

Универзитет у Крагујевцу

Факултет техничких наука Чачак



ПРОЈЕКТОВАЊЕ ДИГИТАЛНИХ СИСТЕМА

Пројектни задатак: Брзиномер

Предметни наставник:

проф. др Синиша Ранђић

Студент:

Гајић Богдан 171/2013

У Чачку, 2017. године

Садржај:

1. Увод	3
2. Опис компоненти	4
2.1. Мотор једносмерне струје	4
2.2. Диск са прорезима и оптокаплер.....	5
2.3. Easy PIC 6	6
3. Реализација пројекта	8
3.1. Електрична шема.....	9
3.2. Код и објашњење	10
4. Изглед завршеног пројекта	20

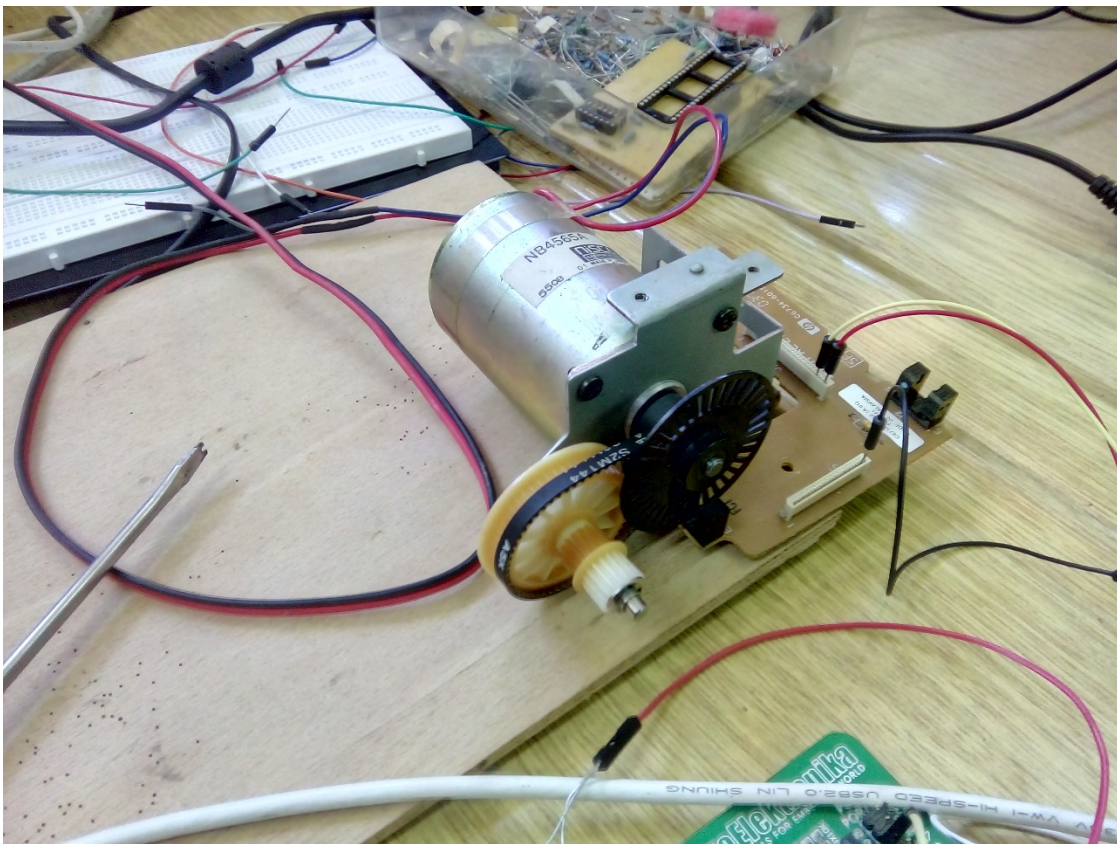
1. Увод

Познавање брзине обртања [обртај/минут] је основна информација у сваком систему код којег се дешава обртно кретање. Она се даље може користити ради контроле, управљања, регулације неког процеса или се трансформисати у брзину кретања [метар/секунда]. Циљ овог пројектног задатка је реализација функционалног брзиномера помоћу развојног окружења Easyrisc 6. При реализацији пројекта коришћен је једносмерни мотор, оптокаплерски пар и горе поменуто развојно окружење.

2. Опис компоненти

2.1. Мотор једносмерне струје

Коришћен је мотор једносмерне струје који се напаја напоном напајања од 24 V и брзином обртања око 3600 обртаја у минути. На вратило мотора је постављен диск са прорезима који поседује 36 прореза. Када се мотор напаја напоном напајања од 5 V у празном ходу се обрће брзином око 735 обртаја у минути што одговара отприлике 440 импулса на оптокаплеру. Када се мотор напаја напоном напајања од 12 V у празном ходу се обрће брзином око 1800 обртаја у минути што одговара отприлике 1100 импулса на оптокаплеру. Изглед коришћеног мотора је дат на слици (сл. 1.).

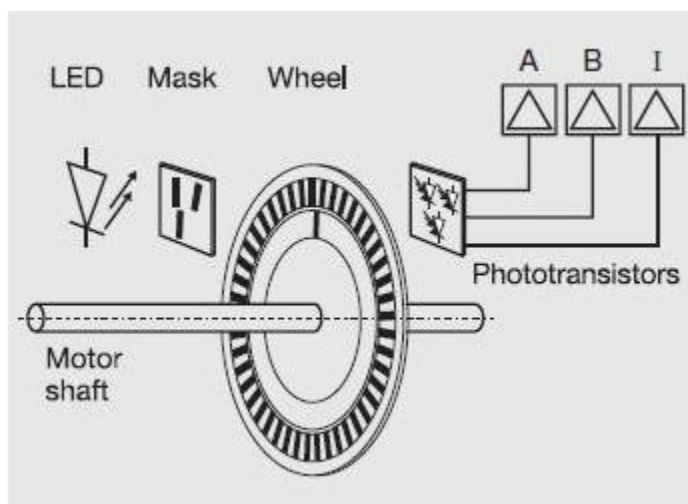


Слика 1. Мотор једносмерне струје

2.2. Диск са прорезима и оптокаплер

Ове две компоненте су од круцијалне важности за реализацију овог задатка, јер нам директно дају информацију о обртању вратила, коју даље можемо користи. Принцип рада је јако сличан као код оптичког инкременталног енкодера са једном фазом. Начин функционисања таквог енкодера је приказан на слици (сл. 2.).

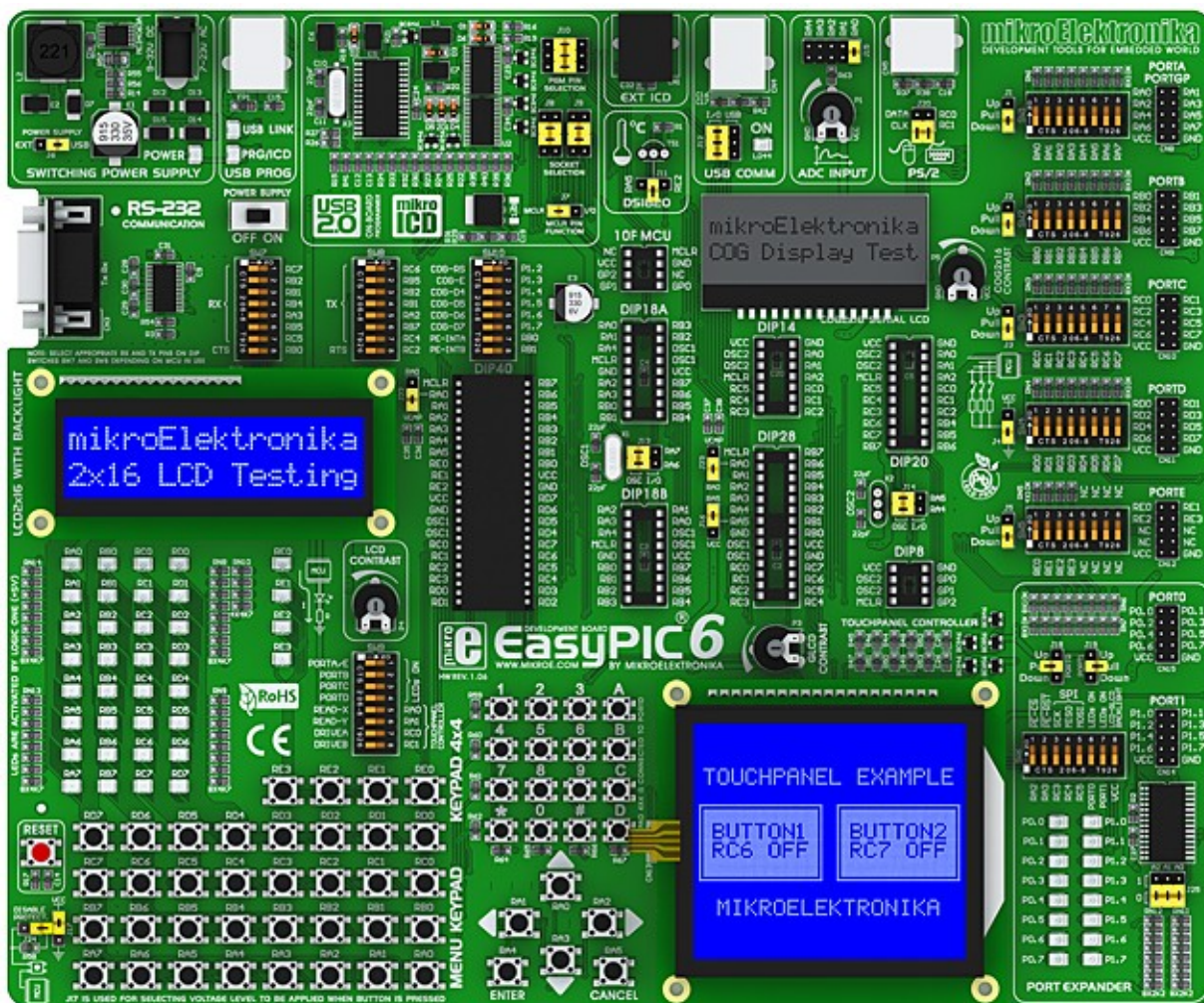
За добијање импулса користили смо оптички пар (оптокаплер) који се састоји од једне ЛЕД диоде и фото-транзистора. Када зрак прође кроз зазор на диску, транзистор генерише импулс. Укупан број зазора је 36, самим тим добијамо 36 импулса по једном обртају.



Слика 2. Принцип рада енкодера

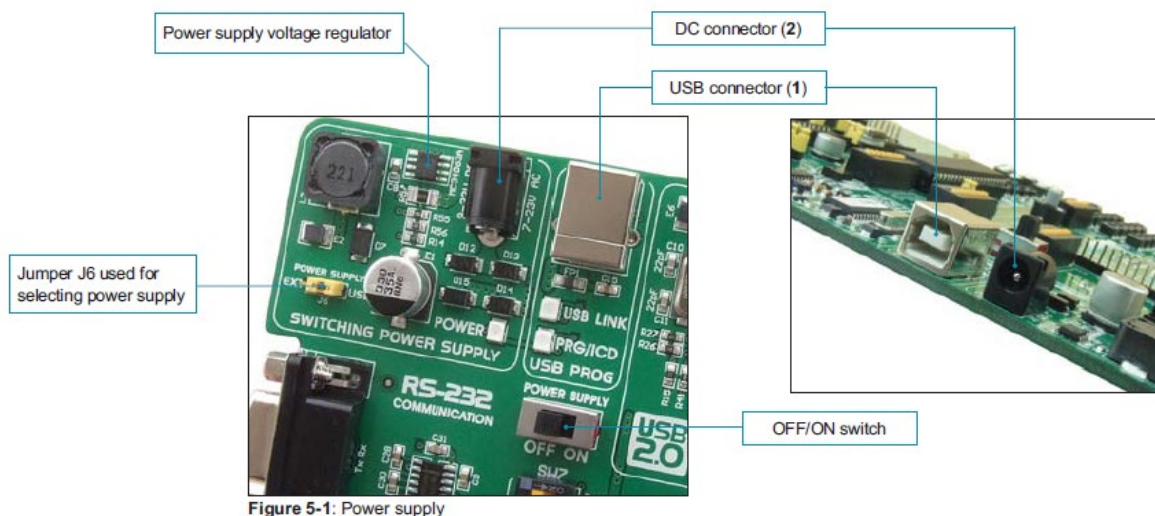
2.3. Easy PIC 6

Easy PIC 6 развојни систем је одлична табла за експериментисање и развој пројеката.



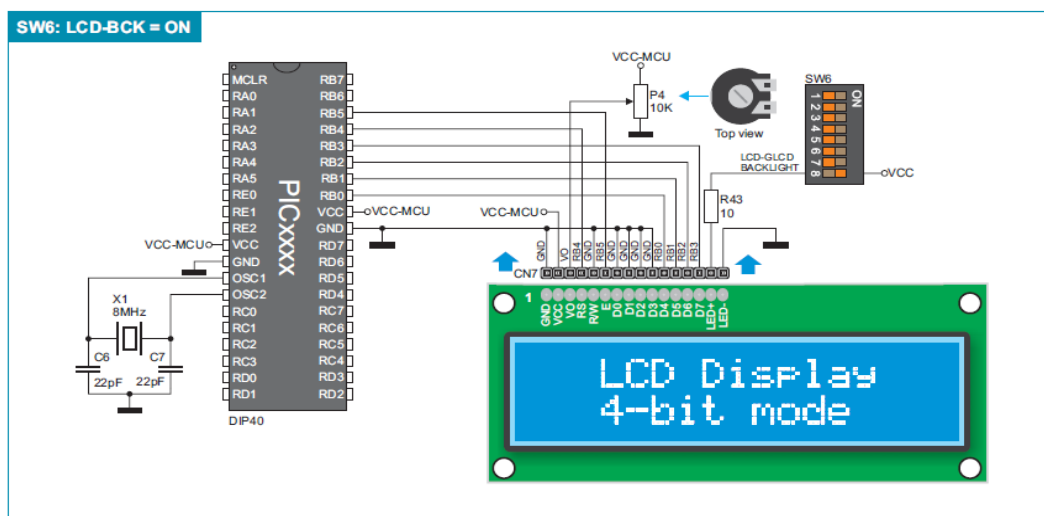
Слика 3. Изглед плоче

За напајање плоче се може користити напајање преко рачунара путем USB-а (+5v) или неко спољно напајање које је конектовано на DC конектор на развојном систему.



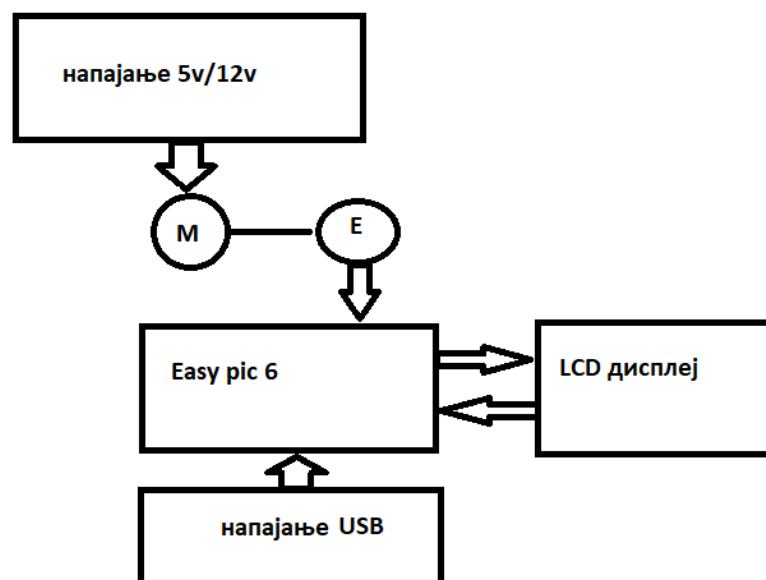
Слика 4. Изглед конектора за напајање

Такође, на плочи се налази и 2x16 LCD дисплеј који смо користили за испис тренутне брзине обртања.



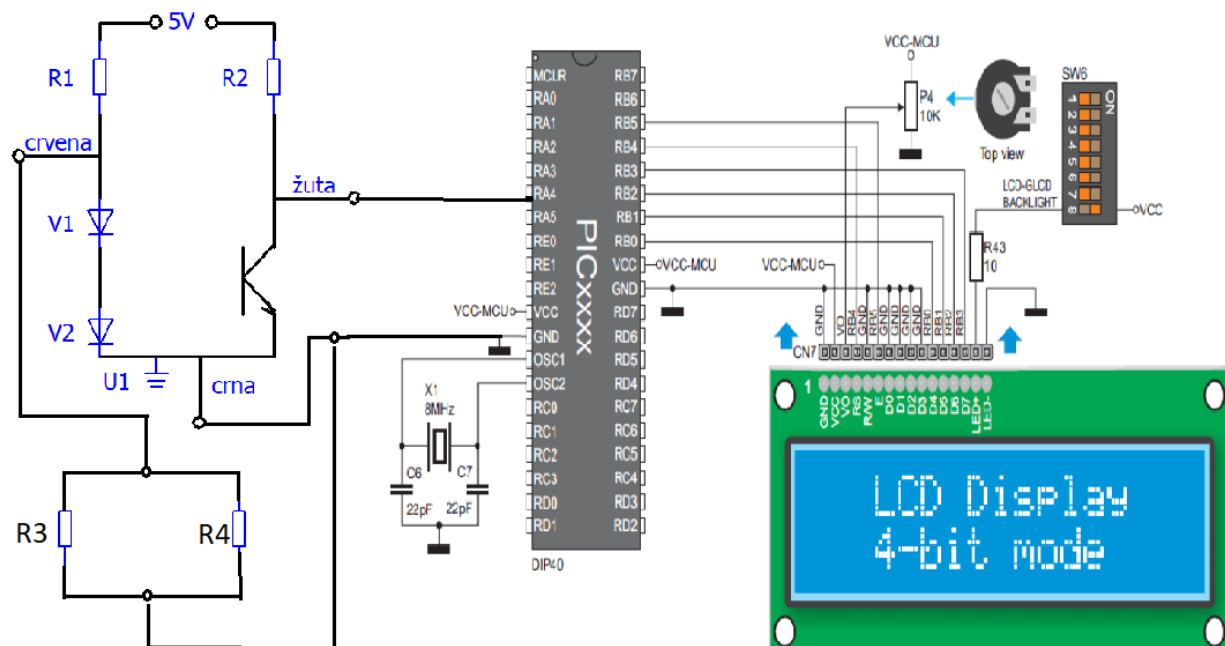
Слика 5. Шема повезивања дисплеја

3. Идејно решење и практична реализација



Слика 6. Блок шема

3.1. Електрична шема

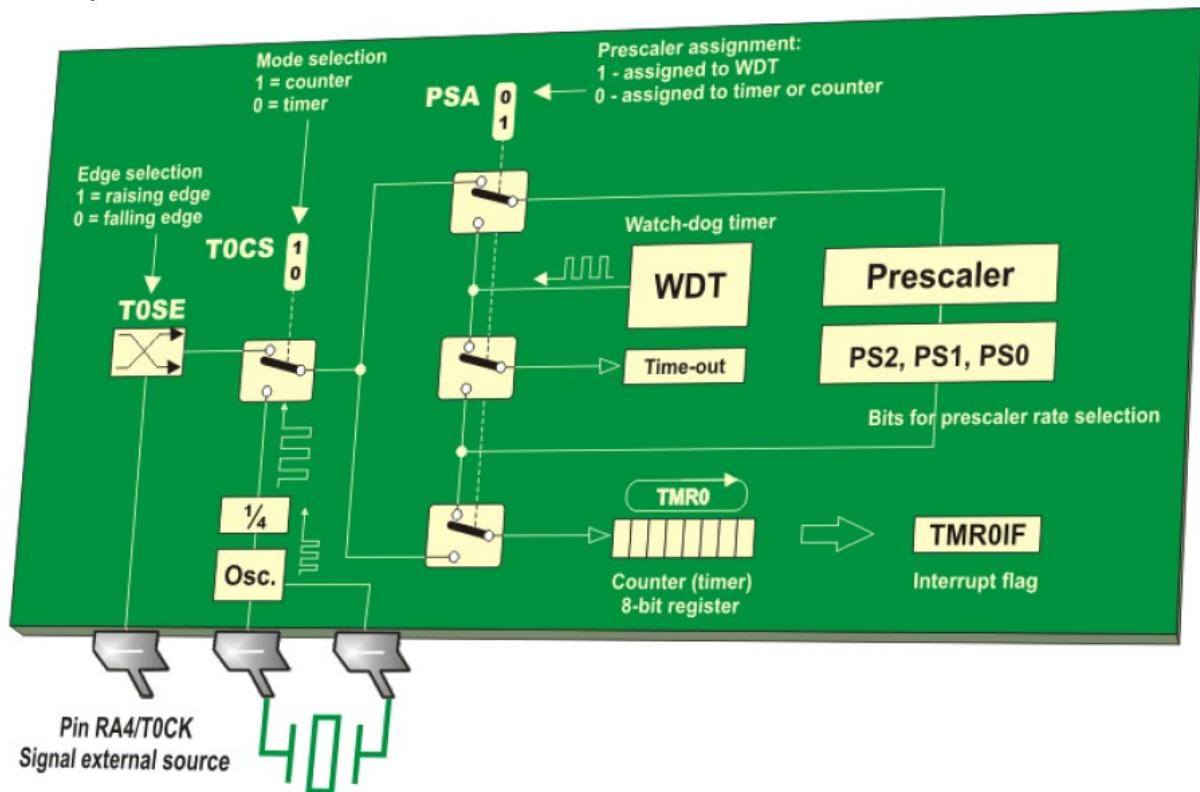


Слика 7. Електрична шема

3.2.Код и објашњење

За извршење овог пројекта било је потребно да се одлучи који ће се тајмер користити за бројање импулса, а који за време.

Timer0 је 8-битни тајмер/бројач унапред поседује прескалер који дели са Watchdog (WDT) тајмером, извор такта тајмера/бројача може бити интерни ($F_{osc}/4$) или екстерни при чему се код екстерног може изабрати ивица на којој ће долазити до тактовања Timer0 тајмера/бројача. Тренутна вредност Timer0 тајмера/бројача може се прочитати или уписати преко осмобитног TMR0 регистра. Када дође до перпуњавања TMR0 регистра(прелазак са 0xFF на 0x00 тј. са 255 на 0) генерише се прекид тајмера T0IF. Тајмер/бројач Timer0 се може користити за прецизно мерење времена или бројање улазних импулса.Структура Timer0 тајмера/бројача приказана је на следећој слици.



Слика 8. Структура Timer0 тајмера/бројача

Рад Timer0 тајмера/бројача контролише се помоћу битова у OPTION_REG регистру, чији је садржај приказан на следећој слици.



Слика 9. Садржај OPTION_REG регистра

Избор начина рада Timer0 тајмера/бројача врши се преко типа сигнала који врши његово тактовање, а који се одабира преко T0CS (Timer0 Clock Select) бита. Уколико је T0CS бит сетован, побудни импулси ће долазити преко RA4/T0CKI пина, па ће Timer0 радити као бројач ако су импулси асинхрони, или као тајмер ако су импулси синхрони. Ако је T0CS бит ресетован, побудни импулси за Timer0 ће долазити од системског такта који је једнак $F_{osc}/4$.

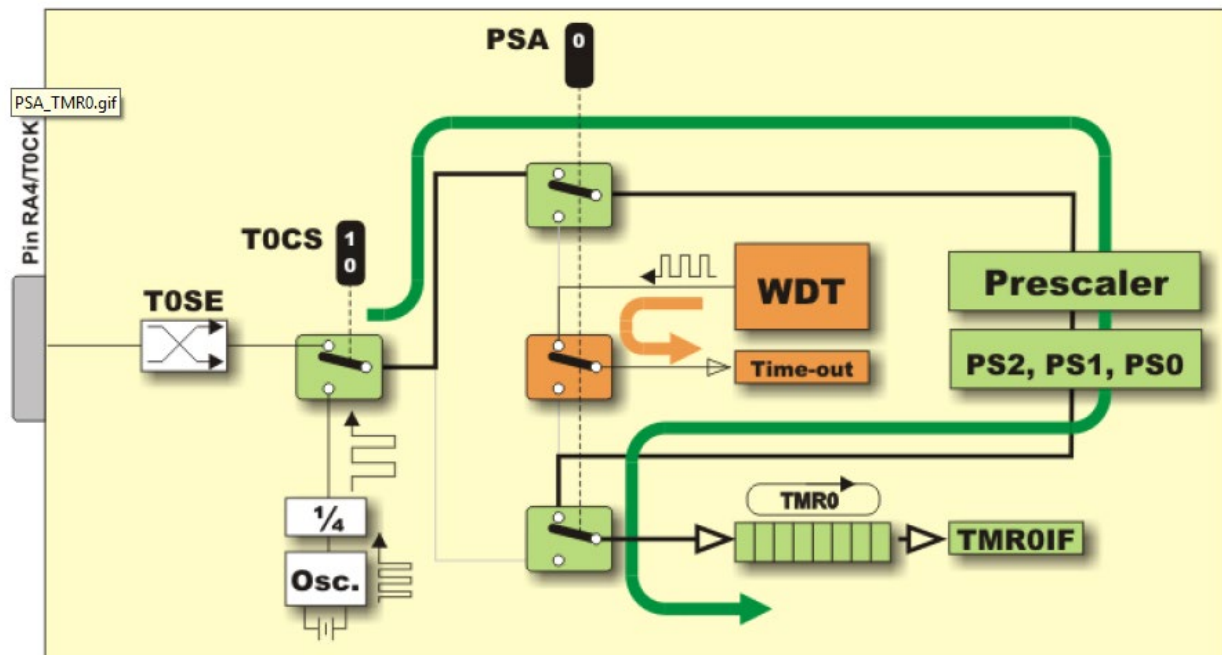
Уколико се Timer0 тактује преко RA4/T0CKI пина, могуће је дефинисати коришћењем T0SE (Timer0 Source Edge Select бита да ли ће се тактовање одвијати на узлазну или силазну ивицу. Ако је T0SE бит сетован, Timer0 ће се инкрементирати на силазној ивици RA4/T0CKI сигнала. Ако је пак T0SE бит ресетован, Timer0 ће се инкрементирати на узлазној ивици RA4/T0CKI сигнала.

Timer0 тајмер/бројач поседује програмабилни прескалер (дигитални делитељ фреквенције) који пропушта један импулс тек када се на улаз прескалера доведе одређени број импулса. Коришћењем прескалера продужава се временски интервал Timer0 тајмера пошто он неће бити тактован сваким побудним импулсом. Прескалер у Timer0

тајмеру/бројачу користи и WatchDog WDT тајмер, па се контрола над прескалером остварује преко PSA (Prescaler Assignment) бита у OPTION_REG регистру. Ако је PSA бит сетован, прескалер је додељен WDT тајмеру, а уколико је PSA ресетован, прескалер је додељује Timer0 тајмеру/бројачу. Степен дељења прескалера, подешава се преко PS2, PS1 и PS0 (Prescaler Rate Select) битова према следећој табели:

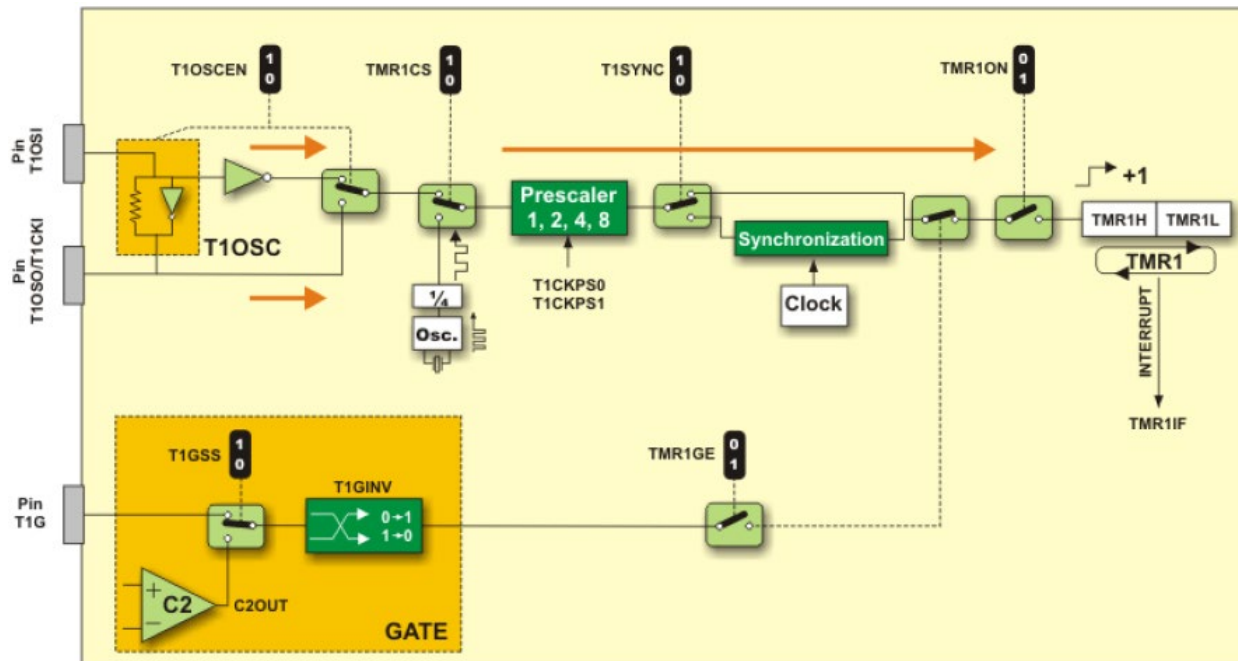
PS2	PS1	PS0	TMR0	WDT
0	0	0	1:2	1:1
0	0	1	1:4	1:2
0	1	0	1:8	1:4
0	1	1	1:16	1:8
1	0	0	1:32	1:16
1	0	1	1:64	1:32
1	1	0	1:128	1:64
1	1	1	1:256	1:128

Табела 1. Степен подешавања прескалера



Слика 10. Коришћење прескалера са TMR0 тајмером/бројачом

Timer1 је 16-битни тајмер/бројач унапред који се инкрементује после одређеног броја импулса при чему је број импулса дефинисан прескалером. Извор такта тајмера/бројача може бити интерни ($F_{osc}/4$) или екстерни. Тренутна вредност Timer1 тајмера/бројача може се уписати или прочитати преко два осмобитна регистра TMR1L и TMR1H. Када дође до препуњавања TMR1 регистра (прелазак са 0xFFFF на 0x0000 тј. са 65535 на 0) генерише се прекид тајмера T1IF. Тајмер/бројач Timer1 се може користити за прецизно мерење времена или бројање улазних импулса. Структура Timer1 тајмера/бројача приказана је на следећој слици.



Слика 10. Структура Timer1 тајмера/бројача



Слика 11. Структура регистра TMR1H и TMR1L

Тактовање Timer1 тајмера/бројача контролише се преко TMR1ON бита у T1CON регистру и GATE сигнала преко TMR1GE бита у T1CON регистру (GATE сигнал омогућава контролу тајмера само када је TMR1ON бит сетован).



Слика 12. структура T1CON регистра за контролу Timer1 тајмера/бројача

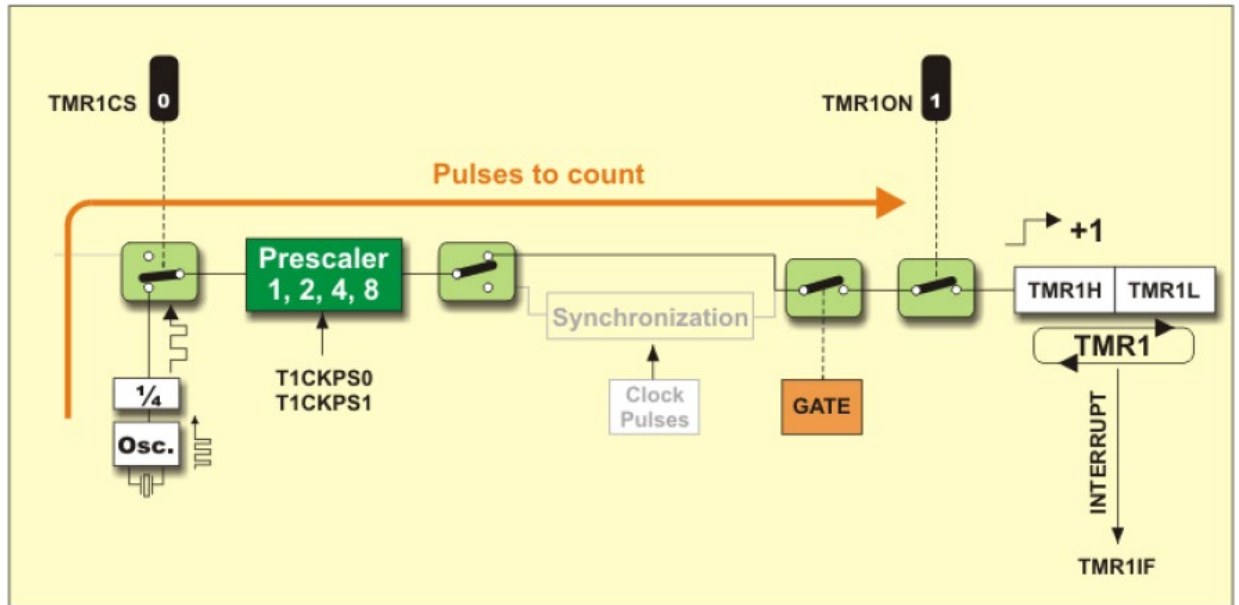
Прескалер Timer1 тајмера/бројача контролисан је преко T1CKPS1 и T1CKPS0 битова у T1CON регистру, према следећој табели:

T1CKPS1	T1CKPS0	Prescaler
0	0	1:1
0	1	1:2
1	0	1:4
1	1	1:8

Табела 2. Степени подешавања за Timer1

Избор такта за Timer1 тајмера/бројача врши се преко TMR1CS (Timer1 Clock Select) бита у T1CON регистру.

Уколико је TMR1CS бит ресетован, побудни импулси за Timer1 ће долазити од системског такта који је једнак $F_{osc}/4$. Том приликом онемогућена је синхронизација са системским тактом.



Слика 13. рад Timer1 тајмера/бројача у бројачком режиму

За бројање импулса смо изабрали 8 битни тајмер TMR0, а за мерење времена 16 битни тајмер TMR1.

Пошто ни један ни други тајмер није идеалан за то, јер ће јако брзо доћи до препуњавања, морали смо да направимо прекиде.

То смо решили на следећи начин:

TMR0 шаље импулсе сваких 0,5 μ s, али пошто је однос прескалера 1:256, то значи да шаље импулс сваких 128 μ s, његово време прекида је сваких 32,768 ms.

TMR1 прикључен је на системски такт $F_{osc}/4=2\text{MHz}$, тако да TMR1 шаље импулсе сваких 0,5 μ s. Прескалер је подешен на 1:1.

$$65\,536 - 32\,768 = 32\,768 \text{ imp}$$

$$\text{Време прекида једнако је } 0,5 \mu\text{s} * 32\,768 = 16\,384 \mu\text{s} = 16,384 \text{ ms}$$

$$\frac{1}{t_{prekida}} = \frac{1 s}{0.016384 s} = 61 \text{ prekid}$$

Из ове једначине видимо да је потребан 61 циклус прекида за TMR1 да би добили једну секунду.

```
- void interrupt() {  
  • if (TMR1IF_bit) {  
  •     cnt++; // increment counter  
  •     TMR1IF_bit = 0; // clear TMR0IF  
  •     TMR1H = 0x80;  
50  •     TMR1L = 0x00;  
  •     }  
  •  
  • if (TMR0IF_bit) {  
  •     imp=imp+256; // increment counter  
  •     TMR0IF_bit = 0; // clear TMR0IF  
  •     }  
  • }  
  • }
```

Слика 14. Прекиди

Бројач TMR0 броји импулсе до 256, и након што дође до прекида, вредност се додаје претходној вредности и ресетује се бројач како би могао поново да извршава бројање. Слично је и са TMR1 само што је он 16 битни тајмер и када дође до прекида инкрементира се променљива cnt.

Сигнал доводимо на RA4 порт.

```
• void main() {  
60   ANSEL = 0;           // Configure AN pins as digital  
•   ANSELH = 0;  
•   C1ON_bit = 0;       // Disable comparators  
•   C2ON_bit = 0;  
•  
•   TRISA.B4=1;         //RA4/TOCKI input  
•   OPTION_REG.T0CS=1;  
67   OPTION_REG.PSA=1;  
•  
•   Lcd_Init();         // Initialize LCD  
70   Lcd_Cmd(_LCD_CLEAR); // Clear display  
•   Lcd_Cmd(_LCD_CURSOR_OFF); // Cursor off  
•  
•   Lcd_Out(1,6,"Brojac"); // Write text in first row  
•   Lcd_Out(2,5,"obrtaja"); // Write text in second row  
•  
•   Delay_ms(2000);  
•  
•   Lcd_Cmd(_LCD_CLEAR); // Clear display  
•   Lcd_Out(1,6,"Brzina"); // Write text in first row  
80   Lcd_Out(2,13,"rpm"); // Write text in first row  
•  
•   PORTD = 0xFF;       // Initialize PORTB  
•   TRISD = 0;          // PORTB is output  
•   T1CON = 1;          // Timer1 settings  
•   TMR1IF_bit = 0;     // clear TMR1IF  
•   TMR1H = 0x80;       // Initialize Timer1 register  
•   TMR1L = 0x00;  
•   TMR1IE_bit = 1;     // enable Timer1 interrupt  
•   TMR0=0;  
90  
•   cnt = 0;            // initialize cnt  
•   INTCON = 0xE0;      // Set GIE, PEIE, TOIE
```

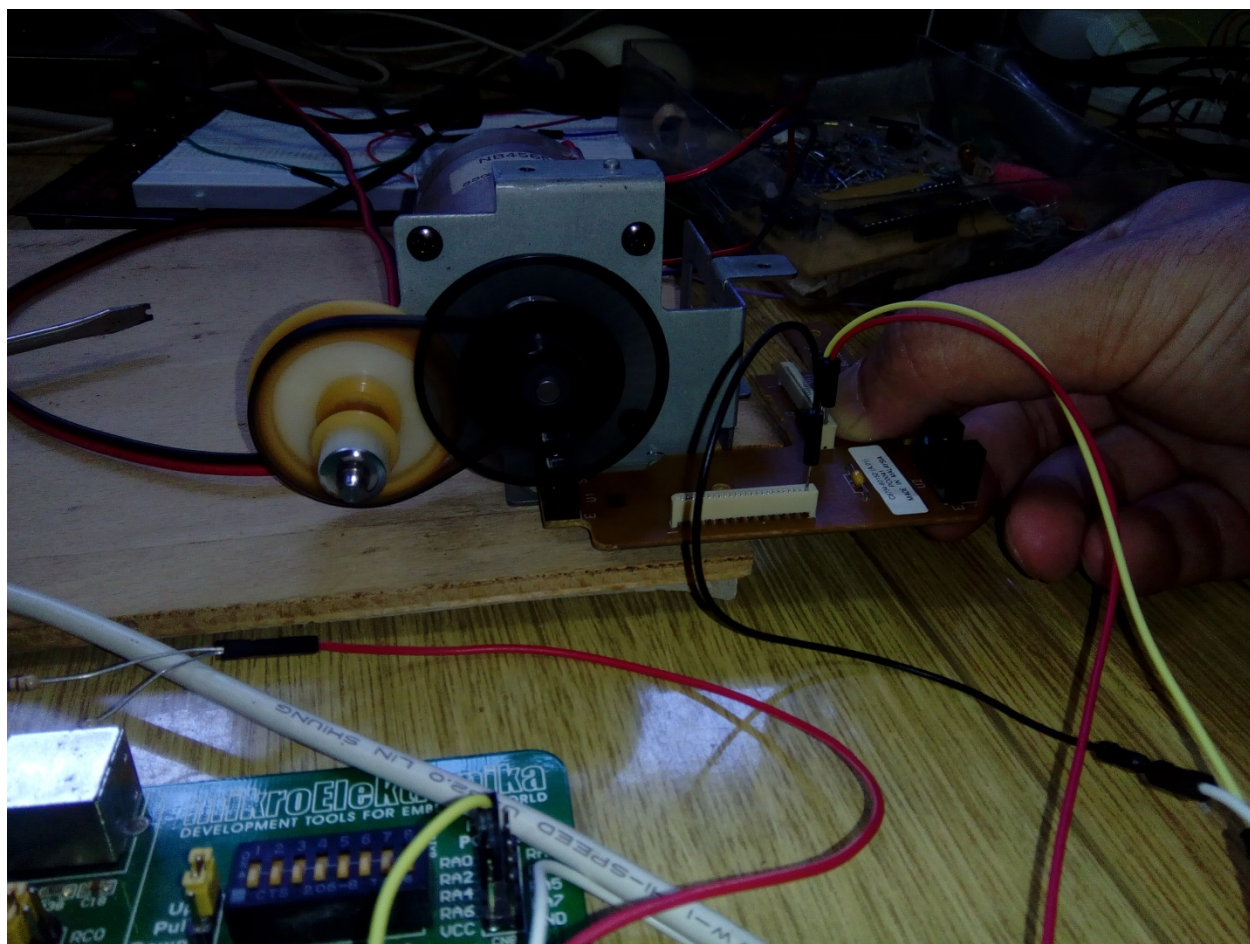
Слика 15. Изглед кода

Након што се деси 61 прекид тајмера TMR1(што је еквивалент 0.999424 s), извршава се ова петља у којој се тренутна вредност регистра TMR0 додаје променљивој `imp`, тако да имамо праву вредност пристиглих импулса. Број обртаја у минути добијамо тако што помножимо импулсе са 60 (да би добили минуте) и поделимо са 36 (број зазора на диску). Након тога се врши ресетовање TMR0 и `imp`.

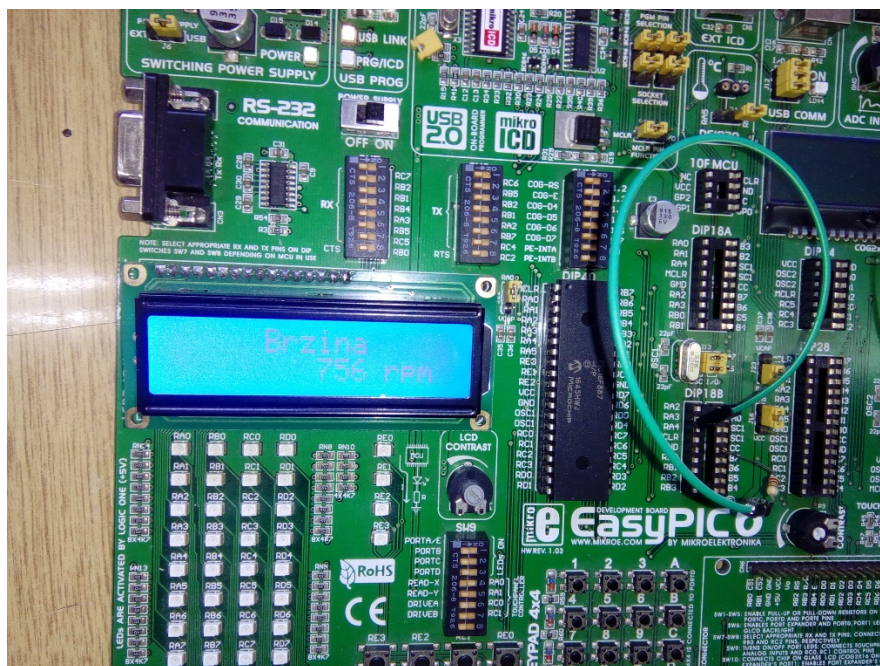
```
do {  
    if (cnt >= 61) {  
        PORTD = ~PORTD;           // Toggle PORTB LEDs  
        cnt = 0;                  // Reset cnt  
        imp=imp+TMR0;  
  
        rpm=(imp*60)/36;  
        LongToStr(rpm, Out_Text);  
        Lcd_Out(2,1,Out_Text);    // Write text in first row  
        TMR0=0;  
        imp=0;  
    }  
} while (1);
```

Слика 16. Петља

4. Изглед завршеног пројекта



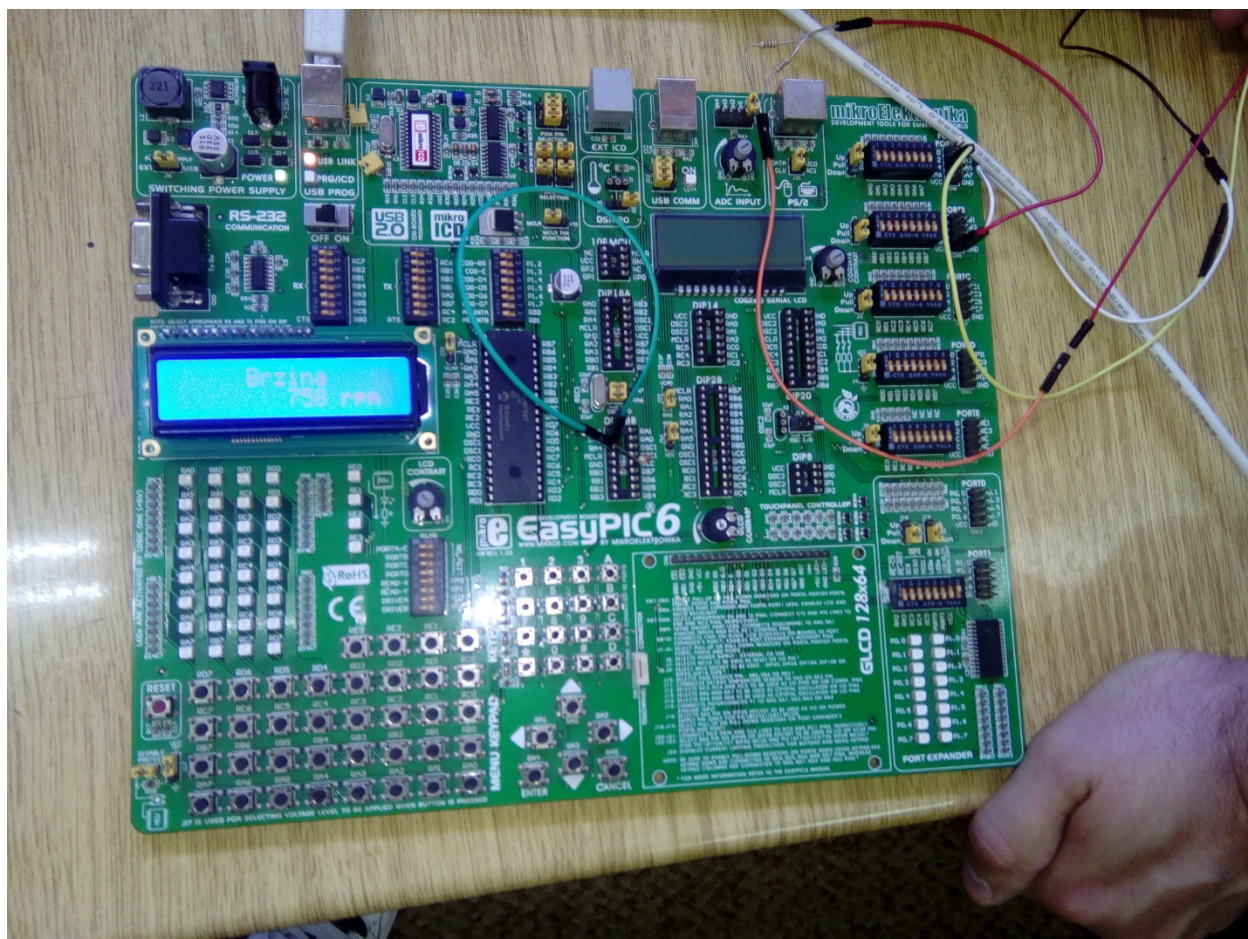
Слика 17. Мотор и оптокаплер



Слика 18. Дисплеј при брзини од 756 rpm-а



Слика 19. Дисплеј при брзини од 700 rpm-а



Слика 20. Брзиномер