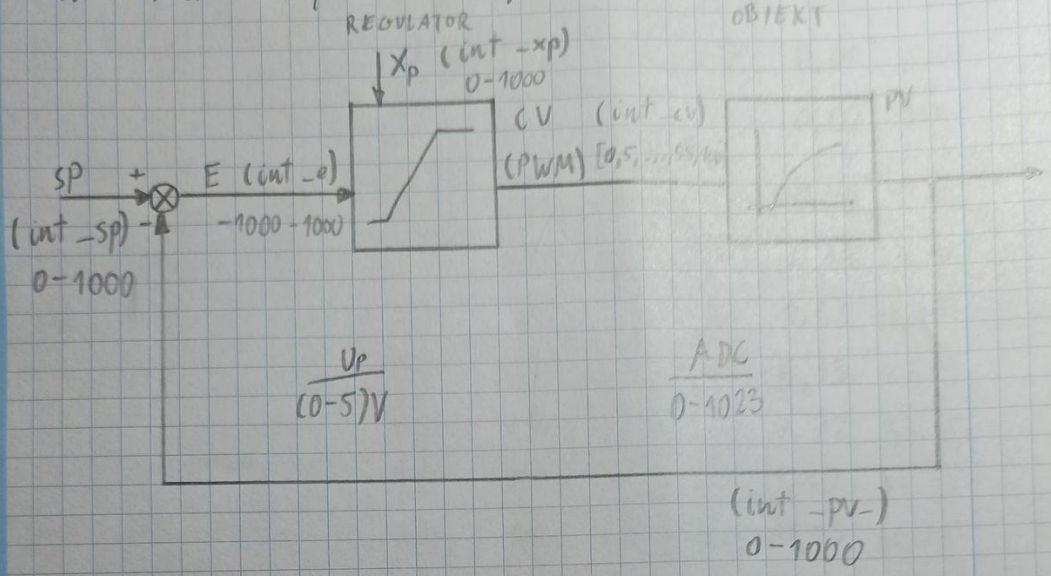
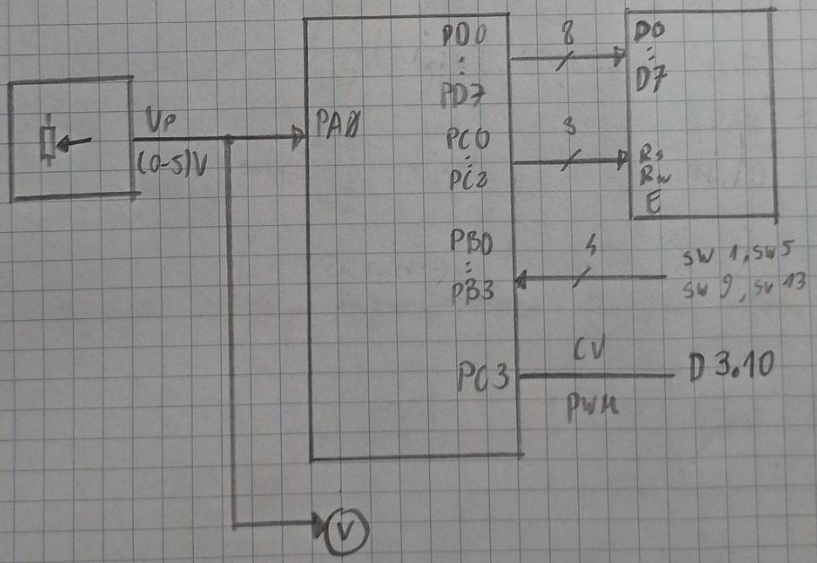
Kacper Gościeniec 248963

1. Schemat blokowy układu regulacji



2. Schemat blokowy połączenia sygnałów w układzie do badania regulatora

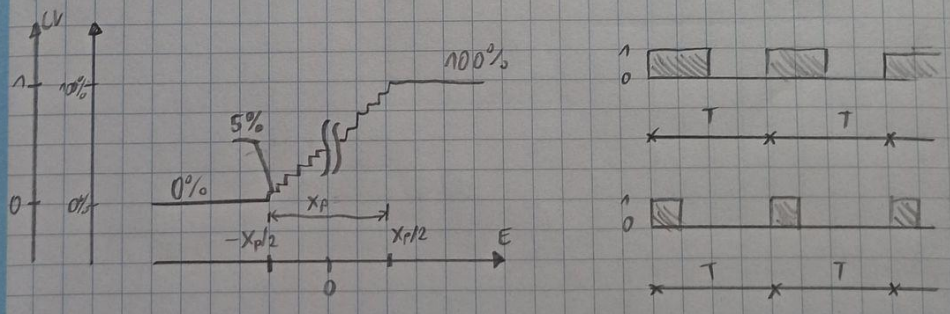


Kacper Grasiewicz
249963

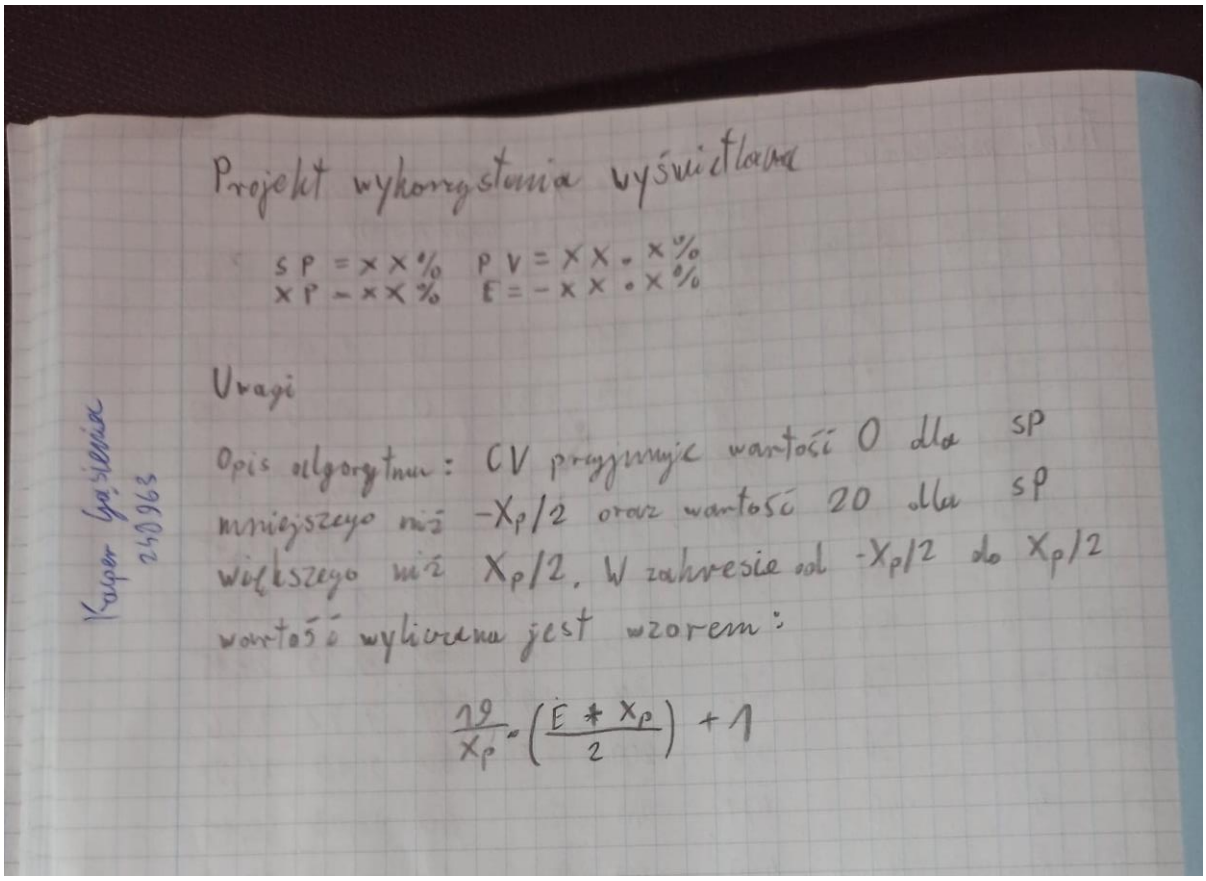
Tabela pomiarów dla $SP = 50\%$, $X_p = 40\%$, $T_0 = 20[s]$
zakres pomiarowy: (0-400)°C / (0-5)V

E[X_p]	E[%]	PV[%]	PV[ADC]	PV[°C]	PV[V]	CV[%]	THI[s]	THI:T$_0$[s] ×100%
-1,00	-40	90	921	360	4,5	0	0,00*	0,00
-0,55	-22	72	737	288	3,6	0	0,00*	0,00
-0,50	-20	70	716	280	3,5	0	0,00*	0,00
-0,45	-18	68	696	272	3,4	5	0,83	4,15
-0,40	-16	66	675	264	3,3	10	1,53	8,15
-0,20	-8	58	593	232	2,9	30	4,33	27,15
-0,10	-4	54	552	216	2,7	40	7,72	38,60
0,00	0	50	512	200	2,5	50	9,10	45,50
0,10	4	46	471	184	2,3	60	12,03	60,15
0,20	8	42	430	168	2,1	70	14,21	71,05
0,40	16	34	348	136	1,7	90	17,86	89,30
0,45	18	32	327	128	1,6	95	18,73	93,65
0,50	20	30	307	120	1,5	100	20,00*	100,00
0,55	22	28	286	112	1,4	100	20,06*	100,00
1,00	40	10	102	40	0,5	100	20,00*	100,00

Charakterystyka regulatora proporcjonalnego / Sterowanie metodą PWM



* dla wartości 0,00$_s$ i 20,00$_s$ brak przebiegu - kolejno stan niski i wysoki na całym okresie



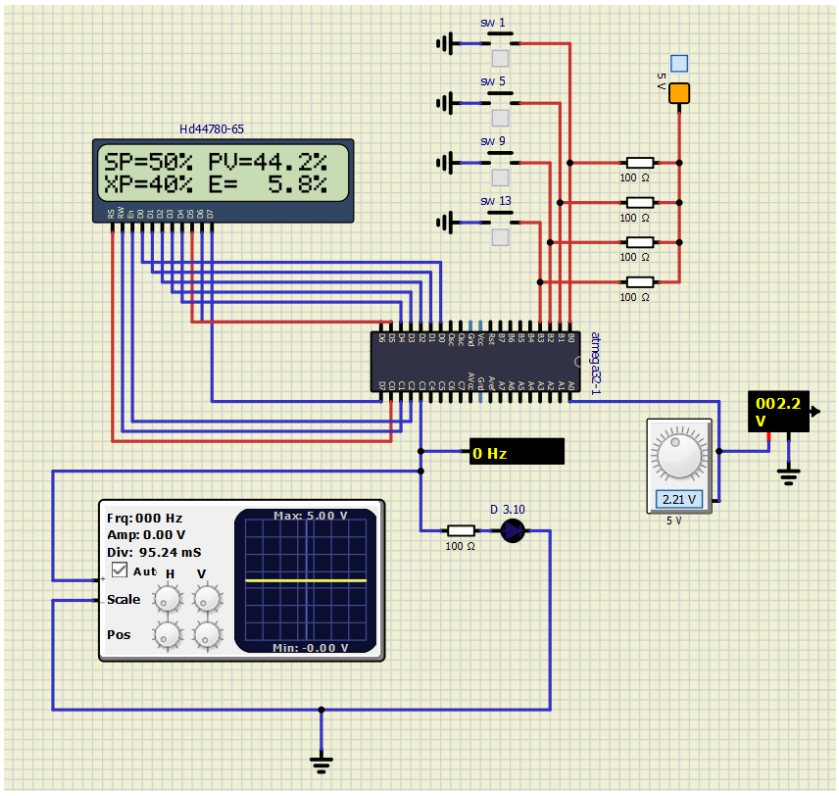
Kod programu (z pominięciem inicjalizacji LCD):

```
int _sp = 50;
int _xp = 40;
int _e;
int int_e;
int dec_e;
float _pv;
int full_pv;
int int_pv;
int dec_pv;
int _T=20;
int _cv = 10;
```

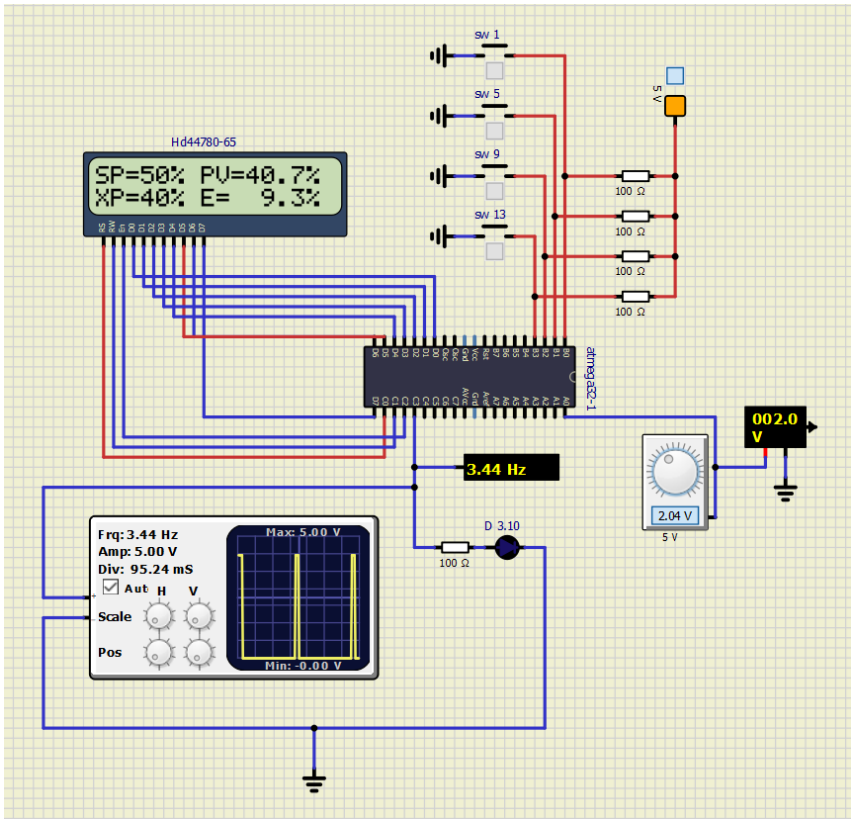
```
int main(void)
{
    char tmp[16];
    int i;

    DDRD = 0xff;
    PORTD = 0x00;
    DDRC = 0xff;
    PORTC = 0x00;
    DDRB = 0x00;
    PORTB = 0xff;
    DDRC = (1<<3)|(1<<4);
    _delay_ms(500);

    LCD2x16_init();
    LCD2x16_clear();

    ADMUX = 0x40;
    ADCSRA = 0xe0;
```

```
    while (1)
    {
        ADCSRA = ADCSRA | (1 << ADSC);
        while (ADCSRA & (1 << ADSC));
        _pv = ADC;
        full_pv = (_pv / 1023.0) * 1000;
        int_pv = full_pv / 10;
        dec_pv = full_pv % 10;
        _e = _sp - int_pv;
        int_e = _e / 10;
        dec_e = _e % 10;

    if(_e < -_xp/2) _cv =0;
    else if(_e >= _xp/2) _cv=20;
    else _cv = (((_e+_xp/2)*19/_xp)+1);

        for (i=0; i<20; i++)
        {
            if(i < _cv && _cv != 0)
            {
                PORTC |= 1 << PINC3; //wlacz diode
            }
            else
        {
                PORTC &= ~(1 << PINC3); //wylacz diode
            }
            delay_ms(10);
        }


       //przyciski
       if (!(PINB & (1 << PB0)))
       {
           _sp = 50;
       }
       if (!(PINB & (2 << PB0)))
       {
           _sp = 40;
       }
       if (!(PINB & (3 << PB0)))
       {
           _xp = 30;
       }
       if (!(PINB & (4 << PB0)))
       {
           _xp = 40;
       }

       LCD2x16_pos(1, 1);
       sprintf(tmp, "SP=%2d%% PV=%2d.%1d%% ", _sp, int_pv, dec_pv);
       for (i = 0; i < 16; i++)
       {
           LCD2x16_putchar(tmp[i]);
       }

       LCD2x16_pos(2, 1);
       sprintf(tmp, "XP=%2d%% E=%3d.%1d%%  ", _xp, _e, dec_e);
       for (i = 0; i < 16; i++)
       {
           LCD2x16_putchar(tmp[i]);
       }
       delay_ms(100);
    }

    return 0;
}
```

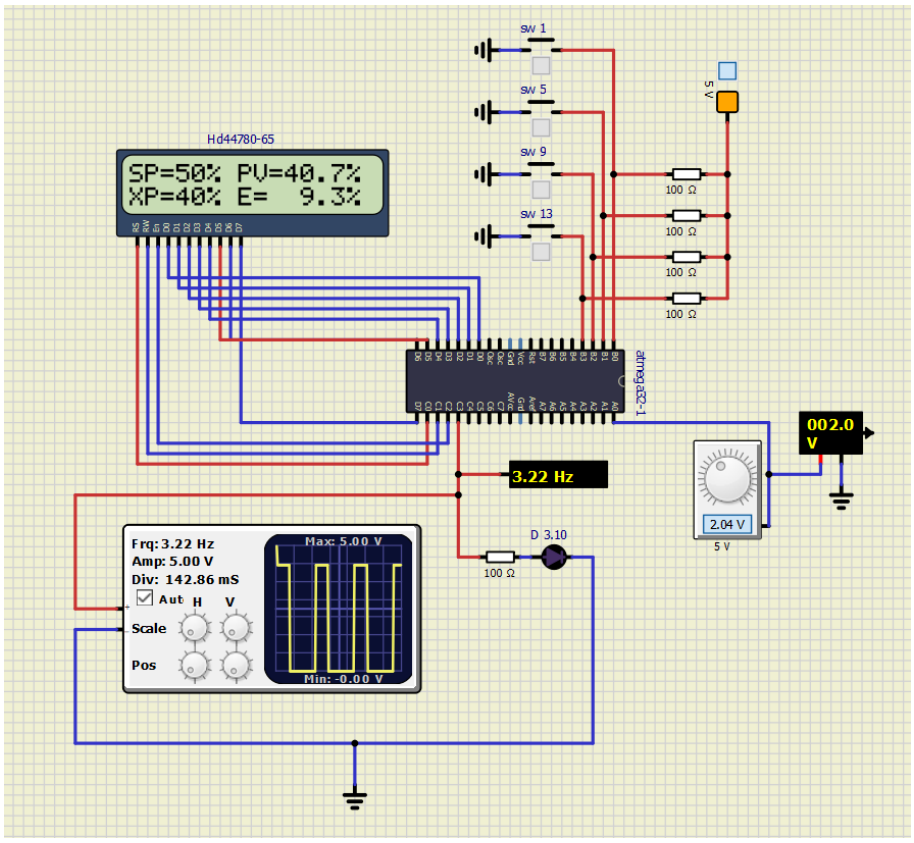
cv = 0


SW 1
SW 5
SW 9
SW 13
5 V
Hd44780-65
SP=50% PV=44.2%
XP=40% E= 5.8%
100 Ω
100 Ω
100 Ω
100 Ω
atmega32-1
002.2 V
0 Hz
2.21 V
5 V
D 3.10
100 Ω
Frq: 000 Hz
Amp: 0.00 V
Div: 95.24 mS
Aut
H
V
Scale
Pos
Max: 5.00 V
Min: -0.00 V

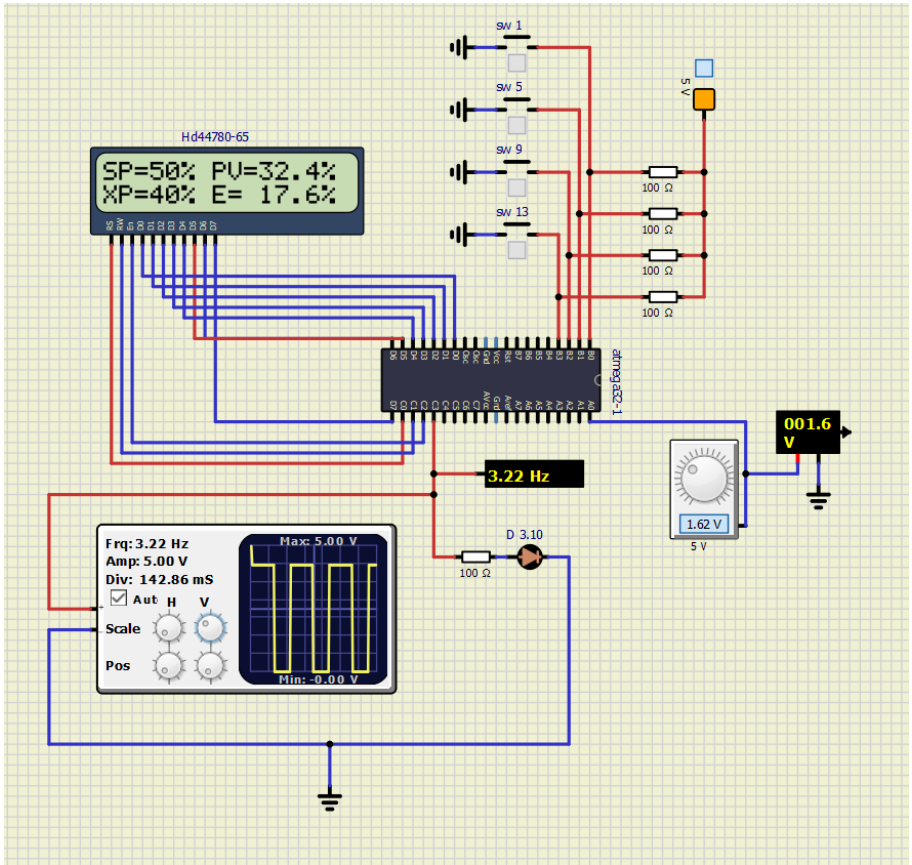
cv = 2 (10%)


SW 1
SW 5
SW 9
SW 13
5 V
Hd44780-65
SP=50% PV=40.7%
XP=40% E= 9.3%
100 Ω
100 Ω
100 Ω
100 Ω
atmega32-1
002.0 V
3.44 Hz
2.04 V
5 V
D 3.10
100 Ω
Frq: 3.44 Hz
Amp: 5.00 V
Div: 95.24 mS
Aut
H
V
Scale
Pos
Max: 5.00 V
Min: -0.00 V

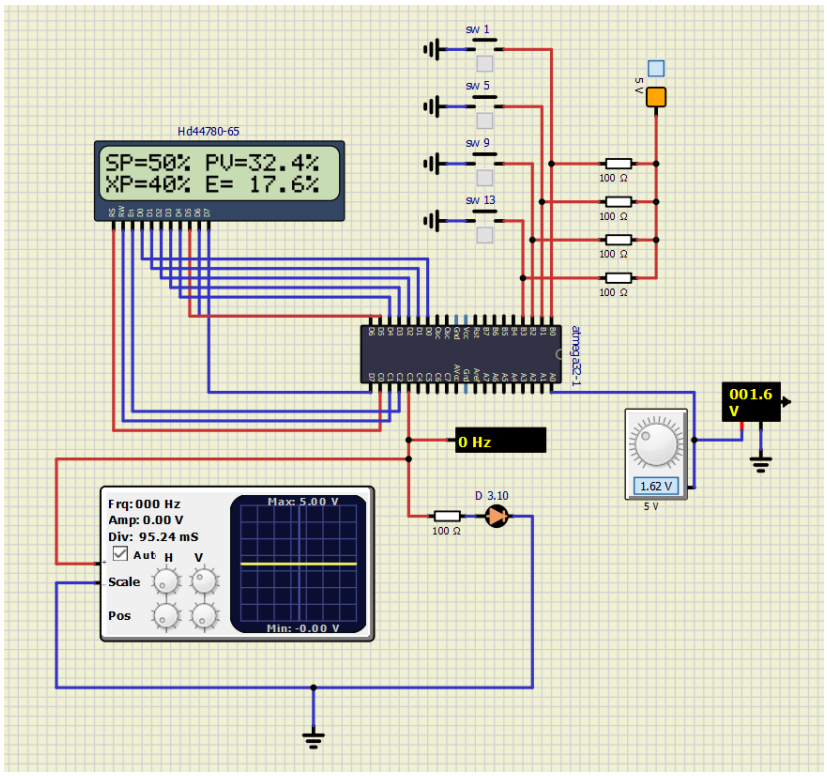
cv = 10 (50%)



cv = 18 (90%)



cv = 20 (100%)


sw 1
sw 5
sw 9
sw 13
5 V
Hd44780-65
SP=50% PV=32.4%
XP=40% E= 17.6%
100 Ω
100 Ω
100 Ω
100 Ω
atmega328-pu
001.6 V
0 Hz
1.62 V
5 V
D 3.10
100 Ω
Frq: 000 Hz
Amp: 0.00 V
Div: 95.24 mS
Aut
H
V
Scale
Pos
Max: 5.00 V
Min: -0.00 V