

Tema: Proiectarea motorului pas cu pas

Scopul lucrării: Proiectarea motorului pas cu pas utilizând funcțiile limbajului C.

Sarcina lucrării:

Se va crea un program “Motor pas cu pas” care va afișa pe primul rind al LCD-ului cuvântul “Viteza =”, iar pe al doilea rind va afișa viteza în procente.

Mod de lucru:

În mediul de simulare ISIS, ce intră în componenta pachetului Proteus 7.6, asamblăm schema ca în figura 1. Ca unitate programabilă s-a ales MC ATmega8. În schema este inclus și un LCD 2x16 pe controller ATmega8.

Schema asamblată în Protelus.

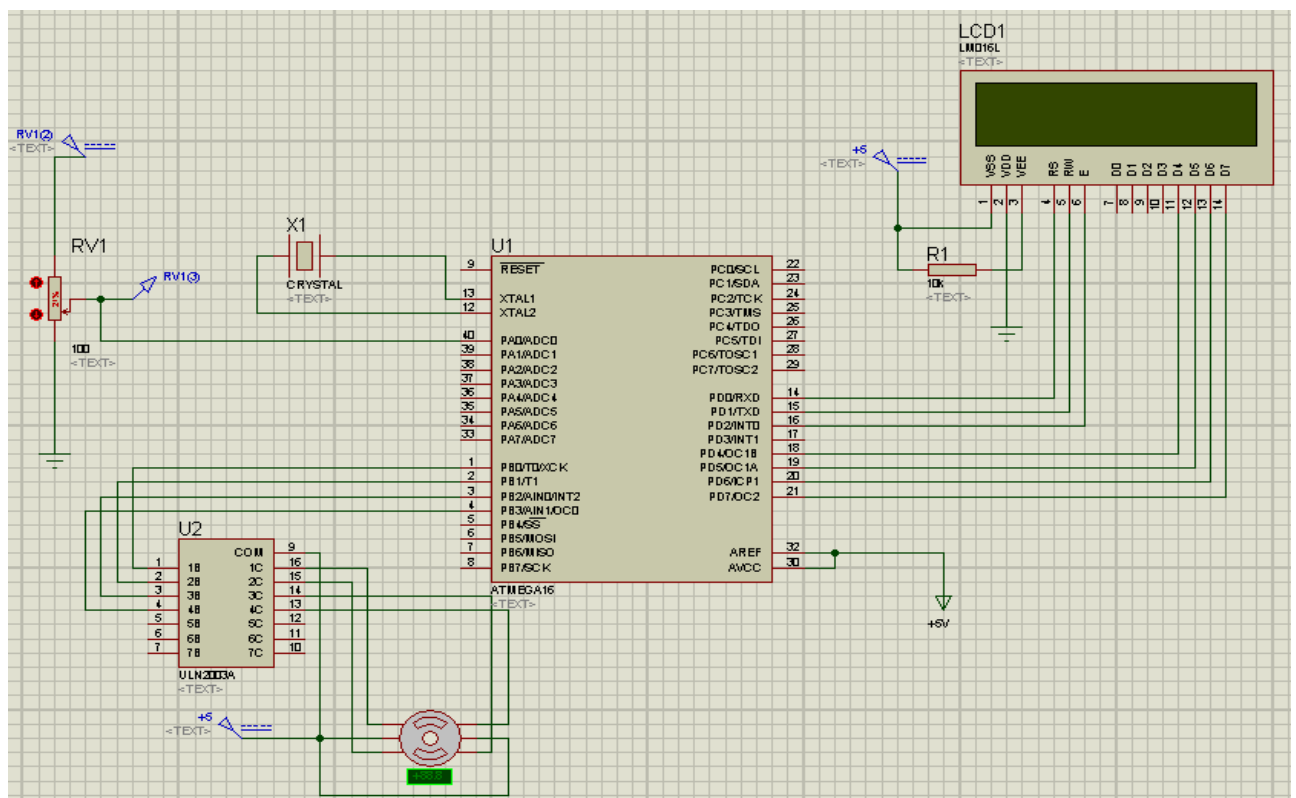


Fig.1 Schema dispozitivului “Motor pas cu pas”.

2. Deschidem CodeVisionAVR, accesăm *New->Project->ATmega*. În

fereastra apărută (fig.2) efectuăm setările în fila MCU ale **CP1E525MCU081xxLL**

Mod	Coala	Nr Docum.	Semn.	Data	Litera	Coala	Coli
A efectuat							
A verificat	Dergaci O..					2	6
Contr. tehn.					<i>Gestionarea cu LCD-display.</i>		
Contr. norm							
Aprobat							
					UTM FRT SER-081		

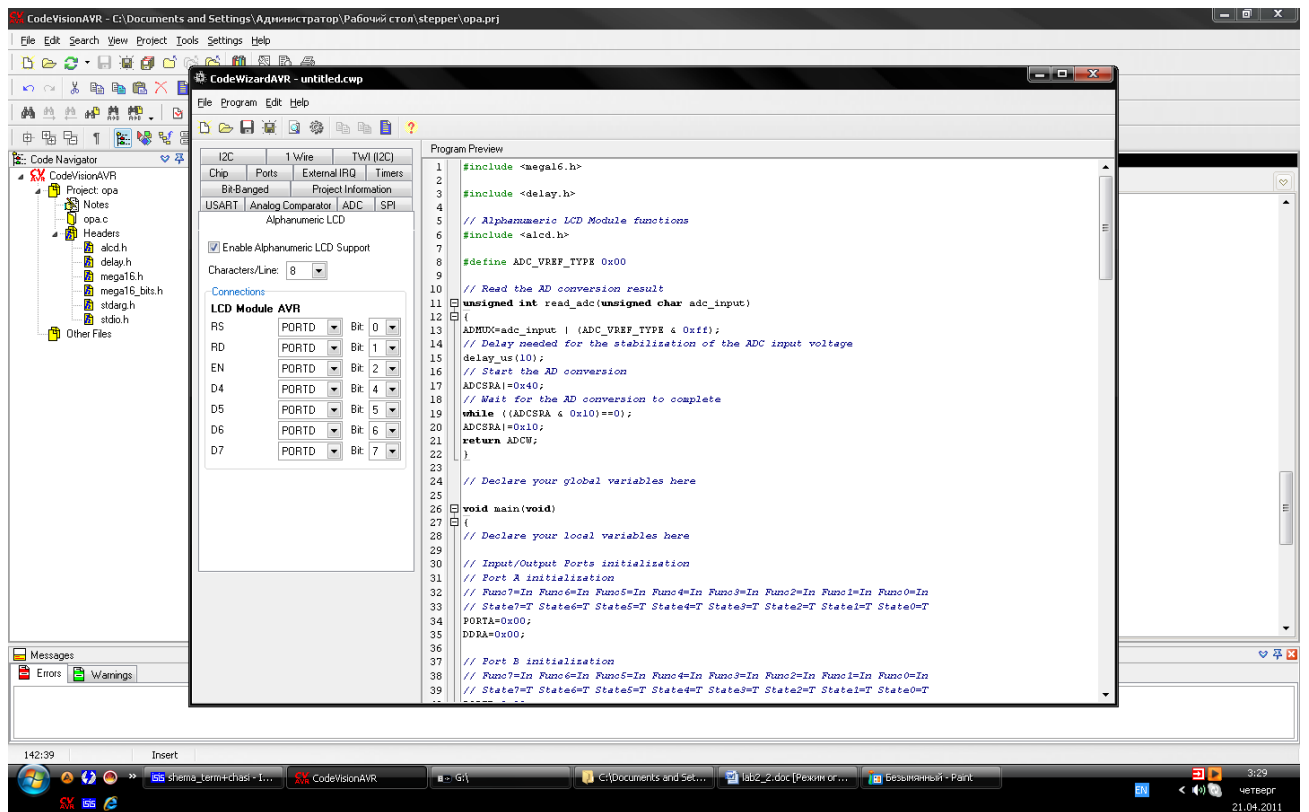


Fig.2 Setarile initiale ale programului in CVAVR.

Apoi salvăm documentul creat. Compilatorul va genera un cod conform setărilor alese. Comentariile se pot șterge pentru a economisi spațiu.

3. Listingul programului:

```
#include <mega16.h>
#include <stdio.h>
#include <delay.h>
#include <alcd.h>
#define ADC_VREF_TYPE 0x00
// Read the AD conversion result
float x;
long int y,i,d;
char buff[16];
void buff_clear(void) {
int n;
for (n=0;n<16;n++) {buff[n]="";
```

					<i>CPAE 525.1 081 xx LL</i>	Coala
Mod	Coala	№ Docum	Semn.	Data		3

```

};

};

void motor(int s) {
PORTB=0;
PINB.0=1;
delay_ms(s);
PORTB=0;
PINB.1=1;
delay_ms(s);
PORTB=0;
PINB.2=1;
delay_ms(s);
PORTB=0;
PINB.3=1;
delay_ms(s);
}

unsigned int read_adc(unsigned char adc_input)
{
ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
// Delay needed for the stabilization of the ADC input voltage
delay_us(10);
// Start the AD conversion
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCW;
}

// Declare your global variables here

void main(void)

```

					<i>CPAE 525.1 081 xx LL</i>	<i>Coala</i>
<i>Mod</i>	<i>Coala</i>	<i>Nº Docum</i>	<i>Semn.</i>	<i>Data</i>		4

```

{
PORTA=0x00;
DDRA=0x00;
PORTB=0x00;
DDRB=0x00;
PORTC=0x00;
DDRC=0x00;
PORTD=0x00;
DDRD=0x00;
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;
MCUCR=0x00;
MCUCSR=0x00;
TIMSK=0x00;
UCSRB=0x00;

```

					<i>CPAE 525.1 081 xx LL</i>	<i>Coala</i>
<i>Mod</i>	<i>Coala</i>	<i>Nº Docum</i>	<i>Semn.</i>	<i>Data</i>		5

```

ACSR=0x80;
SFIO=0x00;
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x83;
SPCR=0x00;
TWCR=0x00;
lcd_init(16);
while (1)
{
i=read_adc(0);
x=100*i/1024;
y=x/1;
lcd_clear();
buff_clear();
lcd_gotoxy(4,0);
lcd_puts("Viteza =");
lcd_gotoxy(2,1);
sprintf(buff,"%u procente",y);
lcd_puts(buff);
d=110-y;
motor(d);
}
}

```

4. Screenshoturi:

- **Compilarea programului**

					<i>CPAE 525.1 081 xx LL</i>	<i>Coala</i>
<i>Mod</i>	<i>Coala</i>	<i>Nº Docum</i>	<i>Semn.</i>	<i>Data</i>		6

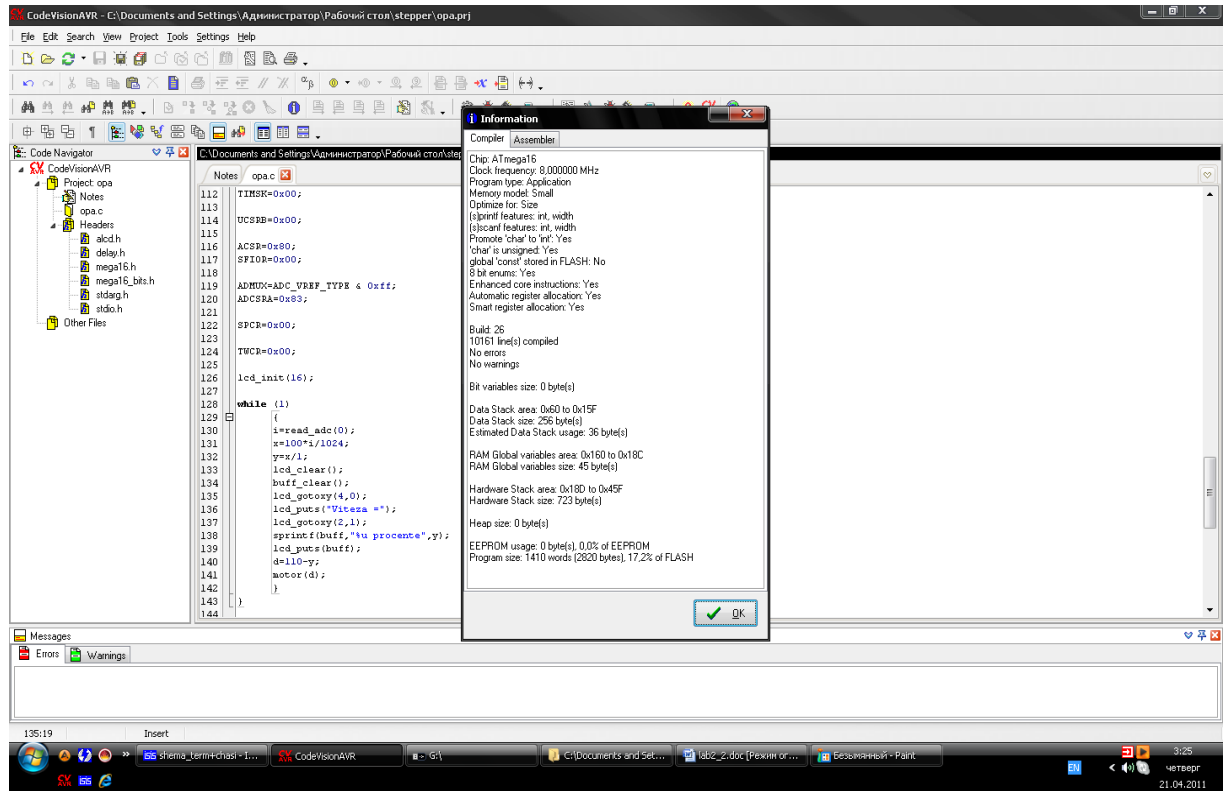


Fig.3 Rezultatul compilarii

- **Lucru motorului pas cu pas**

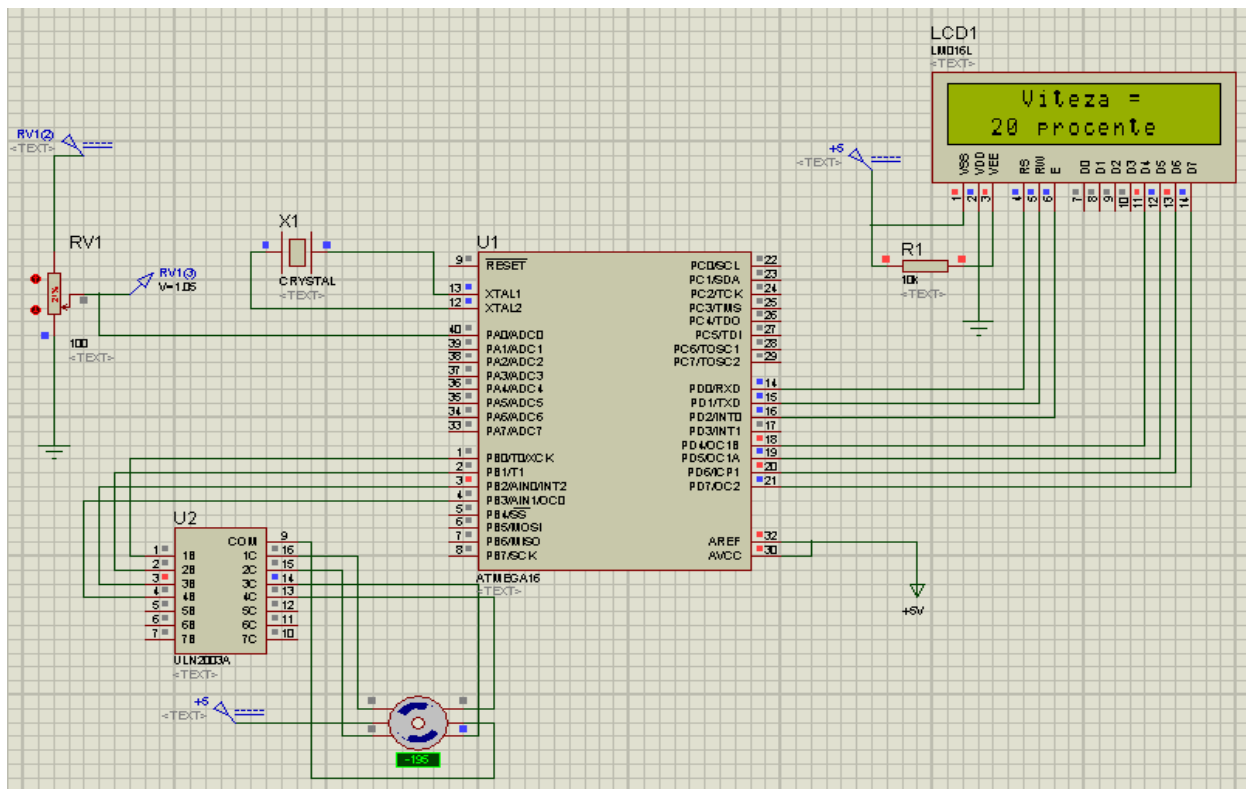


Fig.4 Screenshot cu functionarea dispozitivului.

CONCLUZIE:

						Coala
						7
Mod	Coala	Nº Docum	Semn.	Data	CPAE 525.1 081 xx LL	

După executarea acestei lucrări de laborator pot face următoarele concluzii:
Am căpătat deprinderi practice în gestionarea unui LCD cu ajutorul microcontrolerului. Utilizarea unui LCD este mult mai rezonabilă decât utilizarea indicatoarelor LED segmentate, deoarece programarea lor este cu mult mai simplă.

					<i>CPAE 525.1 081 xx LL</i>	<i>Coala</i>
<i>Mod</i>	<i>Coala</i>	<i>№ Docum</i>	<i>Semn.</i>	<i>Data</i>		8