LABORATOR 10 – PWM

Până acum toate aplicațiile scrise la laborator sau la curs au avut de îndeplinit o singură sarcină. Am calculat funcții logice în laboratoarele 3 şi 4, am afișat caracterul preluat de la tastatură în laboratorul 7, am controlat interfonul în laboratorul 8 sau am afișat timpul în laboratorul 9. În realitate o aplicație micro nu are de îndeplinit o singură funcție, ci mai multe. Un exemplu de funcție secundară executată de foarte multe aplicații micro este afișarea timpului. Cum în multe aplicații este necesară măsurarea timpului (cuptorul cu microunde încălzește mâncarea timpul programat) este foarte simplu să se afișeze timpul în standby.

O altă caracteristică a aplicațiilor scrise până acum constă în timpul scurt de execuție. Timpul maxim de execuție a buclei principale este probabil de câteva zeci de microsecunde pentru aplicațiile funcții logice și tastatură și ajunge la aproape o milisecundă la ceas. Ceasul necesită 1 ms nu pentru ca am face multe calcule ci pentru că avem de afișat 16 caractere.

După cum s-a menționat anterior, majoritatea aplicațiilor reale au de îndeplinit mai mute funcții iar unele dintre aceste funcții necesită timpi de execuție care pot ajunge de ordinul secundelor. În acest laborator vom scrie o aplicație care are de îndeplinit două funcții dintre care una necesită un timp mare de execuție.

Scopul lucrării

Se va scrie o aplicație care are de îndeplinit două sarcini:

1. Să controleze prin intermediul tastelor ‚C' şi ‚D' luminozitatea unui LED,
2. Să simuleze o sarcină care necesită un timp lung de execuție. Timpul lung de execuție se va obține prin apelarea funcției `_delay_ms(5000)`. Această sarcină simulată necesită un timp de execuție de 5 secunde.

După implementare, afișajul LCD va arăta astfel:

C / L	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	A	s	t	e	p	t										
2	L	=	7	5	%											

figura 1

Desfăşurarea lucrării

Pasul 1: Crearea proiectului

Se va crea un proiect nou cu numele **pwm**. Pentru crearea acestui proiect procedați după cum urmează:

a) **Se va crea** proiectul cu numele **pwm**.

b) **Copiați LCD.c** şi **LCD.h** din folderul **kbd** în folderul proiectului **pwm**.

c) **Adăugați** la proiect fișierele copiate la pasul b)

d) **Eliminați fișierul main.c** din proiect.

e) **Adăugați** la proiect fișierul **pwm.c** din platforma de laborator**.**

În pwm.c a secțiunea de afișare a caracterului preluat de la tastatură a fost eliminată.

Deoarece fiecare sarcină are un loc propriu unde afișează, în acest program nu se va folosi `clrLCD()`. Poziționarea se va face cu `gotoLC(…)` și afișarea noilor valori se va face prin suprascriere.

Pasul 2: Crearea schemei

Faceți o copie a schemei **kbd.simu** și redenumiți copia ca **pwm.simu**. **Conectați un LED** pe pinul OC0. **Conectați osciloscopul** pe pinul OC0.

Pasul 3: LED cu luminozitate variabilă

Controlul luminozității prin PWM este prezentat în prelegerea 5, capitolul **3.5** „**Modul fast PWM – fast Pulse Width Modulation**” și capitolul **3.6** **„Intensitate LED controlată prin PWM”**. În implementarea din acest laborator diferența față de curs constă numai în frecvența ceasului procesor: în curs f_{CLK_CPU} este 14,4MHz iar la laborator este 8MHz.

Procedura de setarea timerului 0 este foarte asemănătoare cu setarea timerului 2 din laboratorul 9 – ceas. Registrele Timerului 0 sunt descrise în documentația ATMega16A începând de la pagina 82. Pentru a determina valorile biților din registrul TCCR0 procedați după cum urmează:

a) Determinați valorile biților WGM00 şi WGM01 pentru a programa timerul 0 în mod fast PWM.

Bit	7	6	5	4	3	2	1	0	
	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	TCCR0
Read/Write	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Table 14-2. Waveform Generation Mode Bit Description(1)

Mode	WGM01 (CTC0)	WGM00 (PWM0)	Timer/Counter Mode of Operation	TOP	Update of OCR0	TOV0 Flag Set-on
0	0	0	Normal	0xFF	Immediate	MAX
1	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	1	0	CTC	OCR0	Immediate	MAX
3	1	1	Fast PWM	0xFF	BOTTOM	MAX

b) Determinați *p* astfel încât T_{cycle} să fie **cât mai aproape de 10 ms**. Determinați valorile biților CS02, CS01 şi CS00 pentru *p*-ul calculat.

Bit	7	6	5	4	3	2	1	0	
	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	TCCR0
Read/Write	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Table 14-6. Clock Select Bit Description

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$clk_{I/O}$/(No prescaling)
0	1	0	$clk_{I/O}$/8 (From prescaler)
0	1	1	$clk_{I/O}$/64 (From prescaler)
1	0	0	$clk_{I/O}$/256 (From prescaler)
1	0	1	$clk_{I/O}$/1024 (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

c) Determinați valorile biților COM01 și COM00 care controlează comportarea pinului OC0 conform tabelului următor.

Table 14-4. Compare Output Mode, Fast PWM Mode(1)

COM01	COM00	Description
0	0	Normal port operation, OC0 disconnected.
0	1	Reserved
1	0	Clear OC0 on compare match, set OC0 at BOTTOM, (non-inverting mode)
1	1	Set OC0 on compare match, clear OC0 at BOTTOM, (inverting mode)

d) Programați TCCR0 cu valorile determinate la pașii a-c. **Documentați** setarea lui TCCR0 la fel cum ați documentat setarea lui TCCR2 în laboratorul 9 sau la fel ca în programul de la pagina 20 din prelegerea 5.

e) Setați pinul OC0 ca ieșire. Nu modificați biții din DDR programați anterior.

Luminozitatea LED-ului se va modifica prin intermediul tastelor ‚**C**'(reşte) şi ‚**D**'(escreşte). Luminozitatea se modifică în trepte de 5% din luminozitatea maximă. Apăsarea lui ‚C' mărește luminozitatea cu 5% iar apăsarea lui ‚D' scade luminozitatea cu 5%. Creșterea și descreșterea sunt saturate, adică luminozitatea nu poate crește mai mult de 100% și nu poate scădea sub 0% indiferent de numărul de apăsări ale tastelor ‚C' și ‚D'.

Registrul OCR0 care controlează luminozitatea are 8 biți și astfel valorile luminozității sunt în intervalul 0 - 255. Luminozitatea care se introduce de la tastatură este în plaja 0-100. Din acest motiv este nevoie să calculăm valoarea ce se va scrie în registru în funcție de valoarea L setată de la tastatură.

Pentru L=100 în registru trebuie scris 255 iar pentru L=0 trebuie scris 0. Valoarea care se scrie în registru se calculează în funcție de L cu regula de trei simplă.

Comportarea numărătorului în mod fast PWM este un pic diferită fața de cele explicate anterior astfel încât valoarea care se va înscrie în OCR0 trebuie ajustată.

În modul neinversat OC0 este setat la:

- ‚0' **pe următorul puls** de ceas după detectarea egalității dintre TCNT0 și OCR0. Mai exact, există o întârziere de un puls de ceas între detectarea egalității și resetarea OC0.
- ‚1' **pe următorul puls** de ceas după pulsul care a comutat numărătorul este în starea zero.

Întârzierea de un puls de ceas conduce la **o corecție de -1** față de valoarea calculată cu regula de trei simplă.

Structura codului pentru variația luminozității este comentată în pwm.c. Decomentați și completați.

Afișați valoarea luminozității ca în figura 1. Noua valoare se scrie peste vechea valoare. Valoarea inițială a luminozității este 50%.

Testați dacă se modifica luminozitatea LED-ului!

Dacă funcționează, apelați profesorul pentru verificarea funcționării. Veți prezenta și calculele pentru *p*–ul și Tcycle-ulul timerului 0.

Pasul 4: Simulare calcul.

Pe lângă controlul luminozității se cere executarea unei secvențe de instrucțiuni care face un calcul. Această secvență ar trebui să dureze 5 secunde. Pentru că nu are importanță ce calcul se face ci numai durata, vom simula executarea calculului prin apelarea `_delay_ms(5000)`.

Simularea de calcul cu durata de 5 secunde se va executa dacă se apasă o tastă numerică. Mai multe detalii în programul din platformă.

Pe durata 5 secunde cât se face simularea de calcul , ce se întâmplă cu controlul luminozității? De ce?

Dacă funcționează și aveți un răspuns la întrebările de mai sus, apelați profesorul pentru verificarea finală.