

제어시스템 모델링

[실습1-1 Encoder Read]

이름 : 김 용 현

학번 : 2017006262

실습이론

DC 모터

자기장을 이용하여 한 방향으로 지속적인 토크를 발생시키는 장치.

전기적 모델링 : $V_a = I_a \cdot R_a + K_e \cdot \dot{\theta}_m$

기계적 모델링 : $T_m = K_t \cdot I_a = (J_r + J_l / N^2) \cdot \ddot{\theta}_m$

$$\begin{array}{lcl} \text{Time Domain} & \rightarrow & \text{Frequency Domain} \\ V_a = i_a R_a + k_e \dot{\theta}_m & V_a(s) = i_a R_a(s) + k_e \dot{\theta}_m(s) & \Rightarrow I_a(s) = \frac{V_a(s)}{R_a} - \frac{k_e \dot{\theta}_m(s)}{R_a} \\ T_m = K_t i_a = (J_r + J_l / N^2) \ddot{\theta}_m & J s^2 \theta_m(s) = K_t i_a(s) & \Rightarrow J s^2 \theta_m(s) = \frac{K_t V_a(s)}{R_a} - \frac{K_t k_e \dot{\theta}_m(s)}{R_a} \end{array}$$

전달함수 $G(s) = S \cdot \theta_m(s) / V_a(s) = K_t / (R_a \cdot J s + K_t \cdot K_e)$

PWM(Pulse width modulation)

디지털 신호의 경우 0.5와 같은 표현이 불가능하고 0과 1만 가능하다. 이를 위해 펄스폭의 변조를 이용하여 아날로그적 출력 값을 가지게 하는 것을 PWM이라고 한다. 실험에 사용하는 아두이노의 경우 0부터 255까지의 해상도 가능.

모터 드라이버

모터 회전방향을 변경시켜 주기 위한 장치, 전류의 방향을 물리적으로 변경시켜 주며 PWM으로 모터 전압이나 전류를 조정할 수 있다.

#CW=시계방향 회전, CCW=반시계방향 회전

Encoder

모터에 장착하여 회전량을 측정하는 센서. 실습에 사용하는 엔코더는 자기식 엔코더로, N극과 S극이 번갈아 인식되면서 Encoder A, Encoder B의 값을 증감시켜 회전량을 측정한다.

실습코드(아두이노)

```
#include <SSD1306.h>
#include <MsTimer2.h>
// OLED Setup
#define OLED_DC 5
#define OLED_CLK 8
#define OLED_MOSI 7
#define OLED_RESET 6
SSD1306 oled(OLED_MOSI, OLED_CLK, OLED_DC, OLED_RESET,
0);
//MOTOR driver pin
#define PWM 9
#define IN1 10
#define IN2 11
//Encoder Pin
#define ENCODER_A 3
#define ENCODER_B 2

int cart_encoder = 0;
int pwm_u = -100;

//=====OLED
FUNCTION=====
=//
uint32_t oled_pow(uint8_t m,uint8_t n){
    uint32_t result=1;
```

```

    while(n--){result*=m;
    return result;
}
void OLED_ShowNumber(uint8_t x,uint8_t y,uint32_t num,uint8_t
len){
    u8 t,temp;
    u8 enshow=0;
    oled.drawchar(x-6,y,' ');
    for(t=0;t<len;t++){
        temp=(num/oled_pow(10,len-t-1))%10;
        oled.drawchar(x+6*t,y,temp+'0');
    }
}

void OLED_ShowNumber_Minus(uint8_t x,uint8_t y,int
num_,uint8_t len){
    uint32_t num = (uint32_t)(-1*num_);
    u8 t,temp;
    u8 enshow=0;
    oled.drawchar(x-6,y,'-');
    for(t=0;t<len;t++) {
        temp=(num/oled_pow(10,len-t-1))%10;
        oled.drawchar(x+6*t,y,temp+'0');
    }
}

```

```
void OLED_Show()
{
    oled.drawstring(00,1,"====ENCODER_READ====");
    oled.drawstring(00,2,"encoder:");
    if(cart_encoder<0)
        OLED_ShowNumber_Minus(70,2,cart_encoder,4);
    else
        OLED_ShowNumber(70,2,cart_encoder,4);
    oled.drawstring(00,4,"Encoder A:");
    if(digitalRead(ENCODER_A)==HIGH)
        oled.drawstring(70,4,"HIGH");
    else
        oled.drawstring(70,4," LOW");
    oled.drawstring(00,5,"Encoder B:");
    if(digitalRead(ENCODER_B)==HIGH)
        oled.drawstring(70,5,"HIGH");
    else
        oled.drawstring(70,5," LOW");
    oled.display();
    oled.drawstring(00,6,"PWM :");
    if(pwm_u<0)
        OLED_ShowNumber_Minus(70,6,pwm_u,3);
    else
        OLED_ShowNumber(70,6,pwm_u,3);
    oled.drawstring(00,7,"DIR :");
    if(pwm_u>0)
```

```
        oled.drawstring(70,7,"CW");
    else
        oled.drawstring(70,7,"CCW");
}

void Set_PWM(int motor)
{
    if(motor >= 0)
    {
        digitalWrite(IN1,HIGH);
        digitalWrite(IN2,LOW);
        analogWrite(PWM,motor);
    }
    else
    {
        digitalWrite(IN1,LOW);
        digitalWrite(IN2,HIGH);
        analogWrite(PWM,motor);
    }
}

void doEncoderA()
{
    if(digitalRead(ENCODER_A) == digitalRead(ENCODER_B))
    {
```

```

        cart_encoder =  cart_encoder +1;
    }
    else
    {
        cart_encoder = cart_encoder -1;
    }
}

void doEncoderB()
{
    if(digitalRead(ENCODER_A) == digitalRead(ENCODER_B))
    {
        cart_encoder =  cart_encoder -1;
    }
    else
    {
        cart_encoder = cart_encoder +1;
    }
}

void control(){
    Set_PWM(pwm_u);
}

void setup() {
    //display setup

```

```
int fff = 1;
TCCR1B =(TCCR1B & 0xF8) | fff;
oled.ssd1306_init(SSD1306_SWITCHCAPVCC);
oled.clear();    // clears the screen and buffer
```

```
//motor driver setup
pinMode(IN1, OUTPUT);
pinMode(IN2, OUTPUT);
pinMode(PWM, OUTPUT);
// motor STOP
digitalWrite(IN1, 0);
digitalWrite(IN2, 0);
digitalWrite(PWM, 0);
```

```
//Encoder Setup
pinMode(ENCODER_A, INPUT);
pinMode(ENCODER_B, INPUT);
```

```
// Serial Setup
Serial.begin(2000000);
delay(200);
```

```
//Timer Setup
MsTimer2::set(1, control);
MsTimer2::start();
```

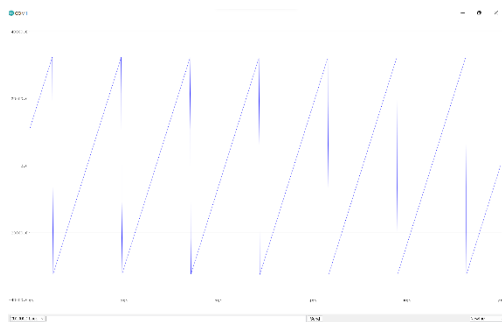


```
//Interrupt Setup
attachInterrupt(0, doEncoderA, CHANGE);
attachInterrupt(1, doEncoderB, CHANGE);

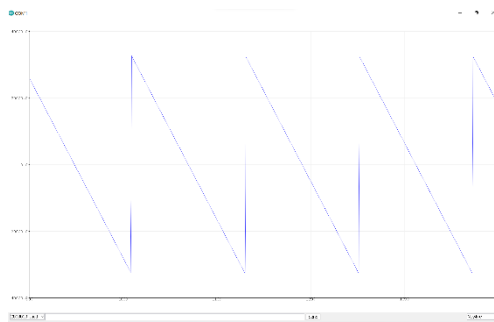
}

void loop(){
  OLED_Show();
  Serial.println(cart_encoder);
}
```

결론



pwm = -100



pwm=100

PWM을 -100으로 잡을 경우 CCW(반시계 방향)으로 회전하고 엔코더 값이 증가하다 특정 값에서 엔코더가 초기값으로 돌아가고 다시 증가하는 모습을 볼 수 있었다.

PWM을 100으로 잡을 경우 CW(시계 방향)으로 회전하며 엔코더 값이 감소하다 특정 값에서 엔코더가 초기값으로 돌아가고 다시 감소하는 모습을 볼 수 있었다.

- A를 이용해 B를 배웠다.

본 실습을 통해 아두이노에서 pwm을 이용하여 아날로그 신호를 출력하고, 이를 이용하여 DC 모터를 제어하는 방법을 배우게 되었다. 또한 전자기식 encoder의 작동 방식을 배우고 아두이노 코딩을 통해 encoder의 카운트하여 변화하는 모습을 확인할 수 있었다.