Osnove mikroprocesorske elektronike

Atmel Studio: GCC projekt

Priprava projekta

Atmel Studio je razvojno okolje za razvoj programov za Atmelove mikrokrmilnike AVR in ARM. Združuje urejevalnik besedil, prevajalnik oziroma zbirnik ter razhroščevalnik (debugger). Preden lahko začnemo z delom moramo ustvariti projekt:

- 1. Nov projekt ustvarimo z izbiro »File/New/Project... File Edit View VAssistX Project Debug Tools Window Help New

 Ctrl+Shift+N
- V oknu, ki se pojavi, izberemo skupino predlog »C/C++«, izberemo predlogo »GCC C Executable Project«, v polje »Name:« vpišemo ime projekta in v polje »Location:« mapo, kjer hočemo imeti novi projekt.



3. V naslednjem oknu izberemo, kateri procesor bomo uporabljali (ATmega328PB).

Device Selection						×
Device Family:	All	v			Search for device	۶
Name	App./Boot Mem	ory (Kbytes)Data Memory	(bytes)EEPROM (bytes	Device Info:		^
ATmega325 ATmega3250 ATmega3250A ATmega3250P ATmega3250PA ATmega325A ATmega325P ATmega325PA ATmega325PA	32 32 32 32 32 32 32 32 32 32 32 32	2048 2048 2048 2048 2048 2048 2048 2048	1024 1024 1024 1024 1024 1024 1024 1024	Device Name Speed: Vcc: Family: Datasheet (Device Page Supported Ti	e: ATmega328PE N/A N/A ATmega Summary) e	3
ATmega328P ATmega328PB ATmega3299 ATmega32900 ATmega32900A ATmega3290PA ATmega3290PA ATmega3290PA ATmega329PA	32 32 32 32 32 32 32 32 32 32 32 32 32	2048 2048 2048 2048 2048 2048 2048 2048	1024 1024 1024 1024 1024 1024 1024 1024	 Atmel-IC AVR Dra AVRISP AVRISP AVRONI EDBG EDBG M JTAGICE 	E aon mkli El SD 33	~
					ОК	Cancel

Ko je čarovnik za nov projekt zaključen, se prikaže okno z odprto datoteko, kamor bomo pisali program. Čarovnik je že pripravil minimalni program, ki ga lahko takoj prevedemo in naložimo na procesor.



Izberimo še orodje za odkrivanje napak – v tem primeru simulator:

1. Kliknemo na »Project/ImeProjekta Properties...«



2. V prikazanem oknu v razdelku »Tool« izberemo želeno orodje:

Vaja1 - AVRStudi		
File Edit View	VAssistX Project Build Debug Tools Window He	lp
: 🛅 - 🔁 🔛 - 🛛	🖌 🛃 👗 🖧 🖄 🖄 👘 - (*) - 💭 - 🖳 🔛 🕪	
😨 🐱 😹 🍋 t	? ? 0, <u>*</u> iš , i 2 = 1 , 3 , i 1 , 1 =	\$
Vaja1 🗙 Vaja1.as	m	
Build	Configuration: N/A	arma
Build Events		
Toolchain	Selected debugger/programmer	
Device	Interface:	
Tool	AVR Simulator	
Advanced	00A200018080 reset	
	Memory Settings	
	Preserve EEPROM	

Stopnja optimizacije

Stopnja optimizacije močno vpliva na velikost programa in hitrost izvajanja programa, hkrati pa tudi na neposredno primerljivostjo C programa s strojno kodo – če je stopnja optimizacije 0, je sledenje programu z razhroščevalnikom enostavno, če je stopnja optimizacije karkoli drugega, pa zelo težko. Stopnjo optimizacije nastavimo pri lastnostih projekta:

1. Kliknemo na »Project/ImeProjekta Properties...«

+ vajai - Avitstudio							
File Edit View VAssistX	Proje	ect	Build	Debug	Tools	Window	Help
🛐 • 🔁 🗉 • 💕 🚽 🏈	8.	ASF	Wizar	d			
🔁 🖾 🐺 🍋 🔓 🧐 🗤	3	Show All Files					
Vaja1.asm ×		Run	Custo	m Tool			
/*		Set	as Star	tUp Proje	ct		
* Vaja1.asm		Vaja	a1 Prop	erties		Alt+F7	

2. V prikazanem oknu v razdelku »Toolchain« izberemo kategorijo »AVR/GNU C Compiler« in podkategorijo »Optimization«:



3. Izberemo želeni nivo optimizacije in shranimo projekt.

***	F7	prevede program
	Alt+F5	prevede program in ga naloži v okolje za odkrivanje napak (simulator, DRAGON,)
	Ctrl+Shift+F5	prekine odkrivanje napak (prekine tudi povezavo do orodja, npr. DRAGONa)
SI	F11	- izvede en ukaz
	F5	požene program
11	Ctrl+F5	ustavi program

Ikone in bližniice

Okna za odkrivanje napak

I/O View (Debug/Windows/IO	View):	
Vaja1 (Running) - AVRStudio		
File Edit View VAssistX Project Build	Debug Tools Window Help	
: 🗊 - 🕮 🗃 - 📂 🔜 🕼 X 🗈 🖎 (Windows •	Breakpoints Alt+F9
: 🔁 📴 🖓 🍋 암 암 야. 🔬 🛍 🚽 🐺	Start Debugging and Break Alt+F5	Processor View
Vaja1.asm ×	Stop Debugging Ctrl+Shift+F5	🛐 I/O View
	a)	

Slika 2: a) Vklop prikaza perifernih naprav in b) prikaz stanja bitov porta B.

Processor View (Debug/Windows/Processor View):

Processor		• 4 ×
Name	Value	
Program Counter	0x0000003	*
Stack Pointer	0x08FF	
X Register	0x0000	
Y Register	0x0000	
Z Register	0x0000	-
Status Register	ITHSVNZC	
Cycle Counter	0	
Frequency		
Stop Watch		
Registers		
R00	0x00	
R01	0x00	
R02	0x00	
R03	0x00	
R04	0x00	
R05	0x00	
R06	0x00	
R07	0x00	
R08	0x00	Ŧ
🔄 IO View 🔤 P	rocessor 💐 Solution E	Properties

Slika 3: Prikaz stanja procesorskega jedra.

Disassembly (Alt+8 ali Debug/Windows/Disassembly):

Disasser	mbly							•
Address	s:							1
 ♦ Viev ♥ ♥ ♥ ♥ 	wing O Show Show Show	ption code sour line i	e byte ce co num	es ode bers			✓ Show address ✓ Show symbol names	
	19:		ldi	r16,0b1000000	1			^
0000	00049	01.	e8		LDI	R16,0x81	Load immediate	
0000	20:	05	out	PORTB, PIG	OUT	0×05 016	Out to T/O losation	
0000	21.	05.1	141	c16 0b0100001	9	0X05,810	out to 1/0 location	
0000	0004B	02.	e4	10,000100001	LDT	R16, 0x42	Load immediate	
	22:		out	PORTB, r16				
0000	0004C	05.	b9		OUT	0x05,R16	Out to I/O location	
	23:		ldi	r16,0b0010010	0			
0000	9004D	04.	e2		LDI	R16,0x24	Load immediate	
	24:	4	out	PORTB, r16				
0000	0004E	05.1	b9		OUT	0x05,R16	Out to I/O location	_
0000	25:		Idi	r16,0b0001100	0	D4.5 0.40	Land Jame Hate	
0000	26.	08.1	er	DODTR a16	LUI	K16,0X18	Load immediate	
0000	20.	05	ha	POKID, 110	OUT	0x05 816	Out to T/O location	
0000	27:	0.5.1	rim	main loop	001	0x003yn10		
	00051	f7.	cf		RJM	P PC-0x0008	Relative jump	
	No so	urc	e fi	le				
0000	00052	ff.	ff		NOP	Undefin	1ed	
0000	00053	ff.	ff		NOP	Undefin	ied	
0000	00054	ff.	ff		NOP	Undefin	ied	
0000	00055	TT.	TT		NOP	Undefin	led	
0000	00050	££ .	5.5		NOP	Undefin	ied and	
0000	0057	ff .	ff		NOP	Undefin	ned and a set of the s	
0000	30059	ff.	ff		NOP	Undefin	ned	
0000	0005A	ff.	ff		NOP	Undefin	ned	
0000	0005B	ff.	ff		NOP	Undefin	red	
0000	9005C	ff.	ff		NOP	Undefin	red	
0000	0005D	ff.	ff		NOP	Undefin	ned	
0000	0005E	ff.	ff		NOP	Undefin	ied	
0000	0005F	ff.	ff		NOP	Undefin	ied	
0000	00000	tf.	TT		NOP	Undefin	ied	
0000	20061	TT.	ГТ с.с.		NOP	Undefin	lea	\sim
<							>	

Slika 4: Prikaz dejanskega zapisa programa v procesorju.