


[프로젝트 일정]





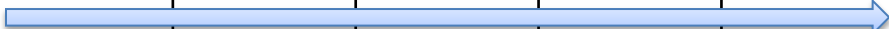

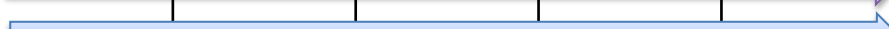

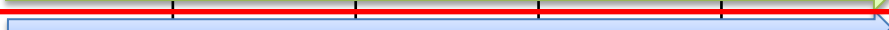


* 금주 목표 : Peripheral 제어 구현 완료 및 freeRTOS 로 이식 완료

* 달성 여부 : 미흡

* 개선 방안 : 기구 문제(시제품으로 기구 고정 및 조립 어려움)

3D 모델링, 3D프린터 구비 하였고, 차주 인쇄 및 Peripheral완료 추가됨

-  방향 설정, 자료 조사
-  부품 선정 및 구매
-  Peripheral 제어 코드 수정
-  Water Valve 기구 설계
-  Ethernet + freeRTOS 코딩
-  디버깅
-  기구 3D 설계 및 프린팅

Mon	Tue	Wed	Thu	Fri
25	26	27	28	29
				
				
				
1	2	3	4	5
				
				
8	9	10	11	12
				
				
15	16	17	18	19
				
				
				
22	23	24	25	26
				

3/26 ~ 4/26

[WEEK 3 : 진행 상황]

☑ 프로젝트 2차 : 기구

Hide completed items

Delete

79%

- ☑ ~~밸브 스텝모터 : 샤이 샤프트 구매 > 3D 프린팅~~
You have unsaved edits on this field. [View edits](#) - [Discard](#)
- ☑ ~~밸브 스텝모터 : 샤프트 연결 방법 찾기 : 3D 모델링해서 3D 프린터로 출력한 다. 3D틀 설치, 모델링, STEP파일생성 > ~ 4/8~~
- ☑ ~~밸브 스텝모터 : 모터 위치 선정 : 축변경없이 다이렉트로 물림~~
- ☐ 밸브 : 정밀 밸브 회전 반경 확인(1초 1드랍 ~ 6초 1드랍 까지) : 몇 바퀴도나 (밸브 하단부 모양에 따라 드랍 차이이기때문에 6mm 튜브 끼고 테스트) 최대 회전반경은 11바퀴. 사용할 범위 : 완전 닫힘 ~ 최초 1바퀴 정도 까지. 테스트1) 물 완충 : 90도~180도 에서 가용 드랍 나옴. 2)물 바닥 : 180도~300도에서 가용 드랍 나옴.
- ☐ 기판 위치 선정
- ☑ ~~케이블 연결 통로 구성 방법 찾기(전기 설계) : 사각 몰드 (덕트) 구매 완료 15x11mm~~
- ☑ ~~로드셀 위치 선정 : 막대모양사용, 물통 지지대 밑에 위치함.~~
- ☑ ~~로드셀 : 막대모양으로할지 체중계에들어가는형태로할지 : 막대모양으로선정 함(측정 무게 차이 때문. 체중계모델은 최소단위40g 정도이고 단독으로쓰기에는 기구설계에매함)~~
- ☐ 로드셀 고정 기구 : 발생문제(1.볼트m5 오프라인 매장에 없음. 온라인구매후 기다림. 2.바닥 고정이 안 될 경우 오차가 심하다) 해결방법: 3d프린트로 하부 지지대, 상부지지대 설계 및 인쇄.
- ☐ 드랍 카운트 센서 위치 선정 + 고정 기구 찾기 : 고정기구없음 3d 프린트.
- ☑ ~~물통 : 기구 바깥에서 물 넣을 수 있게 + 단열 구조 열 설계 할 것임지... : 기구바깥으로 물통 뺀다.~~
- ☑ ~~도어 : 힌지 위치 선정 : 2개 사용, 기존과 같은 위치~~
- ☑ ~~도어 : 도어스위치 위치 개수 및 위치 선정 : 도어 맨 위에 4개 위치하여 배선 용이하게 함~~
- ☑ ~~도어 : 도어스위치 배선 방법 : 덕트 사용~~
- ☑ ~~UV-LED : 위치 선정 : 원두컵 고정판 밑으로..?~~
- ☑ ~~UV-LED : 배선 방법 : 사각 덕트 15x11로 정리~~
- ☑ ~~LCD : 도어 내부 배선 공간 확보 문제 : 도어 최상단에 위치시켜 위로 뺀다.~~

☑ 프로젝트 2차 : 프리페럴

Hide completed items

Delete

80%

- ☑ ~~스텝모터 : 구동 최소 각도 값 확인 : 28BYJ 48(모터중앙이아닌 옆으로 모터축 있다. 기어드모터이다.)스펙상 32스텝/1회전이나, 기어드모터라서 이론상 32*64 = 2048스텝(1회전당)임. 즉 0.176도/step~~
- ☑ ~~스텝모터 : 드라이브없이 직접 구동으로 할 수 있을지 확인 : 안됨. 달링턴 ULN2003필요. < 달링턴C(up to 500mA, with free wheel diode)~~
- ☑ ~~스텝모터 기본 구동 확인 > 8STATE 구동 1000 1100 0100 0110 0010 0011 0001 1001 > 2048step에 반바퀴 회전 확인 완료~~
- ☑ ~~로드셀 : HX711 드라이버 코드 파서 구동 확인~~
- ☑ ~~로드셀 : 캐패시터 달고 값 흔들림 재 확인 > 값 8340000 수준에서 유지 되 나, 마구 구부렸다가 다시 측정하면 이 기준으로 회복안됨. > 비틀림 회복 문제? 시간이 오래 걸리나 확인~~
- ☐ 로드셀 무게 0~1kg 저울 구매하여 측정값과 비교, 측정 신뢰 최소 단위 확인. -> 볼트 고정해도 지지대가 불안정-> 3D설계, 프린팅. -> 10g단위 측정 확보.
- ☑ ~~UV-LEC : 구동 회로 따로 필요할지 확인 < 소모전류 30mA. 필요없음~~
- ☑ ~~IR : 리모콘 버튼 인식 코드 수정 : 리모콘 시그널 비트화 & 저장 > 시그널 비트화 하면서 어떤키눌렀는지를 저장 > 코드작성 완료.~~
- ☐ IR : 리모콘 1 버튼 인식 -> 17개 버튼 인식 코드 수정 : 테스트 하기 (목표 구성대로 테스트는 완료. 개선필요.개선점1:버튼 다른것으로인식->대조 비트 수 늘리기, 개선점2: bitbanging interval값조정 혹은 정확한타이머 도움 받기) 개선점3: 논리연산->비트연산으로 수정
- ☑ ~~로드셀 : HX711에 2CH 연결해서 각각 값 받을 수 있는지 확인 : PD_SCK에 주는 펄스 수로 제어가능하다. PULSE 25: CH1(gain 128), PULSE 26: CH2, PULSE 27(gain 32) : CH1(gain 64)~~

기구

: 3D 설계 및 프린팅.

프리페럴

: IR 버튼17개 구현, 밸브 스텝모터, 로드셀 구현

[WEEK 3 : 상세 진행 상황, 문제 & 해결방안]

발생문제 1

물 밸브 미세 각도 조정
모터, 샤프트 필요



해결방안

샤프트 : 3D 프린트 및 테스트

모터 : 스텝 모터로 각도 제어 및 확인



Half Step Sequence 코드로
4096 Steps/Rev 확인 완료
→ 0.087%/step 확인 완료

```
void rtNotification(rtiBASE *rtireg, uint32 notification)
{
    total++;
    gpioSetBit(gioPORTB, 6, 1);

    if(total >= 2048) // 2번 바퀴 돌았다.
    {
        gpioSetBit(gioPORTB, 6, 0);
        cnt = 0;
    }
    else{
        gpioSetBit(gioPORTB, 6, 1);
    }

    switch(cnt) {
        case 0: gpioSetBit(gioPORTA, 0, 1);
                gpioSetBit(gioPORTA, 1, 0);
                gpioSetBit(gioPORTA, 2, 0);
                gpioSetBit(gioPORTA, 5, 0);
                cnt = 1;
                break;
        case 1: gpioSetBit(gioPORTA, 0, 1);
                gpioSetBit(gioPORTA, 1, 1);
                gpioSetBit(gioPORTA, 2, 0);
                gpioSetBit(gioPORTA, 5, 0);
                cnt = 2;
                break;
        case 2: gpioSetBit(gioPORTA, 0, 0);
                gpioSetBit(gioPORTA, 1, 1);
                gpioSetBit(gioPORTA, 2, 0);
                gpioSetBit(gioPORTA, 5, 0);
                cnt = 3;
                break;
        case 3: gpioSetBit(gioPORTA, 0, 0);
                gpioSetBit(gioPORTA, 1, 1);
                gpioSetBit(gioPORTA, 2, 1);
                gpioSetBit(gioPORTA, 5, 0);
                cnt = 4;
                break;
        case 4: gpioSetBit(gioPORTA, 0, 0);
                gpioSetBit(gioPORTA, 1, 0);
                gpioSetBit(gioPORTA, 2, 1);
                gpioSetBit(gioPORTA, 5, 0);
                cnt = 5;
                break;
        case 5: gpioSetBit(gioPORTA, 0, 0);
                gpioSetBit(gioPORTA, 1, 0);
                gpioSetBit(gioPORTA, 2, 1);
                gpioSetBit(gioPORTA, 5, 1);
                cnt = 6;
                break;
        case 6: gpioSetBit(gioPORTA, 0, 0);
                gpioSetBit(gioPORTA, 1, 0);
                gpioSetBit(gioPORTA, 2, 0);
                gpioSetBit(gioPORTA, 5, 1);
                cnt = 7;
                break;
        case 7: gpioSetBit(gioPORTA, 0, 1);
                gpioSetBit(gioPORTA, 1, 0);
                gpioSetBit(gioPORTA, 2, 0);
                gpioSetBit(gioPORTA, 5, 1);
                cnt = 0;
                break;
    }
}
```

[WEEK 3 : 상세 진행 상황, 문제 & 해결방안]

발생문제 2

IR 리모콘 버튼 인식
버튼 1개만 인식할 수 있음

< 버튼 신호 배열 분류 >

IR 버튼 분류 < 첫글자 0xd7 >

1	2	3	4	5	6	7	8	
0xd7	0xd7	0xd7	0xd5	0xd5	0xd5			letter 7
0xd7	0xd7	0xd7	0xd5	0xd5	0xd5			letter 3
0xd7	0xd7	0xd5						letter 6
0xd7	0xd5	0xdd	0xdd					letter sharp
0xd7	0xd5	0xdd	0xd5					letter 1
0xd7	0xd5	0xdd	0xd7					letter 8
0xd7	0xd5	0xd7	0xd5					letter 9
0xd7	0xd5	0xd7	0xd7					letter 0

```
uint8 letter1[12]={0xd7,0xd5,0xdD,0xd5,0xd7,0xd5,0xd7,0xd5,0xdD,0xdD,0xdD,0xd7};
uint8 letter3[12]={0xd7,0xd7,0xd7,0xd5,0xd5,0xdd,0xd5,0xd5,0xdD,0xdD,0xdD,0xd7};
uint8 letter6[12]={0xd7,0xd7,0xd5,0xd5,0xd7,0xd5,0xd5,0xdd,0xdD,0xdD,0xdD,0xd7};
uint8 letter7[12]={0xd7,0xd7,0xd7,0xd5,0xd5,0xd5,0xd7,0xd7,0xd7,0xd7,0xd7,0xd7};
uint8 letter8[12]={0xd7,0xd5,0xdD,0xd7,0xd5,0xd5,0xd7,0xd5,0xdD,0xd7,0xd7,0xd7};
uint8 letter9[12]={0xd7,0xd5,0xd7,0xd5,0xd5,0xd5,0xdd,0xdd,0xd7,0xd7,0xd7,0xd7};
uint8 letter0[12]={0xd7,0xd5,0xd7,0xd7,0xd5,0xd5,0xd7,0xd7,0xd5,0xd7,0xd7,0xd7};
uint8 lettersharp[12]={0xd7,0xd5,0xdD,0xdd,0xd5,0xd5,0xd7,0xd5,0xd7,0xd7,0xd7,0xd7};
```

IR 버튼 분류 < 첫글자 0xd5 >

1	2	3	4	5	6	7	8	
0xd5	0xdd	0xdd	0xd7					letter star
0xd5	0xdd	0xdd	0xd5					letter 2
0xd5	0xdd	0xd7						letter right
0xd5	0xdd	0xd5						letter down
0xd5	0xd7	0xd7						letter ok
0xd5	0xd7	0xd5						letter 4
0xd5	0xd5	0xdd	0xdd					letter up
0xd5	0xd5	0xdd	0xd5					letter left
0xd5	0xd5	0xd5	0xd7					letter 5

```
uint8 letter2[12]={0xd5,0xdd,0xdd,0xd5,0xd7,0xd5,0xdd,0xd5,0xdD,0xdD,0xdD,0xd7};
uint8 letter4[12]={0xd5,0xd7,0xd5,0xd5,0xdd,0xd7,0xd5,0xdD,0xdD,0xdD,0xdD,0xd7};
uint8 letter5[12]={0xd5,0xd5,0xd5,0xd7,0xd5,0xdd,0xdd,0xdd,0xdD,0xdD,0xdD,0xd7};
uint8 letterstar[12]={0xd5,0xdd,0xdD,0xd7,0xd5,0xd5,0xdd,0xd5,0xdD,0xd7,0xd7,0xd7};
uint8 letterok[12]={0xd5,0xd7,0xd7,0xd7,0xd5,0xd5,0xdd,0xdd,0xd5,0xd7,0xd7,0xd7};
uint8 letterleft[12]={0xd5,0xd5,0xdD,0xd5,0xd5,0xdd,0xdd,0xdd,0xd7,0xd7,0xd7,0xd7};
uint8 letterright[12]={0xd5,0xdd,0xd7,0xd7,0xd5,0xd,0xd7,0xd5,0xdD,0xd5,0xdD,0xd7};
uint8 letterup[12]={0xd5,0xd5,0xdD,0xdd,0xd5,0xd7,0xd7,0xd5,0xd7,0xd7,0xd7,0xd7};
uint8 letterdown[12]={0xd5,0xdd,0xd5,0xdd,0xd7,0xd5,0xdd,0xd7,0xd7,0xd5,0xdD,0xd7};
```

해결방안

버튼 비트 맵 만들고 다른 버튼 부분만 식별하여 버튼 17개 모두 인식시킨다.

* 추가 개선 방안 : 배열, 논리연산 없애고 비트 연산으로 교체

< 코드 >

```
switch (letter[0]){
case 0xd7 :

    switch (letter[1])
    {
        case 0xd7 :

            if(letter[5]==0xd5) remote_now=7;
            else remote_now=3;
            if(letter[2]==0xd5) remote_now=6;
            break;

        case 0xd5 :

            if(letter[2]==0xdd)
            {
                switch (letter[3])
                {
                    case 0xdd : remote_now='h'; // sharp
                    break;
                    case 0xd5 : remote_now=1;
                    break;
                    case 0xd7 : remote_now=8;
                    break;
                }
            }
            else
            {
                switch (letter[3])
                {
                    case 0xd5 : remote_now=9;
                    break;
                    case 0xd7 : remote_now=0;
                    break;
                }
            }
            break;
    }
}

case 0xd5 :
    switch (letter[1])
    {
        case 0xdd :
            switch(letter[2])
            {
                case 0xdd :
                    if(letter[3]==0xd7)
                    {
                        remote_now='s'; //s=star;
                        remote_now=2;
                    }
                    else
                    {
                        case 0xd7 : remote_now='r'; //r=right
                        break;
                        case 0xd5 : remote_now='d'; //d=down
                        break;
                    }
                }
            }
            break;

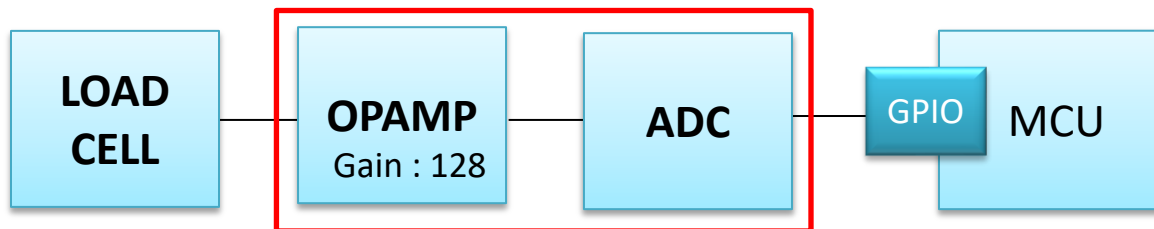
        case 0xd7 :
            if(letter[2]==0xd7) remote_now='k'; //o=ok
            else remote_now=4;
            break;

        case 0xd5 :
            switch (letter[3])
            {
                case 0xdd : remote_now='u'; //u= up
                break;
                case 0xd5 : remote_now='l'; //l= left
                break;
                case 0xd7 : remote_now=5;
                break;
            }
            break;
    }
}
```

[WEEK 3 : 상세 진행 상황, 문제 & 해결방안]

무게 - 측정값

실물 질량과 비교



HX711

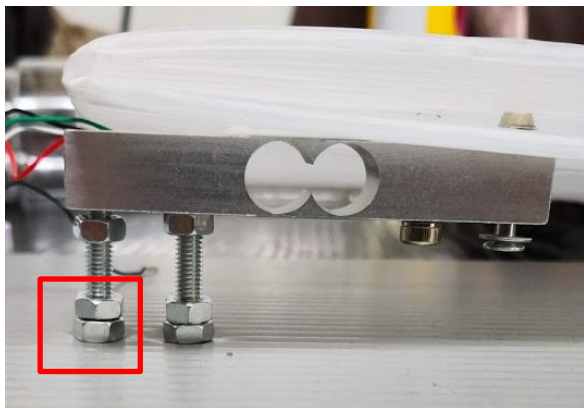
0.0033 V(1kg) ---- 0.4224 V----- 24bit adc ----- 0x128A47

$0.42V : 3.3V = 0x128A47 : 0xFFFFFFFF$

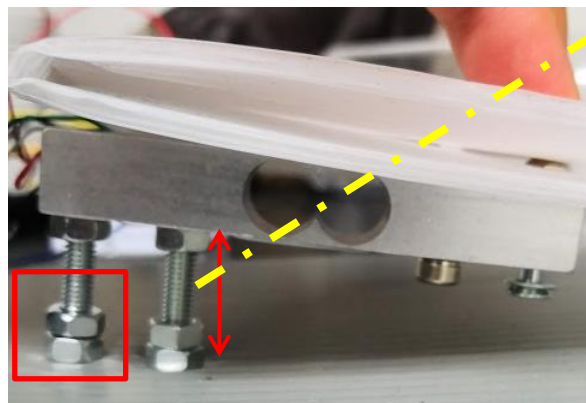
```
/dev/ttyACM0 - P011
x711 adc result = 1216038
x711 adc result = 1215047
x711 adc result = 1219896
x711 adc result = 1234889
x711 adc result = 1236896
x711 adc result = 1252751
x711 adc result = 1259538
x711 adc result = 1247108
x711 adc result = 1125218
x711 adc result = 341955
x711 adc result = 111218
x711 adc result = 110801
x711 adc result = 110746
```

발생문제 3

지지대 들림에 따른 센싱 불안정



< 무게 인가 전 >



< 무게 인가 후 >

해결방안

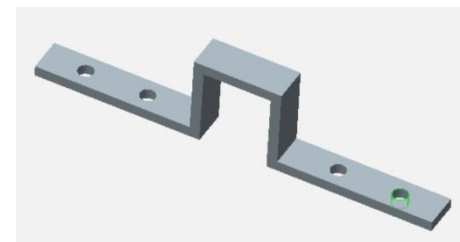
기구 설계

(바닥판 PLA 소재 8mm 로 제작)

(볼트 높이 10mm이하로)

(고정대 기구물 추가 제작)

3D 프린팅



[WEEK 3 : 차주 목표]

* Ethernet 코딩 : MCU <-> DSP 소켓

☒ 프로젝트 2차 : Ethernet lwIP Hide completed items Delete

67%

☒ ~~ccs lwIP 예제 찾기~~

☒ Ethernet 기본 개념, (OSI, TCP/IP)

☐ ccs tutorial code 실행 해 보기 : 동작하지 않음. -> halcogen 수정하여 다시 수행해 볼 것. -> DHCP IP 받아오기 완료. -> freeRTOS 예제 코드도 테스트

Add an item

☒ 프로젝트 2차 : freeRTOS & HET Delete

0%

☐ HET : TIMER 로 쓸 수 있게 해보기. PWM핸들러를 RTI핸들러 처럼 구성해보기. -> r5 manual -- HET -- CNT 조정, 기본주파수 조정. 테스트할 예정

Add an item

* 기구 3D 설계 및 프린팅

* 센서 조립, 및 코드 다듬기

* FreeRTOS로 코드 이식