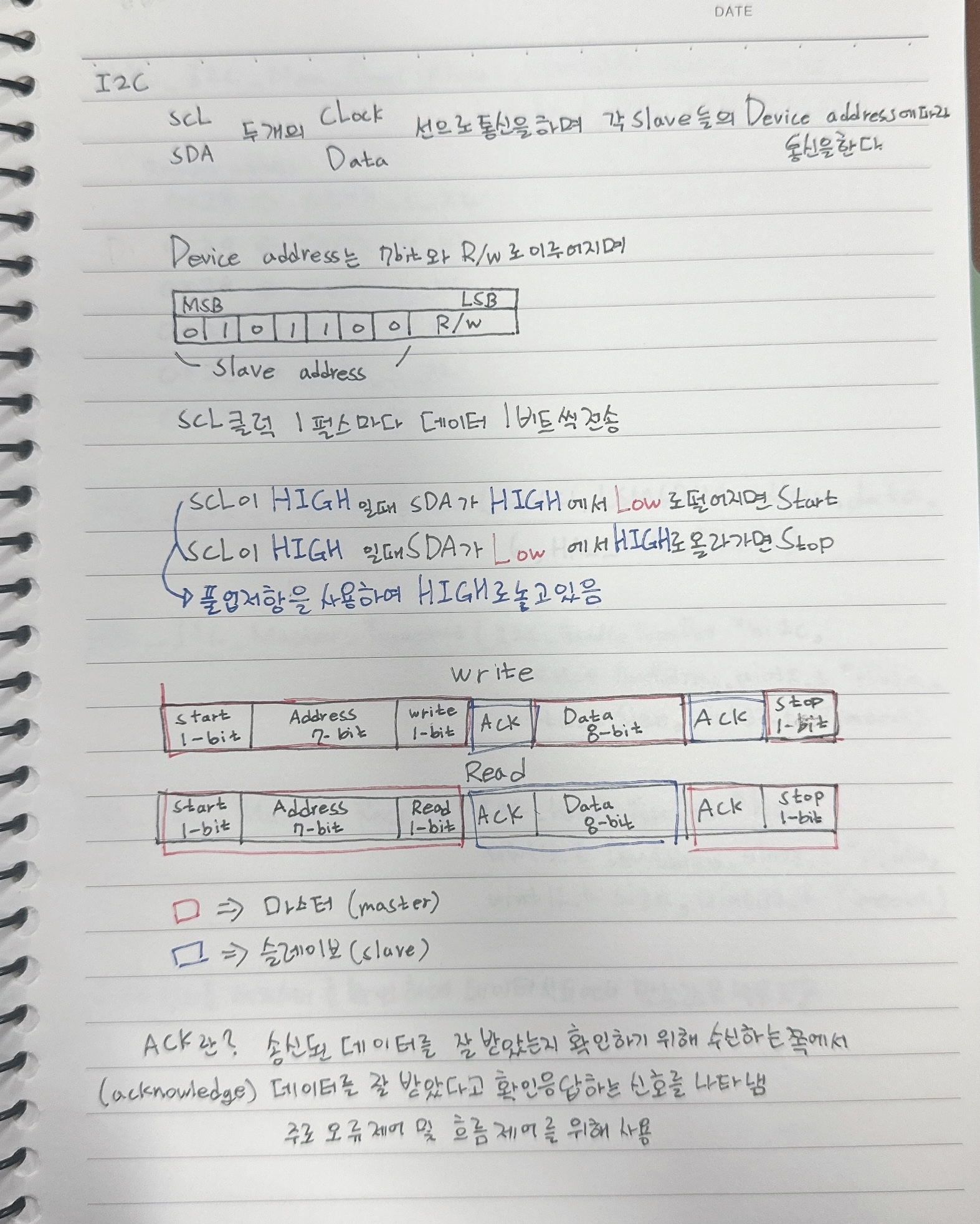
LSM6DSM 센서값 읽기

I2C통신 이해및 적용

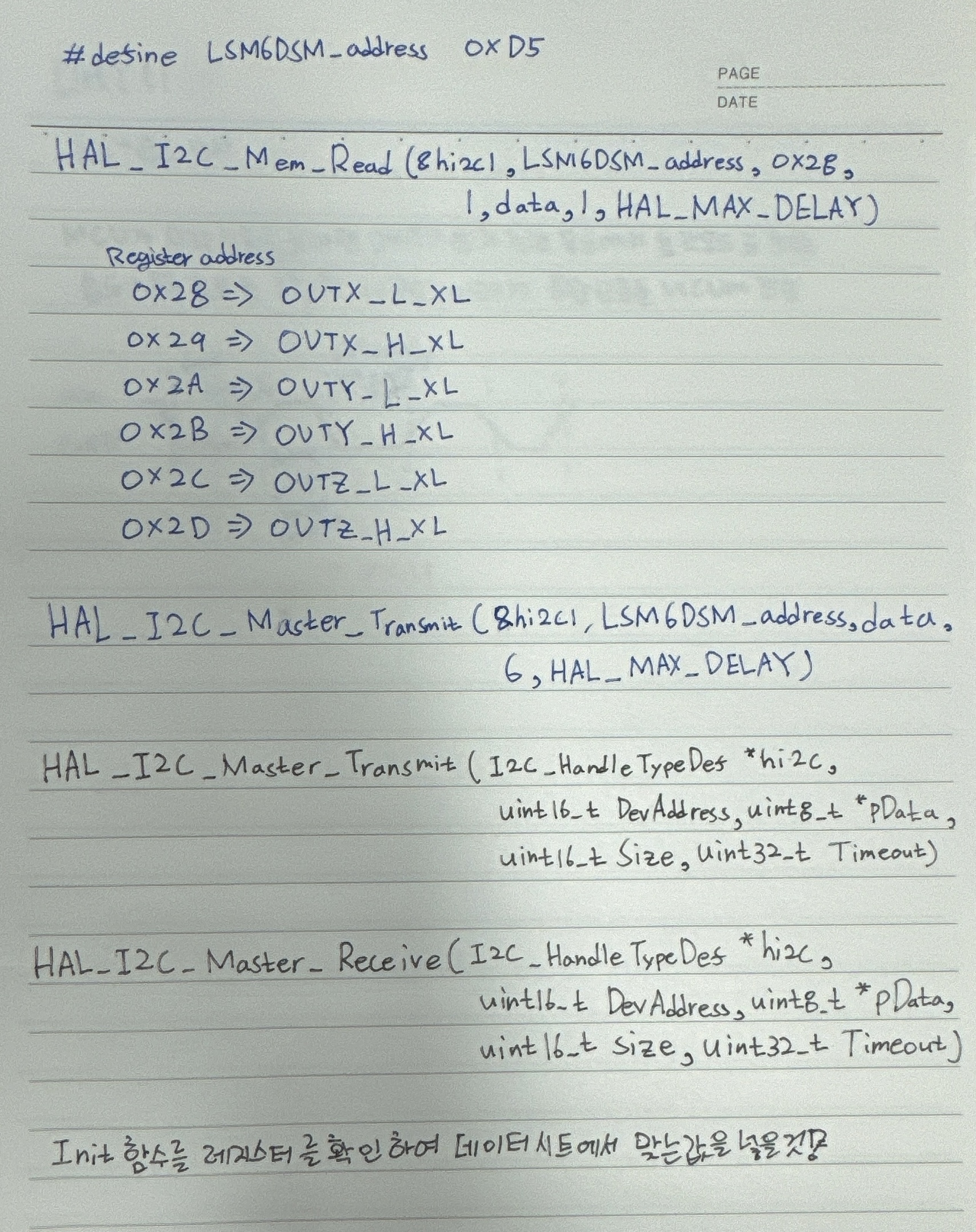
▼

LSM6DSM 센서값 읽기

▼

가속도 센서의 값이 0만 나옴

▼

STM HAL함수 어떠한 함수를 사용해야 되는지 이해 및 적용

▼

그래도 0만 나옴, 문제점을 이해하지 못함

(아래와 같이 사용)

uint8\_t regbuf[6] = {0x10, 0x50};

HAL\_I2C\_Master\_Transmit(&hi2c1,LSM6DSM\_address,regbuf,2,HAL\_MAX\_DELAY);

가속도 값이 나옴

▼

가속도 값으로 기울기를 구함

▼

기울기 값이 이상하게 나와서 데이터 시트를 보고 설정한 scale값 파악

**float** accel\_scale\_factor = 2.0f;

acceleration\_g[0] = acceleration[0] / (accel\_scale\_factor \* 1000.0f);

acceleration\_g[1] = acceleration[1] / (accel\_scale\_factor \* 1000.0f);

acceleration\_g[2] = acceleration[2] / (accel\_scale\_factor \* 1000.0f);

▼

9.8과 유사한 값이 나옴

▼

각축의 기울기 각도를 구함

roll = **atan2**(-acceleration\_g[1], acceleration\_g[2]) \* 180.0 / M\_PI;

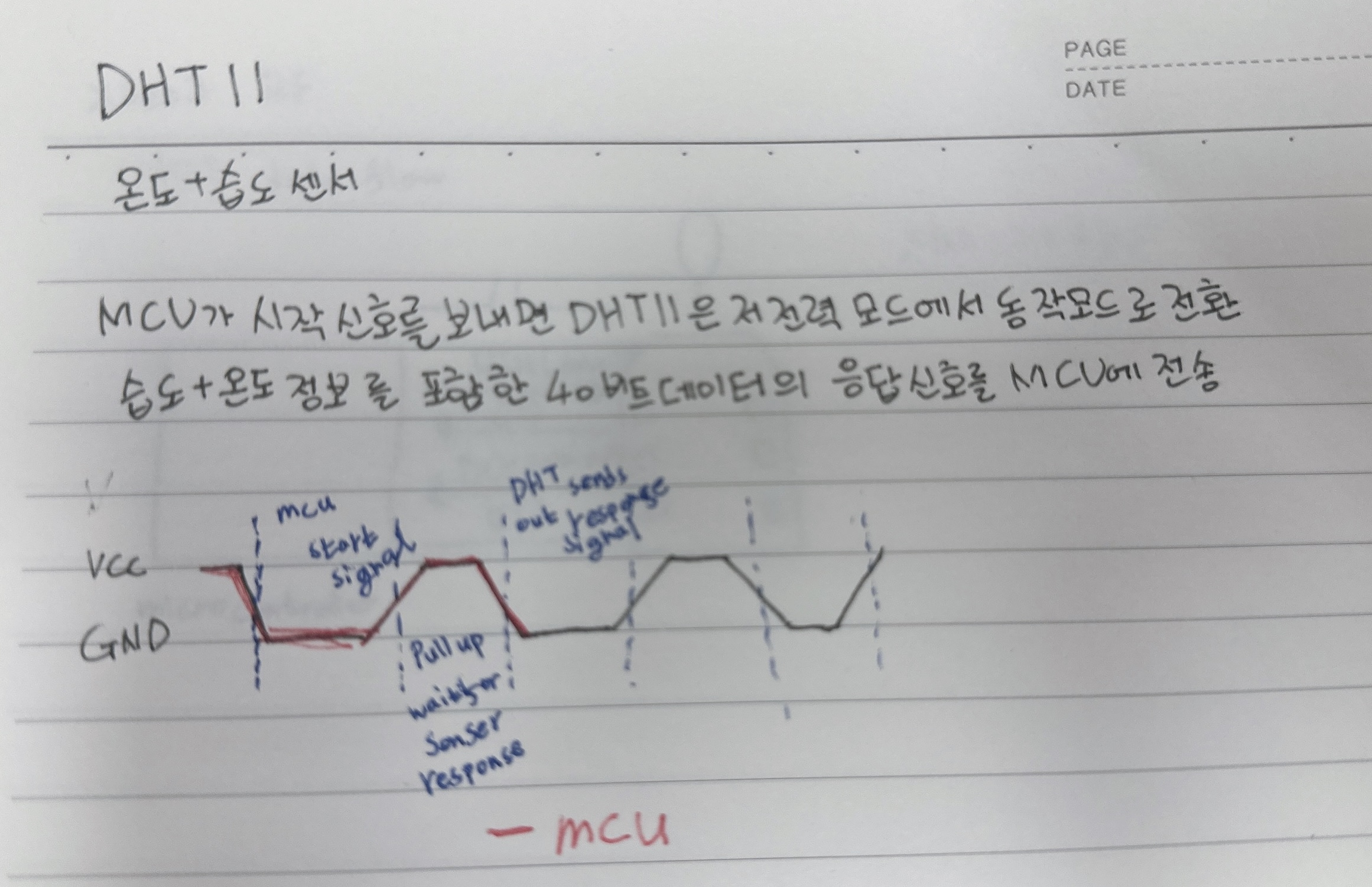
pitch = **atan2**(acceleration\_g[0], **sqrt**(acceleration\_g[1] \* acceleration\_g[1] + acceleration\_g[2] \* acceleration\_g[2])) \* 180.0 / M\_PI;

▼

x,y축의 기울기 각도에 따라 LED변화

DHT11 온도,습도값 읽기

DHT11 데이터시트 이해및 파악

 ▼

코드 적용 하였을때 값이 읽히지 않음

**void DHT11\_Start**(**void**)

{

Set\_Pin\_Output (DHT11\_PORT, DHT11\_PIN); // set the pin as output

HAL\_GPIO\_WritePin (DHT11\_PORT, DHT11\_PIN, 0); // pull the pin low

delay\_us(18000); // wait for 18ms

HAL\_GPIO\_WritePin (DHT11\_PORT, DHT11\_PIN, 1); // pull the pin high

delay\_us(20); // wait for 20us

Set\_Pin\_Input(DHT11\_PORT, DHT11\_PIN); // set as input

}

uint8\_t **DHT11\_Check\_Response**(**void**)

{

uint8\_tResponse = 0;

delay\_us(40);

**if**(!(HAL\_GPIO\_ReadPin (DHT11\_PORT, DHT11\_PIN)))

{

delay\_us(80);

**if**((HAL\_GPIO\_ReadPin (DHT11\_PORT, DHT11\_PIN))) Response = 1;

**else**Response = -1; // 255

}

**while**((HAL\_GPIO\_ReadPin (DHT11\_PORT, DHT11\_PIN))); // wait for the pin to go low

**return**Response;

}

uint8\_t **DHT11\_Read**(**void**)

{

uint8\_ti,j;

**for**(j=0;j<8;j++)

{

**while**(!(HAL\_GPIO\_ReadPin (DHT11\_PORT, DHT11\_PIN))); // wait for the pin to go high

delay\_us(40); // wait for 40 us

**if**(!(HAL\_GPIO\_ReadPin (DHT11\_PORT, DHT11\_PIN))) // if the pin is low

{

i&= ~(1<<(7-j)); // write 0

}

**else**i|= (1<<(7-j)); // if the pin is high, write 1

**while**((HAL\_GPIO\_ReadPin (DHT11\_PORT, DHT11\_PIN))); // wait for the pin to go low

}

**return**i;

}

▼

GPIO OUTPUT에서 LOW펄스를 보낸후 HIGH펄스를 보내지 않고 INPUT 으로 GPIO설정

▼

값이 읽히기 시작 하지만, 값이 한,두번 나오고 읽히지 않음

▼

Timer1에서 사용한 Delay 시간이 정확하지 않음을 파악

▼

Timer1에서 나오는 Delay\_us을 정확히 맞춤

**void** **delay\_us**(uint32\_t time\_us)

{

\_\_HAL\_TIM\_SET\_COUNTER(&htim1,0);

**while**((\_\_HAL\_TIM\_GET\_COUNTER(&htim1))<time\_us);

}

DWT\_Delay로 변경해서 진행

**void** **DWT\_Delay\_us**(**volatile** uint32\_t au32\_microseconds)

{

uint32\_t au32\_initial\_ticks = DWT->CYCCNT;

uint32\_t au32\_ticks = (HAL\_RCC\_GetHCLKFreq() / 1000000);

au32\_microseconds \*= au32\_ticks;

**while** ((DWT->CYCCNT - au32\_initial\_ticks) < au32\_microseconds-au32\_ticks);

}

▼

온도와 습도의 연속적으로 출력 되었지만, 이상한 값이 하나씩 나오게 되었음

▼

데이터시트를 확인하여 8비트 습도 정수데이터 +8비트 습도 십진데이터 + 8비트 온도 정수데이터 + 8비트 온도 십진데이터 +8비트 체크섬을 파악 마지막 8비트 체크섬과 앞서나온 총32비트의 값이 일치하는지 여부 파악, 변경

**if**((uint8\_t)(rh1 + rh2 + temp1 + temp2) != checksum)

{

**return** 0;

}

▼

온도와 습도의 따른 LED변화

외부 인터럽트 사용

stm32l4xx\_hal\_exti.c에서

\_\_weak **void** **HAL\_GPIO\_EXTI\_Callback**(uint16\_t GPIO\_Pin)

함수를 가져와 재정의

이 함수의 인자를 확인하지 못하고

void HAL\_GPIO\_EXTI\_Callback(uint16\_t GPIO\_Pin)

{

if(HAL\_GPIO\_ReadPin(GPIOA, GPIO\_PIN\_11))

{

a++;

if(a==4)

{

a=0;

HAL\_GPIO\_WritePin(GPIOC,GPIO\_PIN\_6,RESET);

HAL\_GPIO\_WritePin(GPIOC,GPIO\_PIN\_7,RESET);

}

}

if(HAL\_GPIO\_ReadPin(GPIOA, GPIO\_PIN\_12))

{

b++;

if(b==4)

{

b=0;

HAL\_GPIO\_WritePin(GPIOC,GPIO\_PIN\_8,RESET);

HAL\_GPIO\_WritePin(GPIOA,GPIO\_PIN\_8,RESET);

}

}

이렇게 함수 사용

▼

**enum** status {

*stop* = 0,

*start* = 1

}current\_status = *stop*;

**void** **HAL\_GPIO\_EXTI\_Callback**(uint16\_t GPIO\_Pin)

{

**if**(GPIO\_Pin == GPIO\_PIN\_11)

{

**if**(HAL\_GPIO\_ReadPin(GPIOA, GPIO\_PIN\_11))

{

current\_status = *start*;

}

}

**if** (GPIO\_Pin == GPIO\_PIN\_12)

{

**if**(HAL\_GPIO\_ReadPin(GPIOA, GPIO\_PIN\_12))

{

current\_status = *stop*;

}

}

}

이렇게 함수의 인자를 사용하여 외부인터럽트 함수 사용

Timer사용

Timer 1사용 – Delay 함수를 사용하기 위하여

Timer 6사용 – Timer을 사용하여 매 시간마다 센서들의 값을 읽어올 수 있게 구성

APB1 Timer, APB2 Timer 기본 클럭 4MHz 이지만 80MHz로 변경

Timer 1 -> 1us

Clock Source : Internal Clock

Counter Setting

Prescaler : 79

Counter Period : 0xffff-1 = 65535

Timer 6 -> 1ms

Mode : Activated

Counter Setting

Prescaler : 199

Counter Period : 400

stm32l4xx\_hal\_tim.c에서

\_\_weak **void** **HAL\_TIM\_PeriodElapsedCallback**(TIM\_HandleTypeDef \*htim)

함수를 main함수로 불러와 사용(weak 표시(처리)되어있는 함수는 사용자가 재정의할 수 있음

**void** **HAL\_TIM\_PeriodElapsedCallback**(TIM\_HandleTypeDef \*htim)

{

**if**(htim->Instance == TIM6) //1ms

{

**if**(current\_status == *start*)

{

**if**(timer\_cnt++ > 100)

{

acc();

**if** (dht11\_timer\_cnt ++ >10)

{

dht11\_data();

dht11\_timer\_cnt = 0;

}

timer\_cnt = 0;

}

}

}

}

main문 안에

HAL\_TIM\_Base\_Start(&htim1);

HAL\_TIM\_Base\_Start\_IT(&htim6);

함수를 사용하여 timer 스타트 함

UART 통신

**int** **\_write**(**int** file, **unsigned** **char** \* p, **int** len)

{

HAL\_UART\_Transmit(&huart1, p, len, 10);

**return** len;

}

이 함수를 사용하여 printf문을 사용하였다.

하지만 HAL\_UART\_Receive\_IT(); 함수를 사용하여 인터럽트를 사용하였을 때 printf문이 잘 실행되지 않았다.

▼

uint32\_t **DWT\_Delay\_Init**(**void**)

{

/\* Disable TRC \*/

CoreDebug->DEMCR &= ~CoreDebug\_DEMCR\_TRCENA\_Msk; // ~0x01000000;

/\* Enable TRC \*/

CoreDebug->DEMCR |= CoreDebug\_DEMCR\_TRCENA\_Msk; // 0x01000000;

/\* Disable clock cycle counter \*/

DWT->CTRL &= ~DWT\_CTRL\_CYCCNTENA\_Msk; //~0x00000001;

/\* Enable clock cycle counter \*/

DWT->CTRL |= DWT\_CTRL\_CYCCNTENA\_Msk; //0x00000001;

/\* Reset the clock cycle counter value \*/

DWT->CYCCNT = 0;

/\* 3 NO OPERATION instructions \*/

\_\_ASM **volatile** ("NOP");

\_\_ASM **volatile** ("NOP");

\_\_ASM **volatile** ("NOP");

/\* Check if clock cycle counter has started \*/

**if**(DWT->CYCCNT)

{

**return** 0; /\*clock cycle counter started\*/

}

**else**

{

**return** 1; /\*clock cycle counter not started\*/

}

}

여기서 클럭을 변경했다가 다시 돌아오는 과정에서

MX\_USART1\_UART\_Init();

이 함수가 잘 실행되지 않아 발생 된 문제로 보임

▼

DWT\_Delay\_Init();

함수 뒤에 MX\_USART1\_UART\_Init(); 함수를 사용하여

다시 초기화를 하였다.

(※만약 되던 것이 잘 안되었을 경우 위에서부터 하나씩 주석 처리 해보면서 어디 문제였는지 찾아볼 것)

▼

HAL\_UART\_Receive\_IT(&huart1, &rxbuffer, 5);

main 함수 안에 이 함수를 사용하였더니 1을 여러번 눌러야지 start가 되었음

▼

HAL\_UART\_Receive\_IT(&huart1, &rxbuffer, 5); 이것을

HAL\_UART\_Receive\_IT(&huart1, &rxbuffer, 1); 로 변경하니 1을 한번 입력받았을 때 바로 callback 함수로 들어갈수 있었다.

**void** **HAL\_UART\_RxCpltCallback**(UART\_HandleTypeDef \*huart)

{

**if** (huart->Instance == USART1)

{

**switch** (cmd\_state)

{

**case** 1 :

**if**(rxbuffer == 0x22)

{

**sprintf**(buffer,"status = start");

HAL\_UART\_Transmit(&huart1, (uint8\_t\*)buffer, **strlen**(buffer), 10);

current\_status = *start*;

cmd\_state ++;

}

**else** **if** (rxbuffer == 0x99)

{

**sprintf**(buffer,"status = stop");

HAL\_UART\_Transmit(&huart1, (uint8\_t\*)buffer, **strlen**(buffer), 10);

current\_status = *stop*;

**return**;

}

**else**

{

**return**;

}

**break**;

**case** 2 :

**if** (rxbuffer == 0x70)

{

//acc 만 내보내준다

//acc(&accx,&accy,xAngle);

angle = (accx +accy) /2;

**printf**("angle = %d\r\n",angle);

cmd\_buf[0] = 0x70;

cmd\_state = cmd\_state +2;

}

**else** **if** (rxbuffer == 0x71)

{

//temp랑 rh를 보내준다

//dht11\_data(&Temperature,&Humidity,temp\_setting,RH\_setting);

**printf**("Temp = %d\r\n",Temperature);

**printf**("RH = %d\r\n",Humidity);

cmd\_buf[0] = 0x71;

cmd\_state = cmd\_state +2;

}

**else** **if** (rxbuffer == 0x72)

{

//acc,temp,rh 보내준다

acc(&accx,&accy,xAngle);

angle = (accx +accy) /2;

//dht11\_data(&Temperature,&Humidity,temp\_setting,RH\_setting);

**printf**("angle = %d\r\n",angle);

**printf**("Temp = %d\r\n",Temperature);

**printf**("RH = %d\r\n",Humidity);

cmd\_buf[0] = 0x72;

cmd\_state = cmd\_state +2;

}

**else** **if** (rxbuffer == 0x73)

{

//acc 값을 받아 acc\_setting에 넣는다

cmd\_buf[0] = 0x73;

cmd\_state ++;

}

**else** **if** (rxbuffer == 0x74)

{

//temp 값을 받아 temp\_setting에 넣는다,rh 값을 받아 rh\_setting에 넣는다

cmd\_buf[0] = 0x74;

cmd\_state ++;

}

**else** **if** (rxbuffer == 0x75)

{

//acc 값을 받아 acc\_setting,temp 값을 받아 temp\_setting,rh 값을 받아 rh\_setting에 넣는다

cmd\_buf[0] = 0x75;

cmd\_state ++;

}

**else** **if** (rxbuffer == 0x76)

{

cmd\_buf[0] = 0x76;

cmd\_state = cmd\_state +2;;

}

**else** **if** (rxbuffer == 0x77)

{

cmd\_buf[0] = 0x77;

cmd\_state = cmd\_state +2;;

}

**else** **if** (rxbuffer == 0x78)

{

//하고싶은거

cmd\_buf[0] = 0x78;

cmd\_state = cmd\_state +2;;

}

**else** **if** (rxbuffer == 0x79)

{

//하고싶은거

cmd\_buf[0] = 0x79;

cmd\_state = cmd\_state +2;;

}

**else**

{

rxbuffer = 1;

}

**break**;

**case** 3:

**if**(cmd\_buf[0] == 0x73)

{

**sprintf**(buffer,"enter acc");

HAL\_UART\_Transmit(&huart1, (uint8\_t\*)buffer, **strlen**(buffer), 10);

HAL\_UART\_Receive(&huart1, &xAngle, 2, 10);

HAL\_UART\_Transmit(&huart1, &xAngle, 2, 10);

//acc(&accx,&accy,xAngle);

angle = (accx +accy) /2;

**printf**("angle = %d\r\n",angle);

cmd\_state ++;

}

**else** **if**(cmd\_buf[0] == 0x74)

{

**sprintf**(buffer,"enter temp,rh");

HAL\_UART\_Transmit(&huart1, (uint8\_t\*)buffer, **strlen**(buffer), 10);

HAL\_UART\_Receive(&huart1, &TEMP, 2, 10);

HAL\_UART\_Transmit(&huart1, &TEMP, 2, 10);

HAL\_UART\_Receive(&huart1, &RH, 2, 10);

HAL\_UART\_Transmit(&huart1, &RH, 2, 10);

temp\_setting = TEMP;

RH\_setting = RH;

//dht11\_data(&Temperature,&Humidity,temp\_setting,RH\_setting);

**printf**("Temp = %d\r\n",Temperature);

**printf**("RH = %d\r\n",Humidity);

cmd\_state ++;

}

**else** **if**(cmd\_buf[0] == 0x75)

{

**sprintf**(buffer,"enter acc, temp, rh");

HAL\_UART\_Transmit(&huart1, (uint8\_t\*)buffer, **strlen**(buffer), 10);

HAL\_UART\_Receive(&huart1, &xAngle, 2, 10);

HAL\_UART\_Transmit(&huart1, &xAngle, 2, 10);

HAL\_UART\_Receive(&huart1, &TEMP, 2, 10);

HAL\_UART\_Transmit(&huart1, &TEMP, 2, 10);

HAL\_UART\_Receive(&huart1, &RH, 2, 10);

HAL\_UART\_Transmit(&huart1, &RH, 2, 10);

temp\_setting = **atoi**((**char**\*)&TEMP);

RH\_setting = **atoi**((**char**\*)&RH);

cmd\_state ++;

}

**else**

{

cmd\_state = 1;

}

**break**;

**case** 4:

**if**(rxbuffer == 0x33)

{

cmd\_state ++;

}

**else**

{

cmd\_state = 1;

}

**break**;

**case** 5 :

**if** (rxbuffer == 0x44)

{

**sprintf**(buffer,"data over");

HAL\_UART\_Transmit(&huart1, (uint8\_t\*)buffer, **strlen**(buffer), 10);

cmd\_state = 1;

}

**else**

{

cmd\_state = 1;

}

**break**;

//HAL\_UART\_Receive\_IT(&huart1, pData, Size);

HAL\_UART\_Receive\_IT(&huart1, &rxbuffer, 1);

}

}

이렇게 UART문을 짜보았고

/\*

0 start 0x22

1 cmd 0x70~0x77

0x70 read acc

0x71 read temp

0x72 read rh

0x73 read acc,temp,rh

0x74 get acc

0x75 get temp

0x76 get rh

0x77 get temp,rh,acc

0x78

0x79

2 data

3 checksum 0x33

4 end 0x44

\*/

이런 형태의 방식을 사용해 보았다

▼

하지만 0x7x로 값을 보냈을경우 case3번으로 들어가서 바로 나와야되지만 0x74를 두번 입력해야지만 들어가는 문제점을 파악하였다.