

개인 작업 일지

| | | | |
|---------------------------|------------------------|----------|-----|
| 프로젝트명 | 스마트 화분 Smart Flowerpot | | |
| 팀명 | 2학년 2조 | | |
| 작업자 | 학과 | 학번 | 이름 |
| | 컴퓨터 공학과 | 20141640 | 이석원 |
| 모든 내용은 팀 작업 일지에서 발췌 했습니다. | | | |

| | | | |
|------|-------------------------|------|---------|
| 작업계획 | 무선랜을 통한 TCP 데이터 전송 실험 | | |
| 작업기간 | 2017년10월27일 09:00~12:00 | 작업장소 | W6 301호 |

작업내용



예제 프로그램, echo_server 응용 프로그램, AP(공유기), "USB to serial", USB 확장 케이블, 부품 및 데이터시트, DC 12V/1A adaptor, 라즈베리 파이

컨트롤 보드와 라즈베리 파이간 통신 실험

라즈베리 파이에서 데이터를 받은 모습입니다.

```
Send data len: 11
rx len: 11
hello: 241

Send data len: 11
rx len: 11
hello: 242

Send data len: 11
rx len: 11
hello: 243

Send data len: 11
```

컨트롤 보드가 라즈베리 파이로부터 확인 응답을 받은 모습입니다.

```
TX:hello:217

TX:hello:218

TX:hello:219
TX:hello:220
```

atmega작동 코드 분석

```
while(1)
{
    //TODO:: Please write your application code
    wifiMain();

    if ( isElapsed())
    {
        sprintf(strTemp,"hello:%d\r\n",counter++); //가장 핵심적인 부분
        wifiSendData(strTemp, strlen(strTemp)); //가장 핵심적인 부분
        debugprint("TX:%s\r\n",strTemp); //가장 핵심적인 부분
    }
}
```

서버 프로그램 작동 코드 분석

```

126 | while(1)
127 | {
128 |     readBufSize = read(clnt_sock, rcv_buf, RX_DATA_MAX); //소켓이로부터 문자를 읽어서 rcv_buf에 저장 하는데 그 크기는 최대 크기는 2048BYTE다
129 |
130 |     if(readBufSize > 0)
131 |     {
132 |         printf("rx len:%d\n", readBufSize); //read()함수는 정상적으로 읽었다면 읽은 BYTE수를 반환 한다.
133 |
134 |         rcv_buf[readBufSize] = 0; //맨 뒷자리를 0으로 셋팅
135 |         printf("%s\n", rcv_buf); //읽어온 대로 출력
136 |
137 |         // echo TX
138 |         writeBufSize = write(clnt_sock, rcv_buf, readBufSize); //받은 데이터를 clnt_sock으로 되돌려 줍니다.
139 |         printf("Send data len: %d\n", writeBufSize); //되돌려준 데이터의 길이를 알려줍니다.
140 |         if(writeBufSize < 0)
141 |         {
142 |             printf("echo write() error\n");
143 |             break;
144 |         }
145 |     }

```

| | | | |
|------|--------------------------------------|--|--|
| 작업계획 | 컨트롤 보드로부터 받은 데이터를 라즈베리파이에서 파일로 저장하기. | | |
|------|--------------------------------------|--|--|

| | | | |
|------|-------------------------|------|---------|
| 작업기간 | 2017년11월03일 09:00~12:20 | 작업장소 | W6 301호 |
|------|-------------------------|------|---------|

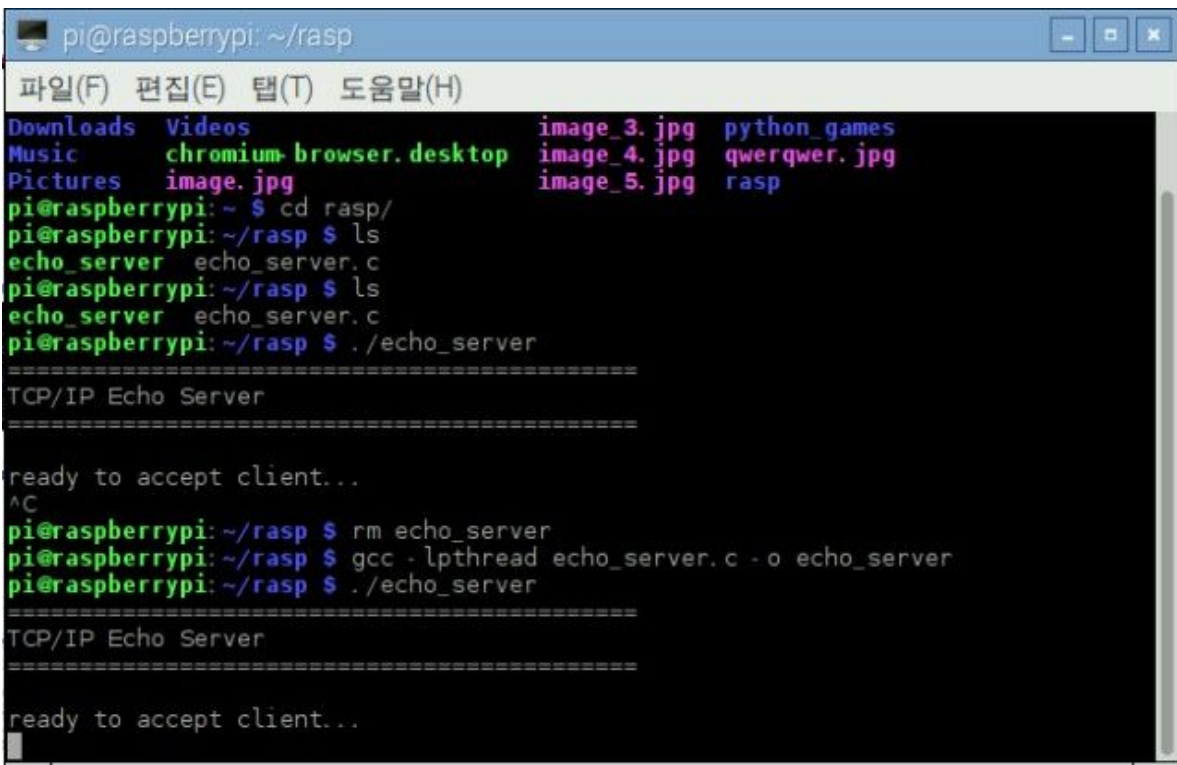
작업내용

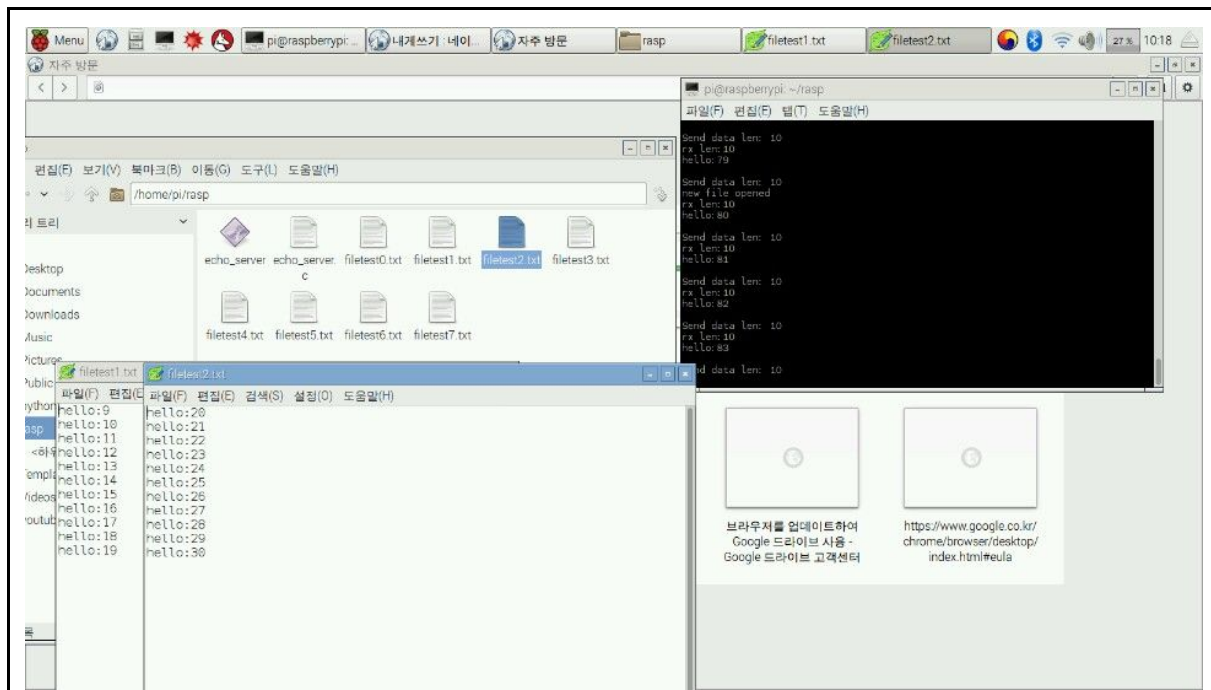
```

122 | char filename[20] = "filetest0.txt"; //파일 이름 -----수정 v
123 |
124 | FILE* stream = fopen(filename, "w"); //파일 오픈
125 | if(stream == NULL) //오픈 확인
126 | {
127 |     printf("file open error \n");
128 |     exit(1);
129 | } //-----수정 ^
130 |
131 | while(1)
132 | {
133 |     if(ftell(stream) > 1000) //각 파일의 크기 검사 -----수정 v
134 |     {
135 |         printf("new file opened \n");
136 |         if(fclose(stream) == EOF) //크기를 벗어나면 파일닫음
137 |         {
138 |             printf("file close error \n");
139 |             exit(1);
140 |         }
141 |
142 |         filename[8]++; //새로운 파일을 만들기 위한 단계
143 |
144 |         stream = fopen(filename, "w"); //새 파일 오픈
145 |
146 |         if(stream == NULL) //오픈 확인
147 |         {
148 |             printf("file open error \n");
149 |             exit(1);
150 |         }
151 |     } //-----수정 ^
152 |
153 |     readBufSize = read(clnt_sock, rcv_buf, RX_DATA_MAX); //소켓이로부터 문자를 읽어서 rcv_buf에 저장 하
154 |
155 |     if(readBufSize > 0)
156 |     {
157 |         printf("rx len:%d\n", readBufSize); //read()함수는 정상적으로 읽었다면 읽은 BYTE수를 반환 한다.
158 |
159 |         rcv_buf[readBufSize] = 0; //맨 뒷자리를 0으로 셋팅
160 |         printf("%s\n", rcv_buf); //읽어온 대로 출력
161 |
162 |         fprintf(stream, "%s", rcv_buf); //파일 입력 -----수정 =====
163 |     }

```

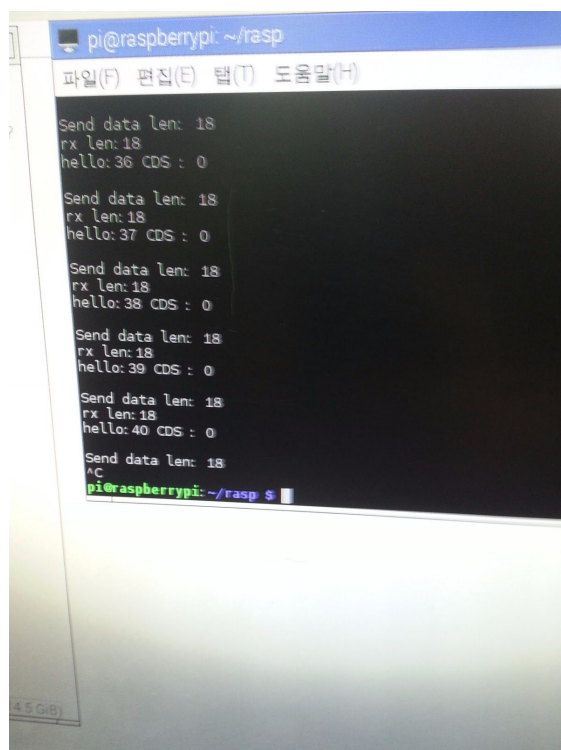
서버프로그램 수정

| | | | |
|--|--|------|---------|
| 작업계획 | 컨트롤보드에 연결된 온습도 조도 센서를 통해 LED를 통제하고 온도 습도 조도 각 데이터를 컨트롤 보드와 라즈베리파이의 wifi를 통해 라즈베리파이로 전송 후 데이터를 파일로 저장 | | |
| 작업기간 | 2017년11월10일 09:00~12:30 | 작업장소 | W6 301호 |
| 작업내용 | | | |
| 라즈베리 파이를 TCP/IP 서버로 설정하고 이용하는 것은 저희에게 아직 추상적이고 알고있는 정보가 너무 적어서 보류하고 컨트롤 보드의 동작과 라즈베리파이와의 통신에만 집중 하기로 했습니다. | | | |
| 이번엔 usb를 가져와서 라즈베리파이에서 데이터를 파일로 저장하는 수정된 echo_server파일을 컴파일하고 실행했습니다. | | | |
|  | | | |
| 컴파일성공해서 wifi연결 대기하는 모습입니다. | | | |



와이파이 연결도 잘되고 데이터도 잘 저장 하고 있습니다. 에코서버 프로그램은 단순히 숫자를 1초마다 1을 증가하고 있습니다.

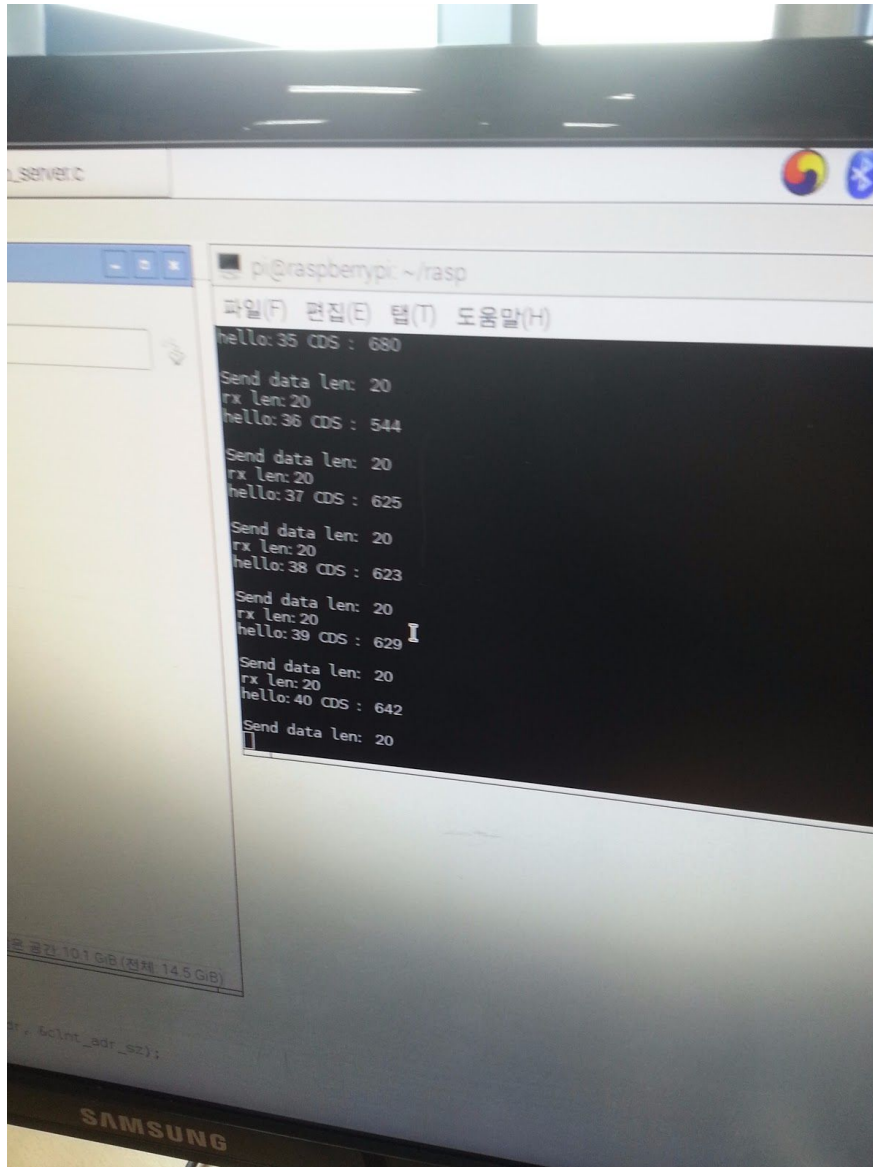
그리고 나서 컨트롤보드로 조도센서를 읽어서 wifi로 전송하는 작업을 했습니다.



컨트롤 보드에 프로그램을 적재하고 파일저장을 성공했습니다. 그러나 뭐가 문제인지 CDS값이 계속 0으로 나왔습니다.



CDS가 0이기 때문에 LED도 항상 불이 들어옵니다.



그렇게 한참동안 원인을 찾았지만 어이없게 핀을 잘못 꽂아서 이상한 값이 나왔던 것입니다. 위 화면은 성공적인 데이터 전송 화면입니다.

| | | | |
|---|-----------------------------------|------|---------|
| 작업계획 | 지난 시간에 이어서 온도와 습도 데이터를 라즈베리파이로 전송 | | |
| 작업기간 | 2017년11월17일 09:10~12:30 | 작업장소 | W6 301호 |
| 작업내용 | | | |
| 지난 시간에 진행했던 장치 그대로 다시 연결 하고 CDS가 제대로 되나 확인후 온습도 데이터전송 작업을 진행했습니다. 코드를 직관적으로 봤을때 문제가 없었고 연결한 핀도 이상이 없었습니다. 그래서 핀설정에 관하여 잘모르기 때문에 자세히 들여다 봤습니다. | | | |

```

    debugPrint( "Start LOOPWRITE ",

    int counter = 0;
    /*-----led부분제거*/
    LED_Init ();          // PORTB = GREEN, PORTE = RED---
    AdcInit(1);           // PORTF = ADC1
    TIMER_Init();
    |
    sei(); //-----

```

이부분은 메인함수에서 CDS입력데이터의 핀을 설정하는 부분입니다.

```

.....
#define TEMP_PORT      F
#define SCK            0
#define DATA          1

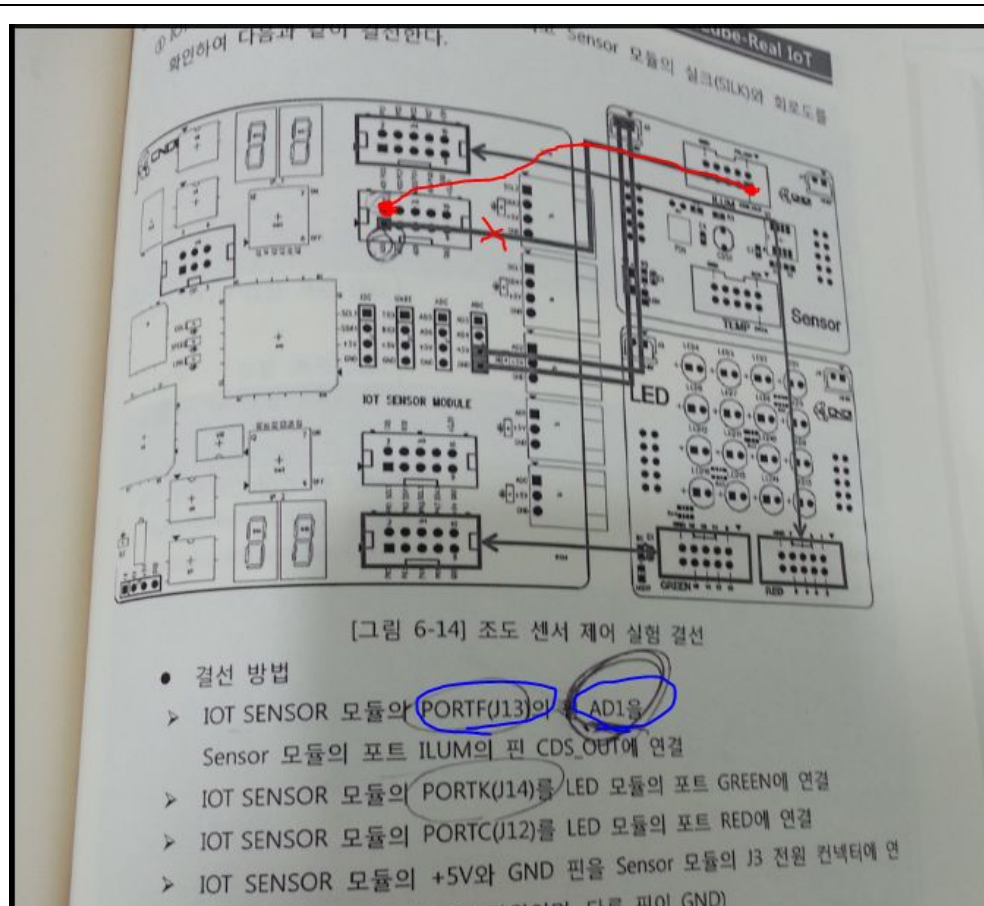
#define NOACK          0
#define ACK            1

unsigned short val_temp, val_humi;
unsigned short SHT11_humi, SHT11_temp;
unsigned short error, checksum;
unsigned char sensing_type;

|
static void I2C_data_output (void)
{   pinMode(TEMP_PORT, DATA, OUTPUT); }
static void I2C_data_input()
{   pinMode(TEMP_PORT, DATA, INPUT); }
static void I2C_sck_output (void)
{   pinMode(TEMP_PORT, SCK, OUTPUT); }
static void I2C_sck_input (void)
{   pinMode(TEMP_PORT, SCK, INPUT); }

```

이부분은 온도와 습도입력데이터의 핀을 설정하는 부분입니다. 저희가 주목한점은 핀을 지정하는 함수가 서로 다르다는 것입니다. 교재에 보면 pinMode()함수로 핀을 입력 또는 출력으로 구성할지 정하고 digitalWrite()와 digitalRead()함수는 디지털핀의 입출력을 수행하는 함수 입니다. 그리고 adclnit() 함수는 adc채널을 초기화 하는 함수라고 합니다. adc는 Analog Digital Converter의 약자로 adclnit()으로 핀을 설정하면 해당핀은 아날로그 데이터를 읽습니다. 그런데 한가지 의문점은 F포트가 어떤 포트인지에 대한 정보가 없다는 것입니다. 디지털 입출력 포트는 A~L포트까지이고 아날로그 입력 포트는 정보가 없고 ucAdcChannel : 0(ADC0)~7(ADC7)까지 라고만 나와있습니다. 데이터 시트에도 포트에 대한 정보는 없었습니다. 그리고 의문을 더욱 미궁으로 빠지게한 것은 토양 수분 센서에서의 핀 설정 입니다.



```

debugPrint( "START LOOP WITH ",
  int counter = 0;
  /*-----led부분제거*/
  LED_Init ( );          // PORTB = GREEN, PORTE = RED---
  AdcInit(1);            // PORTF = ADC1
  TIMER_Init( );
  |
  sei(); //-----

```

CDS센서 데이터를 읽는 핀설정과 결선 방법입니다. 분명히 여기서 결선된 부분은 portF의 1번 핀 입니다.

soilmoisture_sensor - AtmelStudio

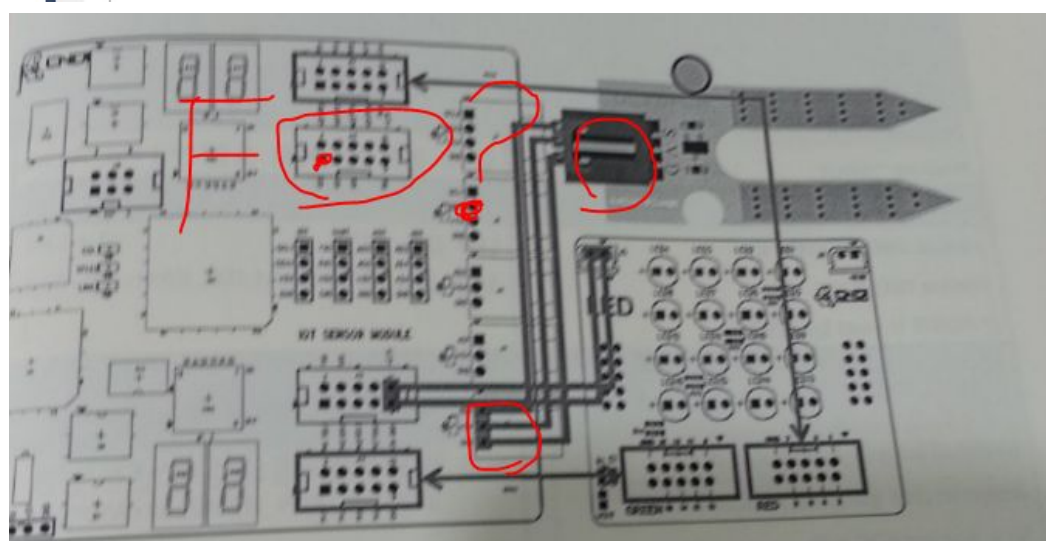
```

File Edit View VAssistX Project Build Debug Tools Window Help
main.c x
LED_Init.for
    digitalWrite(RED_PORT, i, HIGH);
    else if (color == GREEN)
        digitalWrite(GREEN_PORT, i, HIGH);
    }
    else
    {
        if (color == RED)
            digitalWrite(RED_PORT, i, LOW);
        else if (color == GREEN)
            digitalWrite(GREEN_PORT, i, LOW);
    }
}

void main (void)
{
    // debugInit();

    LED_Init (); // PORTB = GREEN, PORTE = RED
    AdcInit(0); // PORTF = ADC0
    TIMER_Init();
    sei();
}

```

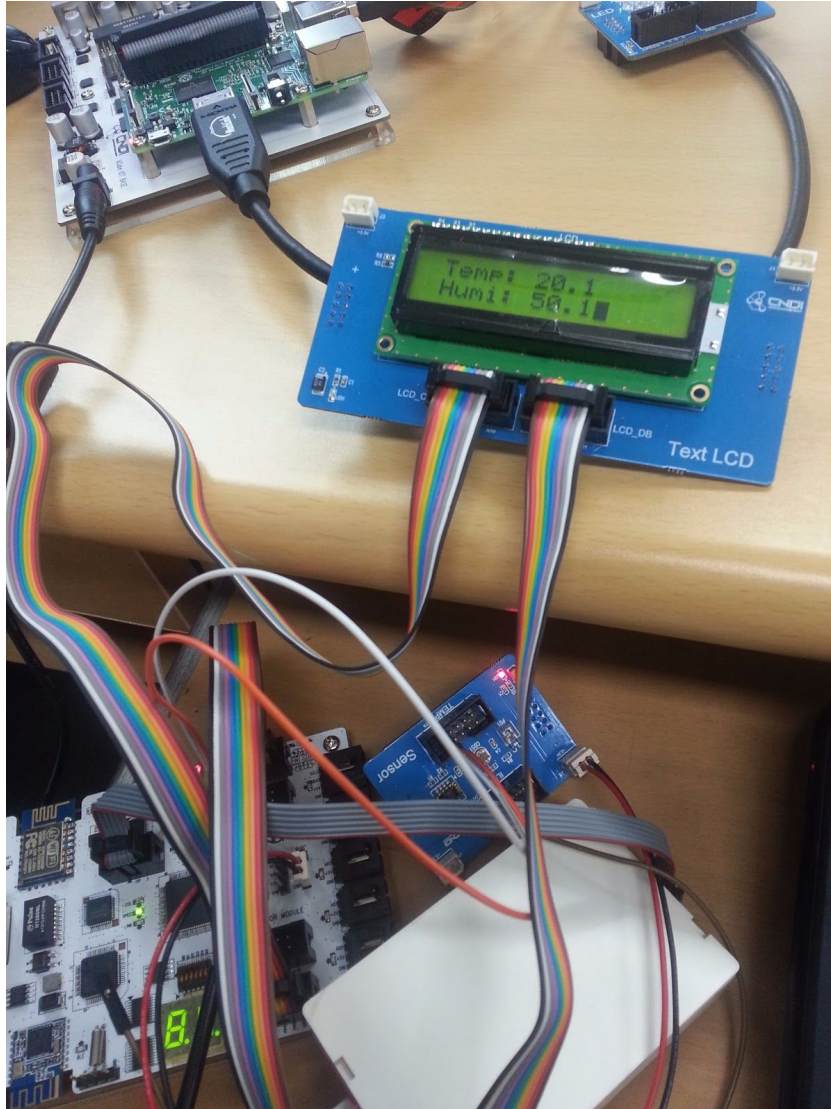


[그림 6-17] Soil Moisturer Sensor 제어 실험 결선

- 결선 방법
 - IOT SENSOR 모듈의 AD0핀을 Soil Moisturer 센서 모듈의 Signal 핀에 연결
 - IOT SENSOR 모듈의 +5V핀을 Soil Moisturer 센서 모듈의 V(Vcc) 핀에 연결
 - IOT SENSOR 모듈의 GND핀을 Soil Moisturer 센서 모듈의 G(GND) 핀에 연결

분명히 F포트의 0번 핀이면 빨간색 점찍은 부분에 결선 해야하지만 알수 없는 핀에 꽂은 그림입니다.

의문점이 풀리기는 커녕 늘어만 가는 상황에서 조교수님들 한테 조언을 구했습니다. 핀문제가 아니라 코드에 문제가 있을 수 있으니 CDS와 온도 습도를 구분해서 따로 실습을 해보라는 조언을 듣고 다시 처음부터 온도와 습도데이터 입력부터 차근차근 해보기로 정했습니다.

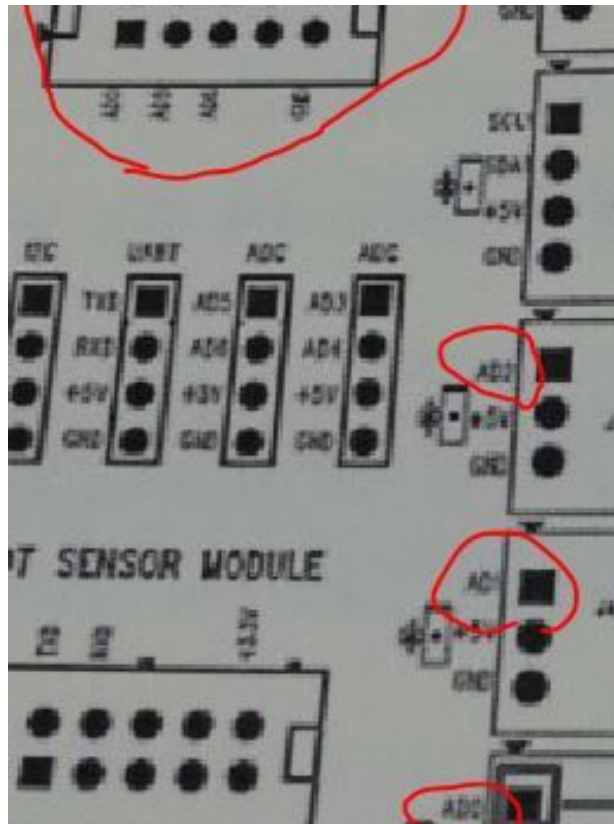


목표 달성도, 앞으로 개선점

목표에 한참 못 미치는 결과를 얻었습니다. 핀설정에 대한 정보는 턱없이 부족하고, LCD에 온습도가 잘 나오는 걸 보면 포트의 문제뿐만 아니라 소스파일에서 수정할 부분도 상당합니다. 제출 기간내에 프로젝트를 마칠수 있을지 걱정입니다.

| | | | |
|------|--|------|---------|
| 작업계획 | 포트 번호 문제 해결과 지난 시간에 이어서 온도와 습도 데이터를 라즈베리파이로 전송 | | |
| 작업기간 | 2017년11월22일 17:30~19:00 | 작업장소 | W6 301호 |

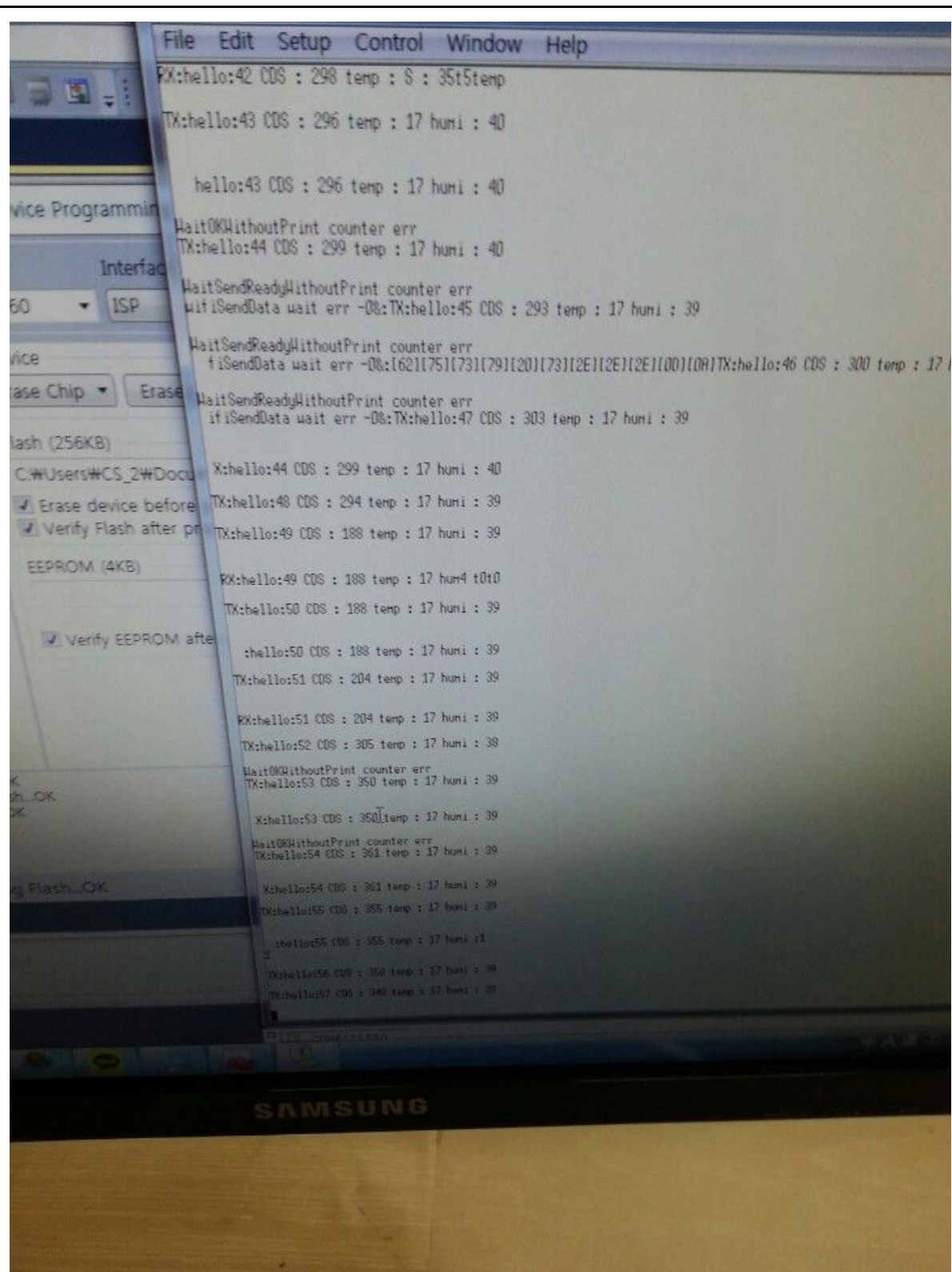
작업내용



캡처 화면에는 완전히 보이지 않지만 빨간 동그라미 표시한 부분의 핀 이름이 동일 합니다. 그래서 같은 F포트의 핀이 2개씩 존재한다고 추측 했습니다.

```
Workingdir/ATmega/WifiTest/WifiTest/sensor.c
1 부분 : 27-38 줄
27 27 *****
28 28
29 29 #include "sensor.h"
30 30 + #define TEMP_PORT F
31 31 + #define SCK 2
32 32 + #define DATA 5
33 33 +
34 34 + #define NOACK 0
35 35 + #define ACK 1
36 36
37 37 unsigned short val_temp, val_humi;
38 38 unsigned short SHT11_humi, SHT11_temp;
```

같은 F포트의 서로다른 핀 설정을 하고 프로그램을 실행 했습니다.



성공적으로 Humi, temp 값을 읽었습니다.

목표 달성도, 앞으로 개선점

컨트롤 보드에 적힌 동일한 핀명은 포트도 동일하고 핀도 동일하다는 중요한 정보를 얻은 성공적인 실습이었습니다.

| | | | |
|----------|--|------|---------|
| 작업 계획 | 포트 번호 문제 해결과 지난 시간에 이어서 온도와 습도 데이터를 라즈베리파이로 전송 | | |
| 작업 기간 | 2017년11월22일 17:30~19:00 | 작업장소 | W6 301호 |

작업내용

```

+ Workingdir/ATmega/WifiTest/WifiTest/soilpart.c
6 + #include "adc.h"
7 + #include "gpio.h"
8 + #include "soilpart.h"
9 +
10 + #define MAX_MOTOR_PIN_NUM 2
11 +
12 + #define MOTOR_PORT F
13 +
14 + #define MT_P 6
15 + #define MT_N 3
16 +
17 + #define ON 1
18 + #define OFF 0
19 +
20 + #define CLOCKWISE 1 // CW
21 + #define COUNTERCLOCKWISE 0 // CCW
22 +
...
143 - cds_gun_AdcValue = AdcRead();
144 - // soil_gun_AdcValue = AdcRead();
145 - // debugprint("gun_AdcValue : %d

```

(CDS, HUMI, TEMP, LED, wifi)와 (SOIL MOISTURE, 모터)를 합치는 작업을 진행했습니다. 모터를 추가하는 부분은 문제가 없는데 토양수분 값을 읽어들이는데 문제가 발생했습니다. cds와 soil moisture는 모두 아날로그 입력을 받아야 하는데 아날로그 입력을 실행하는 함수가 동일하다는 것입니다. 동일한 함수에서 출력되는 값은 같은 것이니 결국 둘중 하나는 포기해야 하는 상황입니다. 팀원들 끼리 무엇을 포기 할지 의논 하는 중에 햇빛은 어차피 창가에 두면 날씨 좋은 날이 많기 때문에 LED의 필요성이 더 낮은 것으로 결론이 나서 LED를 포기 했습니다.

<https://youtu.be/P3wFCKH1e9c>

영상을 보면 아시겠지만 물을 충분히 줘야하는 모터 장치가 채 3초도 안되어 호스를 잠금니다. 팀원들과 상의 해서 적당한 한계 토양수분치를 정하고 모터의 작동 대기 시간을 조정 했습니다.

Workingdir/ATmega/WifiTest/WifiTest/WifiTest.c

```

148 +
149 +     if(soil_gun_AdcValue<=30) { //-----
150 +         Motor_Action(ON, COUNTERCLOCKWISE);
151 +         _delay_ms(1000);
152 +
153 +         Motor_Action(OFF, COUNTERCLOCKWISE);
154 +         _delay_ms(15000);
155 +
156 +
157 +         Motor_Action(ON, CLOCKWISE);
158 +         _delay_ms(1000);
159 +
160 +         Motor_Action(OFF, CLOCKWISE);
161 +
162 +         // _delay_ms(10000);
163 +     }

```

토양수분치는 대략 30이 한계점이고 모터작동 대기시간은 대략 15초 입니다. 하지만 어느 정도의 수치가 적합한지 더 탐구해볼 필요가 있습니다.

| | | | |
|---|------------------------|------|---------|
| 작업계획 | 장치 구현 | | |
| 작업기간 | 2017년12월1일 09:10~17:00 | 작업장소 | W6 301호 |
| 작업내용 | | | |
| 실행 동영상입니다. | | | |
| <p>데이터를 전송 하고 저장하는 부분은 문제가 없지만, 물주는 장치에서 호스를 적당하게 펴고 접는 부분을 찾기가 힘듭니다. 왜냐하면 모터동작의 딜레이시간을 입력해서 개폐장치를 작동하는데 순간적으로 모터가 확확 돌아가서 이 딜레이 시간을 찾기가 힘듭니다. 또한 자칫하면 실이 꼬여버려서 장치가 망가지게 됩니다.</p> | | | |
| <p>또한 물주는 장치의 종대한 문제점이 발견 되었는데, 튜브를 접어도 물이 조금씩 흘러 나옵니다. 이전에 실험 했을때는 작은 페트병에 물을 조금 넣고 해서 잘 됐었는데 이번에 큰페트병에 물을 가득 넣고 해보니 수압이 생각보다 강하여 개폐장치가 제대로 작동 되지 않았지만 관의 모양이 o자형 튜브에서 잘 접히는 I자형 튜브로 교체하니 좀 나아 졌습니다.</p> | | | |