PROGRAMIRANJE MIKROKONTROLERA

Programiranje mikrokontrolera u asembleru predstavlja optimalan nacin programiranja sa stanovišta iskorišćenja ograničenih ugrađenih resursa, kao na primer, programske memorije i dr. S druge strane za složenije projekte program napisan u asembleru može biti komplikovan za čitanje i razumevanje. Zato se danas mikrokontroleri programiraju u višim programskim jezicima, a najpoznatiji i najčešce korišćeni je programski jezik C.

Pretpostavkom da student poseduje osnovna znanja o programskom jeziku C i o

njegovim osnovnim strukturama podataka. Takođe, pretpostavlja se znanje tzv. ANSI C-a. Efikasnost programiranja je povećana i zbog upotrebe elemenata strukturalnog programiranja i širokog skupa operatora.

Proširenja za standardni ANSI C jezik

- Arhitektura 8051 podržava nekoliko fizički odvojenih memorijskih prostora i delova za smeštanje programa. Postoje memorijski prostori koji omogućavaju:
- čitanje ali ne i upis
- upis i čitanje
- upis i čitanje ali brže u odnosu na ostale memorije
- Memoriji za podatke se brzo pristupa jer se koristi 8-bitna adresa. Za pristup internoj memoriji se koriste tri razlicita memorijska tipa:
- data oznacava da se pristupa internoj memoriji podataka uz direktno adresiranje, Što omogucava brz pristup (128B).
- idata označava da se pristupa celoj memoriji za podatke (256B) uz indirektno adresiranje.
- bdata označava da se pristupa lokacijama ukupne velicine od 16 bajtova koje se mogu adresirati po bitovima.
- xdata da se može pristupiti bilo kojoj lokaciji unutar memorijskog prostora (64k) i koristi se kod large memorijskog modela.
- pdata oznacava de se pristupa samo jednoj stranici velicine 256 bajta od spoljašnje memorije za podatke. Ovaj tip se koristi kod compact memorijskog modela.

C51 koristi tri memorijska modela:

- Small Model Kod ovog modela, sve promenljive, podrazumevano, se nalaze u unutrašnjoj memoriji za podatke 8051 sistema tj. koristi se data memorijski tip. Kod ovog modela, promenljivama se pristupa na efikasan nacin. Medutim, sviobjekti koji nisu eksplicitno postavljeni u nekom drugom memorijskom prostoru,moraju da se uklope u okviru unutrašnjeg RAM-a koji je sam po sebi vrlo mali.
- Compact Model Kod ovog modela, podrazumevamo, sve promenljive se nalaze u jednoj stranici spoljašnje memorije za podatke tj. koristi se memorijski tip pdata. Ovaj model može obezbediti maksimalno 256 bajtova promenljivih, jer se koristi indirektno adresiranje kroz registre R0 i R1.
 - Large Model Kod ovog modela, sve promjenljive se nalaze u spoljašnjoj memoriji za podatke (do 64k prostora) tj. koristi se xdata memorijski tip.
 Pristup memoriji kod ovog modela ja u odnosu na ostale modele najsporiji i neefikasan, narocito kod promjenljivih koje sadrže nekoliko bajtova.

-Familija 8051 mikrokontrolera sadrži poseban memorijski prostor za pristup specijalnim funkcijskim registrima (Special Function Registers - SFRs). Ovi registri se koriste u programu za kontrolu tajmera, brojača, serijskog ulaza i izlaza, ulazno-izlaznih portova I periferija. Nalaze se na adresama od 0x80 do 0xFF (128 bajta) i može im se pristupati bitski, bajtovski, ili na nivou reči. Svi nazivi ovih registara su predefinisani u okviru C51 kompajlera.

Tip podataka	Bitovi	Bajtovi	Opseg vrednosti
<u>bit</u>	1		0 to 1
signed char	8	1	-128 to +127
unsigned char	8	1	0 to 255
enum	8 / 16	1 or 2	-128 to +127 or -32768 to +32767
signed short	16	2	-32768 to +32767
unsigned short	16	2	0 to 65535
signed int	16	2	-32768 to +32767
unsigned int	16	2	0 to 65535
signed long	32	4	-2147483648 to +2147483647
unsigned long	32	4	0 to 4294967295
float	32	4	±1.175494E-38 to ±3.402823E+38
<u>sbit</u>	1		0 or 1
<u>sfr</u>	8	1	0 to 255
<u>sfr16</u>	16	2	0 to 65535

Struktura C programa za mikrokontrolere familije 8051

- Prvi red programa (sadrži direktivu #include za čitanje (ubacivanje) izvornog fajla sa deklaracijama za odgovarajuci mikrokontroler. U toku kompajliranja, fajl naveden u direktivi #include se ubacuje u osnovni fajl (u ovom slucaju u main.c). Pored #include značajna direktiva je i #define kojom se definiše makro ili konstanta. Na primer:
- #include<math.h> //ukljucenje matematickih funkcija
- #include<reg51.h> //ukljucenje adresa portova, registara i specijalnih
- bita u memoriji
- #define PI 3.14159 /* definisanje konstante PI, sada se u programu
- svuda piše PI a kompajler zna da je to broj 3.14159 */
- #define Timer0H 0xE4 //Konstanta u kodu Timer0H se zameni sa vrednošcu
- $= 0 \times E4$
- #define Timer0L 0xF0
- Treba zapaziti da je u programu main.c promenljiva i definisana kao unsigned char, što znači da je osmobitna. Kako port P1 ima osam nožica, vrednost koja mu se pridružuje mora biti osmobitna tj. u intervalu od 0 do 255. Pojedinim nožicama porta P1 pristupa se pisanjem P1^x gde x oznacava broj nožice (npr. P1^2=0), gde je x u opsegu od 0 do7.Ovo je bio jednostavan primer programa napisanog u C programu.

KEIL UVISION 3



-Project Workspace vam daje pristup:Datoteci i grupi u okviru projekta,CPU registrima ,Alatima,Tekstualnim predlozima za najčešće korišćene blokove teksta. -Funkcija u projektu za brzu navigaciju

-Output Window daje poruke o grešci i brz pristup

-Memory Window daje pristup memoriji

-The Watch & Call Stack prozor vam omogućava pregled i izmenu programa i prikazuje trenutnu funkciju poziva stabla.

-Radni prozor se koristi za uređivanje datoteka,

-Peripheral Dialog će vam pomoći da proverite status periferija.

→Kada koristite Keil µ Vision3, projekt razvoja ciklusa je otprilike isti kao što je za bilo koji drugi softver.

1-Izradi projekt,odaberi ciljni čip iz baze podataka i konfiguriši(podesi) postavke alata. 2-Napravi source(izvornu) datoteku u C.

3-Izradi dokumentaciju korišćenjem projekt menadžera.

4-Ispravi greške u izvornu datoteku.

5-Ispitaj povezane aplikacije.



Status Bar

-Status Bar prikazan na ekranu prikazuje poruke i informacije o trenutnim

μ Vision komandama i druge informacije kako je objašnjeno u nastavku:

Simulation	Sec.: 0.00000378	L:126 C:21	CAP	NUM	SCRL	OVR	R/W
Debug	Execution	Cursor	E	Edito	or and	Keyb	oard
Channel	Time	Position		Statu	us Info	ormat	ion

→Debug Channel prikazuje aktivan alat za ispravljanje i prikazuje ime ili simulacije kada je program testiran.

→Vreme izvršenja prikazuje vreme trajanja simulacije.

→Pozicija kursora pokazuje trenutnu poziciju u editor datoteci (L: broj linija, C: broj kolona).

→Keyboard Editor prikazuje informacije:

CAP:tipka je trenutno unutar tastature za velika slova. NUM: tipka je trenutno u NUM LOCK statusu. SCRL: tipka je trenutno u zaključanom stanju. OVR: označava pisanje preko editor načina koji zamenjuje znakove kao što ste upisali. R / W čitanje / pristup pisanju;R / O read-only pristup-sprečava modifikacije iz teksta.

Logički analizator

-μ Vision3 Logic Analyzer daje grafički prikaz promene vrednosti varijable(promenljive) ili VTREGs koje ste naveli.



Memorijski prozor

Memory prozor prikazuje sadržaj u raznim područjima memorije. Do četiri različita područja može se pregledati u različitim stranicama. U kontekstu

meni vam omogućava izbor dozvoljenog izlaznog formata.

M	emory											×
	Address: main											
	00024178:	E276	0400	E376	04	00 E366	F7FF	B6E6	0080	B7E6	0000	
	0002418C:	5AE6	0040	D8E6	80	11 2AE6	F63C	28E6	F63C	CEE6	0044	
	000241A0:	88BF	A8E6	0040	Df	DD DODC	4072	0003	2200	F8E6	4418	
	000241B4:	OOCA	3390	F9E0	F .	Decimal				heon	1080	
	000241C8:	86CO	86F4	B01C	8	Unsigned			•	Cha	ne	
	000241DC:	4DFO	84CO	84F4	В	Signed			•	🖌 Int		
	000241F0:	84F4	B01C	F847	0					Lor	na	
	00024204:	OOBD	20EA	430E	8	Ascii				poro	7420	
	00024218:	8229	6B2D	8129	3	Float				β500	OOCA	
	0002422C:	40E2	E4FO	FE46	F:	Double				þoca.	4634	Ţ
		emory #1	Memory	/#2 <u>\</u> Me	moi	Modify Me	mory at 0	00241A6		POFF	1000	

Serijski Window

μ Vision3 nudi dva serijska prozora za serijski ulaz i izlaz. Serijski izlaz

podataka iz simuliranog CPU je prikazan u ovom prozoru.

🚄 Serial #1	
<pre>+**************** REMOTE This program is a simple the AT91M55800A and record and the voltage on the + command -+ syntax Read R [n] Display D Time T hh:mm:ss Interval I mm:ss.ttt Clear C Quit Q Start S</pre>	MEASUREMENT RECORDER ***********************************
Command:	✓ Ascii Mode <u>H</u> ex Mode <u>C</u> lear Window

Stvaranje jednog projekta u uVision 3

Pokrenite Keil uVison3 program.

🕎 µVision 3					- 7 🛛
Eile Edit View Project Debug	Flash Peripherals Tools SVCS Wind	ow <u>H</u> elp			
1 🖆 😅 🖬 🎒 👗 🖻 🛍	<u>2</u> 2 律律 / % % %	A	- M (M ← → (M ⊕	@ 🖪 🔊 🕘 🛞 🖉 🖉	ŋ
*****	·	🛔 🗟 🚍			
Project Workspace 🚽 🗙					
×					
ð.					
Mob					
out Wir					-1
Build (Comm	and λ Find in Files /		4		\
					OVR R/W
🧃 start 🛛 🖾 🍯 🔣	0 🥦 🕽 😇 🕘 🖉 🏷 😭	displej - µVision3 - [D	W µVision3		SR 🚆 🍼 🕒 🎲 21:38

Kreirajte novi dizajn projekta. Odaberite novi projekt iz padajućeg menija

🕎 µVision 3			- 7 🛛
<u>File E</u> dit <u>V</u> iew	Project Debug Flash Peripherals Tools SVCS Window	Help	
	New µ⊻ision Project		
	New Project <u>W</u> orkspace		
S (*) (*) (*)	Import µVision1 Project	P.	
Project Workspace	Open Project		
	<u>Close Project</u>		
	Manage	•	
	Select Device for Target		
	Remove Item		
	Options Al	+F7.	
	Clean target		
	Build target		
	Rebuild all target files		
	😸 Batch Build		
	😸 Translate Ctr	# F7 :	
	👗 Stop build		
	1 I:\seminar\program,\Primer.Uv2 2 C:\Documents and Settings\ZORAN\Desktop\displej.L 3 C:\Keil\C51\Examples\ADI 83x\Blinky\displej.Uv2 4 C:\Keil\C51\Examples\ADI 83x\Blinky\led displej.Uv2 5 C:\Keil\C51\Examples\ADI 83x\Blinky.Uv2 6 C:\Keil\C51\Examples\ADI 83x\Blinky.Uv2 2 C:\Keil\C51\Examples\ADI 83x\Blinky.Uv2 2 C:\Keil\C51\Examples\ADI 83x\Blinky.Uv2	v2	
	8 C:\Keil\C51\RtxTiny2\Examples\Traffic\Traffic.UV2		
×	9 C: (Keil/C51/Examples/Blinky/Blinky/UV2		
÷.	To c. (kei/cor/cxanples/biniky/biniky.ovz		
utput Window			. –
Create a µVision ne	w project	W displat - Wision? TO W wision?	Documenti Microsof
Start		Manapel - Pwanto - To	Bocomaner - Microsoft SK - V 21:30

Otvoriće vam se sledeći prozor koji vas pita da sačuvate(save) kod vašeg

projekta npr.na desktopu. Napišete ime projekta npr.displej.



Nakon što ste kliknuli na save(sačuvaj),otvara se prozor koji će vas pitati koji čip želite koristiti. U ovom slučaju odaberite Atmel AT89S8253 mikrokontroler kao cilj

Select Device for Target 'Target 1'		×
CPU Vendor: Atmel Device: AT8958253 Toolset: C51 Data base	 Use Extended Linker (LX51) instead of BL51 Use Extended Assembler (AX51) instead of A51 Description: 	
AT89LV52 AT89LV55 AT89S2051 AT89S4051 AT89S4D12 AT89S51 AT89S52 AT89S53 AT89S8252 AT89S8252 AT89S8253 AT89S8253 AT89S8253 AT89S8253 AT89S8253 AT89S8253 AT89T1C140 AT91F40416	8051 based Full Static CMOS controller with Three-Level Program Memory Lock, 32 I/O lines, 3 Timers/Counters, 9 Interrupts Sources, SPI, Watchdog Timer, 2 DPTRs, 12K Flash Memory, 2k EEPROM, 256 Bytes On-chip RAM	
	OK Cancel	Help

Pojaviće se novi prozor koji će vas pitati da li želite kopirati standard 8051 kod pokretanja u folder projekta. Osim ako ste stari korisnik, trebate kliknuti No,u suprotnom kliknite Yes.



Možete sada izraditi projekt, ali ne postoji izvorni k^d.



Koristite 'Obnovi sve ciljne datoteke'-Rebuild all target iz

padajućeg menija Project.Sada imate izvorni k^d koji je dodat grupi.



Zatim kliknite na označenu ikonicu koja vam služi za kreiranje novog fajla.Otvoriće vam se nova radna površina "Text 1".



U radnoj površini Text 1 napišite k^d vašeg projekta.



Zatim sačuvajte k^d opcijom Save iz padajućeg menija File.



Naravno sačuvajte k^d pod ekstenzijom src. Ekstenzija sačuvanog fajla zavisi od toga u kom je programu pisan. Ekstenzija src. pokazuje da je kod pisan u programskom jeziku C.

Save As	· · · · · · · · · · · · · · · · · · ·	<
Save in:	🞯 Desktop 💽 🖛 🛍 💣 🎫 -	
My Recent Documents	My Documents Image: Startup.AS1 My Computer Image: Startup.AS1 My Network Places Image: Startup.AS1 Keil_program Image: Startup.AS1 Mikrotalasna Elektronika Image: Startup.AS1	a J
My Documents	Prezentacije STARTUP.LST program, STARTUP.OBJ uhf_pojacalo2_files uhf_pojacalo2. uhf_pojacalo_files uhf_pojacalo.h idisplej UNIVERZITET L idisplej.lnp Upravljanje sec idisplej.M51 Upravljanje sec	+ t J
My Computer My Network Places	isplej.plg Type: M51 File Date Modified: 4.7.2008 19:25 Size: 3,09 KB File name: displej.srd Save as type: Cancel	

Sledeće šta treba uraditi jeste da se taj fajl doda u prozoru Project Workspace 1 to desnim klikom na Source Group 1,pa na opciju Add Files Group.



Pronađite gde vam je smešten taj fajl i kliknite na Add. Zatim idite na opciju 'Obnovi sve ciljne datoteke'-Rebuild all target

iz padajućeg menija Project.



Pojaviće vam se sledeći prozor.Ako postoje greške u programu,kao što je ovde slučaj pisaće vam u Output Window

kakva je greška i u kom se redu nalazi.



Po ispravljanju grešaka ponovo idite na opciju 'Obnovi sve ciljne datoteke'-Rebuild all target iz padajućeg menija Project, i dobićete naredni prozor. Potrebno je da piše 0 Errors (s).

📅 displej - µVision3						
Eile Edit View Project Debug F	lash Peripheral	s <u>T</u> ools <u>S</u> VCS	<u>W</u> indow <u>H</u> elp			
12 😅 🖬 🎒 🕺 🖻 🛍		F 16 % %	n 16 🙀	M → M	a 🔍 🔼 🗩 🗶 🖻	5 CD
🕸 🕮 🛎 🛎 🖾 🕅 🔊	Target 1					
Project Workspace 🔹 👻	Ban					
🖃 🔁 Target 1	E C:Wocur	nents and Se	ttings\ZORAN\Desk	top\displej.src		
🖻 📇 Source Group 1	058	MOV	P1,#Ofeh	; oe 1. leca n	a 0	
STARTUP.A51	059					
i ispiej.src	060	MOV	P1 #Offb			
	062	MOV	PO.#0	· casenie svih ·	displeia	
	063	MOV	P1,#Ofdh	; oe 2. leca na 0		
	064	MOV	P1,#Offh	;		
	065	MOV	PO,#10h	; paljenje displeja	D5	
	066	MOA	P1,#Ofdh	; oe 2. leca na 0		
	067	MOA	P1,#Offh	;		
	068	MOV	A,#01	; upisivanje 1	na displej D5	
	059	LCALL	DO DO	; postavljanje i	naske	
	070	MOV	P1 #Ofeh	i ne i leca n	= 0	
	072	nov	11, worth	, be 1. feca #	1.0	
	073	MOV	P1,#Offh	;		-1
			PO //o	1990 C. 1990	- M	
	displej.src					
× *** WARNING L2: REFER	RENCE MADE	TO UNRESOL	VED EXTERNAL			
SYMBOL: 7C START	C					
MODULE: STARTUP.	obj (?C_SI	ARTUP)				
ADDRESS: 080AH						
🛓 Program Size: data=73	0.0 xdata=0	code=549				
g "displej" - O Error(s	s), 3 Warni	.ng(s).				•
3 K M M M M M M M M M M M M M M M M M M	id À Find in File	is /				•
	[Simulation	L:67 C:20	R/W
🐉 start 🛛 🖪 🧿 🔣 🖉	9 💈 👹	🖲 🖉 🖻 🖉	Doc1.doc	: - Microsoft 🕎 displej - µVision3	👹 untitled - Paint	SR 🖤 🔇 💽 🕵 19:30
					Undered Frank	

Pre simulacije kliknite na Target 1 i desnim klikom miša odaberite opcije za Target 1,da bi se osigurala brzina za ciljni sklop. Brzina zavisi od tipa mikrokontrolera. Za ovaj slučaj izaberite 24.0 MHz ciljno polje i postavite kristal (XTAL) na frekvenciji 24.0 MHz.

Options for Target	t 'Target 1'	X
Device Target Ou	utput Listing User C51 A51	BL51 Locate BL51 Misc Debug Utilities
Atmel AT89S8253		
	Xtal (MHz): 24.0	Use On-chip ROM (0x0-0x2FFF)
Memory Model:	Small: variables in DATA 📃 💌	
Code Rom Size:	Large: 64K program 📃	
Operating system:	None	
		Use multiple DPTR registers
Off-chip Code mer	mory	Off-chip Xdata memory
	Start: Size:	Start: Size:
	Eprom	Ham
	Eprom	Ram
	Eprom	Ram
Code Banking	Start: End:	L'far' memoru tupe support
Panker D	Death Access (0x0000) (0x0000)	Save address extension SEB in interrupts
	Bank Area: Jondood Jondood	
		maal Dafaulta Uula
		Incei Derauits Help

Da biste pokrenuli simulaciju odaberite Start / Stop Debug Sesion opciju iz padajućeg menija Debug.



Prozor projekta uVision sada sadrži pod-prozore. Radni prozor pokazuje trenutni sadržaj najvažnijih registara. Takođe se prikazuje vreme

izvršavanja koda u sekundama. Editor prozor pokazuje C k^d,a

disassembly prozor pokazuje kodove koji grade ceo projekt.

🛂 displej - µVision	n3 - [Disas:	assembly]	_ 7 🗙
<u> </u>	roject <u>D</u> ebu	bug Fl <u>a</u> sh Pe <u>r</u> ipherals <u>T</u> ools <u>S</u> VCS <u>Wi</u> ndow <u>H</u> elp	_ 8 ×
i 🖀 🚅 🖬 🎒 🐰	: B B	↓ ♀ ♀ 律 ≉ 彡 % % % % M → M # ← → M ● ④ ⊡ ● ④ 10 10 10 10 10 10 10 10 10 10 10 10 10	
👫 🗐 🐼 🖓 🛈	₽ {} *(}	▶ ♦ 註 磁 🖻 耍 🦁 🐸 🗏 🖻 🗄 🚾 № 🥕	
Project Workspace	* X	126: 7C STARTUP: LJMP STARTUP1	~
Register	Value	127: -	3
E Regs		128: RSEG ?C_C51STARTUP	
r0	0x00	129:	
r1	0x00	130: STARTUP1:	
r2	0x00	131:	
13	0x00	132: IF IDATALEN <> 0	
r4	0x00		
r5	0x00		
гь -7	0x00		
	UXUU	C:0x0006 00 NOP	
- Jys	0.00	C:0x0007 00 NOP	
h	0,00	C:0x0008 00 NOP	
sn	0x07	C:0x0009 00 NOP	
sp max	0x07	C:0x000A 00 NOP	
PC \$	C:0x00	C:0x000B 00 NOP	
dps	0x00	C:0x000C 00 NOP	
dptr	0x0000	C:0x000D 00 NOP	
states	0	C:0x000E 00 NOP	
sec	0.0000	C:0x000F 00 NOP	
psw	0x00	C:0x0010 00 NOP	
		C:0x0011 00 NOP	
📄 🖹 📕 🕼 📢	} 🦁 .		~
Symbols	* X		
M <u>a</u> sk: *	Case	st 🖹 STARTUP 🧟 Disassembly	
× Running with	Code Si	ize Limit: 2K	
Load "C:\\Doc	cuments	and Settings/\ZORAN/\Desktop/\displej"	
<u>ð</u>			
Nu >			
🖞 ASM ASSIGN Br	reakDisa	sable BreakEnable BreakKill BreakList 🔽 😽	-
	d) Comma	mand / Find in Files / I Memory #1 / Memory #1 / Memory #2 / Memory #3 / Memory #4 /	
For Help, press F1		Simulation [11: 0.0000000 sec	OVR R/V
🛃 start 🛛 🗷	i 🗿 🔣 🤅	🔕 🥵 🕽 😁 🥥 🤣 💁 🧟 🕎 disple) - µVision3 - [D 🛛 🛛 2 Microsoft Office 🔹 🍞 untitled - Paint 🛛 SR 📍 🔇	21:48

Za ispravljanje programa idite na korak Debug→ Step Over. Step Over The command će ispravljati jedan red ili celi potprogram, zavisno od toga gde je pokazivač postavljen.. Pritisnite F11 i pojaviće se mali žuti pokazivač koji vam omogućava da prtite korak po korak-svaku liniju

programa.

🔀 displej - µVis	ion3 - [Disas	sembly]	_ 7 🗙
<u>E</u> ile <u>E</u> dit ⊻iew	Project Debu	ug Fl <u>a</u> sh Peripherals <u>T</u> ools <u>S</u> VCS <u>Wi</u> ndow <u>H</u> elp	_ & ×
1 🖆 🚘 🖬 🎒	X 🖻 🛍	Ω Ω 卓 ∉ ∧ % % % % [🕢 🖌 🖌 🖌 🖌 🖌 🖌 🖌 🖌 🖌 🖌 🖉 💽 💽 🖉 🛞 🕅 🕅	
👫 🗉 🛛 🖓	∂ + {}+ * {}	◇ ※ ☆ ◎ ◎ ◎ ● ■ 〒 ■ ■ ● ● ● ●	
Project Workspace	▲ X	C:0x07F1 00 NOP	~
Register	Value	C:0x07F2 00 NOP	
E Reas		C:0x07F3 00 NOP	
rO	0x7d	C:0x07F4 00 NOP	
r1	0x00	C:0x07F5 00 NOP	
r2	0x00	C:0x07F6 00 NOP	
r3	0x00	C:0x07F7 00 NOP	
r4	0x00	C:0x07F8 00 NOP	
r5	0x00	C:0x07F9 00 NOP	
r6	0x00	C: UXU7FA UU NOP	
r7	0x00		
⊡ — Sys			
а	0x00		
ь	0x00		
sp	0x07		
sp_max	0x07	C.0V. ROW DO HOVER IN T	
PC \$	C:0x08		
dps	UXUU		
i dptr	00000	135: TDATALOOP: MOV BRD.A	
states	0.0000	C:0x0803 F6 MOV RRO.A	
t new	0.0000	136: DJNZ RO, IDATALOOP	
D psw	0,000	C:0x0804 D8FD DJNZ R0, IDATALOOP (C:0803)	
		185: MOV SP,#2STACK-1	
	* 3 🔍	186:	
	0 6	187: ; This code is required if you use L51 BANK.A51 with Banking Mode 4	~
Symbols	▲ X		>
M <u>a</u> sk: *		STARTUP 🤮 Disassembly	
× Running wit	h Code Si	ze Limit: 2K	
1 Load "C:\\I	ocuments	and Settings\\ZORAN\\Desktop\\displej"	
<u>ð</u>			
Nu >			
S ASM ASSIGN	BreakDiss	able BreakEnable BreakKill BreakList 🔽 🕏	-
	Build) Comm	and / Find in Files / Memory #1 / Memory #2 / Memory #3 / Memory #4 /	
For Help, press E1			
r or neip, press P1	70 - 28 - 1		
Start	🗠 🥥 🔊 /	🗢 🤧 😴 🤍 🥮 🖉 🚰 🔤 🙀 displej - µvision3 - [D 🔛 2 Microsoft Orrice 🔹 🦉 Uncitied - Paint 👘 SR 🚆 🔇 j	21:50

Odaberite opciju "project",zatim u tom padajućem meniju izaberite "Opcije za Target 'Target 1'. Odaberite zlaz(output) na kartici i zaokružite 'Kreiraj Hex File' okvir. Zatim idite na opciju 'Obnovi sve ciljne datoteke'-

Rebuild all target iz padajućeg menija Project.

Options for Target 'Target 1'
Device Target Output Listing User C51 A51 BL51 Locate BL51 Misc Debug Utilities
Select Folder for Objects Name of Executable: displej
Create Executable: .\displej
Debug Information Browse Information
Create HEX File HEX Format: HEX-80
C Create Library: .\displej.LIB 🔽 Create Batch File
OK Cancel Defaults Help

Sada idete na opciju "Open file" iz padajućeg menija "File" da bi ste videli generisani HEX fajl,naravno on je snimljen tamo gde vam je izvorni kod pa se može pokrenuti i otuda. Za tip fajla(Files of type) izaberite svi fajlovi(All files).



Sledeće šta treba uraditi jeste da čekirate odnosno označite fajl sa ekstenzijom .hex. To vam je generisani HEX fajl,odnosno heksadecimalna vrednost koju treba prebaciti u mikrokontroler.



Ovde se vidi dobijeni HEX fajl.

🕎 displej - µVision3 - [C:\Do	ocuments and Settings\ZORAN\De	sktop\displej.hex]			- 7 🛛
Eile Edit View Project Debu	ug Fl <u>a</u> sh Pe <u>r</u> ipherals <u>T</u> ools <u>S</u> VCS <u>W</u> ir	ndow <u>H</u> elp			_ & ×
1 🖆 🚅 🖬 🎒 👗 🛍 🛍	2 2 律律 & % % %	·	▲ (* + → * @	0 🔊 🗠 🔊	(h)
🗇 🏥 🛎 🛎 🔛 🔊	Target 1 🔹	A 🗟 🚍			
Project Workspace - ×	:0300000020800F3				
Eriget 1	:02000002100DD	D75817E020000D0			
STARTUP.A51	:1001000075813F7590F	F7580007590FD7590FF75	46 5D		
	:1001200090FE7590FF7	580007590FD7590FF7580	4D		
	:10013000407590FD759 :10014000FE7590FF758	OFF74081201FDF5807590 0007590FD7590FF758020	'73 '9D		
	:100150007590FD7590F	F74091201FDF5807590FE	94		
	:100160007590FF75800 :1001700090FD7590FF7	07590FD7590FF75801075 4011201FDF5807590FE75	16 7C		
	:1001800090FF7580007	590FD7590FF7580087590	E3		
	:1001A000FF758000759	OFD7590FF7580047590FD	5A		
	:1001B0007590FF74001 :1001C0007580007590F	201FDF5807590FE7590FF D7590FF7580027590FD75	3B 66		
	:1001D00090FF740D120	1FDF5807590FE7590FF75	OE		
	: 1001F000FF74021201F	DF5807590FE2103900202	4A		
	:100200093223F065B4F666D7D077F6FBF86DBCF16 :06021000E6EDFD87FFEFA3				
	:0000001FF				
					-
	🖹 displej.src 📋 displej.hex				
× SYMBOL: ?C_STAR	T				
MODULE: STARTUP ADDRESS: 080AH	P.obj (?C_STARTUP)				
<pre> Program Size: data=7 P P </pre>	3.0 xdata=0 code=549				
"displej" - O Error(s), 3 Warning(s).				_
Build Comma	and λ Find in Files /		•		· •
For Help, press F1		Sim	ulation	L:1 C:1	R/W
🦺 start 🔰 🔣 🥥 🛃 /	😂 🧕 🌷 😇 🕘 🌽 🤷	Doc1.doc - Microsoft	🕎 displej - µVision3 - [C	🦉 untitled - Paint	SR 🖞 🔇 🖸 😻 19:32