

# 자율주행 프로젝트

1조

2015111905 배운호

2014111918 김정재

**01 하드웨어**

---

**02 Flow Chart**

---

**03 주요 포인트**

---

**04 장-단점**

---

**05 시도해본 코드**

---

**06 완성 코드**

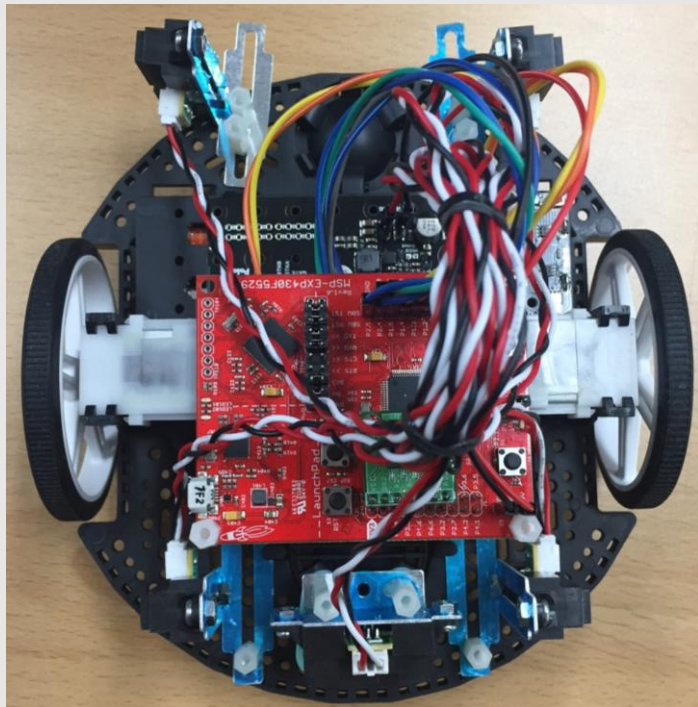
---



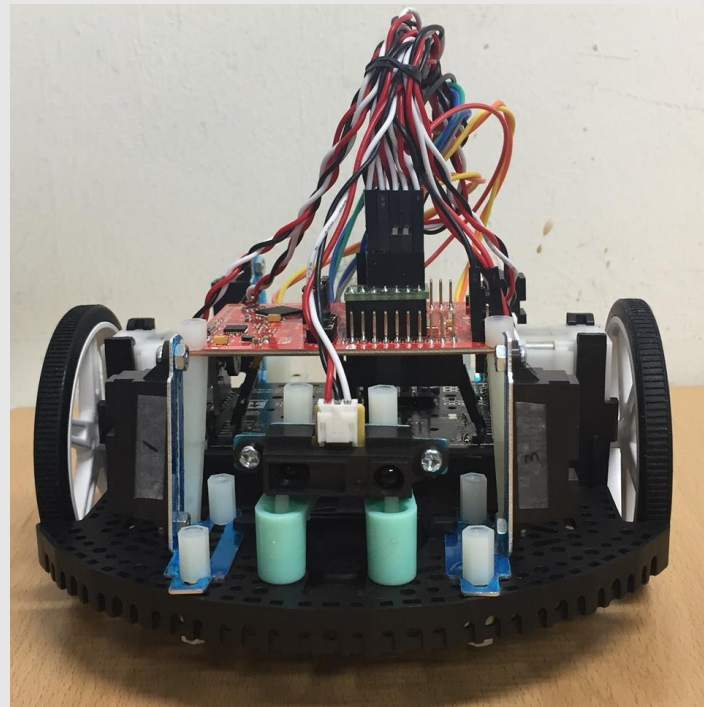
# 01

## 하드웨어

# 1. 하드웨어



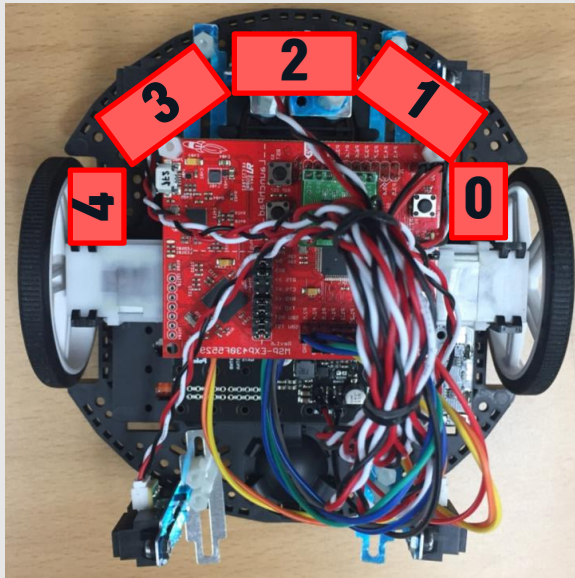
〈 위에서 내려다 본 차체 〉



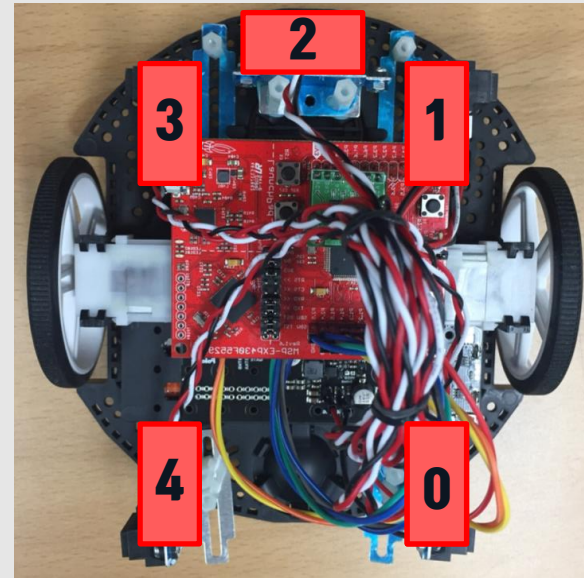
〈 정면에서 바라 본 차체 〉



# 1. 하드웨어



**차체 전방에 거리센서 5개 배치**



**차체 전방에 거리센서 3개 배치  
후방에 거리센서 2개 배치**

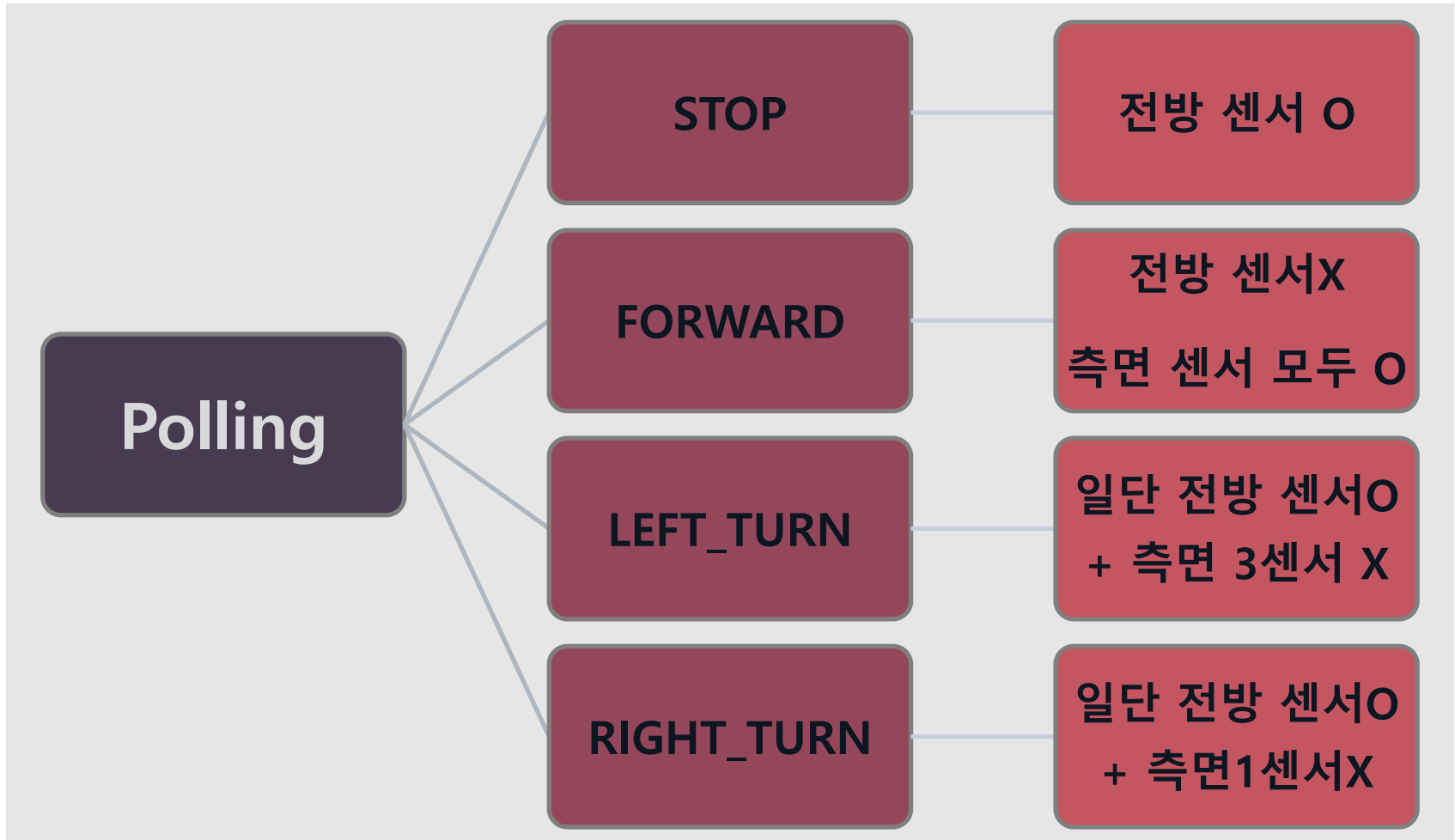


# **02**

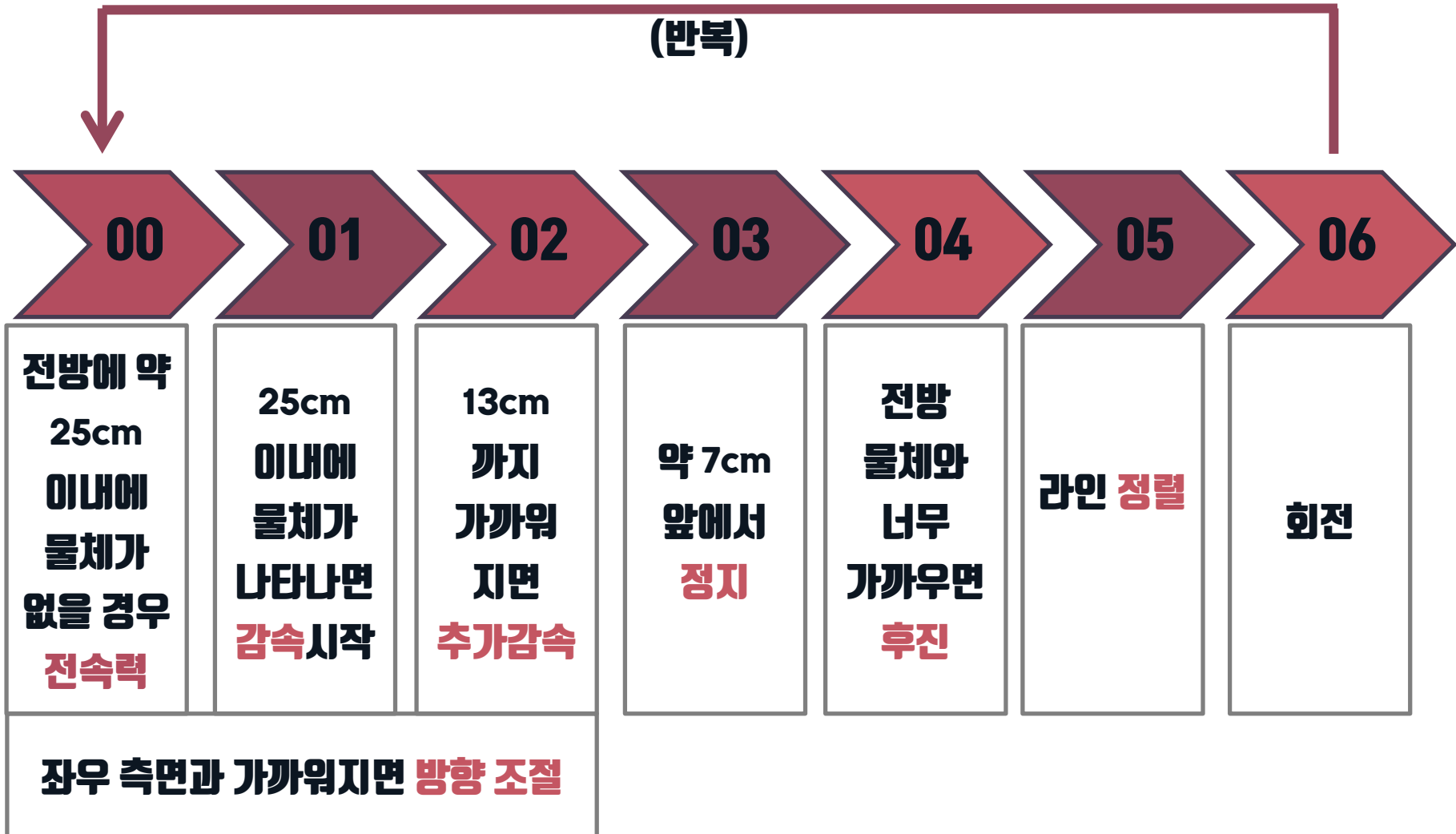
## **Flow Chart**

## 2. FLOW CHART

### ❖ 기존 알고리즘



## 2. FLOW CHART







**03**

**주요 포인트**

### 3. 주요포인트

❖ LINE UP 함수

❖ 가속 함수

❖ Timer Interrupt

❖ Sensor 값 구체화 - **matlab** 사용



# 3. 주요포인트

## ❖ LINE UP 함수

## L-lineup(void)

```
498         }
499         P2OUT |= BIT0; //right DIR reverse
500     }
501     else
502     {
503         P2OUT &= ~BIT0; //right DIR forward
504         TA2CCR1 = 0;
505         TA2CCR2 = 0;
506         break;
507     }
508 }
509 }
510
511 else if(ADC12MEM3<(S[3]-100)) //왼쪽 앞쪽이 벽이랑 멀다.
512 {
513     P2OUT |= BIT2; //left DIR reverse
514     while(1)
515     {
516         if(ADC12MEM4>(S[4]+100)) //왼쪽 뒷쪽이 벽이랑 가깝다.
517         {
518             TA2CCR1 = 800;
519             TA2CCR2 = 400;
520         }
521         else if(ADC12MEM4<(S[4]-100)) //왼쪽 뒷벽도 벽이랑 멀다.
522         {
523             P2OUT &= ~BIT2; //left DIR forward
524             while(1)
525             {
526                 if(ADC12MEM3<S[3]-100) //왼쪽 앞쪽이 기준값에 가까워 질때까지 오른쪽 바퀴만 회전
527                 {
528                     TA2CCR2 = 800;
529                 }
530                 else
531                 {
```

### 3. 주요포인트

#### ❖ LINE UP 함수

#### L-lineup(void)

```
533     }
534     }
535     while(1)
536     {
537         if(ADC12MEM4<S[4]-100) // 왼쪽 뒷쪽이 기준값에 가까워 질 때까지 왼쪽 바퀴만 회전
538         {
539             TA2CCR1 = 800;
540         }
541         else
542         {
543             break;
544         }
545     }
546     P2OUT |= BIT2; //left DIR reverse
547 }
548 else if(ADC12MEM1>(S[1]+100))
549 {
550     TA2CCR1 = 400;
551     TA2CCR2 = 800;
552 }
553 else
554 {
555     TA2CCR1 = 0;
556     TA2CCR2 = 0;
557     P2OUT &= ~BIT2; //left DIR forward
558     break;
559 }
560 }
561 }
562 }
```



### 3. 주요포인트

#### ❖가속 함수를 이용한 가속 구현

```
509 void accel(int L, int R)
510 {
511     if(L>now_pwm_L)
512     {
513         now_pwm_L += 3;
514         TA2CCR1=now_pwm_L;
515     }
516     else if(L<now_pwm_L)
517     {
518         now_pwm_L -= 3;
519         TA2CCR1=now_pwm_L;
520     }
521     else
522     {
523         TA2CCR1 = now_pwm_L;
524     }
526     if(R>now_pwm_R)
527     {
528         now_pwm_R += 3;
529         TA2CCR2=now_pwm_R;
530     }
531     else if(R<now_pwm_R)
532     {
533         now_pwm_R -= 3;
534         TA2CCR2=now_pwm_R;
535     }
536     else
537     {
538         TA2CCR2 = now_pwm_R;
539     }
540 }
```

**+3 씩 값을 조정해 주어 부드러운 가속이 가능하게끔 구현**

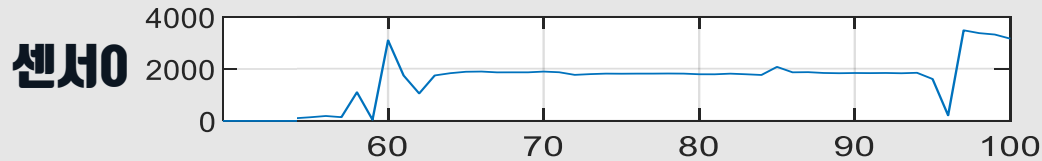
### 3. 주요포인트

#### ❖ Sensor값 구체화

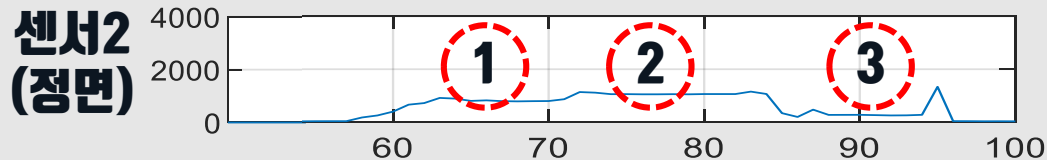
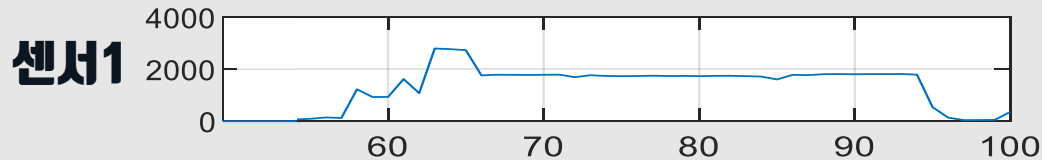
- ❖ 정지 시 정면 거리 센서 값 측정 및 보정
- ❖ 직진 시 자세 보정\_Sensor값 구체화
- ❖ 거리 센서 값 구체화 \_matlab이용

### 3. 주요포인트

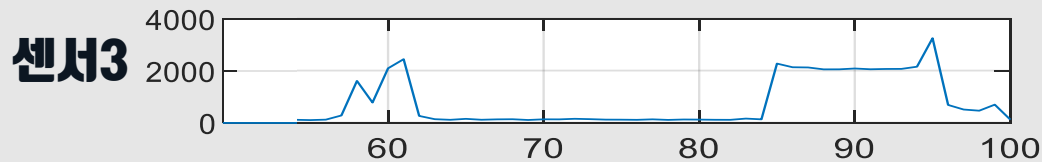
#### ❖ 정지 시 정면 거리 센서 값 측정\_Sensor값 구체화



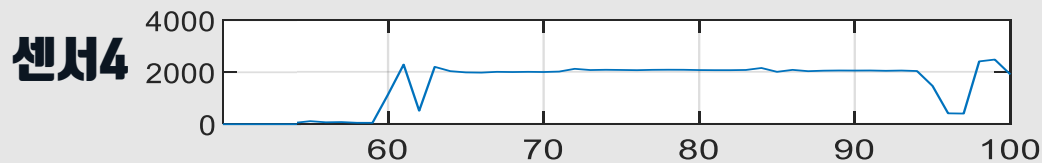
① 차체가 벽면  
약 **24cm** 앞에  
위치한 경우



② 차체가 벽면  
약 **21cm** 앞에  
위치한 경우



③ 차체가 벽면에서  
**30cm 이상**으로  
멀어진 경우



### 3. 주요포인트

#### ❖ 정지 시 정면 거리 센서 값 측정\_Sensor값 구체화

##### < Matlab+excel 이용\_sensor값 추출 >

65	66	67	68	69
1888	1856	1857	1857	1886
1748	1774	1770	1767	1772
821	792	783	792	795
106	121	126	97	123
1969	1999	1991	1999	1992
	경우1	790.5		

75	76	77	78	79
1808	1808	1812	1808	1783
1723	1736	1727	1729	1719
1051	1051	1056	1051	1060
103	122	100	116	112
2063	2076	2080	2078	2067
			경우2	1054.6

88	89	90	91	92
1823	1831	1827	1833	1822
1799	1793	1799	1798	1802
272	274	262	250	255
2048	2080	2051	2062	2064
2051	2048	2051	2039	2049
			경우3	264

#### ① 차체가 장애물

약 24cm 앞에  
위치한 경우

790.5

#### ② 차체가 장애물

약 21cm 앞에  
위치한 경우

1054.6

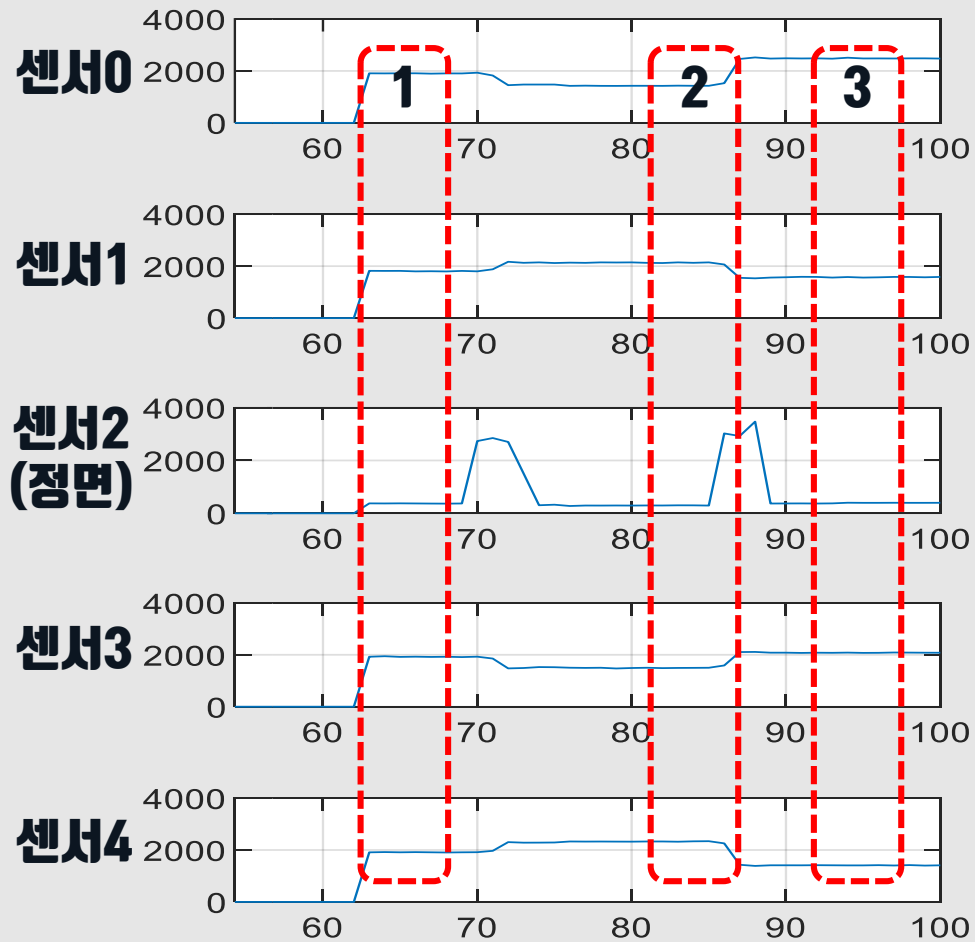
#### ③ 차체가 벽면

30cm이상 으로  
멀어진 경우

264

### 3. 주요포인트

#### ❖ 직진 시 자세 보정\_Sensor값 구체화



① 차체가  
일직선으로  
놓여진 경우

② 차체가  
무측으로  
틀어진 경우

③ 차체가  
좌측으로  
틀어진 경우



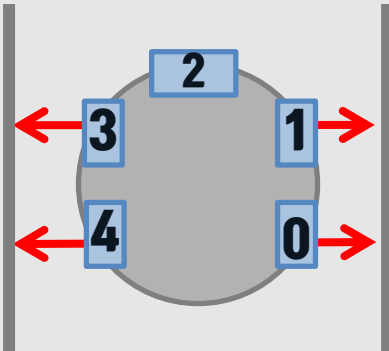
### 3. 주요포인트

#### ❖ 직진 시 자세 보정\_Sensor값 구체화

< Matlab이용\_sensor값 추출 >

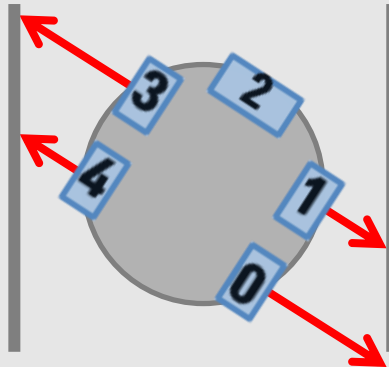
66	67	68
1895	1904	1903
1803	1795	1816
369	366	371
1917	1923	1911
1911	1904	1911

① 차체가 **일직선으로**  
놓여진 경우



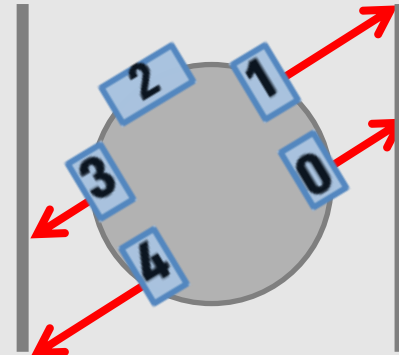
83	84	85
1427	1427	1528
2128	2144	2063
300	293	323
1496	1500	1588
2334	2339	2255

② 차체가 **우측으로**  
틀어진 경우



93	94	95
2508	2480	2481
1580	1559	1569
400	395	395
2083	2073	2076
1407	1407	1416

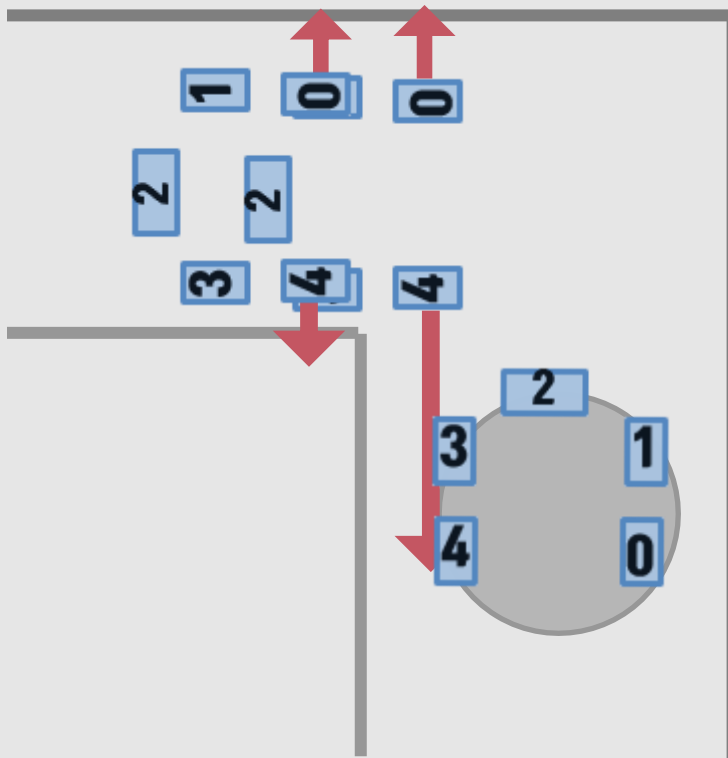
③ 차체가 **좌측으로**  
틀어진 경우



### 3. 주요포인트

#### ❖ 회전 후 자세 보정\_(후면 거리 센서 부착 이유)

〈 후면 거리 센서 부착 이유 〉



차체 **회전** 후,  
4번 거리 센서가  
**평행** 값을 찾은 이후  
주행 시작

### 3. 주요포인트

#### ❖ 거리센서 값 구체화 \_matlab이용

```
%% 데이터 실시간 그래프
close all, clc
fopen(s);

N = 100;
sensor = zeros(5,N);

while 1
    fwrite(s,255,'char');      data = fread(s,1,'char');

    if(data == 255)
        data = fread(s,10,'char'); %10개 데이터를 받는다.
        sensor(1,N) = data(1)*256 + data(2);
        sensor(2,N) = data(3)*256 + data(4);
        sensor(3,N) = data(5)*256 + data(6);
        sensor(4,N) = data(7)*256 + data(8);
        sensor(5,N) = data(9)*256 + data(10);
    else
        disp('에러?');
    end
end
```



### 3. 주요포인트

#### ❖ 거리센서 값 구체화 \_matlab이용

##### < Matlab이용\_sensor값 추출 >

92	93	94	95	96	97	98	99	100
1976	1966	1976	1978	1967	1952	1980	1974	1974
1934	1939	1948	1948	1946	1935	1944	1935	1935
546	563	566	563	571	567	567	564	564
1895	1887	1878	1904	1879	1872	1883	1878	1878
1876	1870	1876	1835	1843	1859	1859	1856	1856

##### < excel이용\_sensor값 평균값 추출 >

1976	1978	1967	1952	1980	1974	1974	1983	오른쪽 뒤
1948	1948	1946	1935	1944	1935	1935	1938	오른쪽 앞
566	563	571	567	567	564	564	624	정면
1878	1904	1879	1872	1883	1878	1878	1879	왼쪽 앞
1876	1835	1843	1859	1859	1856	1856	1951	왼쪽 뒤



**04**

**장 - 단점**



## 4. 장단점

**장점**

**정밀한 자세 보정이 이루어 지고,  
Timer Interrupt 로  
엔코더 값을 비교해서  
pwm값을 추가 보정하였다.  
Accel 함수를 이용해서  
가/감속을 만들었다.**

## 4. 장단점



**단점**

**P제어가 이루어 지지 않아  
부드러운 가/감속에  
어려움이 있다.  
주행 중 자세보정으로 인한  
속도 저하가 존재한다.**



**05**

**시도해본 코드**

# 5. 시도해본 코드

## ❖ P제어

```
want_L = L;  
want_R = R;  
  
if(want_L > L0)    → 가속 코드  
{  
    now_pwm_L = now_pwm_L + (want_L-L0)/5; // (want_velo + velo)/5  
    TA2CCR1 = now_pwm_L;  
    if(now_pwm_R >= 2000)  
    {  
        TA2CCR1 = 2000;  
    }  
}  
else if(want_L < L0) → 감속 코드  
{  
    now_pwm_L = now_pwm_L - (L0-want_L)/5;  
    TA2CCR1 = now_pwm_L;  
    if(now_pwm_L <= 50)  
    {  
        TA2CCR1 = 0;  
    }  
}  
else  
{  
    TA2CCR1 = now_pwm_L;  
}
```

**Kp=500**

**제어량**  
**= Kp \* (목표속도 - 현재속도)**  
**Kp : 비례상수**

## 5. 시도해본 코드

### ❖ 센서 값의 유효범위 설정

타이머 인터럽트로 센서 값 메모리를 읽어와서 변수에 저장  
해놓고, 센서 메모리 값이 특정 값 (4센치) 이하가 되면  
(너무 가까우면 센서 값에 오류가 생기는 점 보완하기 위해)  
센서 변수 값에 메모리 값이 아닌 고정된 상수를 입력해  
사용 그리고 다시 4센치 이상 떨어졌을 때는 센서변수에  
센서 메모리 값 입력

**ADC 값을 를 반복 순차로 읽음**

→ 기대했던 반응 속도보다 떨어져서  
사용하지 못함





**06**

**완성 코드**

## 6. 완성코드

### ❖ 주행 전, 거리 센서 값 측정

```
void main(void)
{
    int i;
    WDTCTL = WDTPW | WDTHOLD; // stop watchdog timer
    switchset(); //switch no = 1.1/2.1
    motorset(); //motor setting
    pwmset(); //pwm setting
    adcset(); //ADC setting
    timerieset(); //timer interrupt setting
    d_inputset(); //load ADC
    encoderset(); //encoder setting
    ledset(); //led setting for logical debugging
    __bis_SR_register(GIE); //global interrupt en
```

```
S[0] = ADC12MEM0;
S[1] = ADC12MEM1;
S[2] = ADC12MEM2;
S[3] = ADC12MEM3;
S[4] = ADC12MEM4;
```

**주행 전 거리 센서 5개  
모두 값을 측정**

## 6. 완성코드

### ❖ 주행 중 직진, LINE UP 함수 적용 L-lineup(void)

```
if(ADC12MEM3 <= S[3]-1000) //왼쪽이 비어있을 경우
{
    L_turn();
    R_lineup();
}
else if(ADC12MEM1 <= S[1]-1000) //오른쪽이 비어있을 경우
{
    R_turn();
    L_lineup();
}
else if((ADC12MEM1 > (S[1]-1000)) && (ADC12MEM3 > (S[3]-1000))) //비어있는 곳이 없을 경우
{
    TA2CCR1 = 0;
    TA2CCR2 = 0;
    __bis_SR_register(LPM0);
}
```

## 6. 완성코드

### ❖ 주행 중 직진, ACCEL 함수 적용

```
352 menu = 1;
353 if(ADC12MEM1>(S[1]+100)) //오른쪽 가까워짐
354 {
355     if(ADC12MEM1>(S[1]+200))
356     {
357         accel(3000,3400);
358     }
359     else
360     {
361         accel(3000,3200);
362     }
363 }
364 else if(ADC12MEM3 > (S[3]+100)) //왼쪽 가까워짐
365 {
366     if(ADC12MEM3 > (S[3]+200))
367     {
368         accel(3400,3000);
369     }
370     else
371     {
372         accel(3200,3000);
373     }
374 }
375 else
376 {
377     accel(3000,3000); //default
378 }
```

L,R 값 지정하여  
적용

## 6. 완성코드

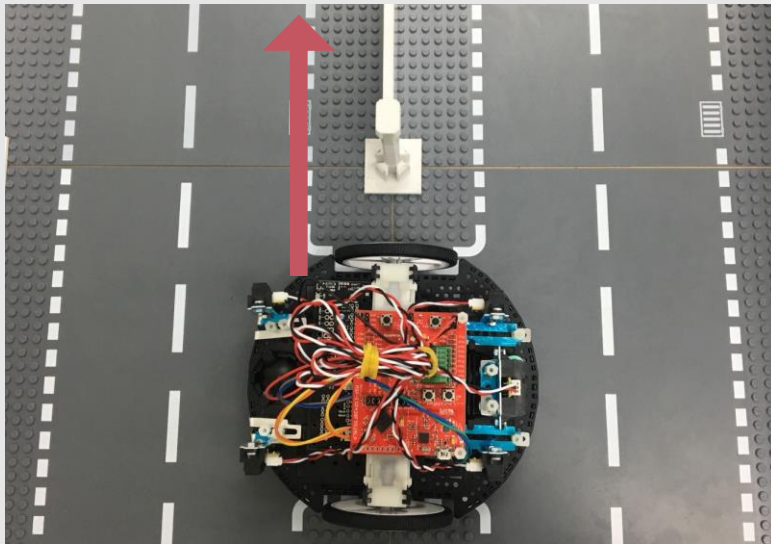
### ❖ 주행 중 회전, ENCODER 이용

```
287 void L_turn(void)
288 {
289     P2OUT |= BIT2; //left DIR reverse
290     R_count = 0;
291     while(1)
292     {
293         if(R_count <= 290)
294         {
295             TA2CCR1 = 2500;
296             TA2CCR2 = 2500;
297         }
298         else
299         {
300             TA2CCR1 = 0;
301             TA2CCR2 = 0;
302             P2OUT &= ~BIT2; //left DIR forward
303             break;
304         }
305     }
306 }
```

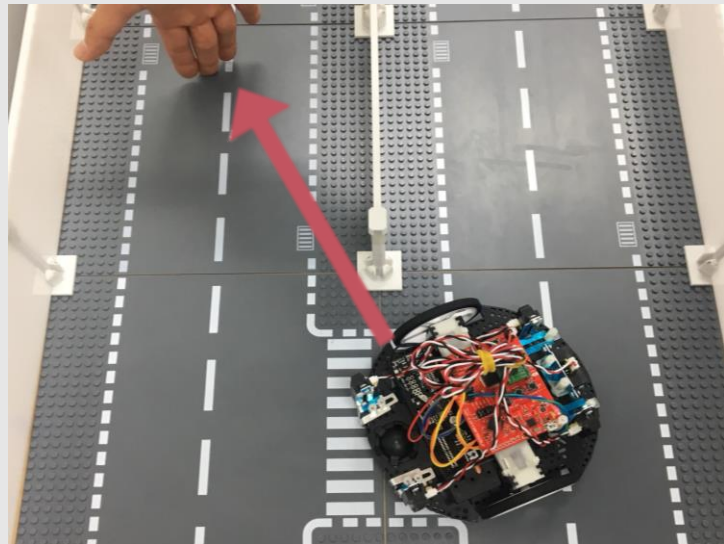
## 6. 완성코드

### ❖ 주행 중 유턴, ENCODER 이용

ADC값 그대로 적용 → ENCODER 이용



**정지상태  
(4번 센서에 쓰레기 값 저장)**



**4번 센서 ADC값에 일정값  
이상이 측정되어야 이동**



## 6. 완성코드

### ❖ 주행 후 정지, LPM모드

```
if(ADC12MEM3 <= S[3]-1000) //왼쪽이 비어있을 경우
{
    L_turn();
    R_lineup();
}
else if(ADC12MEM1 <= S[1]-1000) //오른쪽이 비어있을 경우
{
    R_turn();
    L_lineup();
}
else if((ADC12MEM1 > (S[1]-1000)) && (ADC12MEM3 > (S[3]-1000))) //비어있는 곳이 없을 경우
{
    TA2CCR1 = 0;
    TA2CCR2 = 0;
    __bis_SR_register(LPM0);
}
```

→ LPM모드 적용 (주차)

A 3D wireframe box is centered on the page. Inside the box, the words "Thank" and "you" are stacked vertically in a large, bold, white sans-serif font. The background of the slide is a dark blue-to-red gradient with a subtle pattern of small white dots, resembling a starry night sky.

**Thank  
you**