

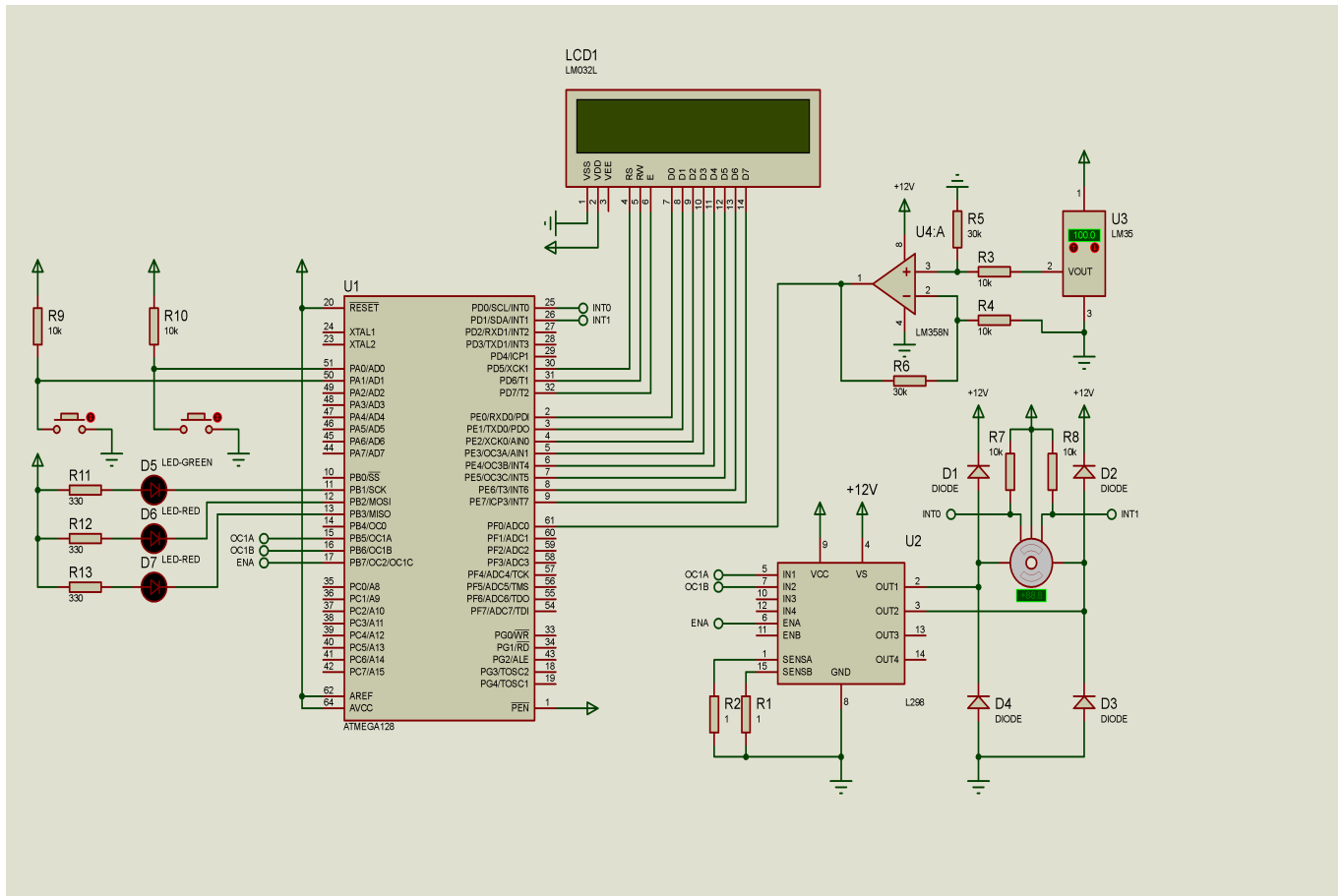
# REPORT

メカトロニクス工学科

実験日：2020年 12月 21日

学籍番号：2016945049

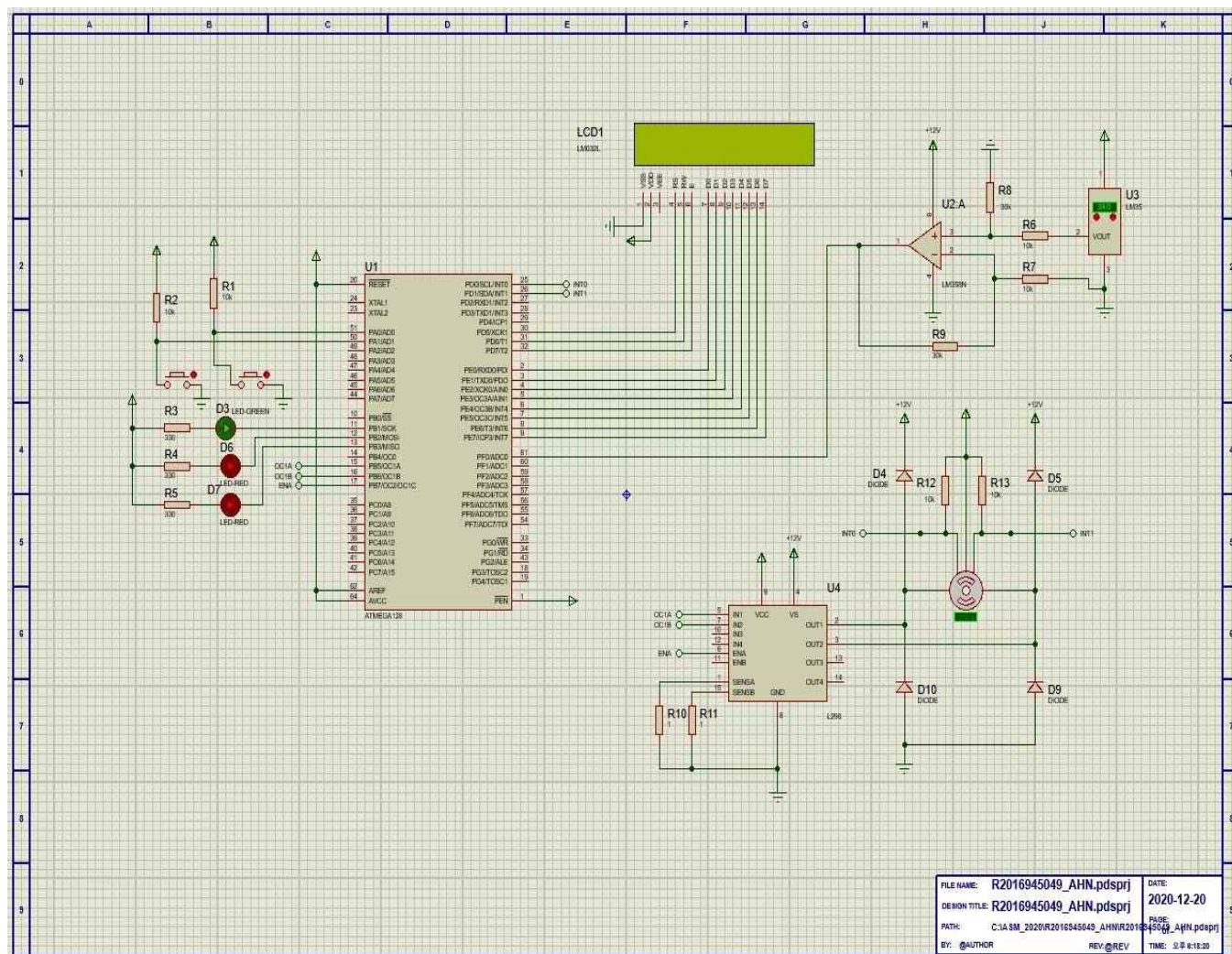
名前：アンソンミン



Atmega128を利用して、次の回路を作ってプログラムコードを作成して実行することで実行の条件が4つあります。

1. LCDに温度とモーターの速度を表示して、モーターの回転パルスも表示する。
2. ボタン1を押すと、モーターが時計回りの方向で回転し、温度が50度以上か、パルス数が200以上なら停止。 - 動作中にボタン1をまた押すと停止
3. ボタン2を押すと、モーターが反時計回りの方向で回転し、温度が25度以下か、パルス数がまた0になったら停止。 - 動作中にボタン1をまた押すと停止
4. モーターが動作したら緑色LCDがオン。 / 時計回りの方向ならD6 / 反時計回りの方向ならD7をオン。

## 1. 回路作り



## 2. ユードビジョンのコード

```
#include <mega128.h>
#include <delay.h>

unsigned int read_adc(unsigned char adc_input) ;

signed char FLAG_RUNs = 0, FLAG_DIRs = 1, KEY1_OLDs = 0, KEY1_NEWs = 0, KEY2_OLDs = 0,
    KEY2_NEWs = 0 ;
signed int  PWM1s = 0 , i=0, k=0;
unsigned int  ADC0 = 0, DAT3 = 0, DAT2 = 0, DAT1 = 0, DAT0 = 0, IMSI = 0 ;
unsigned long Tmr3 = 0, Ovr3 = 0, Rpms = 0;
unsigned long ltmp = 0 ;

interrupt [EXT_INT0] void ext_int0_isr(void)
{
    Tmr3      =(unsigned      long)(TCNT3L)+(unsigned      long)(TCNT3H)*256L      +(unsigned
        long)(TCNT3L)+ Ovr3*65536L;
    Ovr3=0; TCNT3H=0; TCNT3L =0;
}

interrupt [EXT_INT1] void ext_int1_isr(void)
{
}

interrupt [TIM3_OVF] void timer3_ovf_isr(void)
{
    Ovr3 +=1;
}

interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
    TCNT0=0x06;

    KEY1_OLDs = KEY1_NEWs; KEY1_NEWs=PINA.0^0x01;

    KEY2_OLDs = KEY2_NEWs; KEY2_NEWs=PINA.1^0x01;
```

```

if((KEY1_OLDs == 0) && (KEY1_NEWs == 1))
{
    FLAG_RUNs ^= 1 ;
    if(FLAG_RUNs) { PORTB.1 = 0 ; PORTB.2 = 0 ;PORTB.3 = 1 ; }
    else          { PORTB.1 = 1 ; }
}

if((KEY2_OLDs == 0) && (KEY2_NEWs == 1))
{
    if(FLAG_DIRs > 0) { FLAG_DIRs = -1 ; PORTB.2 = 1 ; }
    else              { FLAG_DIRs = 1 ; PORTB.2 = 0 ; PORTB.3=0; }
    if(FLAG_DIRs > 0) { FLAG_DIRs = 1 ; PORTB.3 = 1 ; }
    else              { FLAG_DIRs = -1 ; PORTB.3 = 0 ; }
}

ADC0 = read_adc(0) ;

if(FLAG_RUNs)
{
    i += 1;

    PORTB.7 = 1 ;

    if((FLAG_DIRs > 0)&&(ltmp<50) && (PWM1s<200))
    {
        if (k==0) {
            if (i > 20) { i = 0; PWM1s += 1; }
            if (PWM1s > 200) { PWM1s = 200; k = 1; }
        }
        OCR1AH = PWM1s >> 8 ;
        OCR1AL = PWM1s & 0xFF ;
        OCR1BH = 0;
        OCR1BL = 0;
        PORTB.1 = 0 ;
    }

    else if (FLAG_DIRs > 0)
    {
        OCR1AH = 0; OCR1AL = 0;
        OCR1BH =0; OCR1BL = 0;
        PORTB.1 = 1 ;
    }
}

```

```

    }

else if (ltmp>=25){
{

    PORTB.1 = 0 ;
    OCR1AH = 0;
    OCR1AL = 0;
    OCR1BH = PWM1s >> 8 ;
    OCR1BL = PWM1s & 0xFF ;

    if (k==0) {
    if (i > 20) { i = 0; PWM1s += 1; }
    if (PWM1s > 200) { PWM1s = 200; k = 1; }
    }
    if(k==1){
    if(i >20) { i=0; PWM1s += -1;}
    if(PWM1s<2) {PWM1s=0; k=2;}
    PORTB.1 = 0 ;
    }

    if(ltmp<26){

    OCR1AH = 0; OCR1AL = 0;
    OCR1BH =0; OCR1BL = 0;
    PORTB.1 = 1 ;
    }
    }

}

else
{
    OCR1AH = 0 ; OCR1AL = 0 ;
    OCR1BH = 0 ; OCR1BL = 0 ;
}
}
}

```

```

void LCD_DAT(unsigned char dat)
{
    PORTE = dat ;
    PORTD.5 = 1 ; PORTD.6 = 0 ; PORTD.7 = 1 ; // RS = 1, R/W = 0, E = 1 ;
    delay_us(5) ;
    PORTD.7 = 0 ; // E = 0 ;
    delay_ms(10) ;
}

```

```

void LCD_CMD(unsigned char cmd)
{
    PORTE = cmd ;
    PORTD.5 = 0 ; PORTD.6 = 0 ; PORTD.7 = 1 ; // RS = 0, R/W = 0, E = 1 ;
    delay_us(5) ;
    PORTD.7 = 0 ; // E = 0 ;
    delay_ms(10) ;
}

```

```

void LCD_POS(unsigned x, unsigned y)
{
    unsigned char pos ;
    if(y == 0)      { pos = 0x00 + x ; }
    else if(y == 1) { pos = 0x40 + x ; }
    else if(y == 2) { pos = 0x14 + x ; }
    else            { pos = 0x54 + x ; }

    LCD_CMD(0x80 | pos) ;
}

```

```

void LCD_INIT(void)
{
    LCD_CMD(0x38) ; delay_ms(10) ;
    LCD_CMD(0x38) ; delay_ms(10) ;
    LCD_CMD(0x38) ; delay_ms(10) ;
    LCD_CMD(0x0C) ; delay_ms(10) ;
    LCD_CMD(0x01) ; delay_ms(10) ;
}

```

```
#define ADC_VREF_TYPE 0x00

unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input|ADC_VREF_TYPE;

    ADCSRA|=0x40;

    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCW;
}

void main(void)
{
    PORTA=0x00;
    DDRA=0x00;

    PORTB=0x00;
    DDRB=0x6E;

    PORTC=0x00;
    DDRC=0x00;

    PORTD=0x00;
    DDRD=0xE0;

    PORTE=0x00;
    DDRE=0xFF;

    PORTF=0x00;
    DDRF=0x00;

    PORTG=0x00;
    DDRG=0x00;

    ASSR=0x00;
    TCCR0=0x03;
    TCNT0=0x06;
    OCR0=0x00;

    TCCR1A=0xAB;
```

TCCR1B=0x0A;  
TCNT1H=0x00;  
TCNT1L=0x00;  
ICR1H=0x00;  
ICR1L=0x00;  
OCR1AH=0x00;  
OCR1AL=0x00;  
OCR1BH=0x00;  
OCR1BL=0x00;  
OCR1CH=0x00;  
OCR1CL=0x00;

TCCR2=0x00;  
TCNT2=0x00;  
OCR2=0x00;

TCCR3A=0x00;  
TCCR3B=0x02;  
TCNT3H=0x00;  
TCNT3L=0x00;  
ICR3H=0x00;  
ICR3L=0x00;  
OCR3AH=0x00;  
OCR3AL=0x00;  
OCR3BH=0x00;  
OCR3BL=0x00;  
OCR3CH=0x00;  
OCR3CL=0x00;

EICRA=0x0F;  
EICRB=0x00;  
EIMSK=0x03;  
EIFR=0x03;

TIMSK=0x01;  
ETIMSK=0x04;

ACSR=0x80;  
SFIOA=0x00;

ADMUX=ADC\_VREF\_TYPE;  
ADCSRA=0x83;



```

#asm("sei")
LCD_INIT() ;

LCD_POS(0,0) ; LCD_DAT('T') ; LCD_DAT('M') ; LCD_DAT('P') ; LCD_DAT('=') ;
LCD_POS(0,1) ; LCD_DAT('R') ; LCD_DAT('P') ; LCD_DAT('M') ; LCD_DAT('=') ;
LCD_POS(12,1) ; LCD_DAT('P') ; LCD_DAT('W') ; LCD_DAT('M') ; LCD_DAT('=') ;


while (1)
{
ltmp = (1500L*(long)(ADC0 + 1))/(9216L) ;
    LCD_POS(4,0) ;
    IMSI = ltmp ;
    DAT2 = IMSI/100 ;
    IMSI = IMSI - DAT2*100 ;
    DAT1 = IMSI/10 ;
    DAT0 = IMSI - DAT1*10 ;


    LCD_DAT(DAT2 + '0') ; LCD_DAT(DAT1 + '0') ; LCD_DAT(DAT0 + '0') ;
    LCD_POS(7,0); LCD_DAT('C') ;


    Rpms = 2500000L/Tmr3 ;
    IMSI = Rpms ;
    DAT3 = IMSI/1000 ;
    IMSI = IMSI - DAT3*1000 ;
    DAT2 = IMSI/100 ;
    IMSI = IMSI - DAT2*100 ;
    DAT1 = IMSI/10 ;
    DAT0 = IMSI - DAT1*10 ;


    LCD_POS(4,1) ; LCD_DAT(DAT3+ '0') ; LCD_DAT(DAT2+ '0') ; LCD_DAT(DAT1+ '0') ;
    LCD_DAT(DAT0+ '0') ;


    IMSI = PWM1s ;
    DAT2 = IMSI/100 ;
    IMSI = IMSI - DAT2*100 ;
    DAT1 = IMSI/10 ;
    DAT0 = IMSI - DAT1*10 ;


    LCD_POS(16,1) ; LCD_DAT(DAT2+ '0') ; LCD_DAT(DAT1+ '0') ; LCD_DAT(DAT0+ '0') ;
}
}

```