## MICROCONTROLERE – Lucrarea de laborator 1

#### Scopul lucrarii:

- descrierea modului de lucru cu compilatorul Code Vision AVR si cu programul de depanare Astudio, specifice microcontrolerelor Atmel
- descrierea arhitecturii microcontrolerelor AVR Atmel
- realizarea unui program, in limbaj C, cu programarea porturilor de intrare iesiresi a timerului din structura microcontrolerului Atmel AT90S8515.
- prezentarea placii de evaluare cu microcontroler AVR-Atmel, STK 500
- modul de executie cu simulare (depanare) si pe placa reala

#### Desfasurarea lucrarii

- 1. Se va studia arhitectura microcontrolerului AVR Atmel AT90S8515
- 2. Se va studia modul de operare al programului CVAVR
- 3. Se va studia arhitectura placii de evaluare STK 500
- 4. Se va contrui un proiect nou care sa realizeze programarea porturilor in conformitate cu structura STK 500 si a timerului care va genera intreruperi periodice cu perioada de 20 ms.
- 5. Se vor realiza urmatoarele programe (a-d):
  - a) se va aprinde LED-ul corespumzator butonului SW apasat
  - b) la apasarea butonului SW0 se vor aprinde LED-urile pare, iar la apasarea butonului SW7 se vor aprinde LED-urile impare
  - c) la apasarea butonului SW1, LED-ul LED0 se va aprinde/stinge cu o cadenta de 1 sec.
  - d) se vor aprinde LED-urile LED0-7 pe rind cite 0.5 secunde incepind cu LED-ul 0, in mod cyclic

Toate programele vor fi simulate cu Astudio si rulate in timp real pe STK 500.

# Descrierea modului de lucru cu compilatorul CVAVR si a depanatorului ASTUDIO

 File
 Edit Search View Project Tools Settings Windows Help

 Code Navigator
 Code Templator

 Code Navigator
 Code Templator

 No Project
 Other Files

Se lanseaza in executie programul CVAVR (Code Vision AVR).

#### Crearea unui proiect nou

Se selecteaza din bara de meniu - File->New

SX CodeVisionAVR		
File Edit Search View Pr	aject Tools Settings Windows Help	
🚺 New 🛛 Ozi+N		ㅈ∎@
🈂 Open 🛛 Otrl+O	220 LEEEE 0 8	S 🕸 🕯 S 🚍 🖂 🖬 🖬 👔 😯 🕄
Reopen     Save     Ctrl+5     Save As     Save Al     Shift+Otrl+5     Cose     Cose     Cose Al	dec ( )	
Convert to Library Convert to Library Page Setup Convert Print Preview Print Ctrl+P		
Maccages	1	
	Insert	

In fereastra "Create new file" se selecteaza 'Project"

<b>#</b> CodeVisionAVR	- C 🛛
File Edit Search Wew Project Tools D Code Code A an ignore 第 2000 Code Navigator Code Templater (*) Code VisionAVR No Project Other Files	Settings Windows Help AGE ON & DERX II OF EET X SI ON ON ON ON ONE EEE BEBORIS OF SIGNAL File Type O Source O Project C OK O Project C OK
Messages	
Insert	

#### Se apasa "OK" :

📓 CodeVisionAVR	×
File Edit Search View Project Tools Settings Windows Help         Image: Search View Project Tools Search View Project Tools Search View Project Tools View Proje	
Messages	_
Treast	_

Se opteaza pentru utilizarea "Code Wizard AVR" care va asista crearea proiectului. Programul "Code Wizard AVR" va genera un program sursa care va contine toate programarile resurselor dorite. Programarea acestor resurse se va realize in mod graphic.

Apare o fereastra de dialog pentru selectarea resurselor din microcontroler: Se vor programa doua tipuri de resurse: porturile de intrare – iesire (PORTD – intrare, PORTB – iesire) si timer-ul TIMER0.

Se va selecta "Ports" :

😃 CodeWizardAVR - untitled.cwp 🛛 🔀	🖗 CodeWizardAVR - untitled.cwp 🛛 🔀
File Help	File Help
File Help External IR0 Timero UART Analog Comparator SPI 12C 1 Wire LCD BitBanged Project Information Ohip External SRAM Ports Ohip External SRAM Ports Ohip External SRAM Ports Ohip AT9058515 M Clock: 4.000000 M MHz	File       Help         External IRQ       Timers       UART         Analog Comparator       SPI       I2C       1 Wire         LCD       Bit-Banged       Project Information         Drip       External SRAM       Ports         Port A       Port B       Port C       Port D         Data Direction       Pulup/Output Value       Bit 0       Bit 1       In       T       Bit 0         Bit 1       In       T       Bit 1       Bit 2       Bit 3       In       T       Bit 3         Bit 4       In       T       Bit 3       Bit 4       In       T       Bit 3         Bit 5       In       T       Bit 3       Bit 4       In       T       Bit 3         Bit 6       In       T       Bit 6       Bit 7       In       T       Bit 7

Se selecteaza "Port B" ca iesire cu totii bitii in "1" si "Port D" ca intrare cu totii bitii "Pull up" (P):

Celelalte optiuni sint : "0" – pentru iesire si starea de inalta impedanta ("three state" – T) pentru intrari. Fiecare bit al unui port poate fi programat individual ca intrare sau ca iesire.

Un port I/O este constituit din 3 registre :  $PORTx - contine iesirile ( daca portul este de iesire), PINx - contine intrarile ( daca portul e de intrare) si DDRx - Direction Data Register - care indica pentru fiecare bit daca acesta e intrare ("0") sau iesire ("1"). Sufixul "x" indica portul din microcontroler - x = { A,B, C, D, E ... }.$ 

In programele scrise in limbajul C se pot accesa porturile de intrare – iesire astfel:

- porturile de intrare :
- a= PINx; // se citeste in variabila a valoarea portului de intrare x a= PINx.y; // se citeste in variabila a valoarea bitului y al portului de intrare x
- porturile de iesire :

PORTx=a; // se scrie in portul de iesire x valoarea variabilei a PORTx.y=a; // se scrie in portul de iesire x bitul y valoarea variabilei a

🕏 CodeWizardAVR - untitled.cwp 🛛 🔀	D Co	deWizardA	VR - unt	titled.c	wp [
File Help	File H	ielp			
External IRQ Timers UART	E	demail IRQ	Time		UART
Analog Comparator SPI I2C 1 Wire	Anal	og Comparator	SPI	120	1 Wire
LCD Bit-Banged Project Information	LCD	Bit-Ban	ged F	roject In	iomation
Chip External SRAM Ports	Ch	ip Ext	emal SRA	M	Parts
Port A Port B Port C Port D	Po	rt.A Port B	Part C	Port D	
Data Direction Pullup/Dutput Value		Data Direc	tion Pu	lup/Dut	put Value
Bit0 Out 1 Bit0		B≹O	In	P BR 0	
Bit Out 1 Bit 1		Bit 1	In J	P Bit 1	
Bit 2 Out 1 Bit 2		B≹2	In	P Bit 2	
Bit3 Out 1 Bit3		Bit 3	In J	P Bit 3	
Bit 4 Out 1 Bit 4		Bit 4	In F	P Bit 4	
B&S Out 1 B&S		Bit 5	In F	P Bit 5	
BR6 Out 1 BR6		Bite	In F	P Bit 6	
B≹7 Out 1 B≷7		Bit 7	In F	P BR 7	
·					
	· · · · · · · · · · · · · · · · · · ·		_	_	

Se selecteaza "Timers" -> Timer0; Clock source – "System Clock", "Clock Value" 3.906 kHz, se bifeaza "Overflow Interrupt" si se inscrie in "Timer Value" valoarea "4e". Timerul 0 este programat sa genereze o intrerupere la fiecare 20ms.

CodeWizardAVR - untitled.cwp	😫 CodeWizardAVR - untitled.cwp 🛛 💈
e Help	File Help
Analog Comparator SPI 12C 1 Wire	Analog Comparator SPI 12C 1 Wire LCD Bit/Banged Project Information
Chip External SRAM Ports	Chip External SRAM Ports
External IRQ Timero LIART	External IRQ Timers LIART
mer 0 Timer 1 Watchdog	Timer 0 Timer 1 Watchdog
ock Source: System Clock.	Clock Source: System Clock 💌
lock Value: Timer 0 Stopped	Clock Value: 3.906 kHz 🗸
Overflow Interrupt	C verflow Interrupt
limer Value: 0 h	Timer Value: 4e h

Dupa terminarea setarilor dorite pentru toate resursele se selecteaza din meniu "File" si "Generate, Save and Exit". Se salveaza proiectul.

CodeWizardAVR - untitl	ed.cwp 🔀			
Nie Help				
<ul> <li>New</li> <li>Open</li> <li>Save</li> <li>Save As</li> </ul>	1 Wire formation Ports UART			
Program Preview     Generate, Save and Exit     Exit	×	Sam C Corrollar Sauron D		8
		Save is 🔁 char to	et s	0000
Verflow Interrupt		My Recent Documents Decision Decision		
Timer Value: 4e h		My Documents		
		My Conputer		
		Fierank	1	M Save
L		My Network Save as type	C Compiler files ("	Cancel

Programul generat automat este urmatorul:

Automatic Program Generator © Copyright 1998-2008 Pavel Haiduc, HP InfoTech s.r.l. http://www.hpinfotech.com
Project : Version : Date : 8/1/2008 Author : Freeware, for evaluation and non-commercial use only Company : Comments:
Chip type : AT90S8515 Clock frequency : 4.000000 MHz Memory model : Small External RAM size : 0 Data Stack size : 128
#include <90s8515.h>
char a;

<pre>// Timer 0 overflow interrupt service routine interrupt [TIM0_OVF] void timer0_ovf_isr(void) </pre>
// Reinitialize Timer 0 value TCNT0=0x4E;
// Place your code here
PORTB=a;
}
// Declare your global variables here
void main(void)
{
// Declare your local variables here
// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTA=0x00;
DDRA=0x00;
// Port B initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State/=1 State6=1 State5=1 State4=1 State3=1 State2=1 State1=1 State0=1
DDRB=0xFF;
// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
DDRC=0x00;
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=P State6=P State5=P State4=P State3=P State2=P State1=P State0=P
PORTD=0xFF; DDRD=0x00:
// Limer/Counter 0 initialization // Clock source: System Clock
// Clock value: 3.906 kHz
TCNT0=0x4E,
// Timer/Counter 1 initialization
// Clock source: System Clock // Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge

// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
// External Interrunt(s) initialization
// INITO: Off
// INT1: Off
GIMSK=0x00
MCUCR=0x00:
// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x02;
// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x00,
// Global enable interrupts
#asm("sei")
while (1)
{
// Place your code here
) .

Codul utilizatorului poate fi scris in rutina de servire a intreruperilor sau in programul principal. De asemenea pot fi declarate variabile globale sau locale.

Vom introduce urmatoarele modificari in programul generat automat:

- se va declara o variabila globala a de tip caracter
- in variabila a se va citi portul PORTD
- valoarea lui a se va transmite in portul PORTB

Modificarile sint marcate in programul anterior cu litere ingrasate pe fond colorat.

Programul va fi compilat si apoi rulat cu ajutorul depanatorului ASTUDIO.



Daca nu au aparut erori se genereaza fisierul executabil cu comanda "Build":



#### Simularea si depanarea programului

Se lanseaza depanatorul ASTUDIO (butonul 🚺):



Se deschide, cu butonul "Open" fisierul in format "\*.cof":

Open Project F	ile or Object Fi	le				2 🛛
Look in	C descriere		•	🗢 🗈 💣	<b>H</b> -	
Ny Recent Documents Desktop	Exe Unker Ust Obj Hest.cof					
My Documents						
My Computer						
8						_
My Network Places	File name:	test.col		-		Open
	Files of type:	Project Files, Dbject Files (*.a)	ps,".hex	∴d90,1.at 💌	_	Cancel

Se salveaza proiectul in format ASTUDIO:

Save AVR Studi	o Project File				2 🗙
Save in:	껕 descriere		• • •	💣 💷 •	
My Recent Documents Desktop	Cinker Linker List				
My Documents					
My Computer					
<b>S</b>					
My Network Places	File name:	test_col.aps		9 🗆	Save
	Save as type:	AVR Studio Project Files (".ap	e:	-	Cancel
AVR Studio will on file should be local save the project fil	eate a project for deb ted at the root of the le at the same place	ougging this object file. To ensu original source file project. If the as the object file	re optimal debug e original source i	ging, AVR Studi is not available,	o's project Jou can

Se selecteaza platforma "AVR Simulator" si microcontrolerul "AT90S8515"

Welcome to AVR Stud	Select debug platform and device	Device					
Sindlo 4	AVR Diagon AVR Simulator AVR Simulator V2 (preview) ICE200 ICE40 ICE50 JTAG ICE JTAGICE mkll	AT90PwM3 AT90PwM3B AT9051200 AT9052313 AT9052323 AT9052343 AT9052343 AT9054414 AT9054434 AT9054434 AT9054515 AT9058535	8				
	Dpen platform options						
Ver 4.13.529	K Back Ne	1	Help				

Se apasa butonul "Finish" si se expandeaza PORTB, PORTD si Timer0:



Programul poate fi simulat pas cu pas (cu tasta F11) sau complet (cu tasta F5). De asemenea se pot introduce puncte de intrerupere a programului - *breakpoints* (cu tasta F9). Se pot vizualiza resursele microcontrolerului (registre, memorie, porturi I/O) variabile ale programului.

<u>E</u> dit	<u>V</u> iev	v <u>T</u> ools	<u>D</u> ebug	<u>W</u> indo	W
Х 🗈		<u>T</u> oolbars		•	h /
Ŧ	~	<u>S</u> tatus Ba	r		
	S.	Dis <u>a</u> ssemt	oler		
0x0000	æ	Wat <u>c</u> h		Alt+1	2111
0x025F		<u>M</u> emory		Alt+4	Int
0x0260	F.	Memory 2	1		Por
0x00E0	:==	Memory 3	-		fur +=
0x1802		nomory g	_		ΓΔ=
3158	ōx	<u>R</u> egister		Alt+0	A=C



Valorile porturilor de intrare iesire pot fi modificate in mod grafic:

- inainte de executia rutinei de servire a intreuperilor:

h Dr.V. auferland 2008, 200 Messawar Institutes of	in a literate		<b>E</b> 16		UO View		
Chine house a statistic statistic	iere stest.	v					× 🖬
Clock frequency 4.001010	HHE			<b>_</b>	Name	Add Value	Dits
External RAE mize 0					∃ DANALOG_O	OMPARATOR	
Data Stack size : 128					ALC: NOT		
				-	# SEXTERNAL	NTERRUPT	
finclude (90s8515.h)					# PORTA		
cura o.					🖃 🛃 PORTB		
// Timer D overflow interrupt a interrupt [TIN0_0981 word timer.	ervice : 0 out i	routine			2 POR18	IM8 OFF	
{	0_0+1_1	an ( Horse )			P DO FE	DAT. OFF	
// Reinitialize Timer 0 value TONTO-0-4E:					PIN9	IM16 OWFF	
// Place your code here					A PORTO		
a=PIND; BORTBase;					PDR1D	M2_0/F	
, ) ,				-	🕈 DORD	B-110-00	
4				▶ <i>.</i> /.	📫 PND	B400/12	
· · · · · · · · · · · · · · · · · · ·	latch				× 8,000	INTER_1	
	lame	Value	Туре	Location	i_00.	IN	
	4	0x00 ***	unsigned char	P4	PHDO	9	

- dupa executia rutinei de servire a intreruperilor:



#### Executia in timp real a programului (pe placa de evaluare STK500)

Programul poate fi incarcat in placa de evaluare STK500 pentru a fi executat in timp real cu butonul

Connect failed - Select AVR Programmer					
Platform: STK500 or AVRISP JTAG ICE JTAGICE mkII AVRISP mkII AVR Dragon	Port:	Connect Cancel			
Tip: To auto-connect to the programmer used last time, press the 'Programmer' button on the toolbar.					
Note that the JTAGICE cannot be used for programming as long as it is connected in a debugging session. In that case, select 'Stop Debugging' first.					
Disconnected Mode					

Se selecteaza placa de evaluare STK500 si portul Auto. Conectarea se va face cu butonul "Connect".

Se va efectua programarea memoriei flash interne a microcontrolerului:

Program Fuses LockBits Advanced Board Auto	
C Device	
AT90S8515 Erase Device	
Programming mode	
ISP mode   Frase Device Before Program Verify Device After Program	iming ing
Flash	
Use Current Simulator/Emulator FLASH Memory     Input HEX File	
Program Visitin Band	
EEPROM     ELSe Current Simulator/Emulator EEPBOM Memory	
Input HEX File	
Program Verify Read	
	^
	~

#### Descrierea functionala a porturilor de intrare – iesire (microcontroler AVR)

In figura 1 este ilustrata implementarea tipica a unui port de intrare iesire ai carui biti pot fi configurati individual. Sint necesare 3 registre:

- 1 registru de iesire (PORT sau PORTx, x = A,B,C,D,...)
- 1 registru de intrare (PIN sau PINx, x=A,B,C,D,...)
- 1 registru de directie (DDR sau DDRx, x=A,B,C,D,...)



Figura 1. Port de intrare – iesire configurabil

Scrierea registrului DDR cu 0 sau cu 1 valideaza sau nu bufferele ce au ca iesire pinii extern ai portului. O scriere in port se va efectua la adresa registrului PORT, iar o citire a portului se va realiza printr-o citire de la adresa registrului PIN.

#### Descrierea functionala a Timer-ului (microcontroler AVR)

Circuitul de timp programabil (Timer) este format din doua blocuri: un circuit de prescalare (ce va genera un tact  $T_{CKx}$ , utilizat in cel de al doilea bloc) si un registrunumerator ce va fi decrementat la fiecare front activ al semnalului de ceas generat de blocul de prescalare.

In figura 2 este ilustrat blocul de prescalare:



Figura 2 Blocul de prescalare

 $P_{CKx}$  reprezinta semnalul de tact la intrarea in blocul de prescalare (ceasul de system). Sufixul x = 0,1 este numarul timeru-lui.

Exista un registru de comanda al timer-ului, TCCRx; in acest registru bitii CSx2, CSx1 si CSx0 sint utilizati ca intrari de selectie pentru multiplexorul care genereaza semnalul de ceas catre blocul registru-numarator ce va fi decrementat.

Ca semnal de ceas pentru decrementare poate fi utilizat si un semnal extern, Tx. Structura detaliata a timer-ului 0 este ilustrata in figura 3.





Figura 3. Structura timer-ului 0

Se observa ca exista 2 registre TIFR (Timer Interrupt Flag Register) si TIMSK (Timer Interrupt Mask Register), utilizate pentru a genera intreruperi la fiecare trecere prin zero a registrului numerator. Perioada intreruperilor este calculata prin impartirea perioadei ceasului generat de blocul de prescalare la o constanta prestabilita.





### PLACA DE EVALUARE AVR- STK 500

Placa de evaluare AVR-STK500 permite studierea modului de dezvoltare a aplicatiilor cu microcontrolere Atmel-AVR.

Structura STK500 este prezentata in figura 5.



Figura 5. Structura STK500

Placa are in componenta urmatoarele blocuri:

- bloc de comunicare cu un calculator PC (pe linie seriala RS232)
- microcontroler AVR
- bloc de intrare iesire ( 8 butoane SW0-SW7 si 8 LED-uri LED0-LED7)
- bloc de alimentare
- bloc de programare in circuit a memoriilor interne ale microcontrolerului AVR
- port de comunicatie seriala suplimentar

Arhitectura microcontrolerului AVR este prezentata in figura 6:



Figura 6 Arhitectura microcontrolerului AVR (AT90S8515)

De asemenea placa permite conectarea butoanelor SW si a LED-urilor pe diferite porturi ale microcontrolerului AVR.

Conectarea placii STK500 cu un calculator se realizeaza ca in figura 7.



Figura 7. Conectarea STK500 cu un calculator

Comunicatia intre calculator si STK500 precum si programarea memoriei interne se realizeaza prin intermediul programului Astudio.

Structura butoanelor si a LED-urilor este urmatoarea:

Butoanele SW sint realizate ca in figura 8 (starea de repaus este 1 logic). Toate butoanele SW0-7 sint conectate cu portul PORTD care trebuie configurat ca intrare.



Figura 8. Butoanele SW0-SW7

Apasarea unui buton SW are ca effect transmiterea, pe pinul corespunzator al portului PORTD, a semnlului 0 logic.

LED-urile sint conectate pe portul PORTB, care trebuie configurat ca iesire. Modul de conectare al unui LED este ilustrat in figura 9.



Figura 9. Conectarea LED-urilor

Pentru aprinderea unui LED se va scrie, pe pinul corespunzator al portului PORTB, valoarea 0 logic.