# Temat: Badanie regulatora proporcjonalnego

Michał Prośba 248986

## 1. Zadanie do wykonania

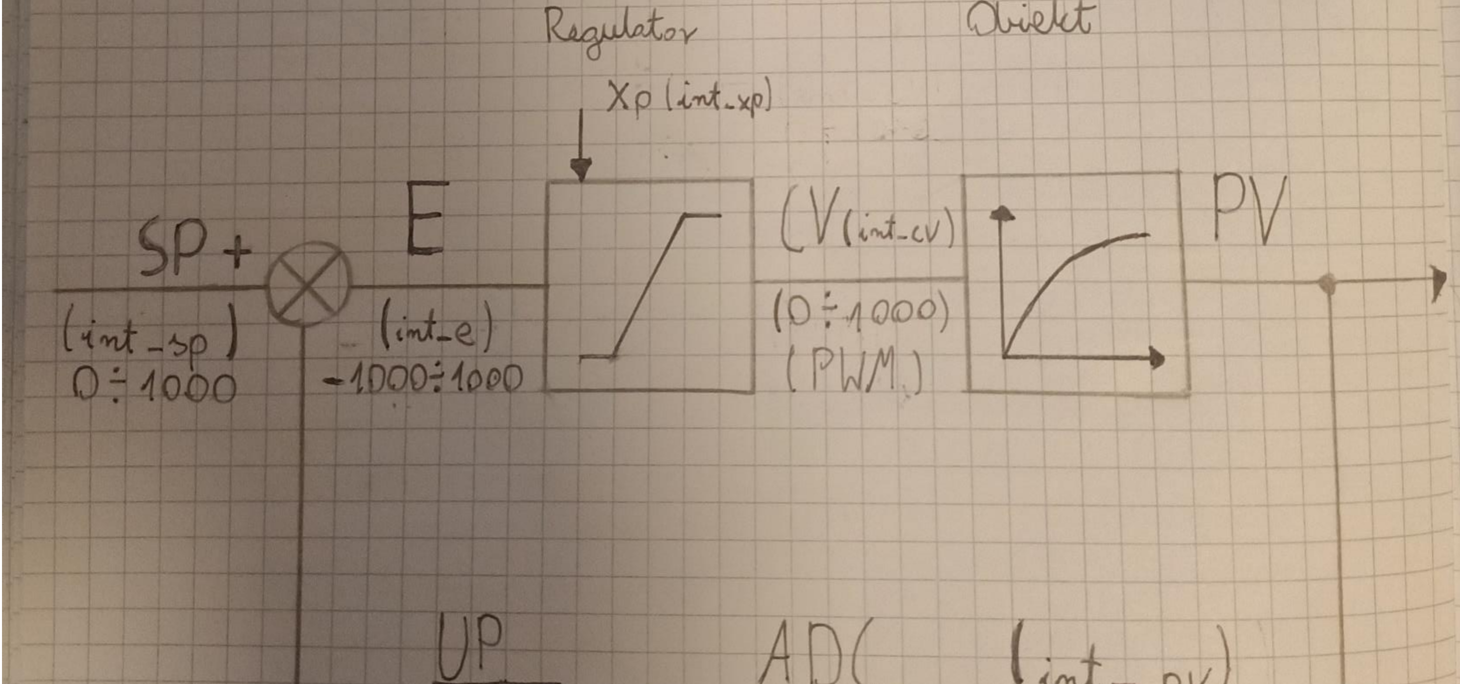
Opracować układ pomiarowy, zmontować układ do badania regulatora, opracować algorytm sterowania w układzie regulacji proporcjonalnej i przetestować regulator w warunkach laboratoryjnych.

Ć.1. Badanie toru wykonawczego PWM

Ć.2. Badanie regulatora

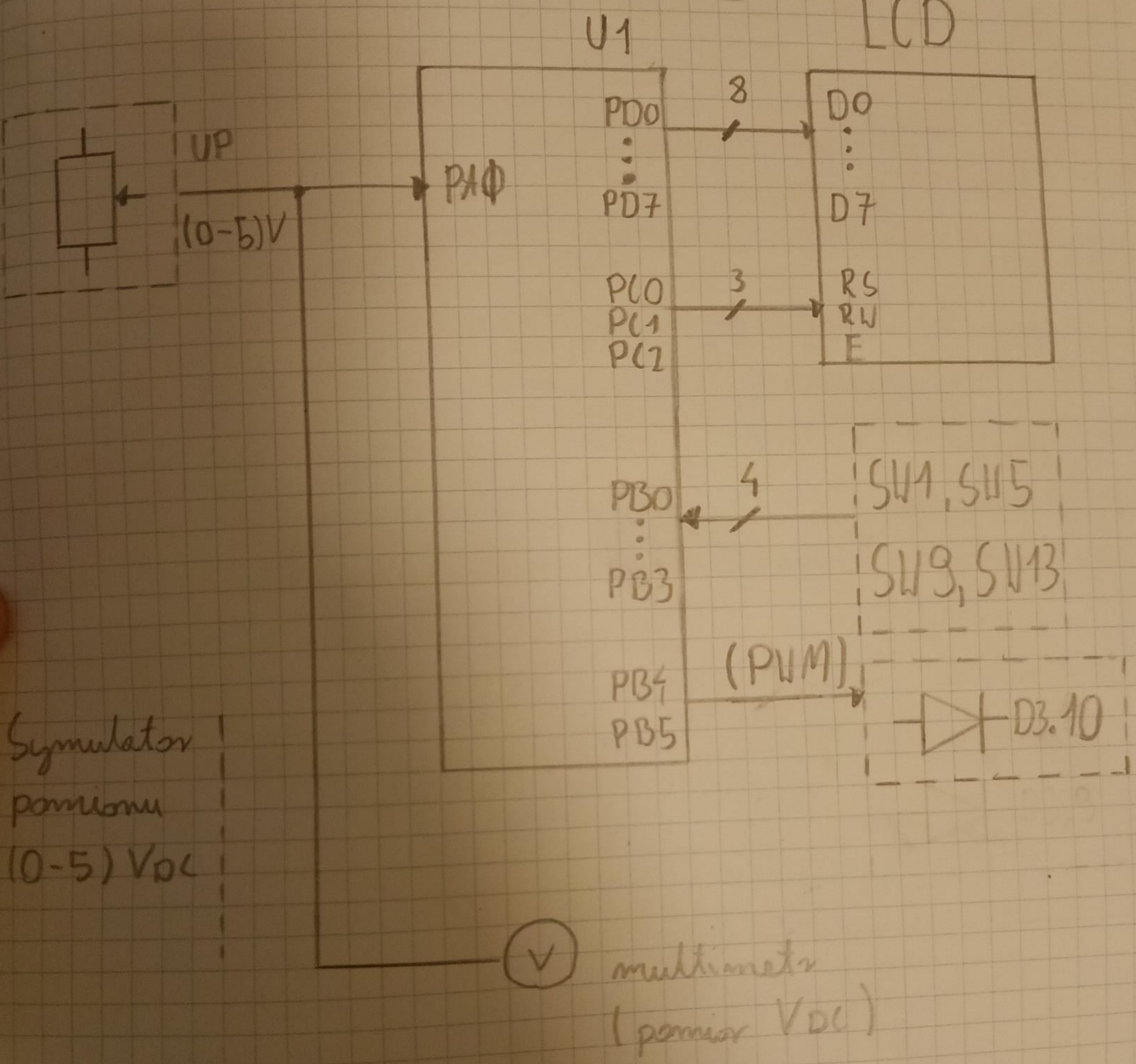
## 2. Założenie projektowe

### 2.1. Schemat blokowy typowego układu regulacji



## 2.2. Schemat blokowy podłączenia sygnałów w układzie do badania regulatora

Michał Prośba 248986



## 2.3. Zadawanie parametrów regulacji Michał Prośla

Zakres pomiarowy (0-400)°C / (0-5)V

a) Po RESET SP=60%, $X_p$=20%

b) Gdy SW1=1, SP=50%

c) Gdy SW5=1, SP=40%

d) Gdy SW9=1, $X_p$=30%

e) Gdy SW13=1, $X_p$=40%

## 2.4. Projekt wykorzystania wyświetlacza LCD

Wariant I SP=xx% PV=xx.x%

$X_p$=xx% E=±xx.x%

## 3. Regula

A'gorzyt

Dane: C

PV/(po

Założenie

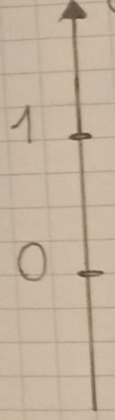
zakres ol

rozdziel

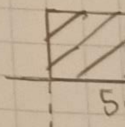
do odmie

testów 10

oraz sp

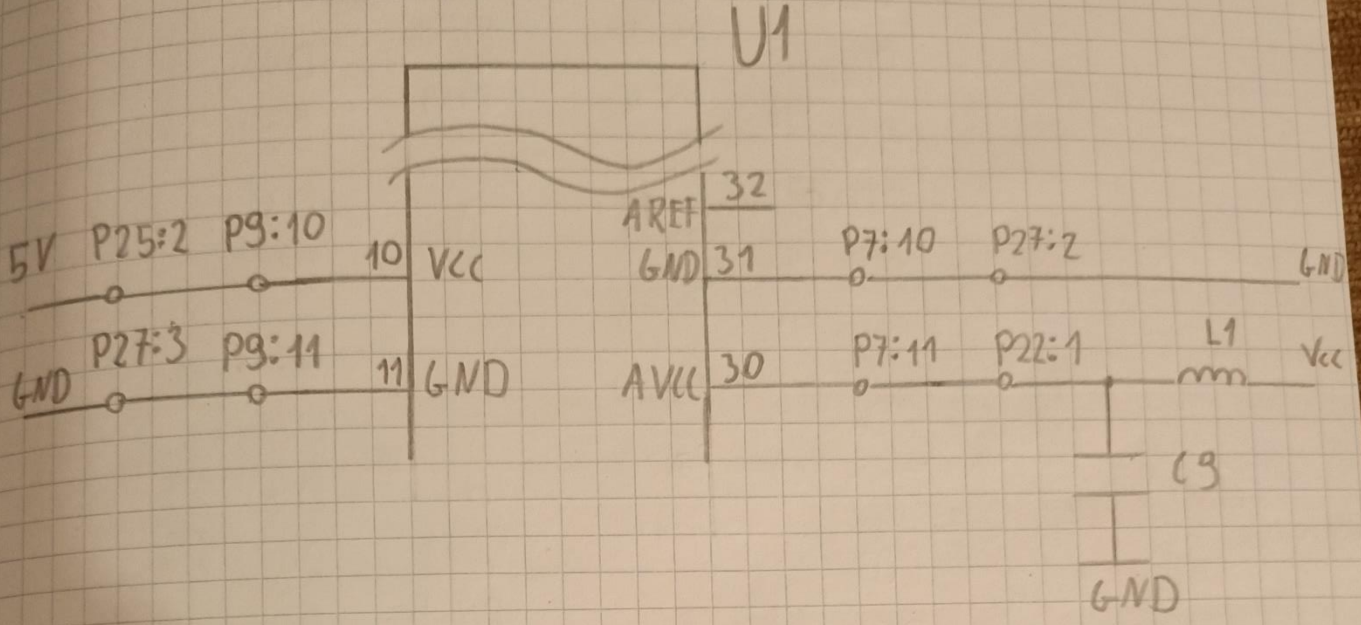


Ste

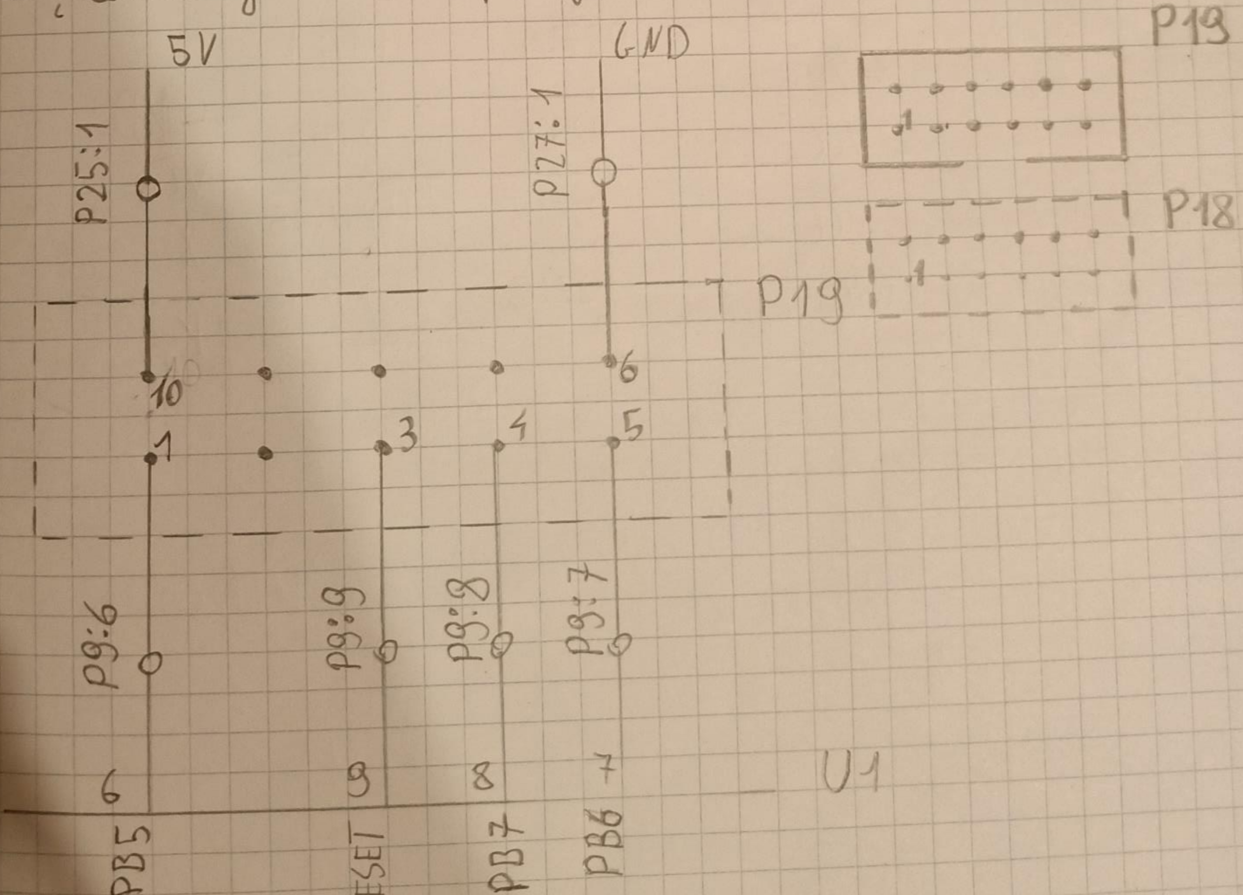


## 2.5. Schematy ideowe połączeń elektrycznych  Michał Prośba

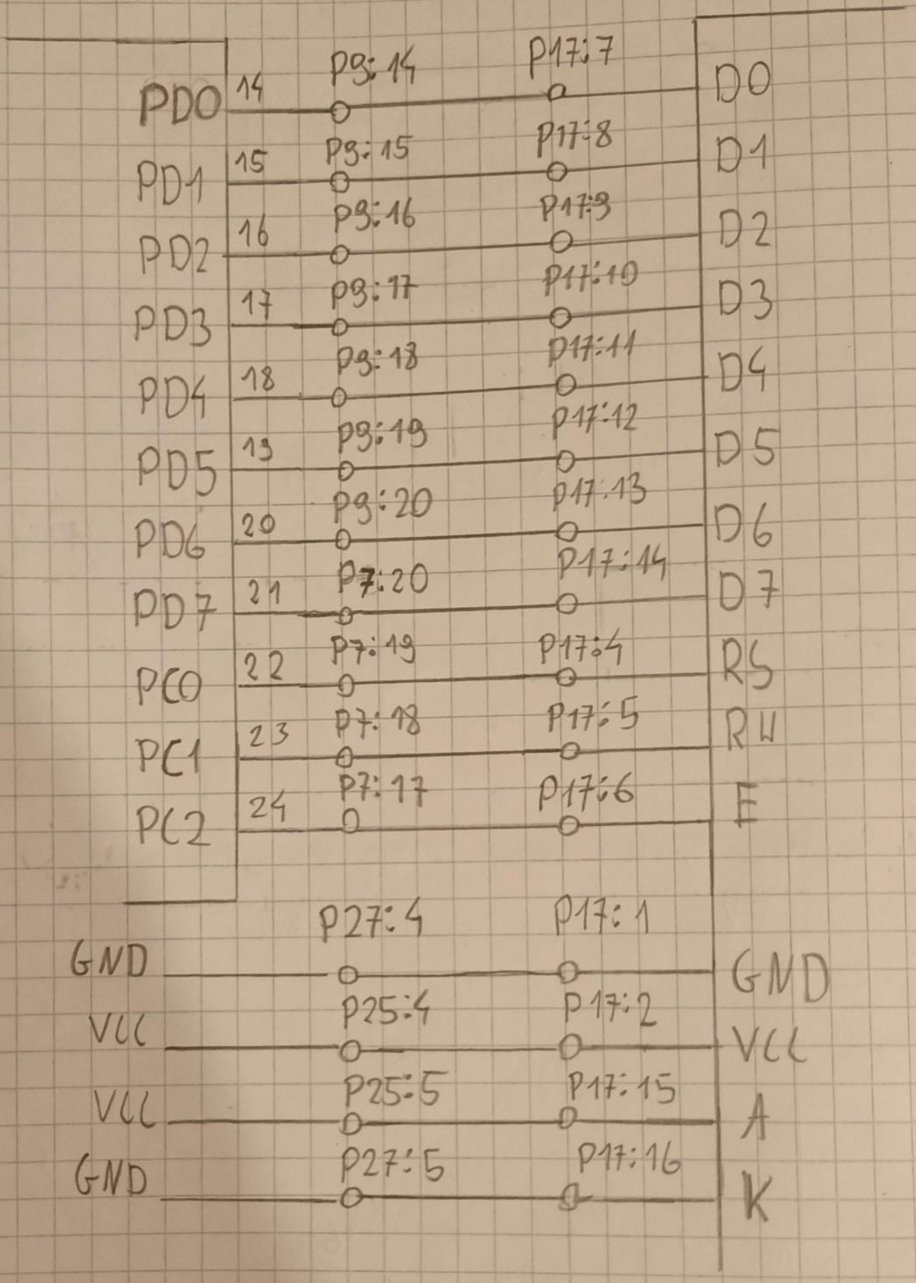
### a) Podłączenie zasilania mikrokontrolera



### b) Podłączenie gniazda programatora



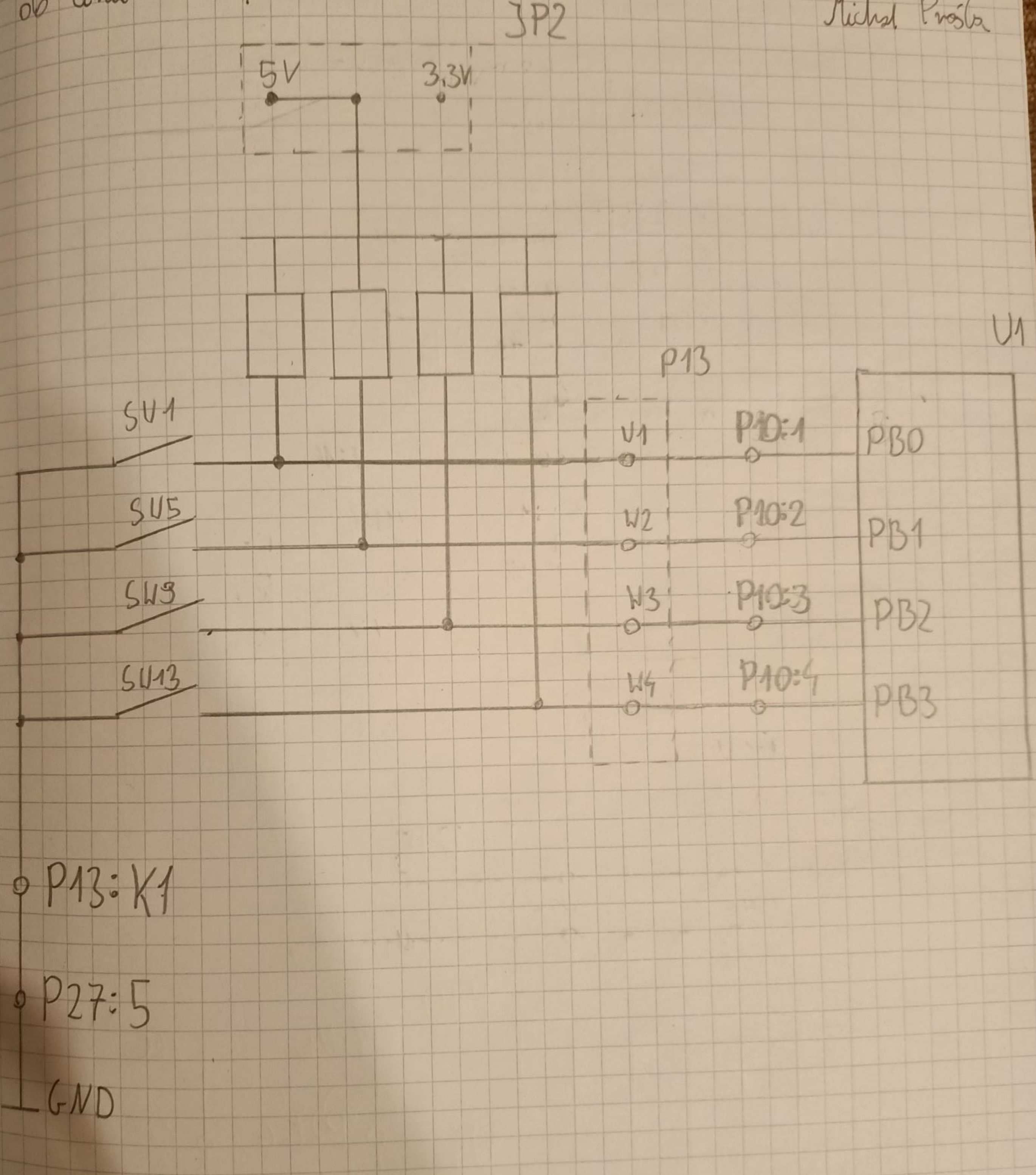
c) Podłączenie wyświetlacza LCD do mikroprocesora Michał Prasiła



Prośla

d) Podłączenie czterech przycisków (SW1, SW5, SW9, SW13) do linii mikroprocesora

Michał Prośla



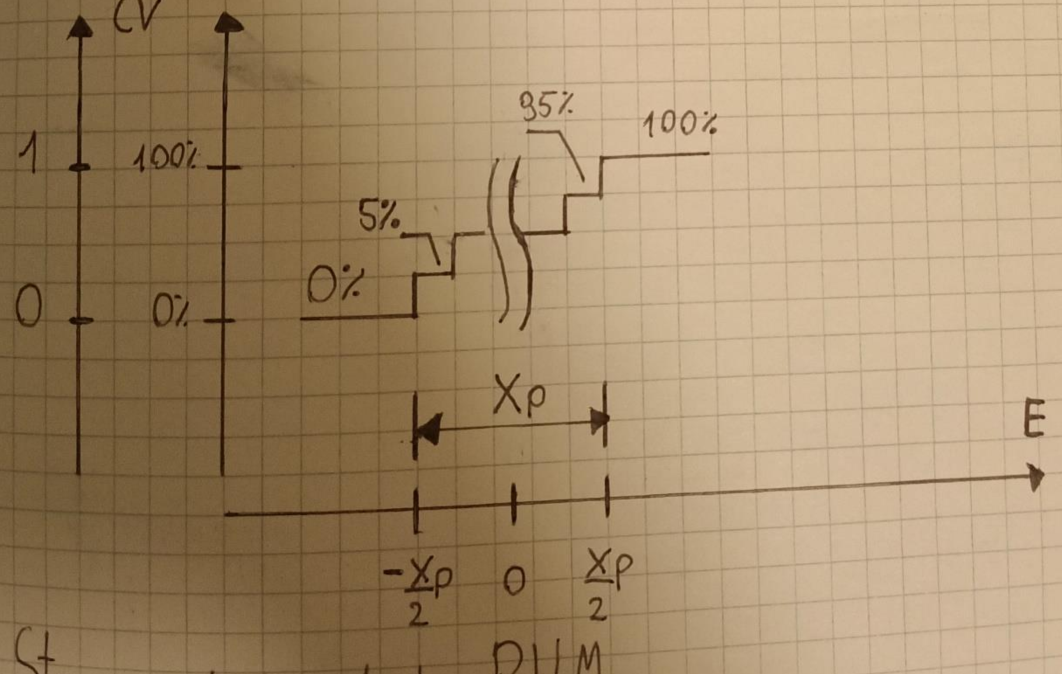
3. Regulator proporcjonalny    Michał Prośba 248986

Algorytm działania:

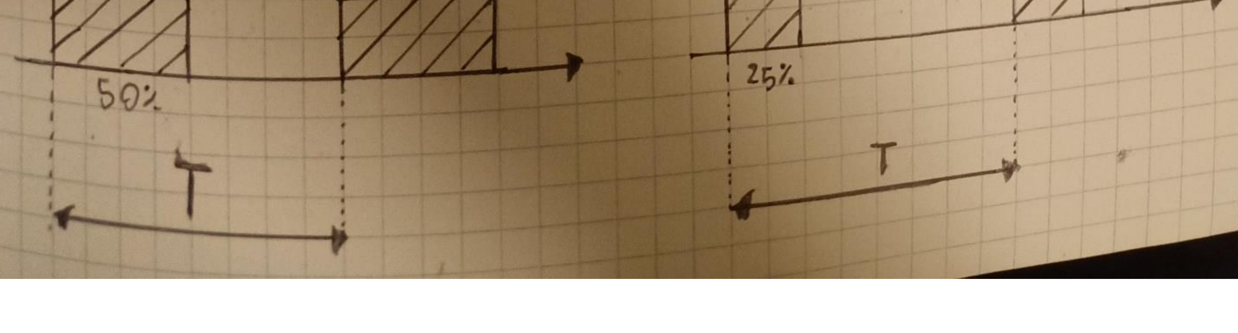
Dane: SP (wartość zadana), Xp (zakres proporcjonalności), PV (pomiar, zmienna procesowa)

Założenie dodatkowe:

okres dla sygnału PWM T0 = 0,2 [s] (dla testów T0 = 20 [s]), rozdzielczość sygnału sterującego 5% (CV = 0%, 5%, 10%, ... 95%, 100%), do odmierzania czasu użyj funkcji delay_ms (ms = 100, albo dla testów 1000). Opisać szczegółowo sposób obliczania sterowania (CV) oraz sposób przejścia na sygnał PUM.



Sterowanie metodą PWM



4. Tabela pomiarowa Michał Prośba 248386

Badanie regulatora dla SP = 60%, $X_p$ = 20%, TO = 200[ms]

Autorzy: Michał Prośba, Patryk Wieczorek

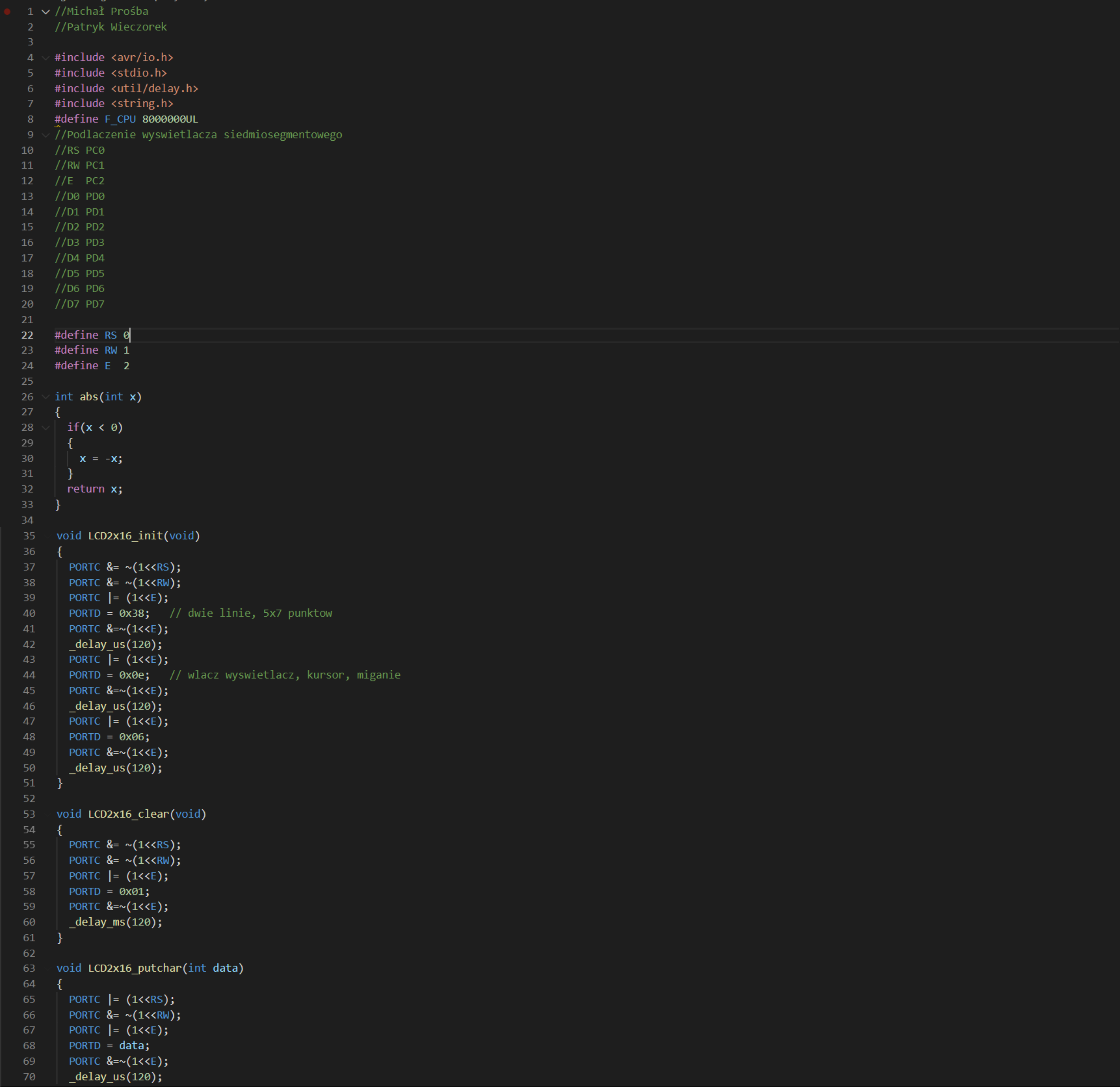
Zakres pomiarowy: (0-400)°C / (0-5)V

| E [Xp] | E[%] | PV[%] | PV[ADC] | PV[°C] | PV [V] | CV[%] | tH [ms]<br>pomiar | tH [s]/200[ms]<br>× 100% |
|---|---|---|---|---|---|---|---|---|
| -1,00Xp | -20,0 | 80,0 | 818 | 320,0 | 4,00 | 0,0 | 0,00 | 0,0 |
| -0,55Xp | -11,0 | 71,0 | 726 | 284,0 | 3,55 | 0,0 | 0,00 | 0,0 |
| -0,50Xp | -10,0 | 70,0 | 716 | 280,0 | 3,50 | 0,0 | 0,00 | 0,0 |
| -0,45Xp | -9,0 | 69,0 | 706 | 276,0 | 3,45 | 5,0 | 10,80 | 5,4 |
| -0,40Xp | -8,0 | 68,0 | 696 | 272,0 | 3,40 | 10,0 | 20,40 | 10,2 |
| -0,20Xp | -4,0 | 64,0 | 655 | 256,0 | 3,20 | 30,0 | 61,20 | 30,6 |
| -0,10Xp | -2,0 | 62,0 | 634 | 248,0 | 3,10 | 40,0 | 80,80 | 40,4 |
| 0,00Xp | 0,0 | 60,0 | 614 | 240,0 | 3,00 | 50,0 | 100,60 | 50,3 |
| 0,10Xp | 2,0 | 58,0 | 593 | 232,0 | 2,90 | 60,0 | 120,40 | 60,2 |
| 0,20Xp | 4,0 | 56,0 | 573 | 224,0 | 2,80 | 70,0 | 139,40 | 69,7 |
| 0,40Xp | 8,0 | 52,0 | 532 | 208,0 | 2,60 | 90,0 | 180,60 | 90,3 |
| 0,45Xp | 9,0 | 51,0 | 522 | 204,0 | 2,55 | 95,0 | 191,20 | 95,6 |
| 0,50Xp | 10,0 | 50,0 | 512 | 200,0 | 2,50 | 100,0 | 200,00 | 100,0 |
| 0,55Xp | 11,0 | 49,0 | 501 | 196,0 | 2,45 | 100,0 | 200,00 | 100,0 |
| 1,00Xp | 20,0 | 40,0 | 409 | 160,0 | 2,00 | 100,0 | 200,00 | 100,0 |

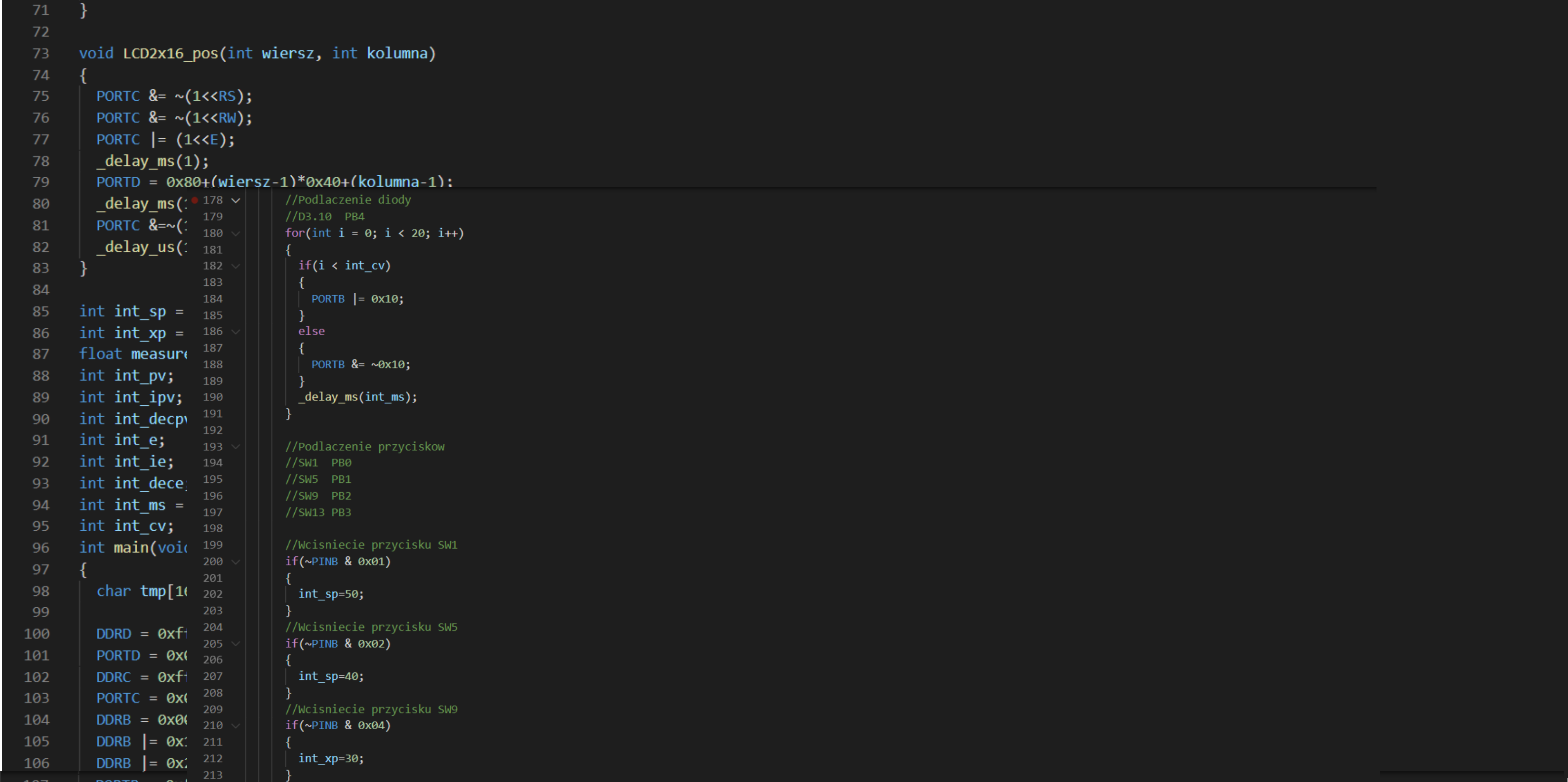
## 5. Uwagi i wnioski Michał Prośba 248986

Progi przełączania regulatora proporcjonalnego zgadzają się z założeniami projektowymi. Dioda D3.10 ma 100% wypełnienia dla $E \geqslant X_p/2$, oraz 0% wypełnienia dla $E \leqslant -X_p/2$. Regulator działa poprawnie.

## 6. Załącznik nr. 1: Kod programu i schemat symulacji

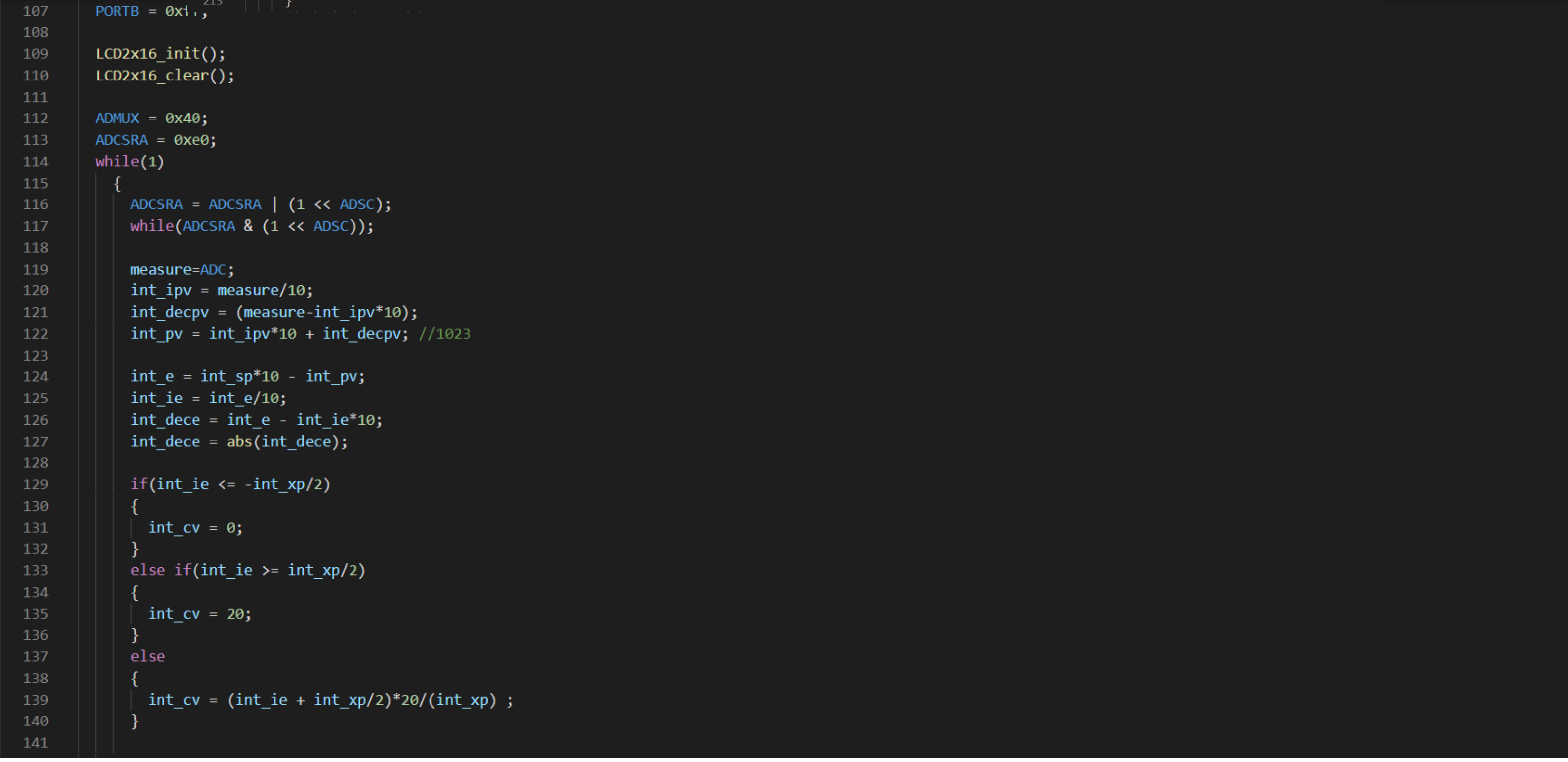


```c
//Michał Prośba
//Patryk Wieczorek

#include <avr/io.h>
#include <stdio.h>
#include <util/delay.h>
#include <string.h>
#define F_CPU 8000000UL
//Podlaczenie wyswietlacza siedmiosegmentowego
//RS PC0
//RW PC1
//E  PC2
//D0 PD0
//D1 PD1
//D2 PD2
//D3 PD3
//D4 PD4
//D5 PD5
//D6 PD6
//D7 PD7

#define RS 0
#define RW 1
#define E  2

int abs(int x)
{
  if(x < 0)
  {
    x = -x;
  }
  return x;
}

void LCD2x16_init(void)
{
  PORTC &= ~(1<<RS);
  PORTC &= ~(1<<RW);
  PORTC |= (1<<E);
  PORTD = 0x38;   // dwie linie, 5x7 punktow
  PORTC &=~(1<<E);
  _delay_us(120);
  PORTC |= (1<<E);
  PORTD = 0x0e;   // wlacz wyswietlacz, kursor, miganie
  PORTC &=~(1<<E);
  _delay_us(120);
  PORTC |= (1<<E);
  PORTD = 0x06;
  PORTC &=~(1<<E);
  _delay_us(120);
}

void LCD2x16_clear(void)
{
  PORTC &= ~(1<<RS);
  PORTC &= ~(1<<RW);
  PORTC |= (1<<E);
  PORTD = 0x01;
  PORTC &=~(1<<E);
  _delay_ms(120);
}

void LCD2x16_putchar(int data)
{
  PORTC |= (1<<RS);
  PORTC &= ~(1<<RW);
  PORTC |= (1<<E);
  PORTD = data;
  PORTC &=~(1<<E);
  _delay_us(120);
```

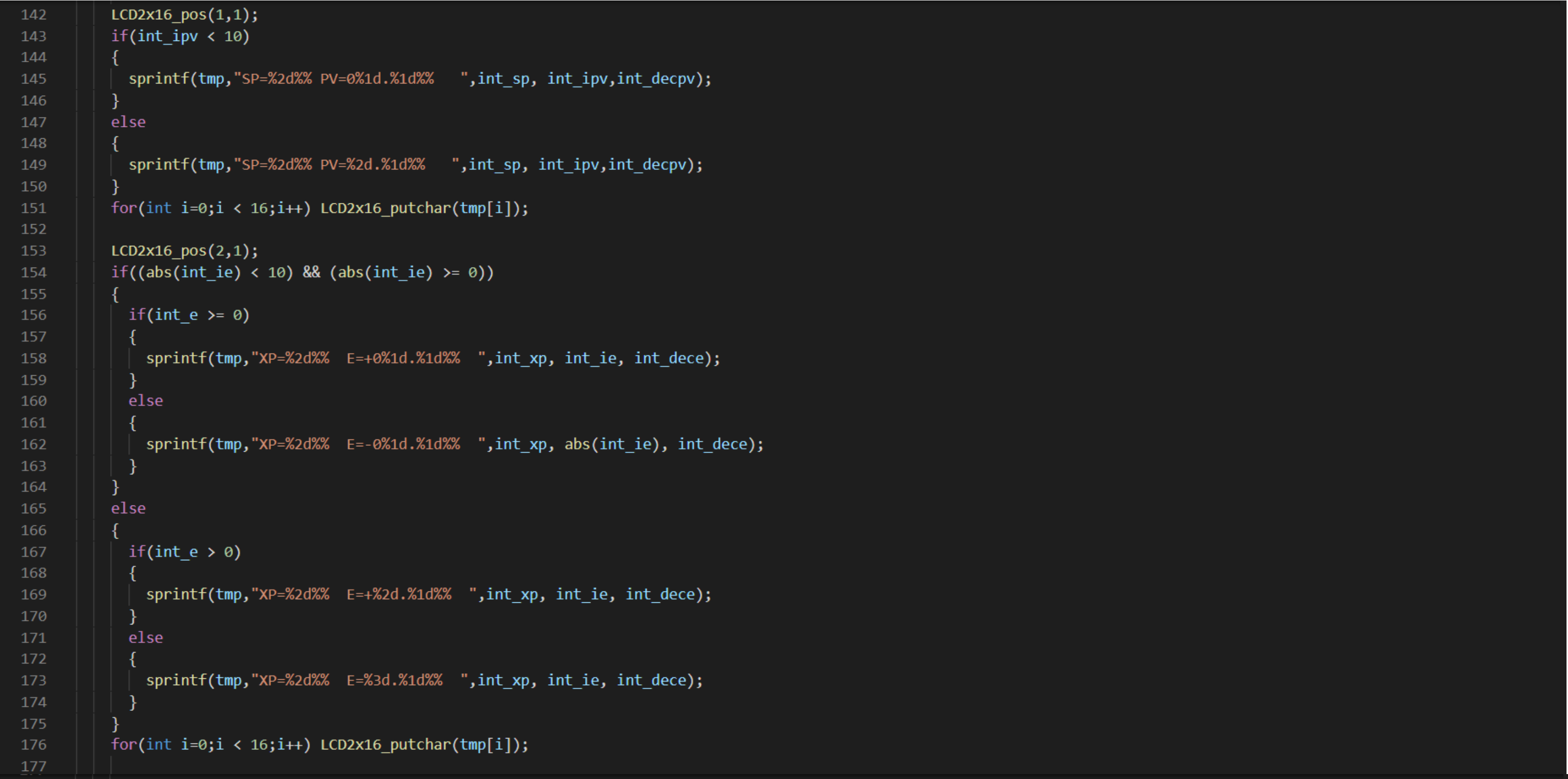


```c
}

void LCD2x16_pos(int wiersz, int kolumna)
{
  PORTC &= ~(1<<RS);
  PORTC &= ~(1<<RW);
  PORTC |= (1<<E);
  _delay_ms(1);
  PORTD = 0x80+(wiersz-1)*0x40+(kolumna-1);
  _delay_ms(
  PORTC &=~(
  _delay_us(
}

int int_sp =
int int_xp =
float measure
int int_pv;
int int_ipv;
int int_decp
int int_e;
int int_ie;
int int_dece
int int_ms =
int int_cv;
int main(voi
{
  char tmp[1

  DDRD = 0xf
  PORTD = 0x
  DDRC = 0xf
  PORTC = 0x
  DDRB = 0x0
  DDRB |= 0x
  DDRB |= 0x
```

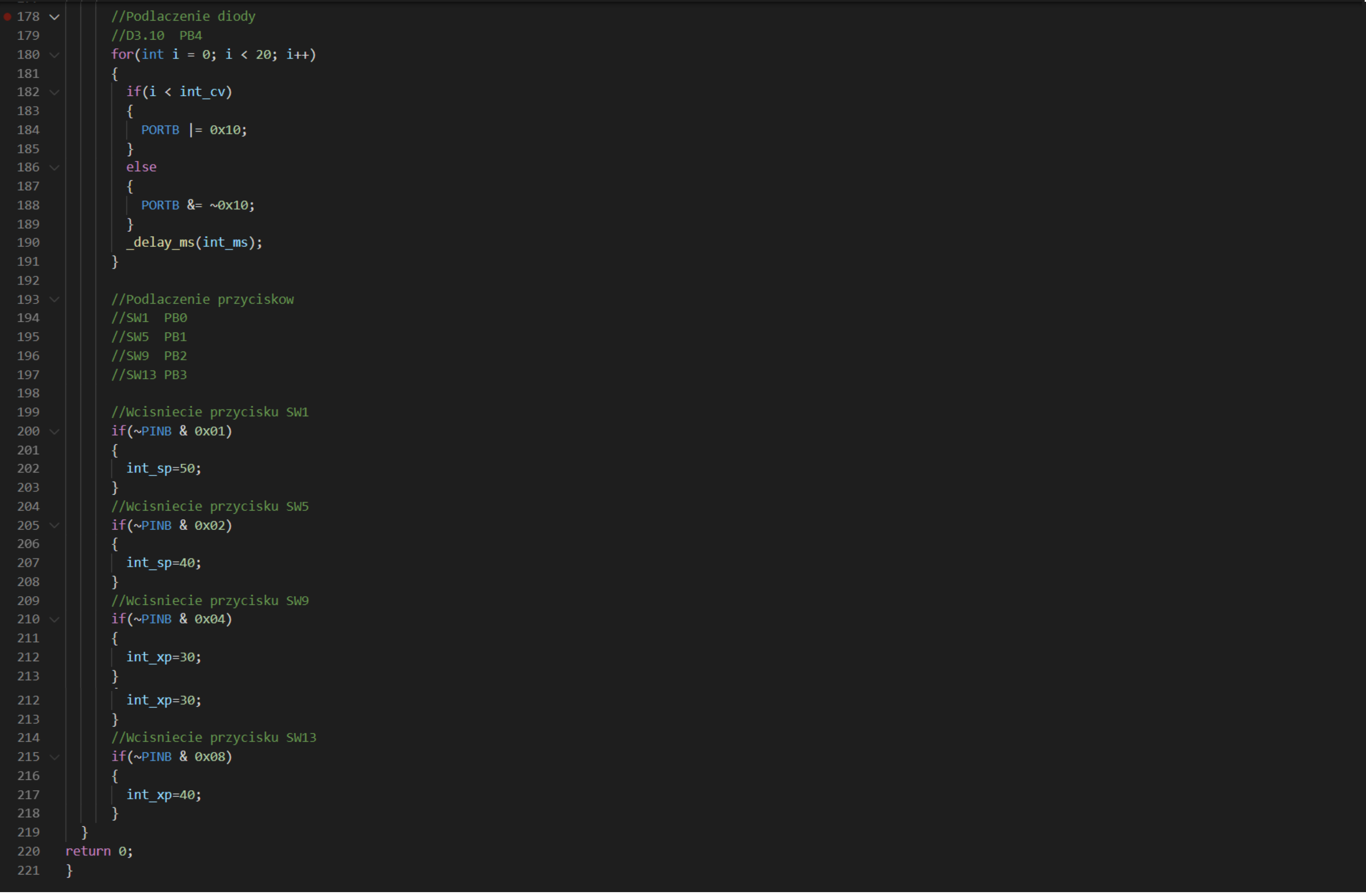
```c
  //Podlaczenie diody
  //D3.10  PB4
  for(int i = 0; i < 20; i++)
  {
    if(i < int_cv)
    {
      PORTB |= 0x10;
    }
    else
    {
      PORTB &= ~0x10;
    }
    _delay_ms(int_ms);
  }

  //Podlaczenie przyciskow
  //SW1  PB0
  //SW5  PB1
  //SW9  PB2
  //SW13 PB3

  //Wcisniecie przycisku SW1
  if(~PINB & 0x01)
  {
    int_sp=50;
  }
  //Wcisniecie przycisku SW5
  if(~PINB & 0x02)
  {
    int_sp=40;
  }
  //Wcisniecie przycisku SW9
  if(~PINB & 0x04)
  {
    int_xp=30;
  }
```



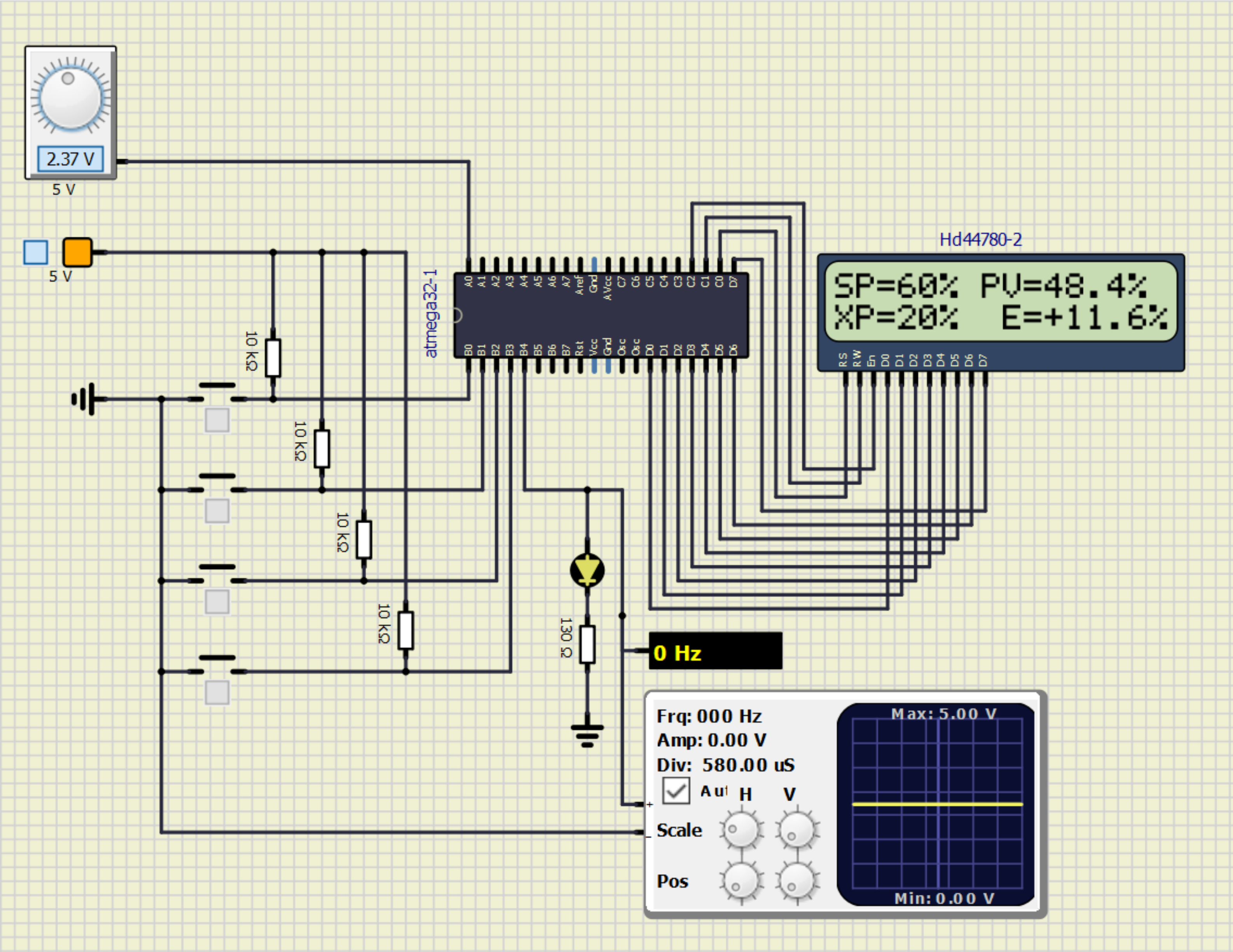
```c
  PORTB = 0x  ;

  LCD2x16_init();
  LCD2x16_clear();

  ADMUX = 0x40;
  ADCSRA = 0xe0;
  while(1)
    {
      ADCSRA = ADCSRA | (1 << ADSC);
      while(ADCSRA & (1 << ADSC));

      measure=ADC;
      int_ipv = measure/10;
      int_decpv = (measure-int_ipv*10);
      int_pv = int_ipv*10 + int_decpv; //1023

      int_e = int_sp*10 - int_pv;
      int_ie = int_e/10;
      int_dece = int_e - int_ie*10;
      int_dece = abs(int_dece);

      if(int_ie <= -int_xp/2)
      {
        int_cv = 0;
      }
      else if(int_ie >= int_xp/2)
      {
        int_cv = 20;
      }
      else
      {
        int_cv = (int_ie + int_xp/2)*20/(int_xp) ;
      }

```



```
    LCD2x16_pos(1,1);
    if(int_ipv < 10)
    {
      sprintf(tmp,"SP=%2d%% PV=0%1d.%1d%%   ",int_sp, int_ipv,int_decpv);
    }
    else
    {
      sprintf(tmp,"SP=%2d%% PV=%2d.%1d%%   ",int_sp, int_ipv,int_decpv);
    }
    for(int i=0;i < 16;i++) LCD2x16_putchar(tmp[i]);

    LCD2x16_pos(2,1);
    if((abs(int_ie) < 10) && (abs(int_ie) >= 0))
    {
      if(int_e >= 0)
      {
        sprintf(tmp,"XP=%2d%%  E=+0%1d.%1d%%  ",int_xp, int_ie, int_dece);
      }
      else
      {
        sprintf(tmp,"XP=%2d%%  E=-0%1d.%1d%%  ",int_xp, abs(int_ie), int_dece);
      }
    }
    else
    {
      if(int_e > 0)
      {
        sprintf(tmp,"XP=%2d%%  E=+%2d.%1d%%  ",int_xp, int_ie, int_dece);
      }
      else
      {
        sprintf(tmp,"XP=%2d%%  E=%3d.%1d%%  ",int_xp, int_ie, int_dece);
      }
    }
    for(int i=0;i < 16;i++) LCD2x16_putchar(tmp[i]);

```



```
      //Podlaczenie diody
      //D3.10  PB4
      for(int i = 0; i < 20; i++)
      {
        if(i < int_cv)
        {
          PORTB |= 0x10;
        }
        else
        {
          PORTB &= ~0x10;
        }
        _delay_ms(int_ms);
      }

      //Podlaczenie przyciskow
      //SW1  PB0
      //SW5  PB1
      //SW9  PB2
      //SW13 PB3

      //Wcisniecie przycisku SW1
      if(~PINB & 0x01)
      {
        int_sp=50;
      }
      //Wcisniecie przycisku SW5
      if(~PINB & 0x02)
      {
        int_sp=40;
      }
      //Wcisniecie przycisku SW9
      if(~PINB & 0x04)
      {
        int_xp=30;
      }
        int_xp=30;
      }
      //Wcisniecie przycisku SW13
      if(~PINB & 0x08)
      {
        int_xp=40;
      }
  }
return 0;
}
```

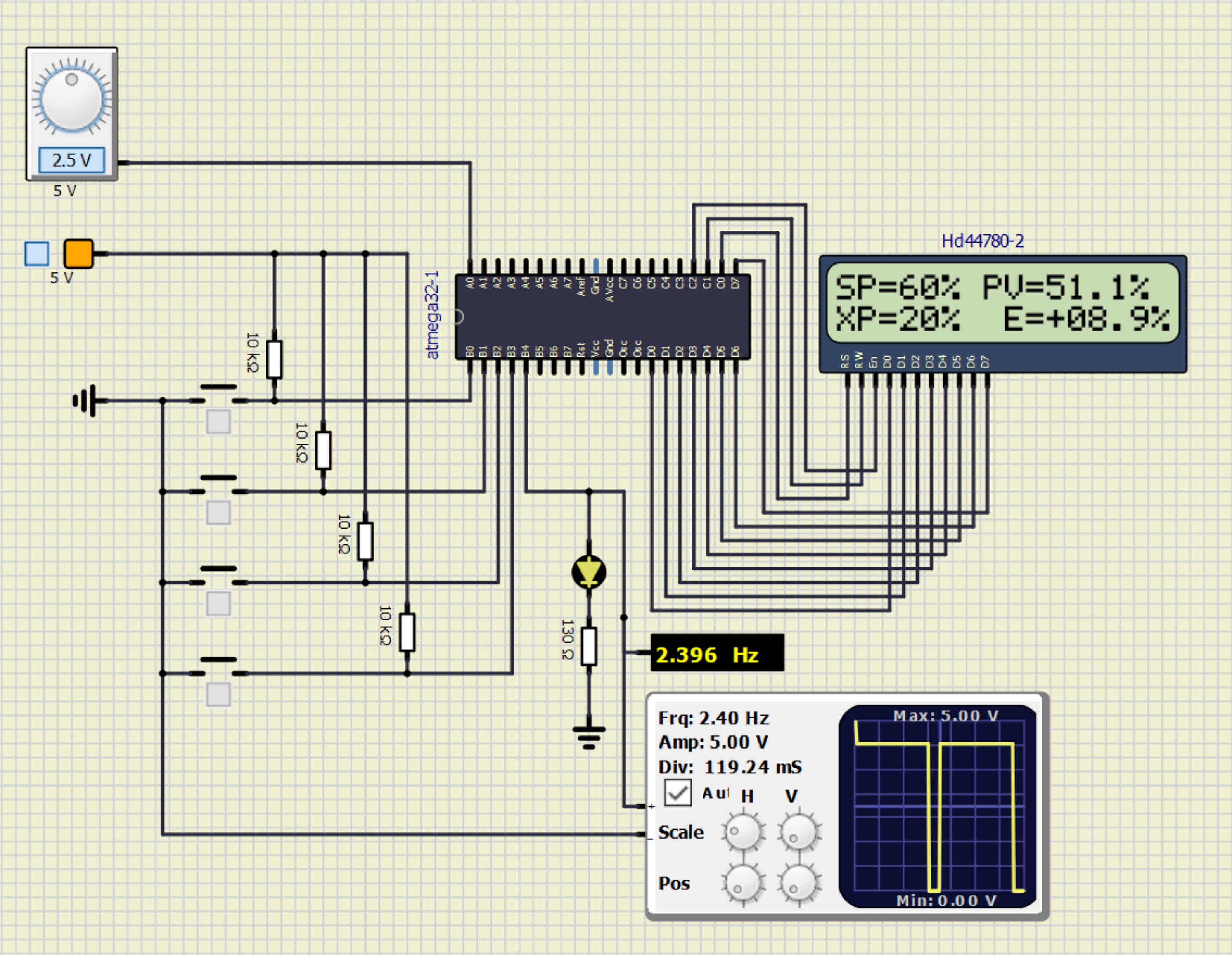


2.37 V
5 V
5 V
10 kΩ
10 kΩ
10 kΩ
10 kΩ
atmega32-1
A0 A1 A2 A3 A4 A5 A6 A7 Aref Gnd AVcc C7 C6 C5 C4 C3 C2 C1 C0 D7
B0 B1 B2 B3 B4 B5 B6 B7 Rst Vcc Gnd Osc Osc D0 D1 D2 D3 D4 D5 D6
Hd44780-2
RS RW En D0 D1 D2 D3 D4 D5 D6 D7
130 Ω
0 Hz
Aut H V
Scale
Pos
Max: 5.00 V
Min: 0.00 V

```
SP=60%  PV=48.4%
XP=20%   E=+11.6%
```

Frq: 000 Hz
Amp: 0.00 V
Div: 580.00 uS
☑ Aut

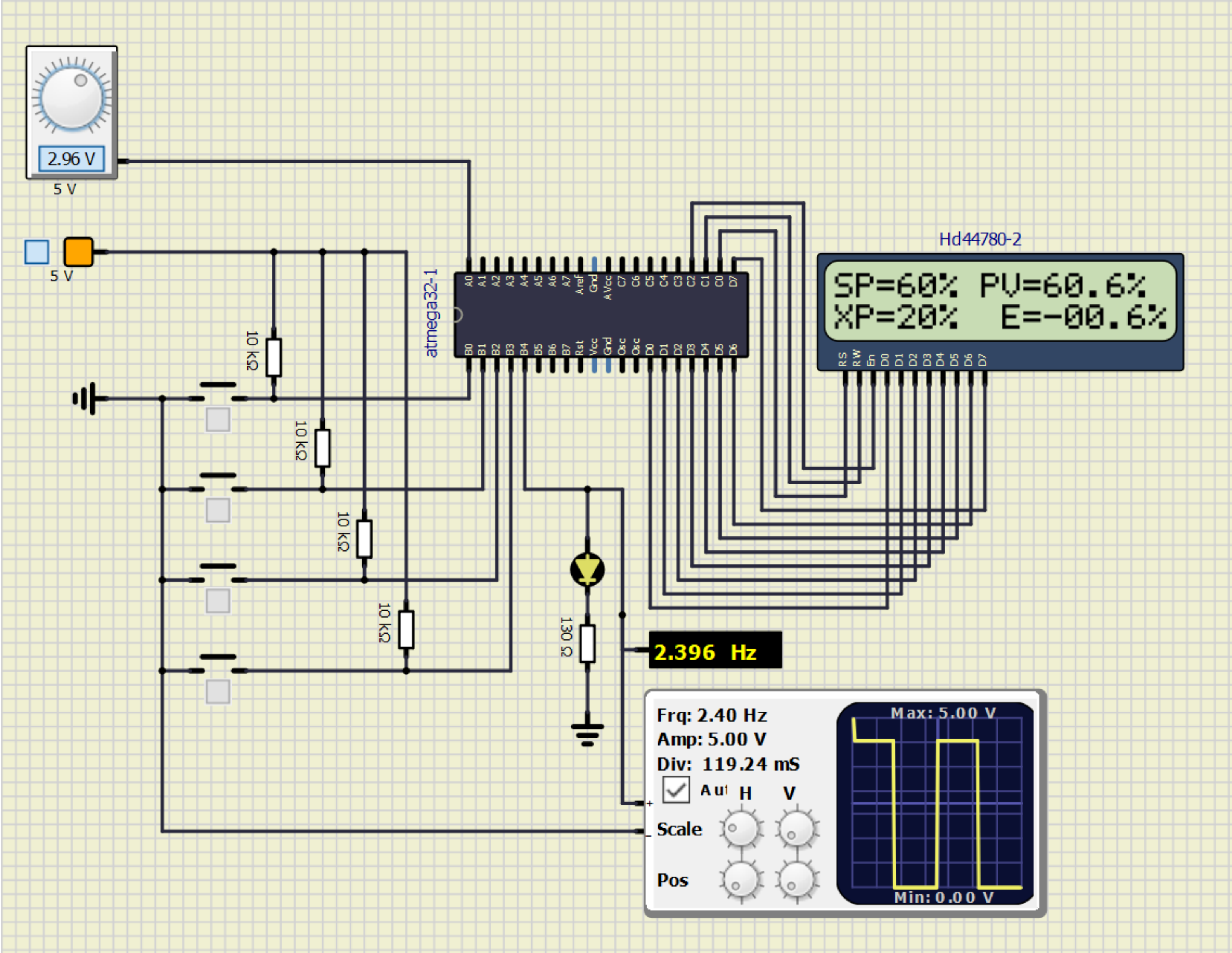
CV – 100%


2.5 V
5 V
5 V
10 kΩ
10 kΩ
10 kΩ
10 kΩ
130 Ω
atmega32-1
A0 A1 A2 A3 A4 A5 A6 A7 Aref Gnd AVcc C7 C6 C5 C4 C3 C2 C1 C0 D7
B0 B1 B2 B3 B4 B5 B6 B7 Rst Vcc Gnd Osc Osc D0 D1 D2 D3 D4 D5 D6
Hd44780-2
RS RW En D0 D1 D2 D3 D4 D5 D6 D7
2.396 Hz
Aut H V
Scale
Pos
Max: 5.00 V
Min: 0.00 V

SP=60% PV=51.1%
XP=20% E=+08.9%

Frq: 2.40 Hz
Amp: 5.00 V
Div: 119.24 mS

CV – około 90%



```
SP=60% PV=60.6%
XP=20%  E=-00.6%
```

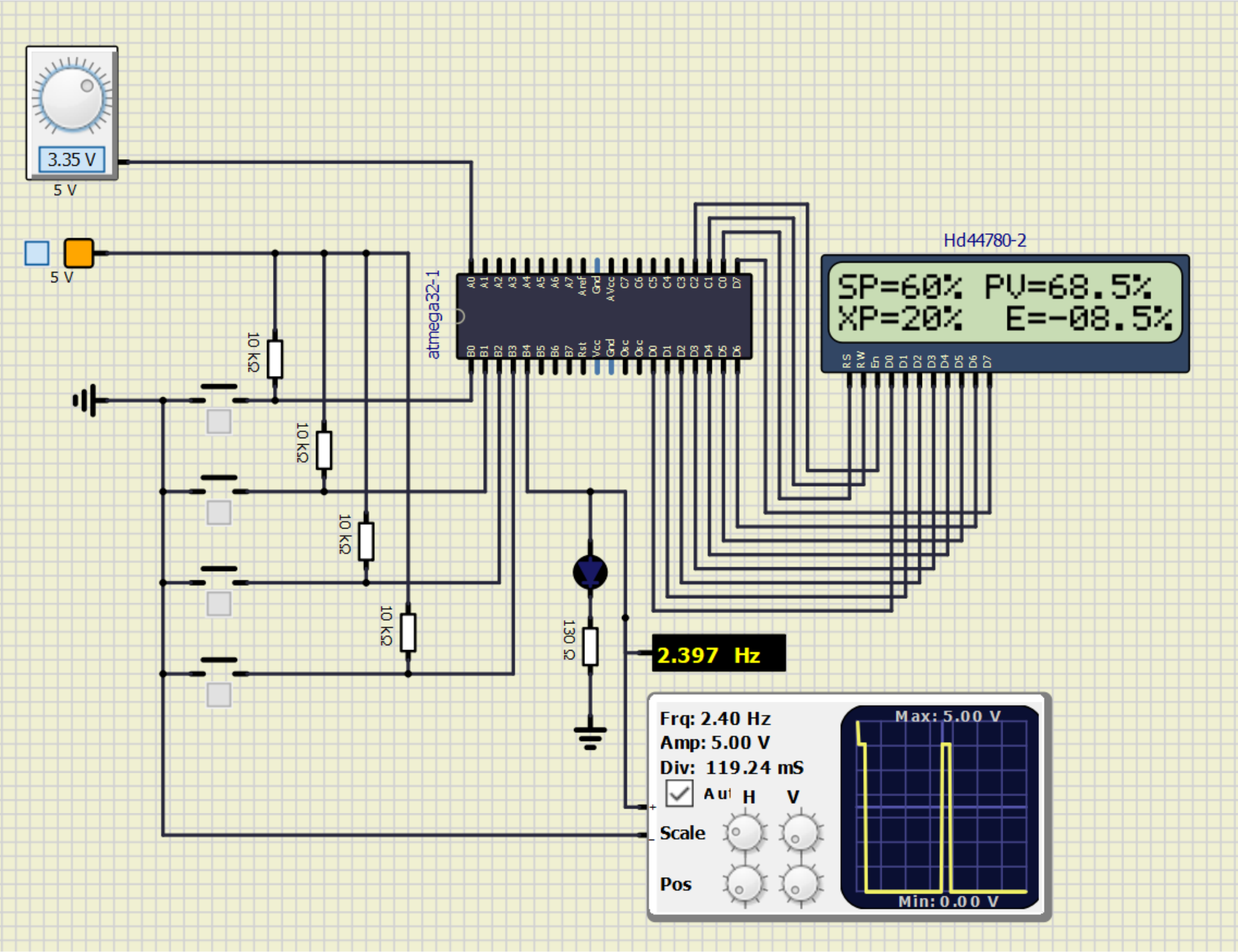
2.396 Hz

Frq: 2.40 Hz
Amp: 5.00 V
Div: 119.24 mS

Max: 5.00 V
Min: 0.00 V

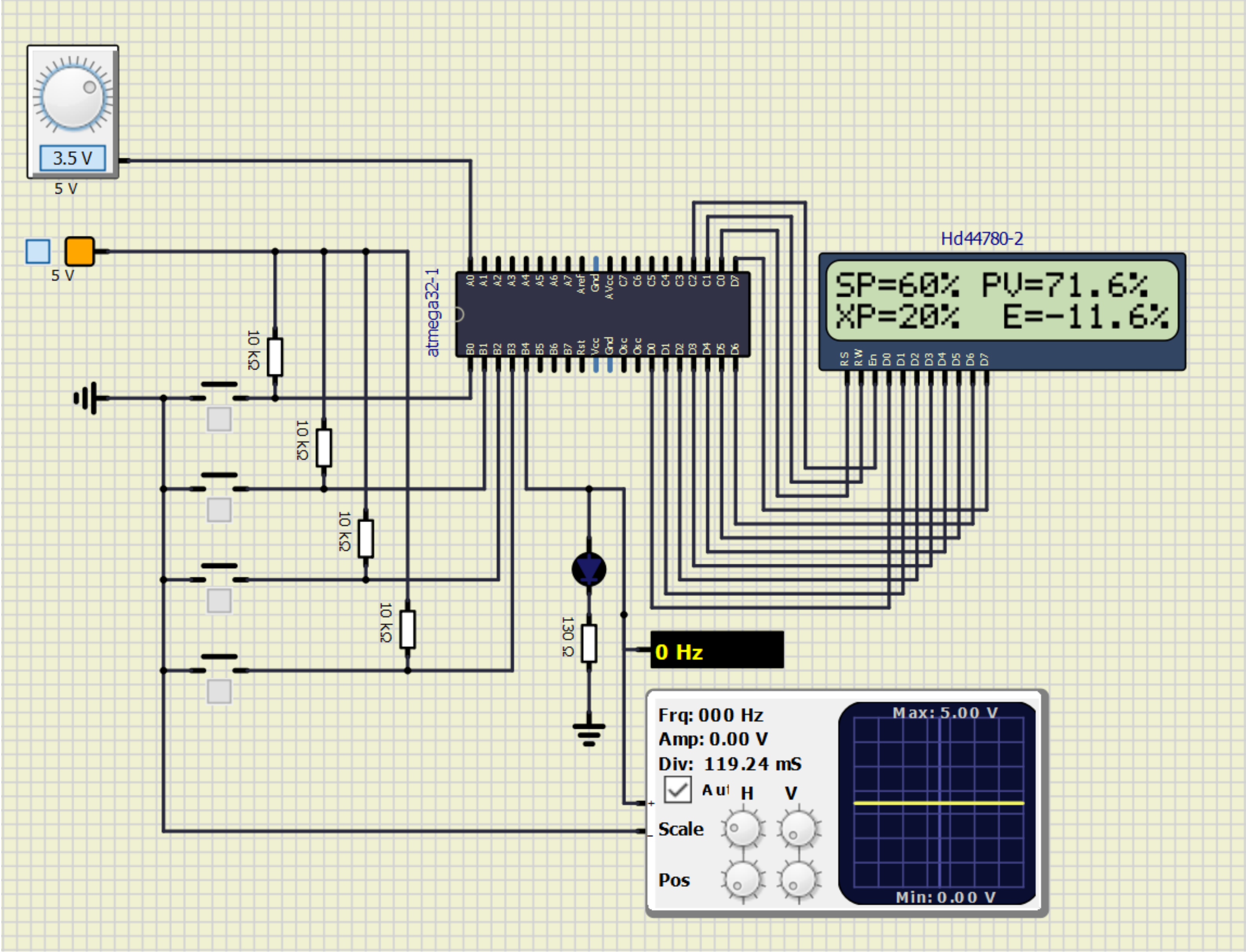
2.96 V
5 V
5 V
atmega32-1
A0 A1 A2 A3 A4 A5 A6 A7 Aref Gnd AVcc C7 C6 C5 C4 C3 C2 C1 C0 D7
B0 B1 B2 B3 B4 B5 B6 B7 Rst Vcc Gnd Osc Osc D0 D1 D2 D3 D4 D5 D6
Hd44780-2
RS RW En D0 D1 D2 D3 D4 D5 D6 D7
10 kΩ
10 kΩ
10 kΩ
10 kΩ
130 Ω
Aut
H
V
Scale
Pos

CV – około 50%



```
SP=60% PV=68.5%
XP=20%  E=-08.5%
```

CV – około 10%



```
SP=60% PV=71.6%
XP=20%  E=-11.6%
```

```
Frq: 000 Hz
Amp: 0.00 V
Div:  119.24 mS
```

CV – około 0%