**“유니스트 적정기술 프로젝트 – STM32 – Vibration”**

**[main.c]**

**코드는 크게 3 부분으로 나뉜다. 1) wifi를 받아오고 연결하는 부분 2) 서버 정보를 받아오고 데이터를 보내는 부분 3) 데이터를 받는 부분이다.**

**Wifi를 받아오고 연결하는 부분은 5,6의 코드 설명에 자세히 적어 놓았으며 서버 정보를 받아오는 부분은 9, 11, 12코드, 데이터를 보내는 부분은 5의 마지막 198-217 코드 부분이다. 8, 13 부분은 데이터를 보정하고 형식에 맞게 바꾸어 주는 부분이다.**

**1. 변수 선언 및 정보 추출**



Line 73-77 : 와이파이 정보 추출

**2. 와이파이 연결 및 데이터 전송**



Line 86-91 : 와이파이 연결

Line 93-99 : 센서 데이터를 받아오고 전송

**3. 변수 선언**



센서 데이터를 정리하기 위한 코드이다.

**4. 함수 선언**



센서 데이터를 단위로 쪼개기 위한 함수를 선언한다.

**5. wifi 연결을 위한 함수 선언**





코드를 살펴보게 되면 State 2가 3부터 한 단계가 넘어갈 때 마다 1증가하고, 다음 단계를 실행하기 위해서는 1만큼 더 증가해야 하는데 위의 코드에는 없다. 이는 데이터를 보내고 응답을 받아와야 다음 단계로 넘어가기 위해, 응답을 받아오고 비교한 뒤 추가로 1을 증가시켜 다음 단계로 넘어가게 설계하였다.

이러한 코드를 하게 된 경위는 처음 wifi를 연결해주는 모듈인 **ESP8266의 초기 세팅**을 해주기 위함이다. 일반적으로는 컴퓨터에 연결하여 터미널을 통해 명령을 주지만 이 과정을 STM을 통해서 하기 위한 코드이다. 150번째 줄 코드에 있는 WIFITEST는 와이파이 모듈이 정상인지 대답을 받기 위해 AT 명령어 중 AT를 보내는 것이고 159번째 줄 RESETWIFI는 AT+RST를, CONNECTWIFI와 CONNECTPC는 와이파이 이름과 비밀번호, connectwifibuffer와 datasendsizenum은 와이파이의 글자 수 정보를 보내는 것이다. 그렇기 때문에 변수로 지정되었다. 글자 수 정보는 모듈 ESP8266에서 데이터를 보낼 때 그 숫자만큼의 데이터만큼만 보내줄 수 있기 때문에 미리 지정을 해주어야 한다. 이를 하지 않으면 보내는 중간에 멈추거나 정보가 누락되어 전송된다. 그 이후의 DATASIZE는 보내는 측정 데이터의 사이즈를 나타내고 이후의 DATASEND를 통해 측정 데이터가 전송된다. Datasizenum은 위의 connectwifibuffer와 datasendsizenum과 같은 이유로 후의 코드에서 결정된다.

마지막의 219번째 줄을 확인하면 State가 있는데, 이는 데이터를 받아서 저장하기 위한 주기로 이를 다시 0으로 돌리어 데이터를 받아오는 주기를 진행한다. 위와 같은 한 주기를 돌게 되면 다시 state2를 0으로 보냄으로써 다시 이 주기를 진행할 수 있게 된다.

**6. 응답 추출 및 비교**



 

State2의 짝수 상태에 대한 코드로 240, 259, 278, 299, 326, 348 번째 줄을 통해 받아오는 대답을 lookup table의 상태와 비교하여 정상 상태일 때 State 2를 다음 주기로 넘어가게 설계하였다. 각각의 if 구문에서 for 구문의 숫자는 대략 큰 숫자를 설정한 상태이므로 조절할 수 있다.

**7. 3축 가속도 센서 활성화**



**8. 3축 가속도 센서 데이터 값 보정**



보정 값은 다음 값을 참고하였다.

<https://www.youtube.com/watch?v=l29xBCKZYi8&ab_channel=MutexEmbedded-Education>

**9. 휴대폰으로부터 와이파이 데이터를 전송**





휴대폰의 어플리케이션에서 SSID, PSWD, ADRS, TAG라는 값을 각각의 와이파이 이름, 비밀번호, 서버 ip 주소, 게이트웨이의 앞에 붙여 놓았다. 따라서 각각의 값 앞에 오는 태그를 읽음으로써 뒤의 값이 사용자가 원하는 값을 알 수 있고 이를 저장한다.

**10. 에러 발생 시 복구**



Countdebug 값을 초당 하나씩 늘려 15초가 지나면 자동으로 주기의 처음으로 돌아간다.

**11. 데이터 글자 수 추출**



위의 설명에서 말했듯이 ESP8266 모듈을 사용할 경우 글자 수를 알아야한다. 그렇기 때문에 각각의 데이터 정보에 대한 글자 수를 세는 코드이다.

**12. 와이파이 데이터 정제**





와이파이 데이터를 맞는 포맷으로 변형시켜 정제하는 과정이다. 또한 ESP8266은 끝에 /r/n을 입력해야 코드의 끝으로 인식한다.

**13. 숫자 10단위로 분리하는 함수**



데이터 전송을 원활하게 하기 위해서 숫자 자릿수를 분리한 코드이다.