

2조_입문설계_중간 보고서	
프로젝트명	식물 재배 환경 알림이
팀원	권민석, 정동호, 박상원, 이석원, 강병운
프로젝트 구상 초안	
목적	센서모듈을 통하여 획득한 재배환경 데이터를 wifi를 통해 라즈베리파이 서버에 저장하고 인터넷으로 라즈베리파이 서버에 접속하여 데이터를 확인할 수 있는 장치입니다.
감시할 데이터	온도, 토양 수분, 조도 센서를 통해 데이터를 취득합니다.
환경 조절	atmega 컨트롤 보드에서 조도 조절과 토양의 수분 조절을 해줍니다.
프로젝트 소스코드	<a href="https://github.com/gwnuysw/RasPiPod">https://github.com/gwnuysw/RasPiPod</a> 의 workingdir파일이 현재 실습중인 파일입니다.
구상도	
<p>The diagram illustrates the system architecture for the 'Plant Growing Environment Notification' project. It shows the following components and their interactions:</p> <ul style="list-style-type: none"> <li><b>User (Top Left):</b> Represented by a cartoon person at a desk with a laptop. A black arrow points from the user to the control board, labeled '웹으로 접속해서 데이터를 볼 수 있다.' (You can view the data by connecting to the web).</li> <li><b>Control Board (Top Right):</b> A circuit board labeled 'Control board'. A black arrow points from it to the user.</li> <li><b>Sensor Module (Bottom Left):</b> A circuit board labeled '온도, 조도 센서' (Temperature, Light Sensor). A black arrow points from it to the control board, labeled '온도, 조도, 토양수분 데이터 전송' (Transmission of temperature, light, and soil moisture data).</li> <li><b>Watering Device (Bottom Center):</b> A yellow watering can labeled '물주는 장치(dc모터를 이용할 예정)' (Watering device (planned to use a dc motor)). A red arrow points from the control board to it, labeled '데이터에 맞춰서 물을 키고, 물을 준다' (Turn on the water according to the data and water).</li> <li><b>LED Module (Bottom Left):</b> A circuit board labeled 'LED'. A red arrow points from the control board to it, labeled '데이터에 맞춰서 불을 키고, 물을 준다' (Turn on the light according to the data and water).</li> <li><b>Plant (Bottom Center):</b> A potted plant labeled '토양 수분 센서' (Soil moisture sensor).</li> <li><b>Text Annotations:</b> <ul style="list-style-type: none"> <li>Next to the control board: '컨트롤 보드로부터 전송 받은 데이터를 웹에서 볼 수 있게 가공한다.' (Processes the data received from the control board so it can be viewed on the web).</li> <li>Next to the sensor module: '센서로부터 얻은 데이터를 서버에 보낸다' (Sends the data obtained from the sensor to the server).</li> </ul> </li> </ul>	
재배할 식물은 방울 토마토	

**1. 잘 자라는 환경조건**

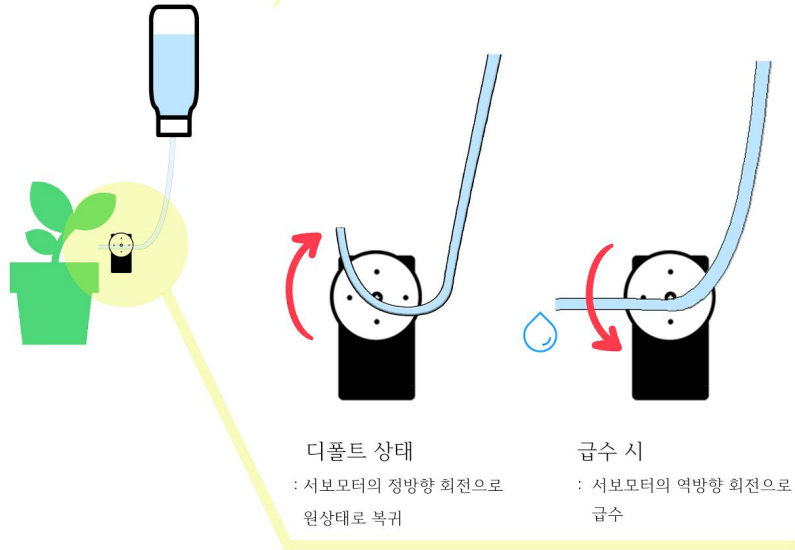
- 싹트는 온도 : 28°C
- 잘 자라는 온도 : 25~27°C, 낮 25~30°C, 밤 18~20°C
- 지온(땅속 온도) 20± 2°C
- 낮 30°C, 밤 20°C 이상이나 13°C 이하에서는 낙과, 열과 및 기형과 발생
- 햇빛의 세기 : 강한 광선을 좋아하는 채소로 햇빛을 충분히 쬌어주는 것이 좋다.
- 토양조건 : 과습에 약함. 양토 또는 식양토가 좋다.

**역할 분담**

물주는 장치를 만들고 토양 수분 정도에 맞춰 작동 시키는 역할	컨트롤 보드와 라즈베리파이 서버의 통신, 라즈베리파이 서버와 외부 인터넷망 사이의 통신을 담당	문서 종합, 작성
1팀 : 권민석, 정동호	2팀 : 박상원, 이석원	3팀 : 강병운

**실습 일자**

일시	2017년10월27일 09:00~12:00	장소	W6 301호	인원	권민석, 박상원, 이석원, 정동호, 강병운
금일 목표	1팀 권민석, 정동호	급수 환경 구상			
	2팀 이석원, 박상원	교재 p328 무선랜을 통한 TCP 서버에 데이터 전송 실험			
	3팀 강병운	보고서 작성			

1팀 실습 내용 : 물주는 장치 고안	
구상1	 <p>디폴트 상태 : 서보모터의 정방향 회전으로 원상태로 복귀</p> <p>급수 시 : 서보모터의 역방향 회전으로 급수</p>
	<p>Q1 : 급수를 하지 않는 평상시에 호스가 안정적으로 고정되는가?</p> <p>Q2 : 서보모터를 얼마만큼 회전시킬것인가?</p>
구상 2	<p>물의 이동 통로 한쪽을 스프링(장치)을 통해 여닫이로 구성하고 서브모터를 통해 물길을 열어주어 급수</p> <p>Q1)평소 급수 하지 않을 때 수압을 버텨 주는 힘의 방향을 바꿔줄 만한 힘이 서보 모터에 있는가?</p>
구상3	<p>구상1, 구상2 의 아이디어에서 착안 호스를 집게로 눌러주어 평소의 상태를 유지 하고 급수가 필요할 때 서브모터를 통해 집게를 열어준다.</p>
실습 도중 문제 해결 방법 또는 앞으로 개선점	

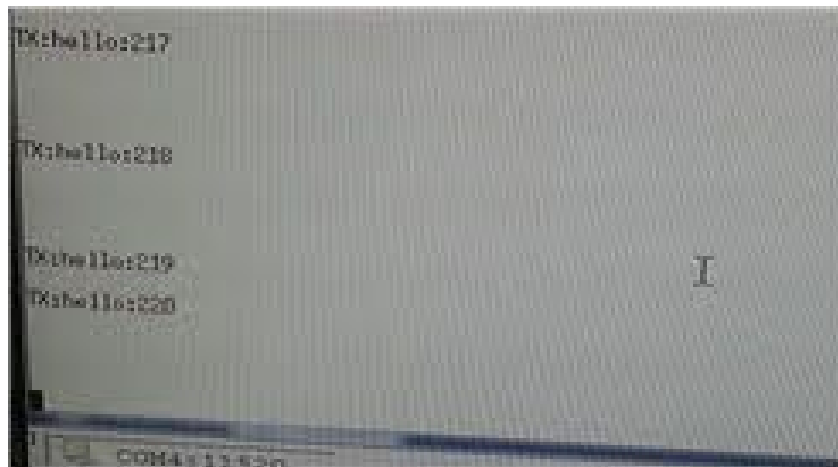
2팀 실습 내용 : 무선랜을 통한 TCP 서버에 데이터 전송 실험	
사용한 장치	<p>예제 프로그램, echo_server 응용 프로그램, AP(공유기), "USB to serial", USB 확장 케이블, 부품 및 데이터시트, DC 12V/1A adaptor</p>

장치를 연결한 모습



라즈베리 파이와 컨트롤 보드간 wifi 통신 확인

컨트롤 보드에서 보낸 데이터를 찍은 사진입니다  
1초마다 데이터를 전송하고 받습니다.  
TX가 보낸 데이터이고 RX가 받은 데이터인데, 디버그 프로그램 문제가 있어서 마우스로 블록을 채워야 RX가 보입니다.



라즈베리 파이에서 받은 데이터와 보낸 데이터를 찍은 사진입니다.  
hello:%d 문자열이 컨트롤 보드로부터 받은 데이터입니다.

```
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
```

실습 도중 문제 해결 방법 또는 앞으로 개선점

실습 장비 중에 USB to serial 케이블이란 것이 있는데 serial쪽 선이 어떻게 전압이고 어떻게 gnd고 어떻게 데이터 인지 구분이 안되서 중간에 헤맨 시간이 많았지만 인터넷에 자료를 찾아서 계속 진행 할 수 있었습니다.



디버깅 메시지가 계속 AP connection fail이 떠서 시간을 많이 허비 했습니다. 차근차근 책과 소스를 읽어보다가 18행과 63행이 의심되서 바꿔보니 접속성공 했습니다. 18행은 라즈베리파이 서버의 ip주소를 입력하고 63행은 이용하는 wifi의 id와 pw를 입력했습니다.

```

17
18 #define SERVER_IP_STR "192.168.1.33" //라즈베리파이 커고 수정해야할 부분
19 #define SERVER_PORT 50001
20
21 static void eventCallback(int eventType, uint8_t* rxBuff, int rxSize)
22 {
23     // ...
24
25     // connect AP
26     debugprint("\r\n");
27     if ( !wifiConnectAP("301", "gwnucomse"))
28     {
29         debugprint("AP connected.\r\n");
30     }
31     else
32     {
33         debugprint("AP connection fail.\r\n");
34     }
35     // ...
36 }

```

## 서버 프로그램 (echo\_server.c) 소스코드 분석

```

126 while(1)
127 {
128     readBufSize = read(clnt_sock, rcv_buf, RX_DATA_MAX); //소켓으로부터 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; //만 한자리를 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     }
146     else if (readBufSize == 0)
147     {
148         printf("readBufSize : 0\n");
149     }
150     else // readBufSize < 0
151     {
152         printf("client(%d) write error.\n", clnt_sock);
153         break;
154     }
155 }
156

```

코드가 길고 복잡하지만 통신관련 코드는 안봐도 됩니다. 캡처한 화면은 실제 데이터를

받아서 화면에 띄우고 다시 보내는 부분입니다. 실질적으로 저 부분만 수정 하면 되는데 데이터라고 해도 단순히 문자열이고 문자열을 주고 받는 것도 잘보면 파일입출력이랑 크게 다르지 않습니다. 어떻게 잘만 손보면 다양한 응용이 가능할 것 같습니다.

```
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
    }
}
```

클라이언트 wifitest.c 코드에서 실질적으로 응용할수 있는 부분의 코드는 상당히 적습니다. strTemp도 단순히 문자열입니다. 마찬가지로 잘만 손보면 다양한 응용이 가능합니다.

```
30 int main(void)
31 {
32     uint8_t strTemp[256];
33     debugInit();
34     wifiInit();
35
36     sei();
37
38     debugprint("wifi test start\r\n");
39     _delay_ms(1000);
```

Specifier	Common Equivalent	Signing	Bits	Bytes	Minimum Value	Maximum Value
int8_t	signed char	signed	8	1	-128	127
uint8_t	unsigned char	unsigned	8	1	0	255
int16_t	short	signed	16	2	-32,768	32,767
uint16_t	unsigned short	unsigned	16	2	0	65,535
int32_t	long	signed	32	4	-2,147,483,648	2,147,483,647
uint32_t	unsigned long	unsigned	32	4	0	4,294,967,295
int64_t	long long	signed	64	8	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
uint64_t	unsigned long long	unsigned	64	8	0	18,446,744,073,709,554,615

### 실습 일지

일시	2017년11월03일 09:00~12:20	장소	W6 301호	인원	권민석, 박상원, 이석원, 정동호
금일 목표	1팀 권민석, 정동호				
	2팀 이석원, 박상원	외부 인터넷 망에서 라즈베리파이 node.js 의 자바 스크립트 파일 접근하기, 컨트롤 보드에서 받은 데이터를 서버프로그램에서			

		파일로 저장하기.
--	--	-----------

1팀 실습 내용	
실습 도중 문제 해결 방법 또는 앞으로 개선점	

2팀 실습 내용	
실습 도중 문제 해결 방법 또는 앞으로 개선점	