

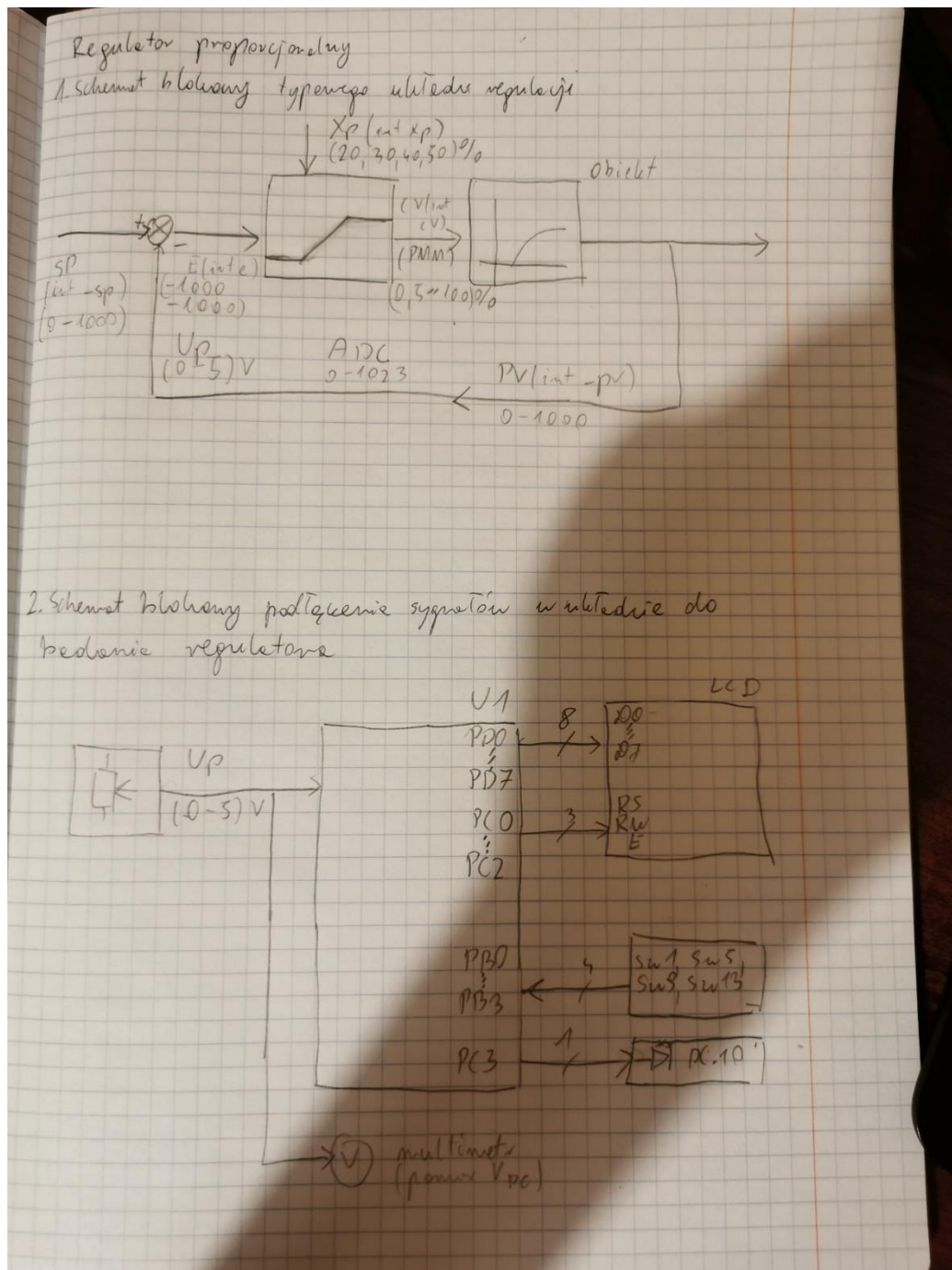
PTM2 – Badanie regulatora proporcjonalnego

Michał Łopatka, 248969

1. Zadanie do wykonania

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

2. Założenia projektowe

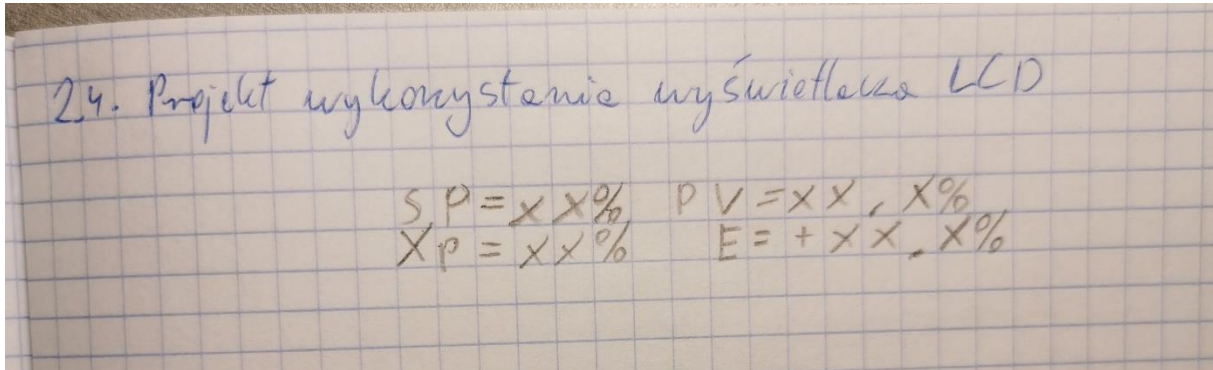


2.3. Zadawanie parametrów regulacji:

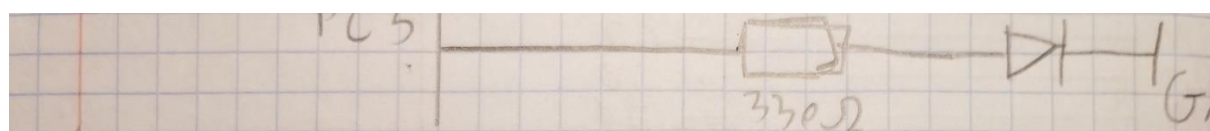
Zakres pomiarowy (0-400) °C/(0-5)V

- a) Po RESET SP=60%, Xp=20%
- b) Gdy SW2=1, SP=50%
- c) Gdy SW6=1, SP=40%
- d) Gdy SW10=1, Xp=30%
- e) Gdy SW14=1, Xp=40%

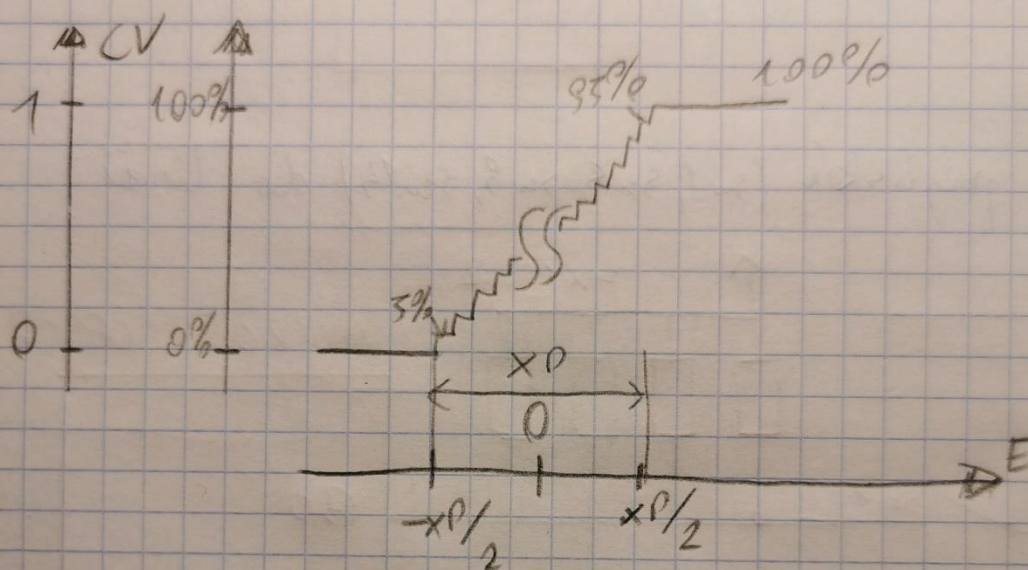
2.4 Projekt wykorzystania wyświetlacza



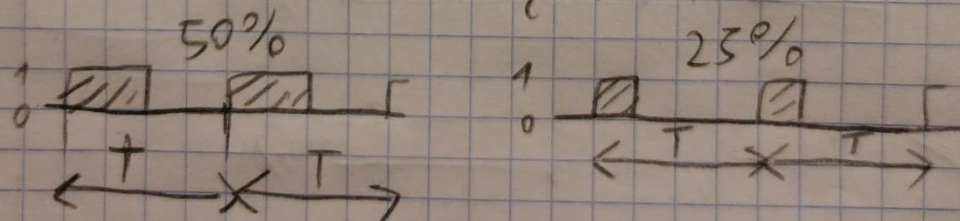
2. Regulator proporcjonalny



3. Regulator proporcjonalny



STEROWANIE METODĄ PWM



4. TABELA POMIAROWA

Badane regulowane dla $SP=40\%$, $XP=20\%$, $T_D=20[s]$

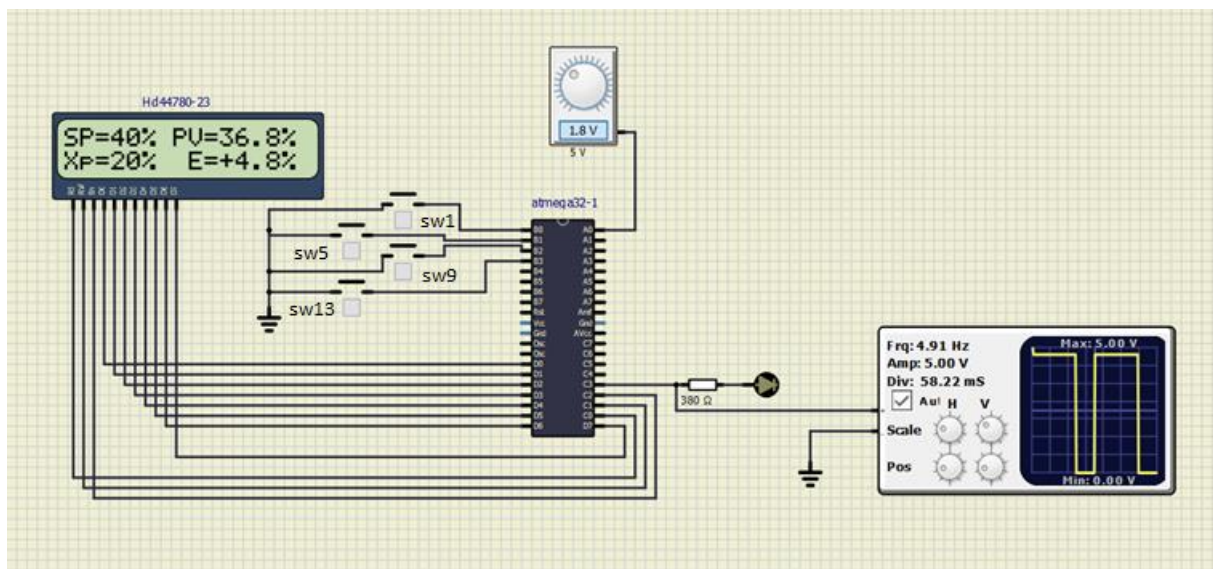
zakres pomiarowy: $(0-400)^\circ C$, $(0-5)V$

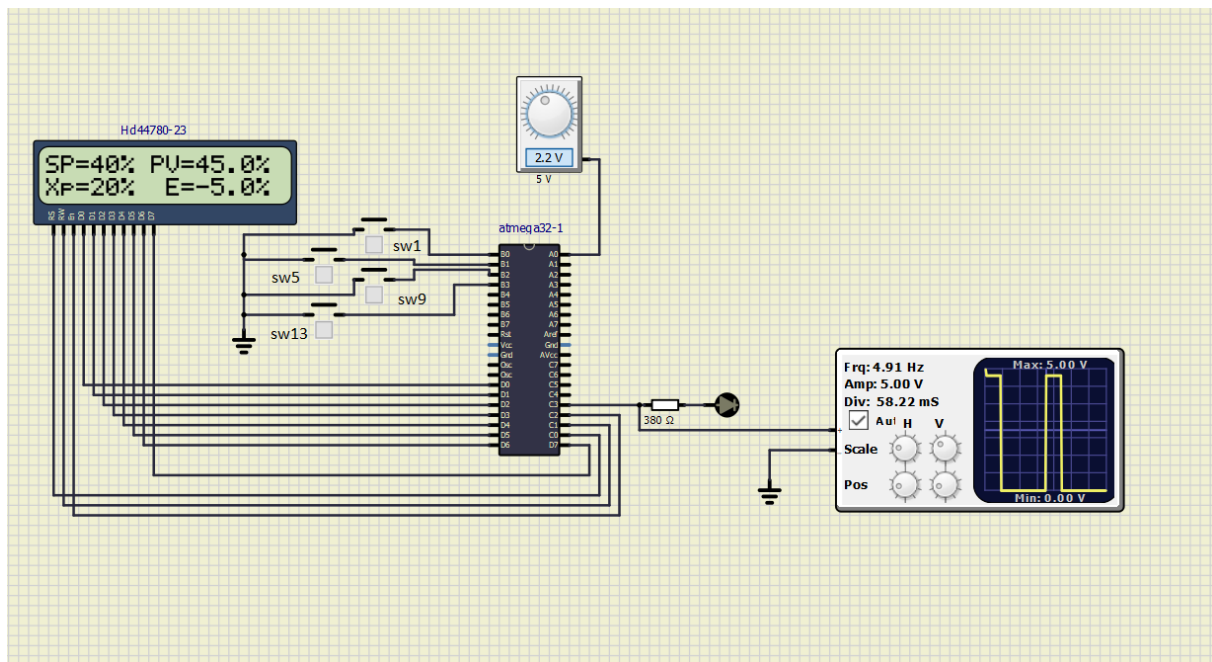
$E[X_P]$	$E[\%]$	$PV[\%]$	$PV[ADC]$	$PV[^\circ C]$	$PV[V]$	$CV[\%]$	$t_H[s]$	$t_H[s]/20[s]$
-1.00 X_P	-20.0	60.0	614	240.0	3.00	0	0.00	0.00
-0.55 X_P	-11.0	51.0	523	204.0	2.55	0	0.00	0.00
-0.30 X_P	-6.0	30.0	311	100.0	2.50	0	0.00	0.00
-0.45 X_P	-9.0	48.0	501	186.0	2.45	5	0.82	4.10
-0.40 X_P	-8.0	48.0	481	182.0	2.40	10	1.89	5.45
-0.20 X_P	-4.0	44.0	450	172.0	2.20	30	6.07	30.35
-0.10 X_P	-2.0	42.0	430	168.0	2.10	40	8.54	42.70
0.00 X_P	0.0	40.0	409	160.0	2.00	50	10.32	51.60
0.10 X_P	2.0	38.0	389	152.0	1.90	60	12.68	63.40
0.20 X_P	4.0	36.0	369	144.0	1.80	70	14.01	70.05
0.40 X_P	8.0	32.0	327	128.0	1.60	80	16.08	80.4
0.45 X_P	9.0	31.0	317	124.0	1.55	85	18.06	90.30
0.30 X_P	6.0	30.0	307	120.0	1.50	100	20.00	100.00
0.35 X_P	7.0	28.0	287	116.0	1.45	100	20.00	100.00
1.00 X_P	20.0	20.0	205	80.0	1.00	100	20.00	100.00

4. Uwagi i wnioski

Dioda D3.10 świeci przez odcinki czasu proporcjonalne do zadanego napięcia. Regulator proporcjonalny ma przebieg schodkowy. Układ pomiarowy działa prawidłowo, występuje jednakże pewien błąd przy pomiarze t_H .

5. Załączniki:





```

ADC = ADC_10bit();
_pv = ADC;
_ipv = _pv/10;
_e = _sp - _ipv; //Michał Łopatka, Mateusz Bartkowiak
_decpv = (_pv-_ipv*10); //czesc po przecinku

```

```

int cv;
cv=5*(50-_ipv);
for(int i=0; i<100; i+=5){
    if(i < cv)
    {
        sbi(PORTC, PC3);
    }
    else
    {
        cbi(PORTC, PC3);
    }
    delay_ms(10);
}

```

```

if(!(PINB & (8<<PB0)))
{
    _sp=50;
}
if(!(PINB & (4<<PB0)))
{
    _sp=40;
}
if(!(PINB & (2<<PB0)))
{
    _h=30;
}
if(!(PINB & (1<<PB0)))
{
    _h=40;
}

```

