

```
#define F_CPU 16000000UL ///util_delay.h 를 위한 헤더
#include <avr/io.h> ///avr을 위한 헤더
#include <avr/iom128.h> ///atmega128을 위한 헤더
#include <util/delay.h> ///_delay_us()를 사용하기 위한 헤더
#include <avr/interrupt.h> ///인터럽트 헤더
ISR(TIMER3_COMPA_vect);
ISR(TIMER1_COMPA_vect);

char CW[4]={0x11, 0x22, 0x44, 0x88};
char R_CW[4]={0x01, 0x02, 0x04, 0x08};
char L_CW[4]={0x10, 0x20, 0x40, 0x80};
char CCW[4]={0x88, 0x44, 0x22, 0x11};
char R_CCW[4]={0x01, 0x02, 0x04, 0x08};
char L_CCW[4]={0x10, 0x20, 0x40, 0x80};
char a[4]={0x11, 0x22, 0x44, 0x88};

unsigned char data;
int i=0;
int j=0;
void Step(unsigned char data);
char recive(void);
void InitTimer3(void);
void InitTimer1(void);
void InitUart(void);

void InitTimer3(void)//
{
    TCCR3A=0b00000000;
    TCCR3B=0b00001100;
    TCCR3C=0b00000000;
    ETIMSK=0b00010000;
}
void InitTimer1(void)
{
    TCCR1A=0b00000000;
    TCCR1B=0b00001100;
    TCCR1C=0b00000000;
    TIMSK=0b00010000;
}
void InitUart(void)
{
```

```
    UBRRH0H=0x00;
    UBRROL=0x67;
    UCSRA=0x00;
    UCSRB=0x10;
    UCSC=0x06;
}
char recive(void)
{
    unsigned char data;
    while((UCSRA & 0x80) == 0x00);
    data=UDR0;
    return data;
}
//ctc 256 timer 1,3 (16000000/(256*(OCR+1))
int main(void)
{
    char data;
    DDRA=0xFF;
    DDRD=0xFF;
    PORTD=0x00;
    DDRC=0xFF;
    PORTC=0xFF;
    InitTimer3();
    InitTimer1();
    InitUart();

    UBRRH0H=0x00;
        UBRROL=0x67;
        UCSRA=0x00;
        UCSRB=0x10;
        UCSC=0x06;
    DDRE = 0x00;
    OCR1AH=0x00;
    OCR1AL=0x7C;
    OCR3AH=0x00;
    OCR3AL=0x7E;
    SREG|=0x80;

    while(1)
    {
data=recive();
```

```
Step(data);  
    }  
    return 0;  
}
```

```
void Step(unsigned char data)  
{  
    int k;  
    if(data=='a'){//전진  
        TIMSK=0x00;  
        OCR3AH=0x00;  
        OCR3AL=0x7E;  
        for(k=0;k<4;k++)  
        {  
            a[k]= F_CW[k];  
        }  
    }  
}
```

```
else if(data=='b'){//후진  
    TIMSK=0x00;  
    OCR3AH=0x00;  
    OCR3AL=0x7E;  
    for(k=0;k<4;k++)  
    {  
        a[k]= B_CCW[k];  
    }  
}
```

```
else if(data=='c'){//앞 좌회전-timer3가 더 빠름  
{  
    TIMSK=0x10;  
    OCR1AH=0x00;  
    OCR1AL=0x80;  
    OCR3AH=0x00;  
    OCR3AL=0x83;  
    for(k=0;k<4;k++)  
    {  
        a[k]= R_CW[k];  
    }  
}
```

```
}  
  
}  
else if(data=='d')//앞 우회전-timer1이 더 빠름  
{  
    TIMSK=0x10;  
    OCR1AH=0x00;  
    OCR1AL=0x83;  
    OCR3AH=0x00;  
    OCR3AL=0x80;  
    for(k=0;k<4;k++)  
    {  
        a[k]= R_CW[k];  
    }  
  
}  
  
else if(data=='e')//뒤 좌회전-timer3가 더 빠름  
{  
    TIMSK=0x10;  
    OCR1AH=0x00;  
    OCR1AL=0x80;  
    OCR3AH=0x00;  
    OCR3AL=0x83;  
    for(k=4;k>0;k--)  
    {  
        a[k]= R_CW[k];  
    }  
}  
else if(data=='f')//뒤 우회전-timer1이 더 빠름  
{  
    TIMSK=0x10;  
    OCR1AH=0x00;  
    OCR1AL=0x83;  
    OCR3AH=0x00;  
    OCR3AL=0x80;  
    for(k=4;k>0;k--)  
    {  
        a[k]= R_CW[k];  
    }  
}
```

```
}  
ISR(TIMER3_COMPA_vect)  
{  
    i++;  
    i=i%4;  
    PORTA=a[i];  
}  
ISR(TIMER1_COMPA_vect)  
{  
    j++;  
    j=j%4;  
    PORTA=L_CW[j];  
}
```

```
#define F_CPU 16000000UL ///  
#include <avr/io.h> ///  
#include <avr/iom128.h> ///  
#include <util/delay.h> ///  
#include <avr/interrupt.h> ///  
#include <math.h>  
#include <stdio.h>  
#include <string.h>  
#include <stdlib.h>  
typedef unsigned char byte;  
/*-----MPU-----*/  
byte MPU6050_read(byte addr);  
void MPU6050_write(byte addr, byte data);  
void getRawData();
```

```
signed int gx = 0, gy = 0, gz = 0, ax = 0, ay = 0, az = 0;
float ax_ang, ay_ang;
byte buffer[12];
void InitUart(void);
unsigned char getch_USART0(void);
void getstr(unsigned char* str);
void putch_USART0(unsigned char data);
void putstr(unsigned char* data);
char data;
int ADC_set();
void ADC_init(void);
unsigned int ADC0onversion(int ch);
unsigned int ADC1onversion(int ch);
unsigned int ADC2onversion(int ch);
void angle_set();

void InitUart(void)
{
    UBRR0H=0X00;
    UBRR0L=0X67;
    UCSR0A = 0X00;
    UCSR0B = (1<<RXEN0)|(1<<TXEN0);
    UCSR0C = 0X06;
}
void Data(unsigned char data)
{
    while((UCSR0A & 0X20)!=0x00);
    UDR0 =data;
}
void putch_USART0(unsigned char data){
    while(!(UCSR0A&(1<<UDRE0)));
    UDR0=data;
}
void putstr(unsigned char* data){
    char i=0;
    for(i=0; data[i] != 0; i++){
        putch_USART0(data[i]);
    }
}
//----- Initialize ADC
void ADC_init(void)
```

```
{
    // 기준전압 설정
    ADMUX |= (1 << REFS1) | (1 << REFS0) ;
    // ADC 허용
    ADCSRA |= (1 << ADEN);
    // 프리스케일러 값 64
    ADCSRA |= (1 << ADPS2) | (1 << ADPS1) | (0 << ADPS0);
}
//----- Analog to Digital
unsigned int ADC0onversion(int ch)
{
    ADMUX=0xC0;
    _delay_ms(10);
    // AD 변환 시작
    ADCSRA |= (1 << ADSC);
    // AD 변환 완료 여부 확인, AD 변환 완료될 때까지 대기, 완료 확인되면 다음 진행
    while(!(ADCSRA & (1 << ADIF)));
    // AD 변환 완료된 데이터 반환
    return ADC;
}
unsigned int ADC1onversion(int ch)
{
    ADMUX=0xC1;
    _delay_ms(10);
    // AD 변환 시작
    ADCSRA |= (1 << ADSC);
    // AD 변환 완료 여부 확인, AD 변환 완료될 때까지 대기, 완료 확인되면 다음 진행
    while(!(ADCSRA & (1 << ADIF)));
    // AD 변환 완료된 데이터 반환
    return ADC;
}
unsigned int ADC2onversion(int ch)
{
    ADMUX=0xC2;
    _delay_ms(10);
    // AD 변환 시작
    ADCSRA |= (1 << ADSC);
    // AD 변환 완료 여부 확인, AD 변환 완료될 때까지 대기, 완료 확인되면 다음 진행
    while(!(ADCSRA & (1 << ADIF)));
    // AD 변환 완료된 데이터 반환
    return ADC;
}
```

```
}

void twi_init(void) ///twi 초기화
{
    TWSR=0x00; ///상태레지스터-SCL주파수 분주비 1 ///상태코드 상위5비트는 Read
only 이다
    TWBR=0x0C; ///비트레이트레지스터-0b00001100 SCL주파수 400kHz
    TWCR|=0x84; ///제어레지스터
}

void mpu6050_init(void) ///mpu 초기화
{
    MPU6050_write(0x6B,0x00); ///mpu는 초기에 슬립모드이다
    MPU6050_write(0x6C,0x00);
    MPU6050_write(0x1B,0b00011000); ///자이로 범위셋팅 2000'/s -> 3,4번 비트 00일때
250'/s부터 2배씩 증가
    MPU6050_write(0x1C,0b00011000); ///가속도 범위셋팅 16g -> 3,4번 비트 00일때 2g
부터 2배씩 증가
    MPU6050_write(0x1A,0x06); ///DLPF 0x06 대역폭5Hz 딜레이19ms
}
int main(void)
{
    DDRF=0x00;
    DDRA=0xFF;
    ADC_init();
    twi_init();
    PORTA=0xFF;

    mpu6050_init();

    InitUart();

    DDRE=0x02;
    UBRR0H=0X00;
    UBRR0L=0X67;
    UCSR0A = 0X00;
    UCSR0B = 0X08;
    UCSR0C = 0X06;
    char data;

    while(1)
```

```
{
    Data('a');
    delay_ms(300);
    Data('b');
    delay_ms(300);

    //angle_set();
}

return 0;
}

void delay_ms(unsigned int m)
{
    unsigned int i,j;
    for(i=0;i<m;i++)
    {
        for(j=0;j<2117;j++);
    }
}

byte MPU6050_read(byte addr) ///mpu6050의 내부레지스터의 값을 수신하여 읽는함수
{
    byte data; ///함수의 반환형과 일치시킨다

    TWCR=0xA4; ///START bit 전송
    while(((TWCR&0x80)==0x00) || ((TWSR&0xF8)!=0x08))
    {
        ; ///1.플래그가 셋 되었고 2.상태코드-START전송완료 두조건이 만족되면 while()을
빠져나옴
    }

    TWDR=0xD0; ///mpu6050의 주소 0b110100x + Write비트 0 x는 mpu6050의 AD0핀
의 상태로 결정됨
    TWCR=0x84; ///플래그를 소프트웨어에서 클리어하려면 1을 입력한다

    while(((TWCR&0x80)==0x00) || ((TWSR&0xF8)!=0x18))
    {
        ; ///1.플래그가 셋 되었고 2.상태코드-확인비트ACK 수신완료 두조건이 만족되면
```

```
while()을 빠져나옴
} ///SLA+W 전송완료

TWDR=addr; ///마스터가 Read하고싶은 레지스터의 주소를 송신
TWCR=0x84; ///플래그를 소프트웨어에서 클리어하려면 1을 입력한다

while(((TWCR&0x80)==0x00) || ((TWSR&0xF8)!=0x28))
{
    ; ///1.플래그가 셋 되었고 2.상태코드-DATA송신완료 두조건이 만족되면 while()을
    빠져나옴
}
TWCR=0xA4; ///REPEATED START

while(((TWCR&0x80)==0x00) || ((TWSR&0xF8)!=0x10))
{
    ; ///1.플래그가 셋 되었고 2.상태코드-REPEATED START전송완료 두조건이 만족
    되면 while()을 빠져나옴
} ///제어권을 잃지않고 계속해서 통신을 지속

TWDR=0xD1; ///mpu6050의 주소 0b110100x + Read비트 1 x는 mpu6050의 AD0핀
의 상태로 결정됨
TWCR=0x84; ///플래그를 소프트웨어에서 클리어하려면 1을 입력한다

while(((TWCR&0x80)==0x00) || ((TWSR&0xF8)!=0x40))
{
    ; ///1.플래그가 셋 되었고 2.상태코드-확인비트ACK 수신완료 두조건이 만족되면
    while()을 빠져나옴
} ///SLA+R 전송완료

TWCR=0x84; ///플래그를 소프트웨어에서 클리어하려면 1을 입력한다
///데이터 수신대기상태
while(((TWCR&0x80)==0x00) || ((TWSR&0xF8)!=0x58))
{
    ; ///1.플래그가 셋 되었고 2.상태코드-DATA수신완료 두조건이 만족되면 while()을
    빠져나옴
} ///TWCR6번비트가 '0'이므로 마스터는 ACK신호를 생성하지않는다 그렇기때문에 상태
코드는 0x58;

data=TWDR; ///수신data를 읽는다
TWCR=0x94; ///STOP bit 전송
```

```
    return data; ///  
}

void MPU6050_write(byte addr,byte data) ///  
{
    TWCR=0b10100100; ///  

    while(((TWCR&0x80)==0x00) || ((TWSR&0xF8)!=0x08))
    {
        ; ///  

    }

    TWDR=0xD0; ///  

    TWCR=0x84; ///  

    while(((TWCR&0x80)==0x00) || ((TWSR&0xF8)!=0x18))
    {
        ; ///  

    }

    TWDR=addr; ///  

    TWCR=0x84; ///  

    while(((TWCR&0x80)==0x00) || ((TWSR&0xF8)!=0x28))
    {
        ; ///  

    }

    TWDR=data; ///  

    TWCR=0x84; ///  

    while(((TWCR&0x80)==0x00) || ((TWSR&0xF8)!=0x28))
    {
        ; ///  

    }
}
```

```
TWCR=0x94; ///  
_delay_us(50);  
}  
  
void getRawData()  
{  
    buffer[0] = MPU6050_read(0x3B);  
    buffer[1] = MPU6050_read(0x3C);  
    buffer[2] = MPU6050_read(0x3D);  
    buffer[3] = MPU6050_read(0x3E);  
    buffer[4] = MPU6050_read(0x3F);  
    buffer[5] = MPU6050_read(0x40);  
  
    buffer[6] = MPU6050_read(0x43);  
    buffer[7] = MPU6050_read(0x44);  
    buffer[8] = MPU6050_read(0x45);  
    buffer[9] = MPU6050_read(0x46);  
    buffer[10] = MPU6050_read(0x47);  
    buffer[11] = MPU6050_read(0x48);  
  
    ax = (int)buffer[0] << 8 | (int)buffer[1];  
    ay = (int)buffer[2] << 8 | (int)buffer[3];  
    az = (int)buffer[4] << 8 | (int)buffer[5];  
    gx = (int)buffer[6] << 8 | (int)buffer[7];  
    gy = (int)buffer[8] << 8 | (int)buffer[9];  
    gz = (int)buffer[10] << 8 | (int)buffer[11];  
  
    ax_ang=-atan2((double)ax,sqrt((double)ay*(double)ay+(double)az*(double)az))*180/M_PI  
; ///  
    ay_ang=atan2((double)ay,(double)az)*180/M_PI;  
    _delay_ms(500);  
}  
int ADC_set()  
{  
    unsigned int ch_ADC0,ch_ADC1,ch_ADC2;
```

```
int cnt=0;

ch_ADC0 = ADC0onversion(0);
ch_ADC1= ADC1onversion(1);
ch_ADC2=ADC2onversion(2);
if(ch_ADC0>250)
{
    cnt++;
}
if(ch_ADC1>700)
{
    cnt++;
}
if(ch_ADC2>700)
{
    cnt++;
}
_delay_ms(1);
return cnt;
}
void angle_set()
{
int gear=0;
getRawData();
gear=ADC_set();
switch(gear)
{
    case 3: //정지
        putchar_USART0('7');
        break;
    case 2://1단속력
        if(ax_ang>60) {//앞
            if(ay_ang>50)//오른쪽
                putchar_USART0('1');
            else if(ay_ang<-50)//왼쪽
                putchar_USART0('2');
            else//전진
                putchar_USART0('3');
        }
        else if(ax_ang<-60){//뒤
            if(ay_ang>50)//오른쪽
```

```
    putchar_USART0('4');
    else if(ay_ang<-50)//왼쪽
    putchar_USART0('5');
    else//전진
    putchar_USART0('6');
    }
else //정지
    putchar_USART0('7');
break:
case 1://2단속력
if(ax_ang>60) {//앞

    if(ay_ang>50)//오른쪽
    putchar_USART0('8');
    else if(ay_ang<-50)//왼쪽
    putchar_USART0('9');
    else//전진
    putchar_USART0('a');
    }
else if(ax_ang<-60){//뒤
    if(ay_ang>50)//오른쪽
    putchar_USART0('b');
    else if(ay_ang<-50)//왼쪽
    putchar_USART0('c');
    else//전진
    putchar_USART0('d');
    }
else //정지
    putchar_USART0('7');
break:
case 0://3단속력
if(ax_ang>60) {//앞
    if(ay_ang>50)//오른쪽
    putchar_USART0('e');
    else if(ay_ang<-50)//왼쪽
    putchar_USART0('f');
    else//전진
    putchar_USART0('g');
    }
else if(ax_ang<-60){//뒤
    if(ay_ang>50)//오른쪽
```

```
    putch_USART0('h');
    else if(ay_ang<-50)//왼쪽
    putch_USART0('i');
    else//전진
    putch_USART0('j');
    }
else //정지
    putch_USART0('7');
    break:
}
gear=0;
}
```