

# 나만의 작은 세그웨이

최종발표

2023. 06. 15

인하대학교 정보통신공학부

박승재 (12191765)



# 연구 목표

- 주제

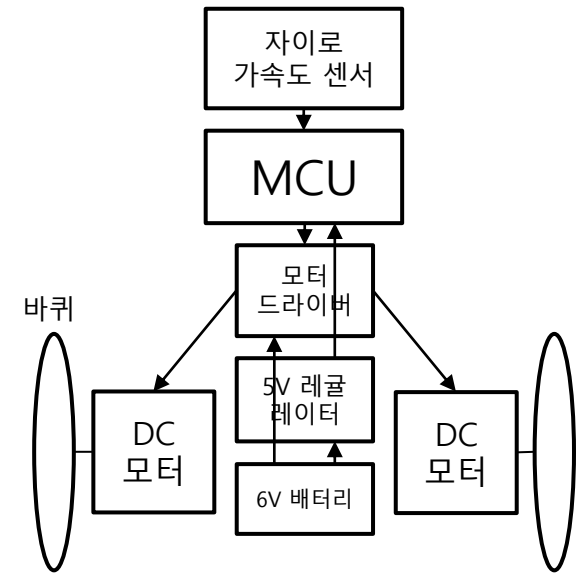
나만의 작은 세그웨이

- 주요 목표

- MCU 및 부품 회로 설계
- 자이로가속도 센서 I2C 통신과 상보필터를 이용한 노이즈 제거
- PID 제어기와 PWM 신호를 이용한 DC모터 회전 제어
- 초음파 센서를 이용한 거리 측정 (TCA 이용)

- 주제 선정 이유

혼자서 하기에는 도전적인 주제이지만, 임베디드 시스템의 핵심적인 개념을 포함하기 때문에 프로젝트 성공과 관계없이 본 과목에서만 얻을 수 있는 지식과 경험을 쌓을 수 있다고 판단했음



# 연구 성과

## 1. MCU 및 부품 회로 설계

OrCAD 이용 (완료)

## 2. 자이로가속도 센서 I2C 통신 + 상보필터를 이용한 노이즈 제거

TWI0 레지스터를 이용해 I2C 통신 코드 작성 (완료)

상보필터를 이용한 노이즈 제거 (완료)

## 3. TCB를 이용한 양방향 PWM 출력

TCB0와 TCB1을 이용해 PWM 출력 생성 (완료)

## 4. PID 제어를 이용한 DC모터 회전 제어

PID 제어 코드는 구현했으나, (완료)

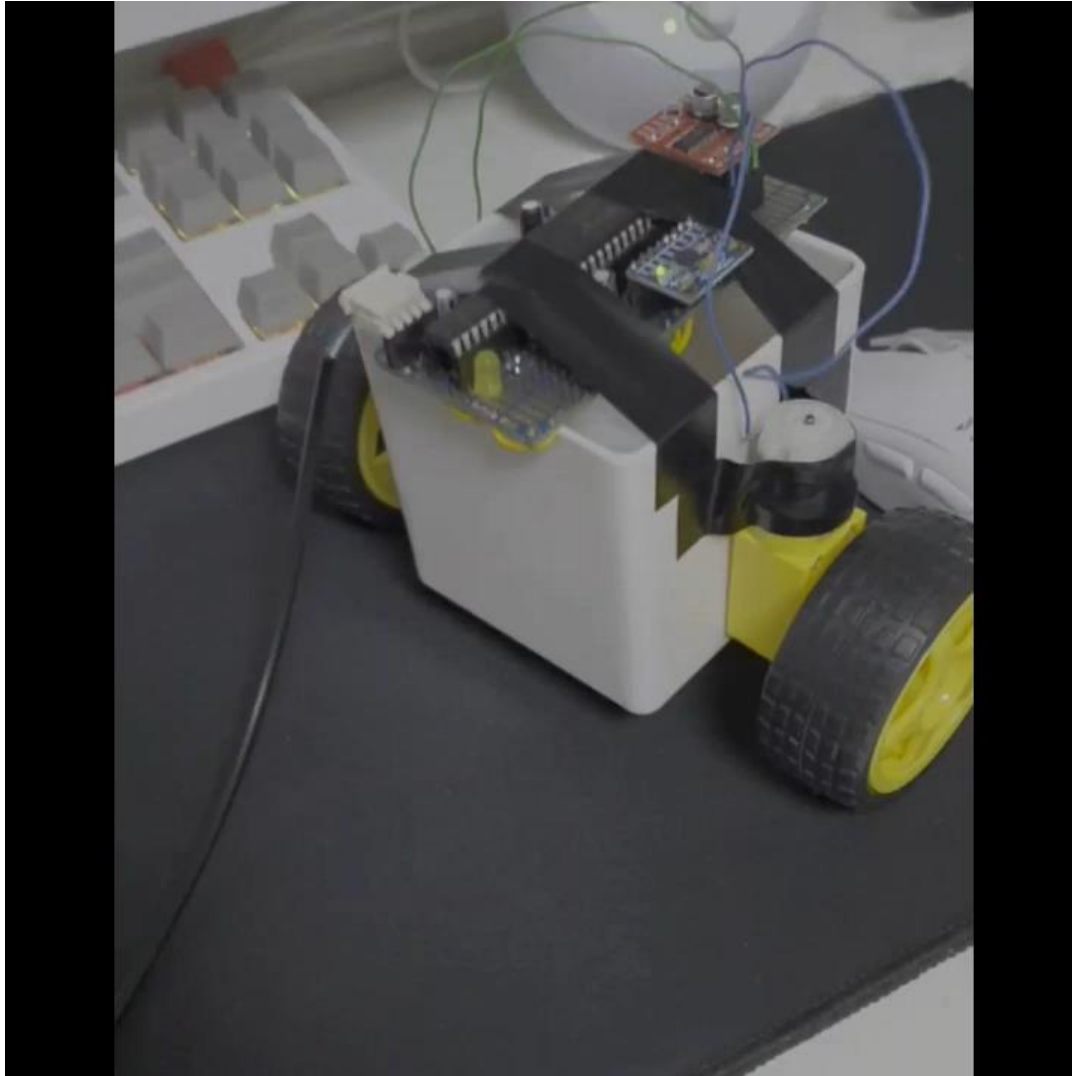
PID 파라미터를 찾지 못해 로봇이 안정적으로 서있지 못함 (진행)

## 5. 초음파 센서를 이용한 거리 측정

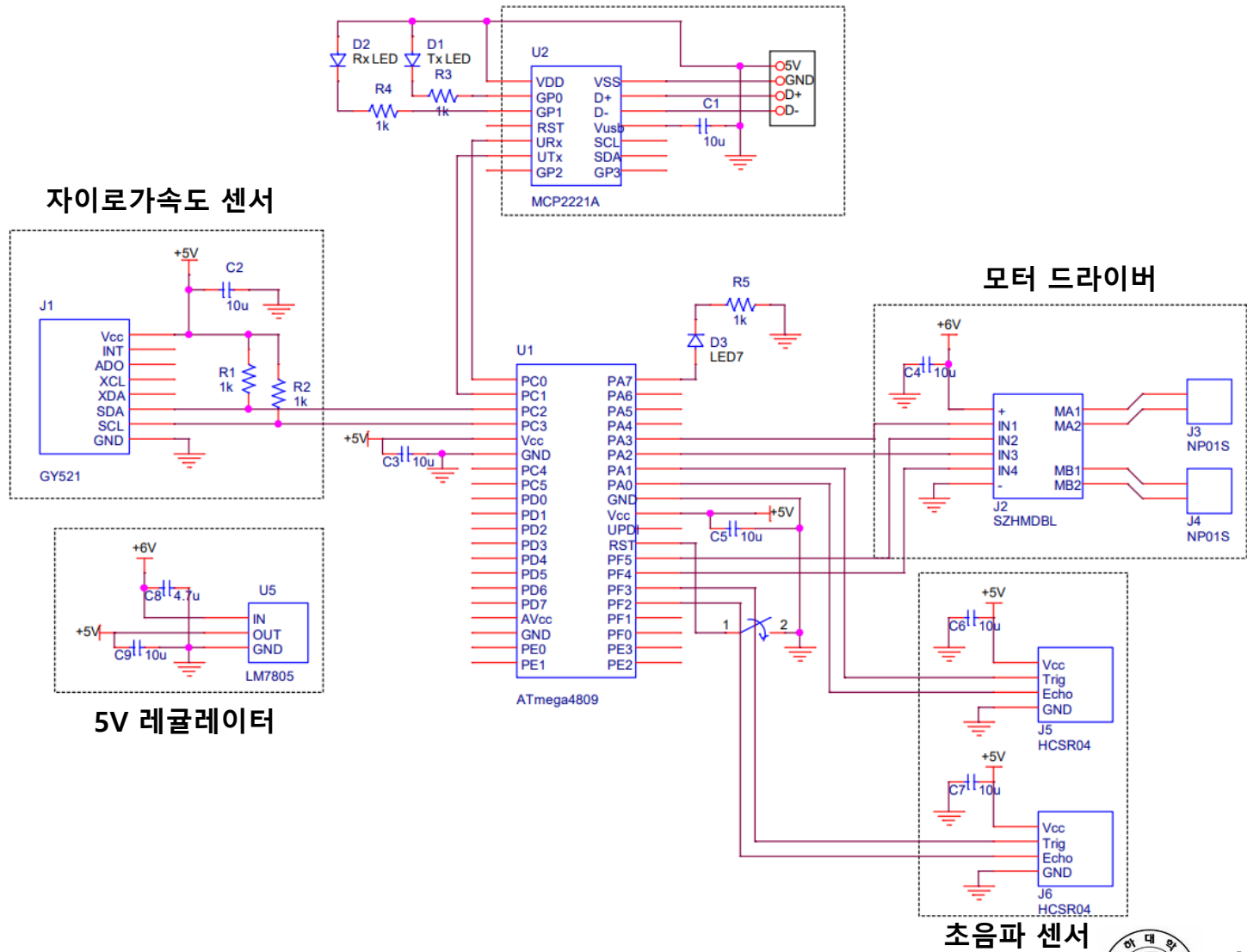
TCA0 이용해 거리 측정 코드는 작성했으나, 초음파 센서 조립 미완료 (진행)



# 연구 성과

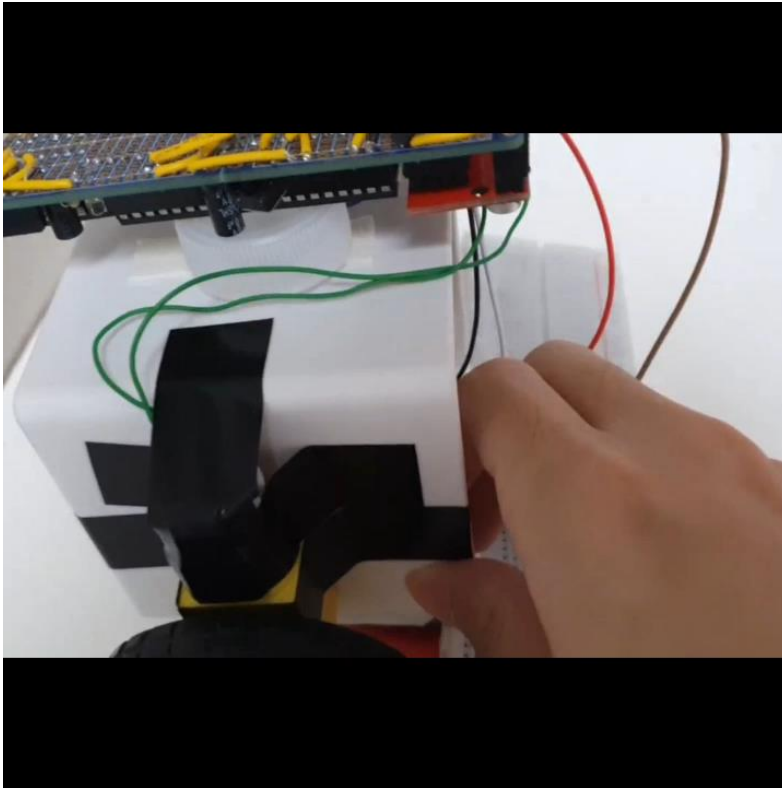


# ① MCU 및 부품 회로 설계



# ① 주요 이슈 1

- 모터로부터의 전압 노이즈가 드라이버에서 증폭되어 시스템 전체에 전파



(1.5V 건전지 4개 직렬연결)

초록색 LED는 전압원(6V)와 GND에 걸리는 전압을 측정하기 위함

전압이 불안정하다는 의미

모터가 작동할 때마다 초록색 LED가 깜빡임  
한 쪽 모터를 분리하면 문제 발생 X

USB 전압원(5V)인 경우에도 문제 발생 X

→ 이번주 화요일에 조교님께 레귤레이터를 받아서 계획에 없던 배터리를 장착했는데,  
이후로 문제가 발생해서 구체적인 원인은 찾지 못함

**모터 드라이버 or 회로 불량으로 추정**



## ② 자이로가속도 센서 I2C 통신

```

91 void _twi_init() {
92     _PORTMUX_TWISPIROUTEA |= _PORTMUX_TWI0_ALT2_bm; // PC2: SDA, PC3: SCL
93     _PORTC_PIN2CTRL |= _PORTC_PIN2CTRL_PULLUPEN_bm; // PC2: PULLUP
94     _PORTC_PIN3CTRL |= _PORTC_PIN3CTRL_PULLUPEN_bm; // PC3: PULLUP
95
96     unsigned int frequency = 400000; // 400kHz I2C 통신 속도는 400kHz
97     unsigned short t_rise = 300; // 300ns
98     unsigned int baud = (F_CPU / frequency - F_CPU / 1000 / 1000 * t_rise / 1000 - 10) / 2;
99     _TWI0_MBAUD = (unsigned char) baud;
100
101     _TWI0_MCTRLA = _TWI_ENABLE_bm;
102     _TWI0_MSTATUS = _TWI_BUSSTATE_IDLE_gc;
103 }
104
105 bool _twi_start(unsigned char device, bool read) {
106     _TWI0_MADDR = device << 1 | (read ? 1 : 0);
107
108     while (!(_TWI0_MSTATUS & (_TWI_WIF_bm | _TWI_RIF_bm)))
109         ;
110
111     if (_TWI0_MSTATUS & _TWI_ARBLOST_bm) {
112         while (!(_TWI0_MSTATUS & _TWI_BUSSTATE_IDLE_gc))
113             ;
114         return false;
115     }
116     if (_TWI0_MSTATUS & _TWI_RXACK_bm) {
117         _TWI0_MCTRLB |= _TWI_MCMD_STOP_gc;
118         while (!(_TWI0_MSTATUS & _TWI_BUSSTATE_IDLE_gc))
119             ;
120         return false;
121     }
122     return true;
123 }

```

```

19 #define _PORTMUX_TWISPIROUTEA (*(volatile unsigned char*) (0x05E0 + 0x03))
20 #define _PORTMUX_TWI0_ALT2_bm ((unsigned char) 0b00100000)
21 #define _PORTMUX_TCBROUTEA (*(volatile unsigned char*) (0x05E0 + 0x05))
22 #define _PORTMUX_TCB1_ALT1_gc ((unsigned char) 0b00000010)
23 #define _PORTMUX_TCB0_ALT1_gc ((unsigned char) 0b00000001)
24
25 #define _TWI0 (*(volatile unsigned char*) 0x08A0)
26 #define _TWI0_MCTRLA (*(volatile unsigned char*) (0x08A0 + 0x03))
27 #define _TWI_ENABLE_bm ((unsigned char) 0b00000001)
28 #define _TWI0_MCTRLB (*(volatile unsigned char*) (0x08A0 + 0x04))
29 #define _TWI_ACKACT_NACK_gc ((unsigned char) 0b00000100)
30 #define _TWI_MCMD_RECVTRANS_gc ((unsigned char) 0b00000010)
31 #define _TWI_MCMD_STOP_gc ((unsigned char) 0b00000011)
32 #define _TWI0_MSTATUS (*(volatile unsigned char*) (0x08A0 + 0x05))
33 #define _TWI_RIF_bm ((unsigned char) 0b10000000)
34 #define _TWI_WIF_bm ((unsigned char) 0b01000000)
35 #define _TWI_RXACK_bm ((unsigned char) 0b00010000)
36 #define _TWI_ARBLOST_bm ((unsigned char) 0b00001000)
37 #define _TWI_BUSERR_bm ((unsigned char) 0b00000100)
38 #define _TWI_BUSSTATE_IDLE_gc ((unsigned char) 0b00000001)
39 #define _TWI0_MBAUD (*(volatile unsigned char*) (0x08A0 + 0x06))
40 #define _TWI0_MADDR (*(volatile unsigned char*) (0x08A0 + 0x07))
41 #define _TWI0_MDATA (*(volatile unsigned char*) (0x08A0 + 0x08))

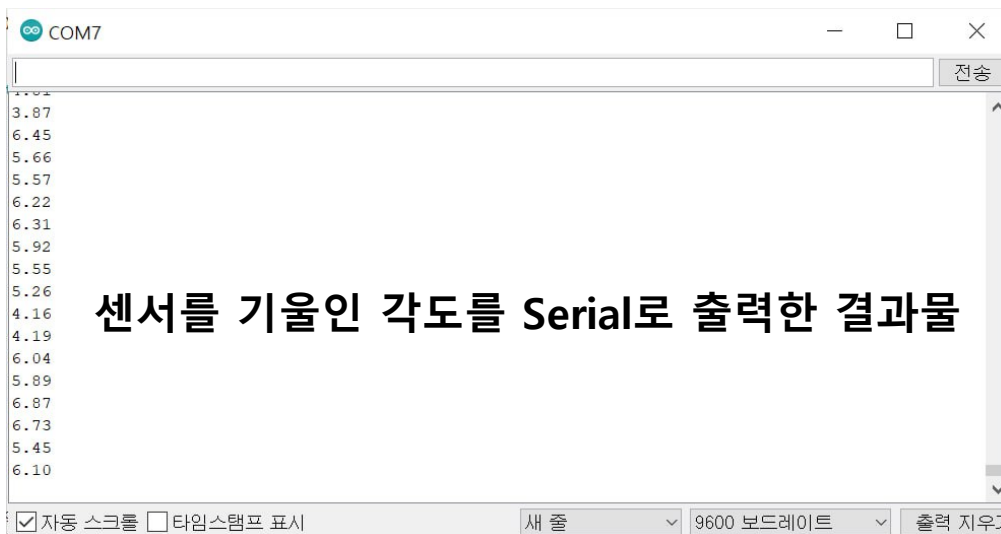
```

데이터시트를 참고해 필요한 레지스터를 volatile 포인터로 정의



## ② 상보필터를 이용한 노이즈 제거

```
71 void loop() {  
72     // fetch sensor data  
73     short raw_ax, raw_ay, raw_az, raw_gx, raw_gy, raw_gz;  
74     mpu6050_fetch(&raw_ax, &raw_ay, &raw_az, &raw_gx, &raw_gy, &raw_gz);  
75  
76     // calculate angle by accel data  
77     float ax = raw_ax - mpu6050_offsets[0];  
78     float ay = raw_ay - mpu6050_offsets[1];  
79     float az = raw_az - mpu6050_offsets[2];  
80     float angle_ax = atan(ay / sqrt(ax * ax + az * az)) * (180 / PI); 가속도 센서에서 가져온 값 (고주파 노이즈)  
81  
82     float gz = ((float) (raw_gx - mpu6050_offsets[3])) / 131; 자이로 센서에서 가져온 값 (저주파 노이즈)  
83     // angle_gx += gz * DT;  
84  
85     // complementary filter  
86     angle_x = ALPHA * (angle_x + gz * DT) + (1 - ALPHA) * angle_ax; 상보필터를 이용한 노이즈 제거  
87  
88     Serial1.println(angle_x);  
89 }
```



COM7

전송

3.87  
6.45  
5.66  
5.57  
6.22  
6.31  
5.92  
5.55  
5.26  
4.16  
4.19  
6.04  
5.89  
6.87  
6.73  
5.45  
6.10

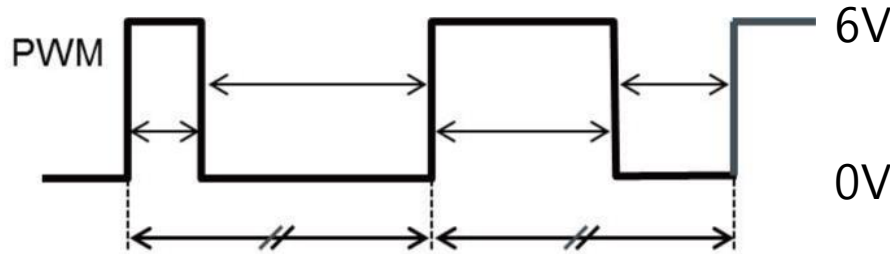
센서를 기울인 각도를 Serial로 출력한 결과물

☒ 자동 스크롤 ☐ 타임스탬프 표시 새 줄 9600 보드레이트 출력 지우기





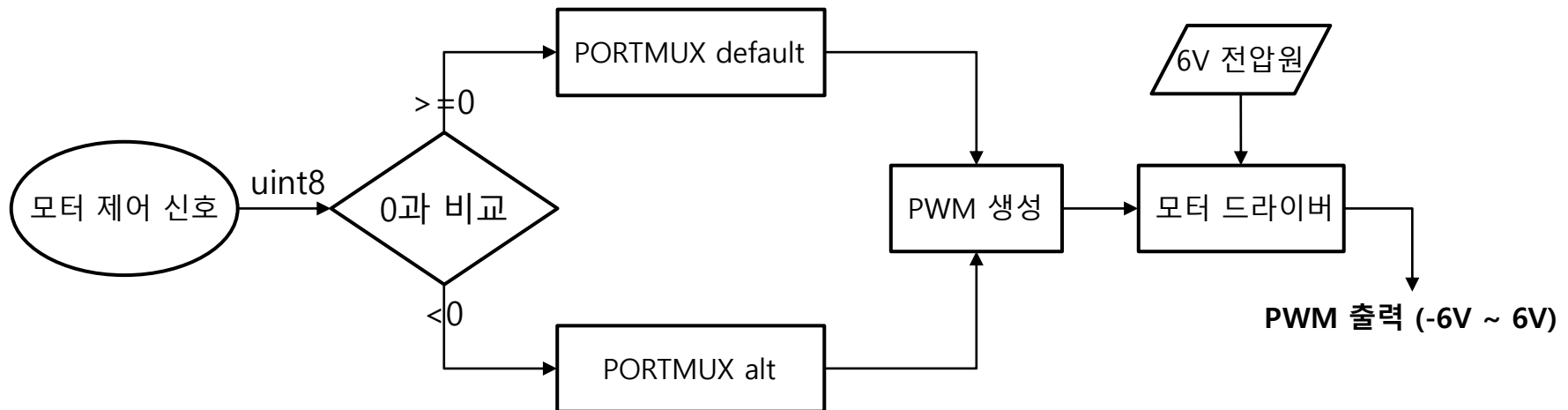
### ③ TCB를 이용한 양방향 PWM 출력



```
void _tcb0_init() {
    _PORTA_DIR |= 0b00000100; // PA2: OUTPUT
    _PORTF_DIR |= 0b00010000; // PF4: OUTPUT
    _PORTA_OUT &= ~0b00000100; // PA2: LOW
    _PORTF_OUT &= ~0b00010000; // PF4: LOW
    _TCB0_CCMPL = 0x00; // Duty
    _TCB0_CCMPH = 0xff; // TOP
    _TCB0_CTRLA = _TCB_CLKSEL_CLKDIV2_gc | _TCB_ENABLE_bm;
    _TCB0_CTRLB |= _TCB_CCMPEM_bm | _TCB_CNTMODE_PWM8_gc;
}
```

실습시간에 배운 PWM 코드는 한 방향 회전만 가능

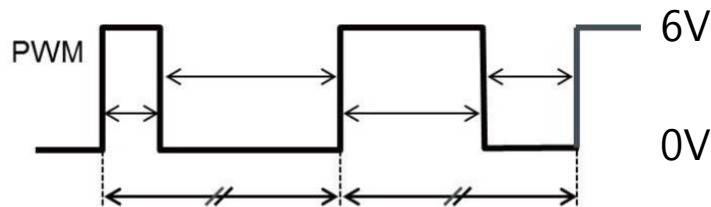
TCB를 이용해 PWM을 생성하면 한 방향으로만 회전 가능



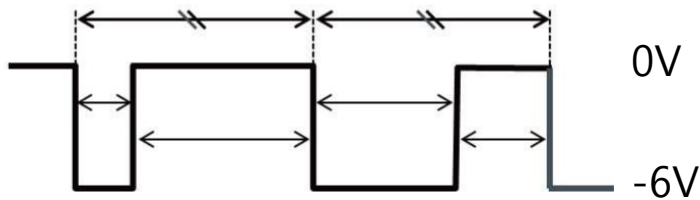
모터의 양 단을 PORTMUX 핀과 연결하면 TCB 하나로 -6V부터 6V까지의 PWM 출력 가능

### ③ TCB를 이용한 양방향 PWM 출력

#### 정방향 회전



#### 역방향 회전



MODEL	VOLTAGE		NO LOAD		AT MAXIMUM EFFICIENCY				STALL	
	OPERATING RANGE	NOMINAL V	SPEED r/min	CURRENT A	SPEED r/min	CURRENT A	TORQUE N.m	OUTPUT W	TORQUE N.m	CURRENT A
NP01S-220	3.0 - 6.0	3.0	50	0.25	34	0.45	0.07	0.71	0.52	0.24

```

unsigned char mx1508_map(float power) {
    int p = power;
    if (p < 0) {
        p = -p;
    }
    p += 158;
    if (p > 255) {
        p = 255;
    }
    return (unsigned char) p;
}

```

```

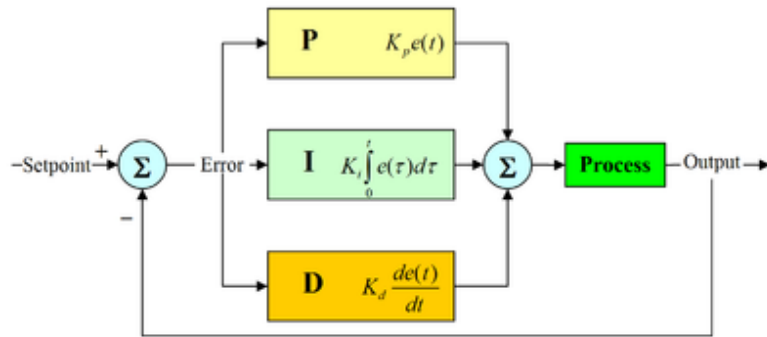
void mx1508_left_run(float power) {
    if (power >= 0) {
        _tcb0_default_pin();
    } else {
        _tcb0_alt_pin();
    }
    _tcb0_set_duty(mx1508_map(power));
}

```

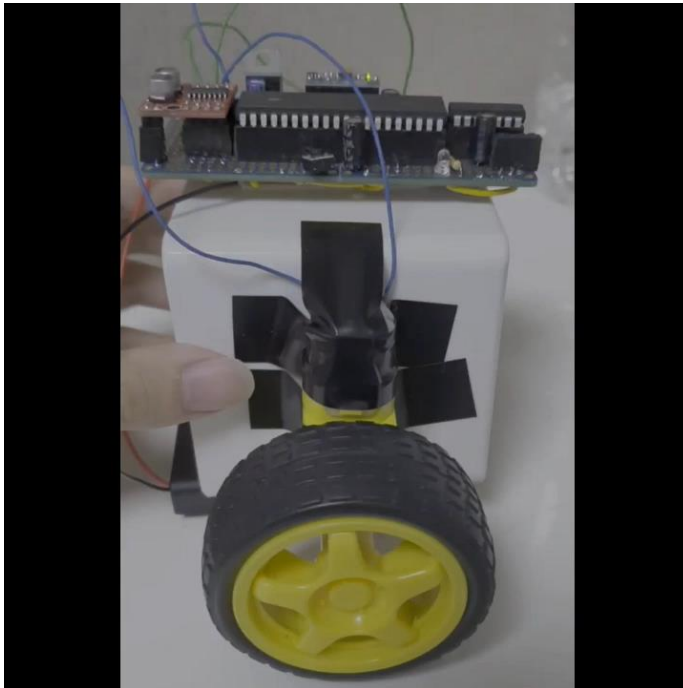
모터를 작동시키기 위한 최소 전압을 맞춰주기 위해 PWM duty에 최소값을 지정



## ④ PID 제어를 이용한 DC모터 회전 제어



```
// pid controller
float sp = 0; // setpoint
float pid_err = sp - angle_x;
float pid_d_err = pid_prev_err - pid_err;
pid_prev_err = pid_err;
float pv = KP * pid_err + KD * pid_d_err;
```

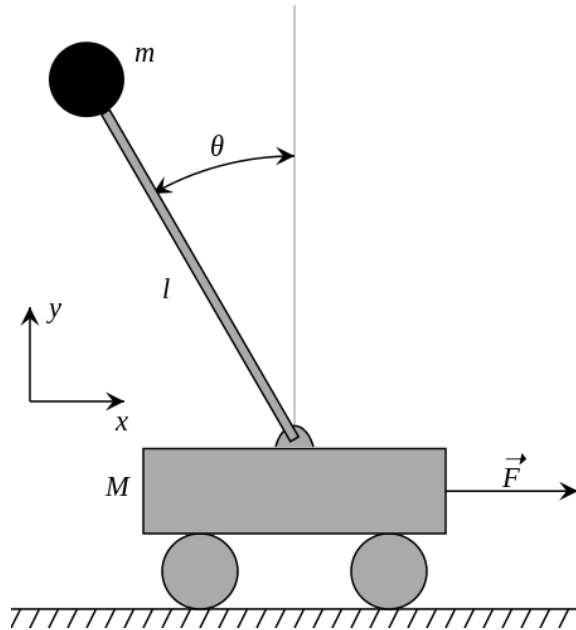


DC모터 자체가 적분기 역할을 하기 때문에  
I 성분은 제외하고, P와 D만 이용해서 제어  
(I 성분 넣고도 해봤는데 큰 차이가 없었음)

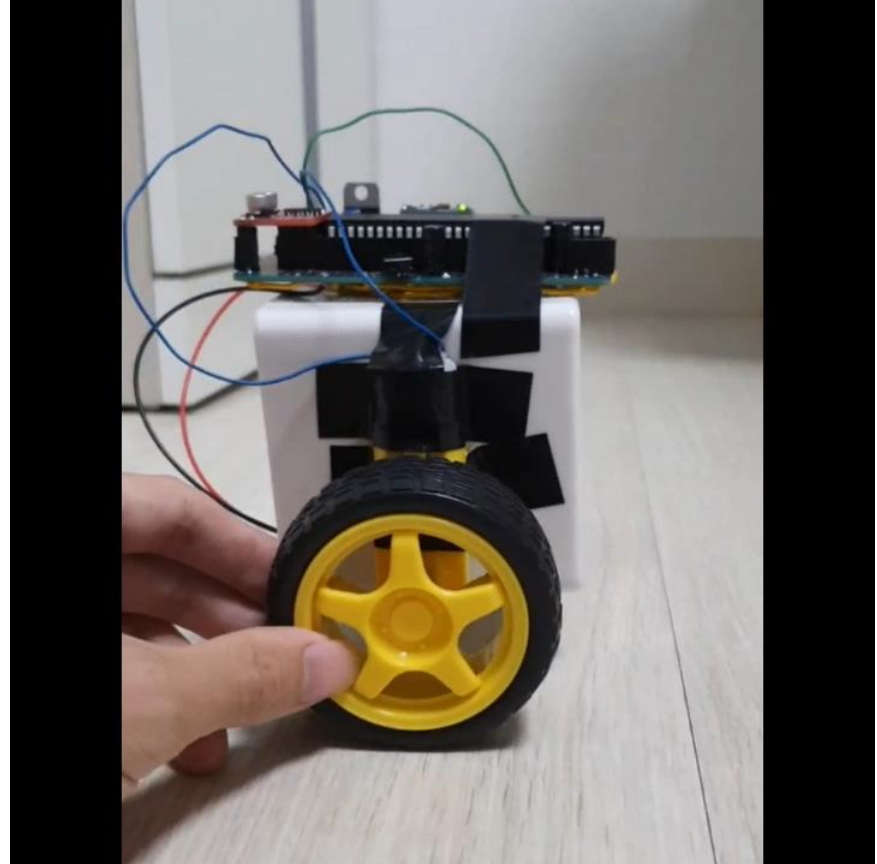
D는 기울기이므로,  $K_d \times \frac{de(t)}{dt}$  꼴인데,  
 $K_d$ 와  $dt$ 는 상수이므로  $K_d' \times de(t)$  형태로 표현

← 로봇을 기울이는 것에 따라  
바퀴가 회전하는 것을 확인 가능

## ④ 주요 이슈 2



밸런싱 로봇은 물리학적으로  
역진자와 동일



로봇을 기울였을 때 자세가 빠르게 안정화되는 PID 파라미터를 찾지  
못해 진동이 많음

## ⑤ 초음파 센서를 이용한 거리 측정

```
unsigned int t = 0;

void setup() {
  Serial1.begin(9600);

  TCA0_SINGLE_PER = 0xff;
  TCA0_SINGLE_INTCTRL = TCA_SINGLE_OVF_bm;
  TCA0_SINGLE_CTRLA = TCA_SINGLE_CLKSEL_DIV256_gc | TCA_SINGLE_ENABLE_bm;

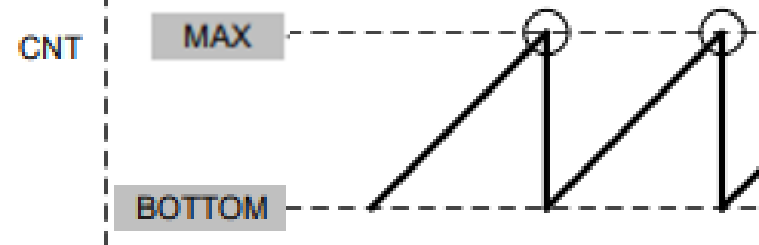
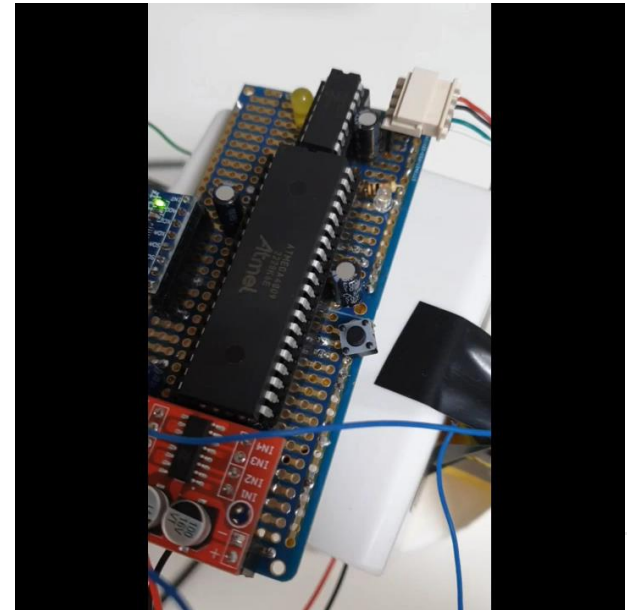
  sei();
}

void loop() {
  unsigned int T = 3; // [s]
  if (t == T * F_CPU / 65536) { // T * F_CPU / (DIV * (PERIOD + 1))
    Serial1.println("3s");
    digitalWrite(7, HIGH);
    delay(100);
    digitalWrite(7, LOW);
    t = 0;
  }
}

ISR(TCA0_OVF_vect) {
  t += 1;
  TCA0_SINGLE_INTFLAGS = TCA_SINGLE_OVF_bm;
}
```

16MHz면 1clock에 62.5ns가 소요되고,  
256배 Prescaler를 사용하므로 1clock당 16us  
Period를 255로 정했으니까 4.096ms마다 Interrupt 발생

3초에 1번 깜빡이는 LED



## ⑤ 초음파 센서를 이용한 거리 측정

```
unsigned int t = 0;
```

```
void setup() {
```

```
    PORTA_DIR &= ~0b00000001; // PA0: INPUT  
    PORTA_DIR |= 0b00000010; // PA1: OUTPUT  
    PORTA_PIN0CTRL |= 0b00000001; // PA0: PULLUP  
    PORTA_OUT &= ~0b00000010; // PA1: LOW
```

```
    TCA0_SINGLE_PER = 0xff;  
    TCA0_SINGLE_INTCTRL = TCA_SINGLE_OVF_bm;  
    TCA0_SINGLE_CTRLA = TCA_SINGLE_CLKSEL_DIV16_gc | TCA_SINGLE_ENABLE_bm;
```

1us마다 인터럽트 호출하도록 16x Prescaler 사용

```
    sei();
```

```
}
```

```
void loop() {
```

```
    unsigned int t_start = t;  
    PORTA_OUT |= 0b00000010; // PA1: HIGH  
    while (t > t_start + 10) {} // 10us 대기  
    PORTA_OUT &= ~0b00000010; // PA1: LOW
```

초음파 Pulse 출력

```
    t_start = t;  
    while (PORTA_IN & 0b00000001 == 0) {} // 초음파가 되돌아 올 때까지 대기  
    unsigned int t_diff = t - t_start;
```

초음파 Pulse 수신

```
    unsigned int distance = t_diff / 58; // 거리
```

```
}
```

```
ISR(TCA0_OVF_vect) { // 1us 마다 호출
```

```
    t += 1;  
    TCA0_SINGLE_INTFLAGS = TCA_SINGLE_OVF_bm;
```

```
}
```



# 연구 기록

<> Code 🔑 Pull requests 🛡 Security 📊 Insights ⚙ Settings <https://github.com/Astro36/ICE3015-project>

main 1 branch 0 tags Go to file Add file <> Code About

Astro36 Add project ppt file c03d1e8 3 minutes ago 31 commits

datasheet	Add datasheet	9 minutes ago
homework	Add homework report	9 minutes ago
.clang-format	Implement TWI function	last month
mysegway.ino	Update main code	51 minutes ago
project_interim.pdf	Add project ppt file	3 minutes ago
project_proposal.pdf		
test_arduino.ino		
test_tca0_hcsr04.ino		
test_tca0_timer.ino		
test_twio_mpu6050.ino		
test_wire_mpu6050.ino		

Commits on May 6, 2023

- Implement PID controller  
Astro36 committed on May 6 843fb63 <>
- Refactor code and add hcsr04 function without pulseln  
Astro36 committed on May 6 e04595c <>

Commits on May 5, 2023

- Implement complementary filter  
Astro36 committed on May 5 9953953 <>

Commits on May 3, 2023

- Implement TWI function  
Astro36 committed on May 3 c7981f1 <>

5월 3일에 첫 번째 커밋

혼자서 진행한 연구 결과를 GitHub에 정리 (개인 포트폴리오로 사용 예정)  
부품별로 동작을 확인할 수 있게 test\_\* 파일과  
메인 코드가 들어있는 mysegway.ino 파일로 분류



# 연구 기록

17 Mar 2023 · by 박승재 0 Comments

Arduino는 문법이 제한된 C++ 언어를 이용해 프로그래밍하며, 코드의 구성부는 크게 `setup` 함수와 `loop` 함수로 나뉘어 있습니다.

`setup` 과 `loop` 의 동작 원리를 알아보기 위해, Arduino Core 코드를 다운로드 받아 `main` 함수를 열어보았습니다.

코드 분석은 Atmel SAM3X를 사용하는 Arduino Due를 기준으로 수행했습니다.

```
cores/arduino/main.cpp:
```

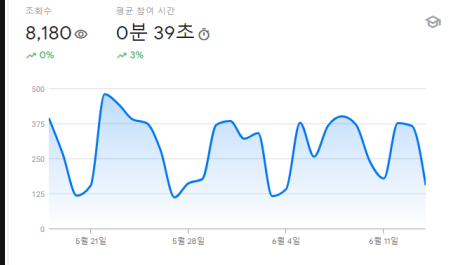
```
/*  
 * \brief Main entry point of Arduino application  
 */  
int main(void) {  
    // Initialize watchdog  
    watchdogSetup();  
  
    init();  
  
    initVariant();  
  
    delay(1);  
  
#if defined(USBCON)  
    USBDevice.attach();  
#endif  
  
    setup();  
  
    for (;;) {  
        loop();  
        if (serialEventRun)  
            serialEventRun();  
    }  
  
    return 0;  
}
```

모든 C++ 프로그램은 `main` 에서부터 시작합니다.

아두이노가 실행되면 가장 먼저 `watchdogSetup` 을 호출합니다.

## 사이트 개요

지난 28일



방문 페이지 + 쿼리 문자열		자연 Google 검색결과 클 릭수	자연 Google 검색결과 노 출수	자연 Google 검색결과 클 릭률	자연 Google 검색 평균 계 재순위	사용자	참여 세션수
		1,479 총계 대비 25.87%	10,887 총계 대비 23.75%	13.59% 평균 대비 +8.9%	6.92 총계 대비 96.39%	1,257 총계 대비 24.87%	934 총계 대비 24.73%
1	/cpp/2020-03-07/cpp-review/	2	28	7.14%	5.89	2	2
2	/cpp/2021-07-25/cpp-pbt-rapidcheck/	0	7	0%	7.00	1	1
3	/cpp/2021-10-31/cpp-iterator-sorting/	1	6	16.67%	4.00	1	1
4	/cpp/2023-01-20/smart-pointer/	2	29	6.9%	7.31	2	2
5	/cpp/2023-02-03/functor/	4	103	3.88%	9.97	3	3
6	/cpp/2023-05-05/winsock2-tcp/	4	45	8.89%	13.96	5	5
7	/cpp/2020-07-04/clion-mingw/	24	272	8.82%	8.50	16	15
8	/cpp/2020-05-26/c-cpp-string/	16	202	7.92%	8.60	15	12
9	/cpp/2023-03-17/arduino-setup-loop/	4	292	1.37%	9.90	4	3

프로젝트를 수행하며 개인적으로 공부한 내용을 스터디 블로그에 작성  
(해당 블로그는 월 조회수 8천 회 정도 나오는 블로그)



정보통신공학부  
임베디드시스템 설계



# 연구 평가

제품 완성은 미흡하나,

프로젝트를 통해 배우고자 한 목표 달성에는 성공했다고 생각함

- MCU 및 부품 회로 설계 → 성공
- 자이로가속도 센서 I2C 통신 → 성공
- 상보필터를 이용한 노이즈 제거 → 성공
- TCB를 이용한 양방향 PWM 출력 → 성공
- PID 제어기를 이용한 DC모터 회전 제어 → 코드 구현 완료 파라미터는 못 찾음
- 초음파 센서를 이용한 거리 측정 → 코드 구현 완료 조립 미완료

각 목표 구현까지의 기간은 정확하게 잡았다고 생각하지만,

하드웨어적인 문제와 PID 파라미터 찾는 시간을 고려하지 못해

완전한 작동에 실패했다고 생각함



# 결론 및 고찰

- 프로젝트 막판에 고장이 나서 실제로 작동하는 것을 보여줄 수 없어서 너무 아쉬움
- 데이터시트와 레지스터를 참고해 라이브러리 없이 코드를 작성하며 시스템을 처음부터 만드는 경험을 할 수 있었음
- 회로 설계에도 시스템을 안정적으로 동작시키기 위한 다양한 기법이 동원된다는 것을 배웠음
- DC모터 제어가 얼마나 어려운지 직접 부딪치며 느꼈음
- 임베디드 시스템은 코딩만이 아니라 반복적인 실험을 통해 완성된다는 것을 깨달음
- 학점과 관계없이 한 학기동안 최선을 다해 유의미한 결과를 남겼다  
고 생각함 (실습 경험, 프로젝트 포트폴리오, 블로그 등)



# 부품 리스트

상품명	수량	금액(원)	비고
기어박스 장착모터 NP01S-220	2	3,960	
바퀴 66파이	2	1,760	
Dual DC 모터 드라이버 [SZH-MDBL-010]	1	1,980	
자이로 가속도 센서 MPU-6050 [SZH-EK007]	2	7,700	1개 파손
초음파 센서 HC-SR04 [SZH0USBC-004]	2	3,960	
만능기판 50*50 [SMEAPS55]	1	1,870	파손
만능기판 97*90	1	3,300	
납땀용, 점프용 전선 (AWG24) 50mm	1	3,410	
AA배터리 건전지 홀더 스위치형 [4개입]	1	1,210	
디바이스마트 배송비	3	8,100	부품 파손으로 인해 3회 주문
총 금액		37,250	

MCU, 캐패시터, 레귤레이터 등의 일부 부품은 조교님께 제공받아 사용  
예산 초과로 인해 일부 부품은 사비로 구매

