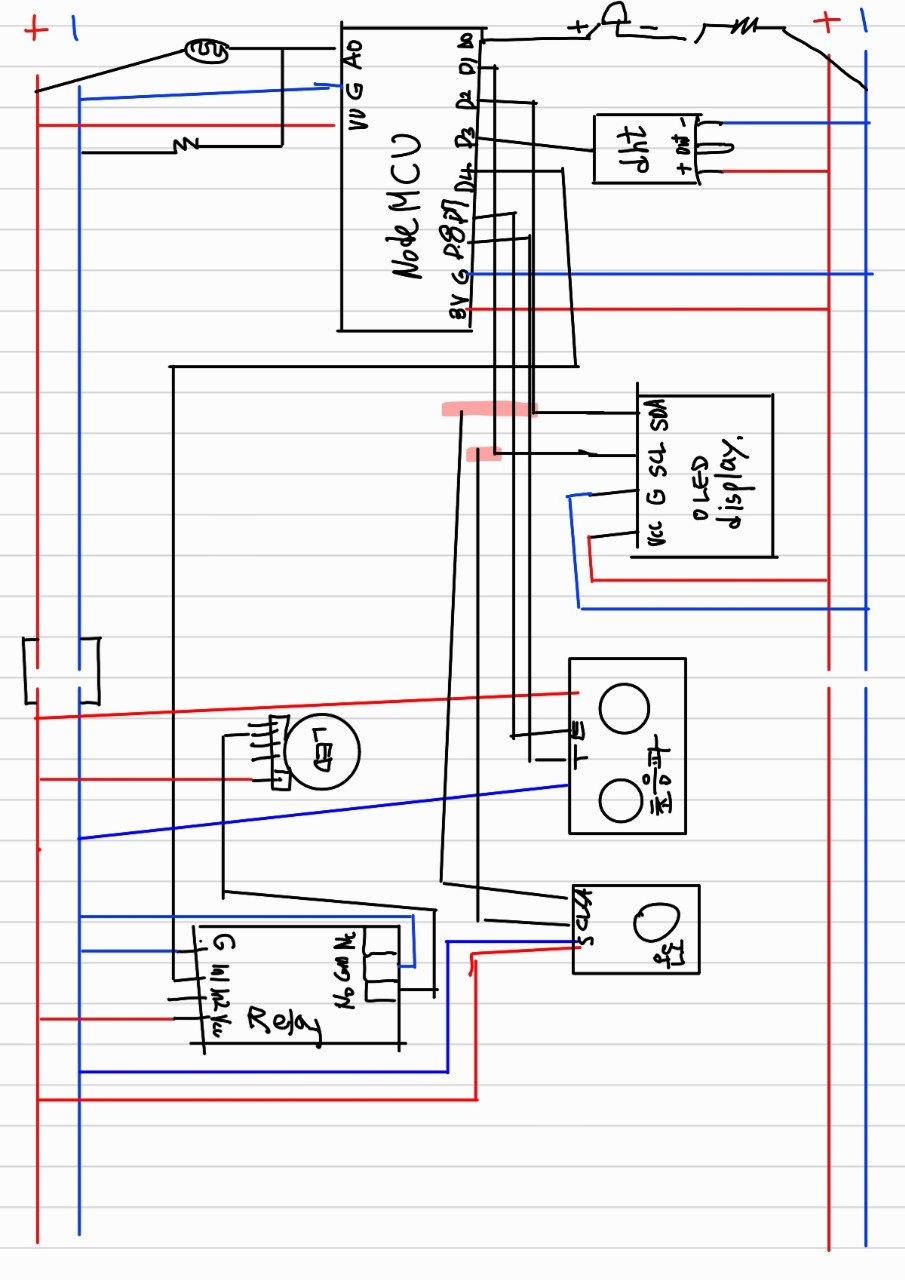
1. **설계 요약 및 정리(선웅)**
2. **역할 분배**
3. **참고 자료**
4. **설계 목표 및 필요 기능**

* **설계 목표**
  + 사용자가 체온 측정하기를 원할 때, 체온을 측정하고 이를 display로 출력하며 비정상적인 온도가 측정된 경우 잘못되었음을 출력하기
  + 각종 센서와 actuator들을 통해, 조도 값, 온도 및 습도 값, USBLED점등 등을 수행하는 iot device 만들기
  + IOT 웹 서버를 구현하고 MQTT를 통하여 iot device간의 통신 수행하기
* **필요 기능**
  + 사용자의 신체가 초음파 센서로부터 20cm이내에 들어오면 측정하고자 함을 감지하고, 5cm이내로 와야 측정할 수 있음을 출력하기
  + 5cm 이내인 경우, 온도센서를 동작하여 체온을 감지
  + 인간의 정상적인 신체 온도 35.9~37.5도로 측정된 경우, 정상적인 온도임을 알리고, 온도 값을 출력하기
  + 비정상적인 수치 값이 나온 경우, 비정상적인 온도임을 알리고, 다시 측정할 것을 알림과 동시에 온도 값을 출력하기
  + Non-blocking 형태로 coding
  + Hw2와 유사

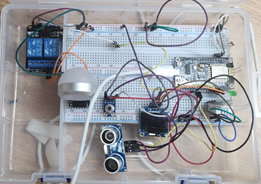
1. **기능 구현 및 실험 결과**

* **IOT device 구현**
  + **회로 구현**
    - nodeMCU를 동작시키기 위한 회로를 아래와 같이 구현한다.

초음파 센서의 Trig 핀과 echo핀은 각각 nodeMCU의 D8, D7에 연결하여 값을 인가하고, 비접촉식 온도센서의 SCL, SDA의 경우는 OLED display와 같은 라인에 핀을 연결하여, 최종적으로 각각 nodeMCU의 D1, D2에 연결한다.

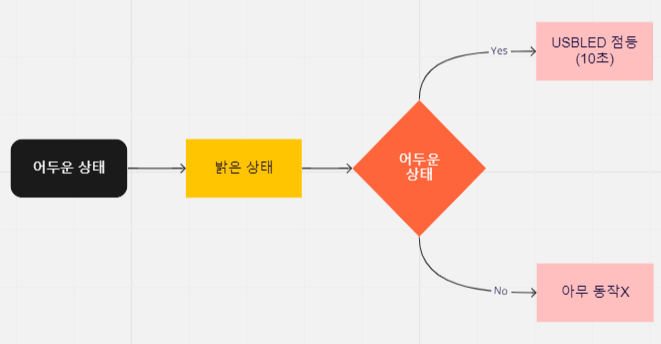


|  |  |
| --- | --- |
| **Ultrasonic wave 센서** | VCC à VU(5V)  Trig à D8  Echo à D7  Gnd à GND |
| **비접촉식 온도 센서** | SDA à D2  SCL à D1 |
| Cds | A0, VCC 연결 |
| Dht22 | + àVCC  Out à D3  - à GND |
| Led | + à D0  - à GND |
| Usbled | 1번째 핀 à VCC  4번째 핀 à relay NO |
| Relay | 전원 및 기기 연결 부분  NO à USBLED 4번째 핀  COM à GND  신호 부분  IN1 à D4 |
| OLED display | SDA à D2 SCL à D1 |

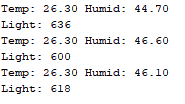


* + **기능 구현**
    - 구현한 기능들을 정리하면 다음과 같다. 조도 센서, dht22센서를 통해 조도 값, 온도 및 습도 값을 읽어오고, 이를 serial monitor에 출력한다.

USBLED의 경우는 다음과 같은 상황에서 점등을 하도록 구현하였다.

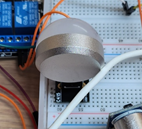
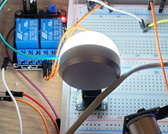


* + **실행 결과**
    - 온도 및 습도 값, 조도 값이 3초 간격으로 출력됨을 확인할 수 있다.

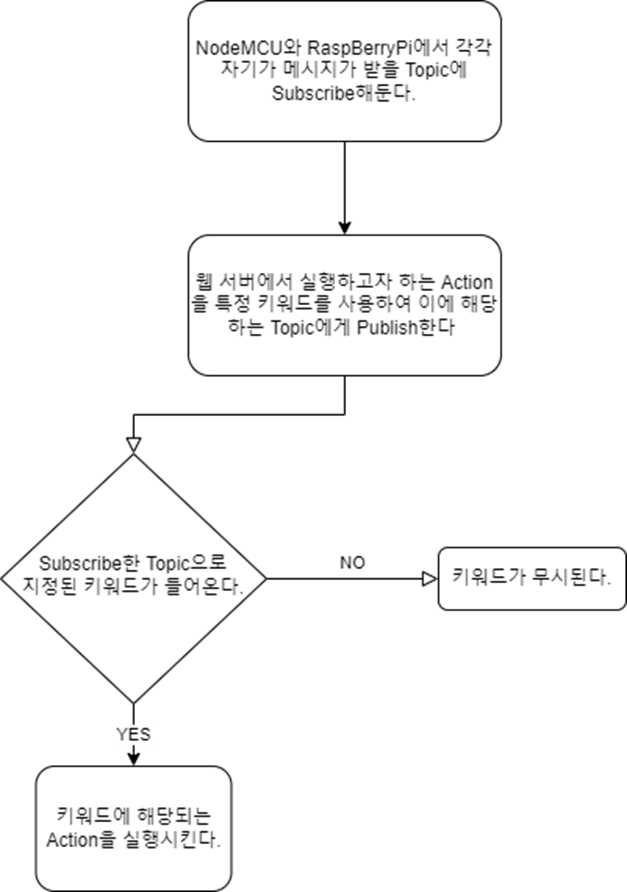


* + - 위의 로직의 경우를 만족하는 상황을 재현하여 Light intensity가 다음과 같이 변하도록 하였고, USBLED가 점등되는 것을 확인할 수 있었다.

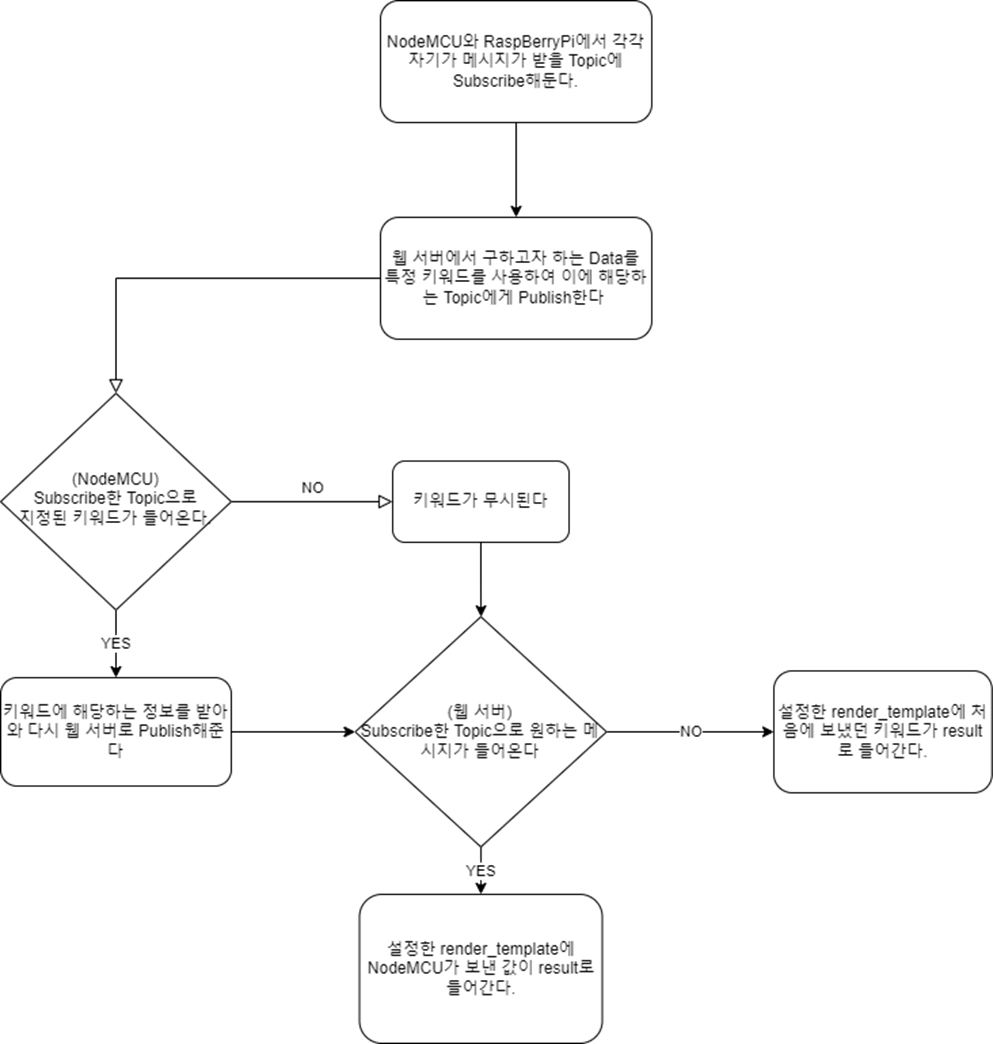
  

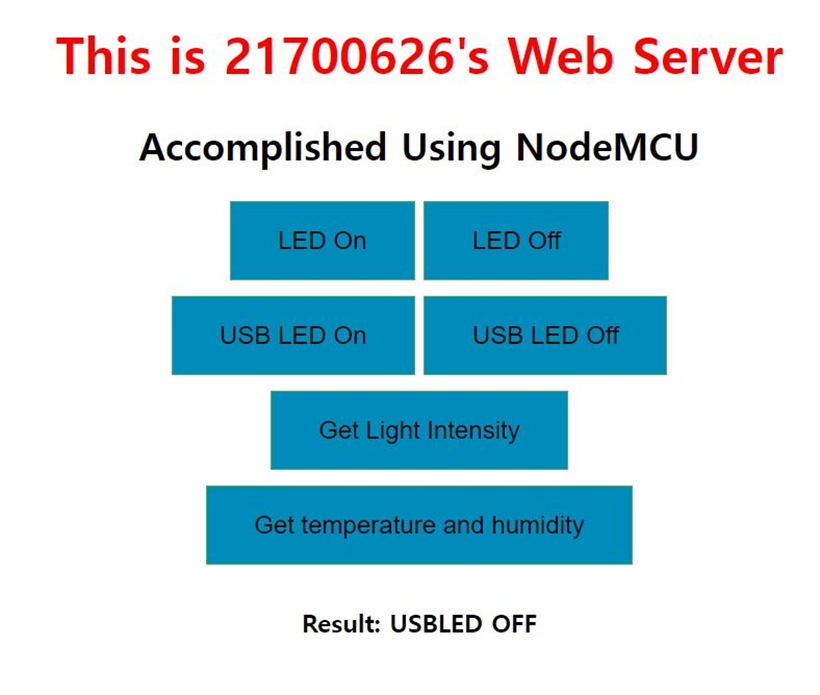
* **IOT 웹서버 구현**
  + **Flowchart**
    - Flask에서 NodeMCU 제어할 시 Process



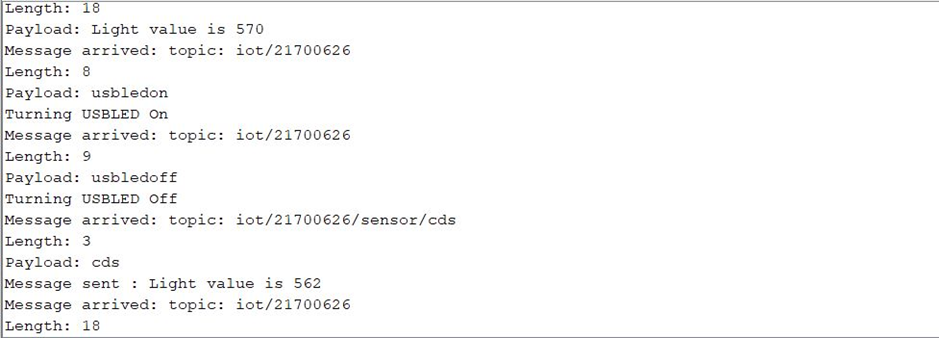
* + - NodeMCU에 특정 Data 요청받은 경우 Process



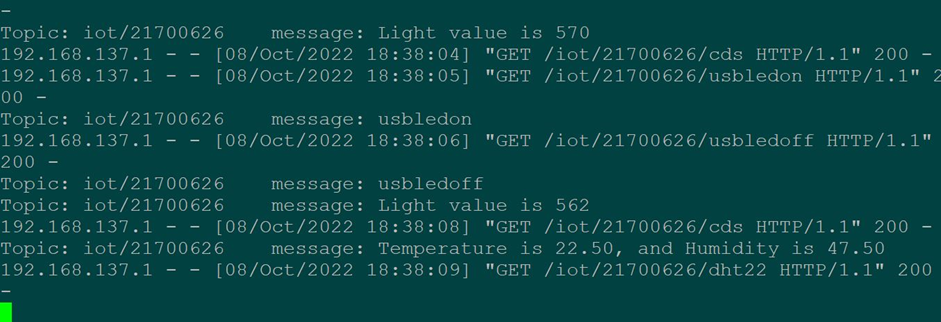
* + **웹서버 화면**
    - 웹서버의 기능은 2 카테고리로 나누어지는 데 하나는 Action을 요청하는 기능이고, 나머지는 Data를 요청하는 기능이다. LED On, Led Off, USB LED On, USB LED Off, Led Toggle 과 USB LED Toggle은 NodeMCU에게 특정 행동을 하도록 요청하고, 이 메세지가 성공적으로 전달 시, 요청이 수행된다. Get Light Intensity 와 Get temperature and humidity는 NodeMCU에 데이터를 요청하고, 이 요청이 성공적으로 전달된다면 필요한 데이터를 받게 된다.



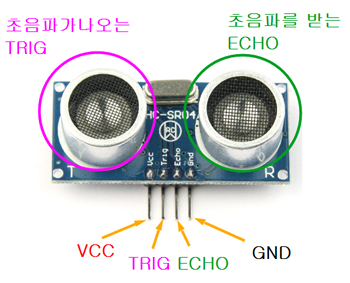
* + **실행 결과**
    - NodeMCU에서 Subscribe 된 topic으로 메시지를 받는 것을 확인할 수 있다.



* + - 웹서버에서 특정 URL을 치거나 버튼을 누르면, NodeMCU으로 요청을 Publish 하고, NodeMCU에서 Publish한 메시지를 Flask에서 받는 것을 확인할 수 있다. 정상 작동을 확인한 이후, topic을 iot/{조번호}로 변경하였고 정상 작동함을 다시 한번 확인할 수 있었다.



* **초음파센서(HC-SR04) 동작**
  + **필요지식 (동작 원리)**
    - 초음파 센서는 약 20KHz이상의 높은 주파수의 소리를 보낸 후, 반사되어 돌아오는 시간 차를 측정해서 거리를 알 수 있는 센서
    - 금속, 목재, 유리, 종이 등의 단단한 물체에는 거의 100% 반사되어 돌아오지만, 옷감과 같은 일부 물질은 초음파를 흡수하여서 정확한 측정 값을 구하기 어려운 단점이 있음
    - 핀맵은 좌측부터 VCC, TRIG, ECHO, GND로 구성
    - TRIG에서 초음파가 발생되고, 반사되어 돌아오는 초음파는 ECHO부분에서 받음
    - 측정 범위는 2cm~4M, 사용 전압: 5V

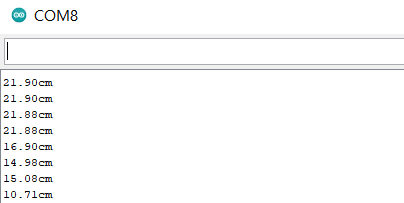


* + **동작 구현**

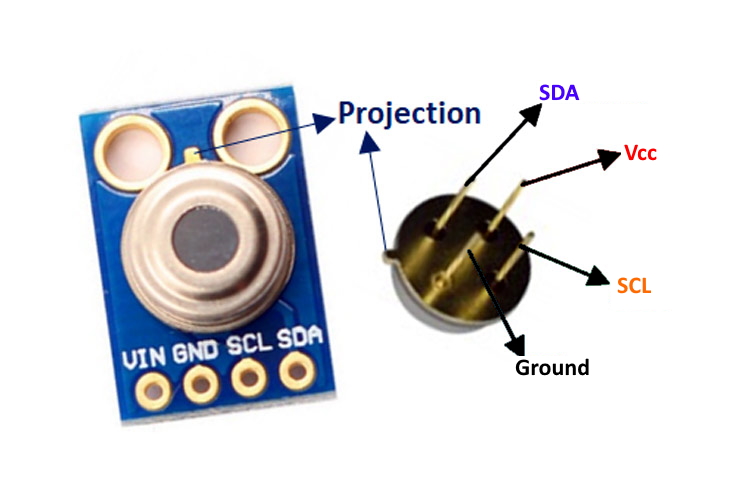
|  |  |
| --- | --- |
| VCC | NodeMCU의 VU 연결(5V) |
| TRIG | NodeMCU D8 연결 |
| ECHO | NodeMCU D8 연결 |
| GND | NodeMCU GND 연결 |

|  |  |
| --- | --- |
|  | TRIG와 ECHO를 각각nodeMCU의 D8, D7에 연결  초음파센서에서 물체까지의 거  리 값을 측정하는 사용자 정의함수 선언  Duration: 초음파가 발산 후, 반사되어 돌아오는 시간을 저장  Distance: 물체와의 거리 값 |
|  | TRIG의 경우, 초음파를 발산하는 부분이기 때문에 TRIG pin을 output으로 설정하여 신호를 출력하도록 함  ECHO의 경우, 반사되어 돌아오는 초음파를 받는 부분이므로 ECHO pin을 input으로 설정하여 신호를 받도록 함 |
|  | 거리를 측정하는 함수를 실행 후, 이를 시리얼 모니터에 출력 |
|  | trig에서 초음파를 발산하는 방법은 일단, TRIG 핀을 LOW로 초기화 시켜주고 HIGH으로 인가하여 초음파를 발생시킴, 이후에 다시 TRIG를 LOW로 하여 신호 출력을 중지함  echo의 경우, 초음파가 돌아오는 시간을 pulseIn함수를 사용하여 측정함  거리를 구하는 공식:  ((초음파속도340m/s\* duration)/10^4) / 2  10^4은 cm로 단위 변환을 위함  /2는 왕복시간이기 때문에 편도 시간만을 고려하기 위함 |
| pulseIn(ECHO, HIGH or LOW) | 아두이노 기본 함수로,  핀이 HIGH로 설정되면 타이머가 작동되고, 초음파를 반환 받을 시에 LOW로 바뀌며 타이머를 중단, 걸린 시간을 반환 |

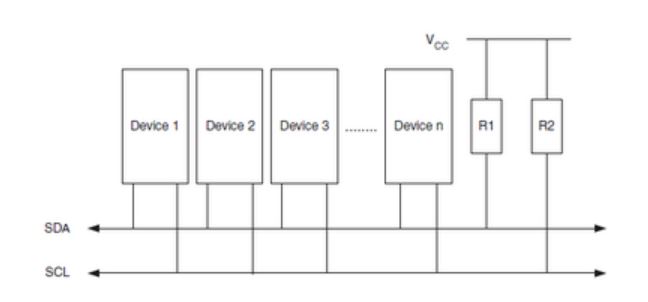
* + **결과**
    - 초음파로부터 물체까지의 거리가 정상적으로 실행됨을 확인할 수 있다.



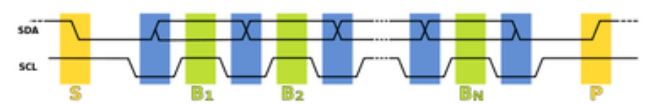
* **비 접촉 온도센서(MLX90614) 동작**
  + **필요지식 (동작 원리)**
    - 비 접촉 온도센서는 근거리 물체에서 방출되는 적외선을 통해 온도를 측정하는 센서
    - 대상 물체의 측정 범위는 약 -70도에서 380도 사이이고, 사용 환경의 측정 범위는 약 -40도에서 125도 사이이다. 적외선 카메라나 적외선 온도센서와 같은 원리로 동작된다. 17비트 ADC와 DSP 유닛을 탑재하여 정확한 측정이 가능하고, 0.1도 단위로 측정된다.
    - 핀맵은 좌측부터 VIN, GND, SCL, SDA로 구성
    - 사용 전압: 3V or 5V



* + **I2C 통신 방식**
    - I2C는 SDA와 SCL를 통해 통신이 되는데, SDA (Serial Data)는 데이터를 주고받기 위해 쓰이고 SCL (Serial Clock)은 데이터 송수신 타이밍 동기화를 위해 쓰인다. Master와 Slave 구조로 이루어지며 Master는 1개지만 Slave는 최대 127개까지 연결할 수 있다. 클럭을 사용하는 동기식 통신이어서 시간에 구속되지는 않지만 Slave가 여러 개 있을 수도 있어서 주소 데이터가 붙기 때문에 데이터가 길어진다. NodeMCU가 Master이고 모듈들(현재 구성에서는 OLED 디스플레이와 비 접촉 온도센서)이 Slave들이다. 주소 데이터를 쓰기 때문에 각기 다른 pin을 쓰는 것이 아니라 지정된 pin 하나에 다 연결한다. (SCL은 D1, SDA는 D2)



* + - I2C 통신은 시작, 데이터, 종료 신호로 이루어져서 각 Device간의 데이터 송수신에 혼란이 없도록 한다. 슬레이브의 주소 값은 시작 신호 이후 나오는 7 bit이며, 8번째 비트는 데이터를 쓰는 신호인지 읽는 신호인지 구분하는 bit이다. 한 클럭에 한 bit씩 데이터 신호를 만든다.



* + **동작 구현**

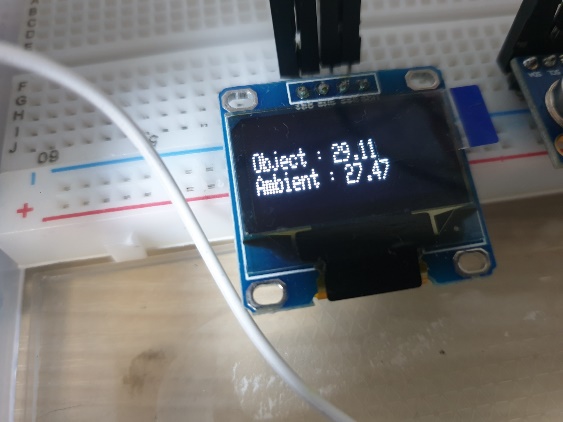
|  |  |
| --- | --- |
| VCC | NodeMCU의 3V 연결 |
| SDA | NodeMCU D2 연결 |
| SCL | NodeMCU D1 연결 |
| GND | NodeMCU GND 연결 |

* + - 비접촉식 적외선 온도센서를 사용하기 위해서 라이브러리를 등록해야 한다. 필요한 라이브러리는 adafruit-mlx90614로 다음의 사이트에 접속하여 라이브러리를 직접 다운받고, 아두이노의 libraries 파일 안에 압축을 풀어준다. <https://github.com/adafruit/Adafruit-MLX90614-Library>

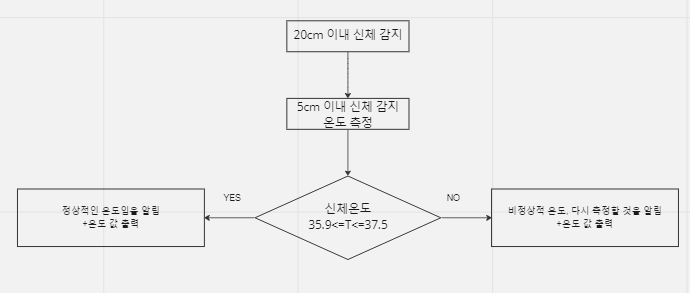
사용하는 함수로는 mlx.readObjectTempC와 mlx.readAmbientTempC가 있다. 여기서 중점적으로 사용할 함수는 전자로, 물체의 온도를 측정하여 반환해주는 함수이다.

|  |  |
| --- | --- |
|  | MLX90614와 OLED Display 사용을 위한 헤더 파일들을 include 시킨 후, 스크린 Dimension과 리셋 버튼, 그리고 I2C 주소를 정의한다. MLX90614 라이브러리를 생성하고 SSD1306에 대하여 선언을 함. |
|  | 온도 센서를 시작하고 OLED 디스플레이 또한 시작한다. 만약 디스플레이 시작 시 오류가 발생한다면 오류 메세지 출력 후 무한 루프 발생. |
|  | 0.5초 마다 displaytext 함수를 실행하여 온도 값을 섭씨로 출력. |
|  | Object 온도와 Ambient 온도를 parameter로 받아온 후, display를 매번 clear 해주며 (아니면 화면이 더러워짐), 문자 크기, 위치, 색을 지정한 후, parameter로 받아온 온도 2가지를 OLED로 출력한다. |

* + **결과**
    - 비 접촉 온도센서에서 물체와 주변 온도를 측정하고 OLED display로 정상 출력하는 것을 볼 수 있다.



* **체온 측정 구현**
  + **기능 구현**
    - 정상적으로 작동하는 초음파 센서와 비접촉식 적외선 온도센서를 이용하여 필요 기능들을 구현하고자 한다. 기능에 대한 flow chart는 다음과 같다.



1. 사용자의 신체가 초음파 센서로부터 20cm이내에 들어오면 측정하고자 함을 감지하고, 5cm이내로 와야 측정할 수 있음을 출력하기
2. 5cm 이내인 경우, 온도센서를 동작하여 체온을 감지
3. 인간의 정상적인 신체 온도 35.9~37.5도로 측정된 경우, 정상적인 온도임을 알리고, 온도 값을 출력하기
4. 비정상적인 수치 값이 나온 경우, 비정상적인 온도임을 알리고, 다시 측정할 것을 알림과 동시에 온도 값을 출력하기
5. Non-blocking 형태로 coding
   * + 이어서, 구현한 코드의 내용과 설명은 다음 표와 같다.

|  |
| --- |
|  |
| 필요한 헤더파일 포함 |
|  |
| 선언문에서 추가된 부분은 DELAY 1000으로,  Display를 출력할 때마다 1초간 출력하기 위해 선언 |
|  |
| 함수 및 전역 변수 선언  Measure\_dist는 초음파센서와 물체와의 거리 값을 가져옴  Displaytext는 출력하고자 하는 문자열을 매개변수로 전달받아 출력함  displayTemp는 온도 값을 delay를 주어 출력해주는 함수  Measure\_temp는 구현된기능에 맞춰 온도를 측정하고, displaytext함수를 사용하여 출력함  cur\_mils\_dist와 cur\_mils\_display는 non\_blocking형태로 거리 값과 출력하는 것에 delay를 주기 위해 사용  flag는 displaytext함수가 출력을 1초동안 하도록 제어하기 위해 사용 |
|  |
| Display를 화면을 초기에 지우기 위한 clearDisplay, display함수를 사용이 추가됨 |
|  |
| 거리를 계속해서 측정  온도는 거리 값에 따라 함수 내에서 측정 |
|  |
| 1초마다 거리 값을 측정하고, 시리얼 모니터로 출력하기 위해 내용이 추가됨  Millis()는 프로그램이 실행된 이후부터 시간을 계속해서 측정하는 데, cur\_mils\_dist(e.g. 0초)+DELAY(1초) 값보다 millis()현재 실행된 시간이 크거나 같게 되면, 1초가 지난 것이기 때문에 cur\_mils\_dist 변수에 DELAY만큼 값을 더해주고, 온도를 측정함 |
|  |
| 거리 값이 20cm이내 인 경우,  5cm이내로 온 경우 온도를 측정하는 데, 손바닥을 온도 센서에 갖다 댄 경우에 측정되는 온도가 계속해서 비정상적인 온도가 측정되어 4도를 더하였음(실제 체온계와 오차 값을 계산하여 추정하는 것도 좋은 방법)  정상적 체온이 측정된 경우, 정상적 체온이라는 알림과 함께 온도를 출력  그렇지 않은 경우, 비정상적인 온도가 측정되었고, 다시 측정할 것을 알림과 동시에 온도도 출력  5cm초과인 경우는 측정을 원한다면 5cm이내로 올 것을 알림  거리가 20cm초과인 경우, display를 지워 둠 |
|  |
| 출력하고자 하는 문자열을 str매개변수로 받아서 출력함 |
|  |
| String을 매개변수 str로 받아서 출력을 실행하는 데, 1초마다 출력을 하기 위해 flag 변수를 새로 선언하여 사용  Flag는 선언과 동시에 false로 선언을 해두었고, false인 경우에 현재 프로그램 실행 시간을 cur\_mils\_display에 받아오고, flag를 true로 바꾸어 loop으로 인해 cur\_mils\_display변수가 계속해서 millis값을 받아오지 못하게 함  Millis()값이 cur\_mils\_display(display함수가 실행초기화된 시점에 측정한 시간값)와 DELAY(1초)값의 합의 이상인 경우, 1초가 지났음을 의미하여 str내용을 출력하도록 구현 |

* + **결과**

|  |  |
| --- | --- |
|  | 거리가 5cm이내인 경우, 온도 값이 측정되어 출력됨을 확인할 수 있다. |
|  | 거리가 20cm 초과인 경우, OLED display에 아무것도 출력되지 않음을 확인할 수 있다. |
|  | 초음파 센서로부터 거리가 20cm 이내인 경우, 온도 측정을 원하면 5cm 이내로 오라는 문구가 출력됨을 확인할 수 있다. |
|  | 5cm 이내인 경우, 정상적인 체온이 측정되면 정상 온도라는 문구와 함께 온도가 출력됨을 확인할 수 있다. |
|  | 5cm 이내인 경우, 비정상적인 체온이 측정되면 비정상 온도, 다시 측정하라는 문구와 함께 온도가 출력됨을 확인할 수 있다. |

1. **설계 요약 및 정리**

* 비 접촉 온도 센서는 적외선을 이용하여 물체와 주변 온도 값을 읽어오고, 이 값들을 128x64 I2C OLED 디스플레이에 정상적으로 출력하였다. 이 작업을 수행하기 위하여 하드웨어 상으론 OLED Display의 SCL과 SDA단자들을 각각 D2, D1에 연결하였고, 비 접촉식 온도 센서의 SCL, SDA 단자들 또한 같은 핀에 연결하였다. I2C 통신의 특성상 같은 핀에 연결하여도 Master-Slave 관계로 서로 방해를 하지 않고 주소 데이터를 사용하여 같이 정상적으로 구동 되었다. 소프트웨어 상에서는 먼저 MLX90614 와 SSD1306의 라이브러리를 같이 가지고 와 (둘 다 Adafruit사의 제품) 선언하고, setup()에서 두 센서들을 시작한다. 그리고, displaytext() 함수를 만들고, 이 함수 안에서 디스플레이 초기화와 메시지 출력을 0.5초 마다 반복했었다. 물체 온도를 가져오기 위해선 readObjectTempC 함수를 사용하였고, 주변 온도는 readAmbientTempC를 사용했다. 디스플레이를 초기화 하는데는 clearDisplay, 출력하기 위해서는 print와 display를 사용하였다.

초음파 센서의 경우에는 물체를 향해 초음파를 쏜 후 다시 돌아오는 시간을 측정하여 거리를 파악할 수 있었다. 이 작업을 수행하기 위해서 하드웨어 상으론 Trig와 Echo가 각각 D8, D7 핀에 연결되었는데, Trig에서는 초음파를 발생시켰고, Echo에서는 돌아오는 초음파를 받는 역할을 수행하였다. 소프트웨어 상으로는 먼저 trig와 echo의 핀을 지정해주고 trig는 OUTPUT, 그리고 echo는 INPUT으로 지정해주었다. 초음파가 돌아오는 시간을 저장할 때는 duration 변수를, 물체와의 거리를 저장하는 데에는 distance 변수를 사용하였다. 1초마다 Measure\_dist 함수를 실행하였고, 함수 안에선 trig를 먼저 LOW로 초기화 시키고 HIGH로 인가하며 초음파를 발생시킨 후 LOW로 바꿔 멈추는 형식으로 trig핀을 사용하였고, echo는 pulseIn 함수를 사용하여 초음파가 돌아오는 시간을 계산하였다. pulseIn 함수는 핀이 HIGH로 된 후 LOW로 바뀔 때 까지의 시간을 측정해 반환하는 함수이다.

위 두 센서들은 동작 방식이 전혀 다르지만 비 접촉/근거리에서 사용 가능하다는 공통점이 있다. 우리는 위 두 센서들을 비 접촉 온도계와 같이 사용하되, 온도계를 더 편하게 사용할 수 있는 기능을 더 추가한다. 초음파 센서로 사용자의 신체가 20cm이내에 들어오는 것을 감지하고, 5cm 안에 들어오는 경우엔 온도를 감지한다. 만약 비정상 온도가 감지된다면 온도 값과 함께 알림이 뜬다. 이를 하드웨어적으로 구현하기 위해서 먼저 앞서 사용한 두 코드들을 합치고 이에 추가적인 부분을 더했다. 먼저 함수들을 4개 구현했는데 measure\_dist에서는 초음파센서로 거리를 측정하고, displaytext에서는 문자열을 parameter로 받아 출력한다. 그리고, displayTemp에서는 온도를 일정 간격으로 출력해주고, measure\_temp에서는 온도를 측정한다. 코드가 non\_blocking으로 진행하기 위해 cur\_mils\_dist와 cur\_mils\_display 변수들을 사용하였다.

Setup은 전에 나온 코드들과 같고, loop에선 measure\_dist와 measure\_temp가 연속적으로 실행된다. Measure\_dist에서는 millis()를 사용해 연속적으로 1초를 non-blocking방식으로 계산한다. Trig는 LOW-HIGH-LOW 방식으로 진행되고 duration과 distance도 이전 방식과 동일하게 계산된다. Measure\_temp에서는 distance를 우선적으로 20cm 이하인지 확인하고, 5cm 이내이면 체온을 측정 후, 정상 범위안에 있는지 확인한다. displayTemp함수 에서는 1초마다 문자열을 출력하는데, 이를 위해 flag를 사용한다. Millis()를 이용해 display함수가 실행 초기화된 시점과 비교해 1초를 계산한 후 flag를 조정한다. 이러한 과정들을 통해 체온 측정기를 정상적으로 구현할 수 있었다. 하지만, 비접촉식 온도 센서로 측정되는 신체 온도 값이 정확하게 측정되지 않아 측정된 값에 적정 값을 더하여 출력하였다는 한계점이 존재한다.

1. **역할 배분**

* **이정언**
  + - 서론의 설계 목표 및 필요기능, 본론의 IOT device 구현, 초음파 센서 동작 구현, 초음파 센서와 비접촉식 온도 센서를 통해 체온 측정기를 구현을 하는 것을 담당하였다.
    - 설계 목표 및 필요 기능은 팀원과의 논의로 단순히 온도를 측정하는 것이 아닌, 사용자의 측정여부를 판단하고 이에 따라 안내를 하는 방식으로 정의하였다.
    - Iot device를 구현하는 파트에서는 기존의 구현된 회로에 초음파 센서와 온도센서를 결합하는 과정을 명시하였다. 회로를 구상하고, 구현하였으며 이전에 진행하였던 각종 센서들을 어떤 논리로 구현했고, 결과는 어떠한지 간략하게 명시하였다.
    - 초음파 센서의 동작 구현 파트에서는 센서의 동작 원리를 조사 및 정리하고, 이를 숙지한 다음에 동작을 위한 코드를 구현하였다.
    - 두 센서를 결합하여 체온을 측정하는 파트에서는 사전에 정의하였던 필요 기능들의 flow chart를 만들어 시각적으로 표현하였고, 이에 대한 설명을 기술하였다. 기능 구현을 위해 두 센서에 대한 코드들을 결합, 수정, 새로 정의하는 등의 작업을 하였고, 이에 대한 설명을 기술하였다.
    - 이정언에 의해 작성된 Iot device관련 코드, 두 센서를 추가 결합하여 만든 체온 측정 아두이노 코드와 장선웅에 의해 작성된 web관련 코드 3가지를 합하여 하나의 아두이노 코드로 동작하도록 흐름을 수정하였다.
    - 마지막으로, 전체적인 보고서의 틀과 흐름, 가독성 체크 등을 하였고, 제출 방식대로 디렉토리를 구성하여 제출하였다.
* **장선웅**
  + - 전체적인 설계 요약과 정리(결론), 본론의 IOT 웹서버 구현, 비접촉 온도 센서 동작 구현을 담당하였다.
    - Web관련된 코드를 작성하였다. (Python코드와 Web server에 사용되는 template)
    - 비접촉 온도센서 동작 구현 파트에서는 센서의 동작 원리를 조사 및 정리하고, 이를 숙지한 후 동작 시키기 위한 코드를 작성하였다. (I2C에 대해서도 다시금 조사하였다.)
    - 웹서버 구현하는 파트에서는 Flow Chart로 연결 방식을 설명하였고. 동작 과정과 결과를 간략히 설명하였다.

1. **참고 자료**

* **초음파 센서**

[**https://m.blog.naver.com/boilmint7/220926404472**](https://m.blog.naver.com/boilmint7/220926404472)

[**https://blog.naver.com/PostView.naver?blogId=lline\_edu&logNo=222219099428&categoryNo=1&parentCategoryNo=0**](https://blog.naver.com/PostView.naver?blogId=lline_edu&logNo=222219099428&categoryNo=1&parentCategoryNo=0)

[**https://www.arduino.cc/reference/en/language/functions/advanced-io/pulsein/**](https://www.arduino.cc/reference/en/language/functions/advanced-io/pulsein/)

* **비 접촉 온도 센서**

[**http://vctec.co.kr/product/%EC%A0%81%EC%99%B8%EC%84%A0-%EC%98%A8%EB%8F%84%EC%B8%A1%EC%A0%95-%EC%84%BC%EC%84%9C-mlx90614-infrared-thermometer-breakout-mlx90614/6765/**](http://vctec.co.kr/product/%EC%A0%81%EC%99%B8%EC%84%A0-%EC%98%A8%EB%8F%84%EC%B8%A1%EC%A0%95-%EC%84%BC%EC%84%9C-mlx90614-infrared-thermometer-breakout-mlx90614/6765/)

[**https://eduino.kr/product/detail.html?product\_no=274&gclid=Cj0KCQjw--2aBhD5ARIsALiRlwDvszIgDLm5VpFGxIdHRQjXauwfa2TmaPI3AkZkgBZZ6XY9Q774DToaAqzZEALw\_wcB**](https://eduino.kr/product/detail.html?product_no=274&gclid=Cj0KCQjw--2aBhD5ARIsALiRlwDvszIgDLm5VpFGxIdHRQjXauwfa2TmaPI3AkZkgBZZ6XY9Q774DToaAqzZEALw_wcB)

[**https://blog.naver.com/yuyyulee/220323559541**](https://blog.naver.com/yuyyulee/220323559541)