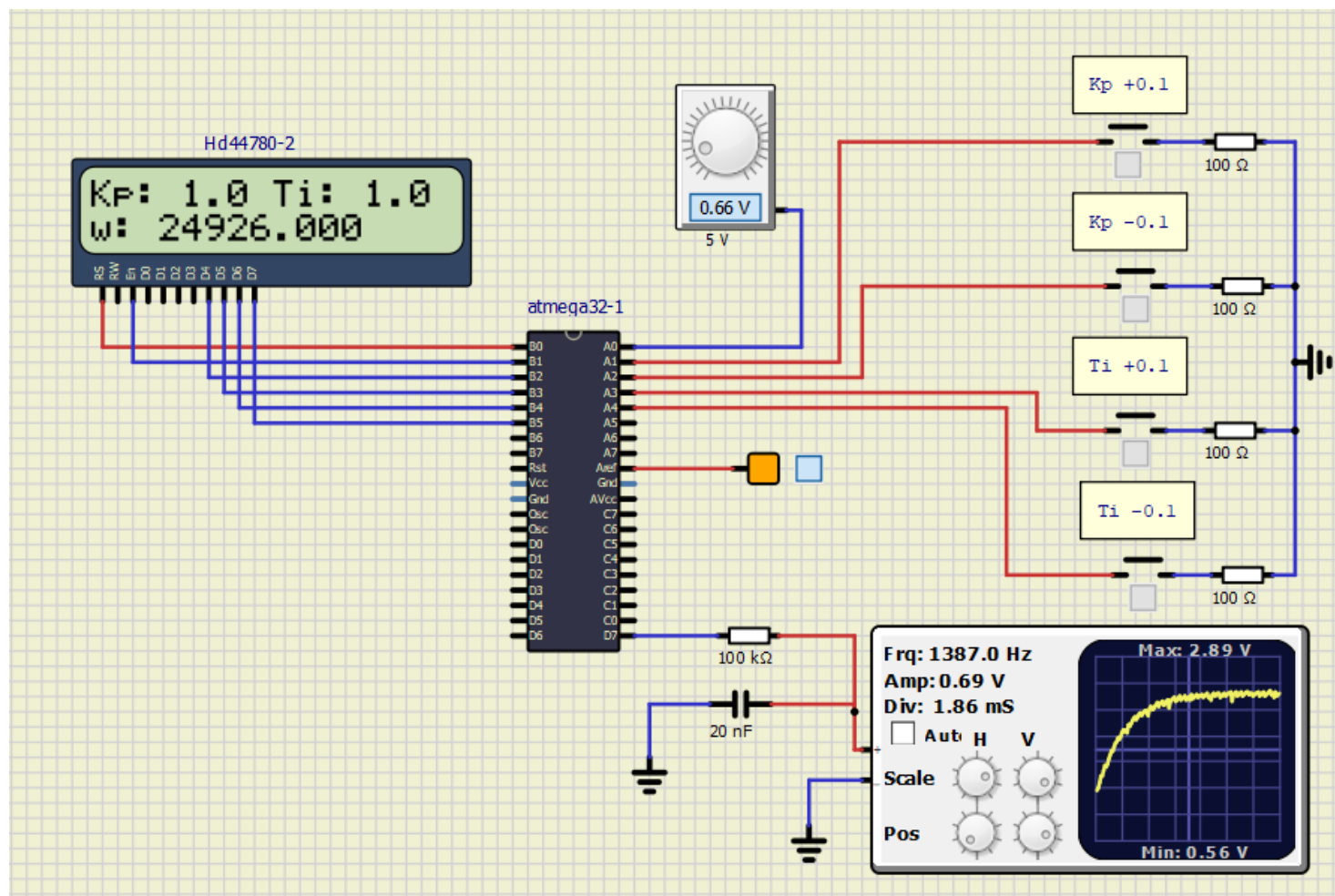


Sprawozdanie

1. Cel ćwiczenia

Celem ćwiczenia było stworzenie regulatora typu PI z regulowanymi wartościami T_i oraz K_p .

2. Schemat układu



3. Kod programu

```
#include <avr/io.h>
#include <stdlib.h>
#include <string.h>
#include <util/delay.h>
#include "GLOBAL.h"
#include "LCD_HD44780.h"
class adc
{
public:
    static void init()
    {
        ADMUX = 0;
        ADCSRA = 1<<ADPS0 | 1<< ADPS1 | 1<< ADPS2 | 1<<ADEN;
    }
    static int32_t czytaj()
    {
        ADCSRA|= (1<<ADSC);
        while (ADCSRA & (1<<ADSC));
        return ADC;
    }
};
```

```

    }
};
void PWM_init()
{
    TCCR0 = (1<<WGM00) | (1<<WGM01) | (1<<COM01) | (1<<CS00);
    TCCR2 = (1<<WGM00) | (1<<WGM01) | (1<<COM01) | (1<<CS00);
    DDRD |= (1<<PD7);
};
void wyswietl (double kp, double Ti, double wynik)
{
    char jeden[16], dwa[16], trzy[16];
    dtostrf(kp,0,1,jeden);
    dtostrf(Ti,0,1,dwa);
    dtostrf(wynik,0,3,trzy);
    LCD_HD44780::clear(); // czyszczenie lcd
    LCD_HD44780::writeText("Kp: ");
    LCD_HD44780::writeText(jeden);
    //LCD_HD44780::goTo(0,1);
    LCD_HD44780::writeText(" Ti: ");
    LCD_HD44780::writeText(dwa);
    LCD_HD44780::goTo(0,1);
    LCD_HD44780::writeText("w: ");
    LCD_HD44780::writeText(trzy);
}
void przyciski (double &kp, double &Ti)
{
    if(bit_is_clear(PINA,1))
    {
        kp+=0.1;
    }
    if(bit_is_clear(PINA,2) && kp>0.1)
    {
        kp-=0.1;
    }
    if(bit_is_clear(PINA,3))
    {
        Ti+=0.1;
    }
    if(bit_is_clear(PINA,4) && Ti>0.1)
    {
        Ti-=0.1;
    }
}
int main()
{
    PWM_init();
    double kp=1.0;
    double Ti=1.0;
    sbi(PORTA,PA1);
    sbi(PORTA,PA2);
    sbi(PORTA,PA3);
    sbi(PORTA,PA4);
    adc::init();
    LCD_HD44780::init();
    int32_t in;
    double wynik=0, wynik1=0, wynik2=0;
    while(true)
    {
        wyswietl(kp,Ti,wynik); // wyswietlanie kp i Ti na wyswietlaczu
        przyciski(kp,Ti);      // zmiana wartosci kp i Ti za pomoca przyciskow
        _delay_ms(500);
        in=adc::czytaj();
        wynik1=in*kp;
        wynik2+=(in/Ti);
        wynik=wynik1+wynik2;
    }
}

```

```

        if(wynik>30000)
        {
            wynik2=0;
        }
        OCR2=(int) wynik % 255;
    }
}

```

4. Wnioski

Na schemacie przedstawiono regulator typu PI z możliwością zmiany wartości k_p , T_i oraz wyjściem PWM. Jednorazowo za pomocą przycisku można zmienić wartości poszczególnych zmiennych o 0,1. Górne ograniczenie wartości wyjściowej zostało ustawione na 30000, aby zapobiec powstawaniu zjawiska windupu, a także, by pamięć zajmowana przez zmienną przechowującą wartość wyjściową nie zajęła całej pamięci mikrokontrolera. Wartości k_p , T_i oraz wartość na wyjściu są poprawnie wyświetlane na wyświetlaczu LCD. Dla lepszej widoczności wykresu na oscyloskopie został dodany filtr RC.

Wraz ze wzrostem wartości T_i spada tempo wzrostu wartości na wyjściu, jednakże dla każdej dodatniej wartości T_i wartość wyjściowa dąży do nieskończoności.

Wraz ze wzrostem wartości k_p wzrasta tempo wzrostu wartości na wyjściu, jednakże tak samo jak w przypadku zmiennej T_i , wartość wyjściowa dąży do nieskończoności.