C Programming in Atmel Studio 6.2 Step by Step Tutorial



Atmel

Powered by Visual Studio

Sepehr Naimi BIHE University

Contents

Introduction	2
Downloading and Installing Atmel Studio	3
Opening Atmel Studio	3
Creating the first project	4
Writing the first C program	6
Compiling	7
Debugging	8
Using Breakpoints	.11
Disassembly	.12

Introduction

This tutorial will teach you how to write, compile, and trace a simple program in Atmel Studio.

Downloading and Installing Atmel Studio

Download the newest version of Atmel Studio from the Atmel website:

http://www.atmel.com/tools/atmelstudio.aspx

Atmel Studio × +	
♦ ♦ @ www.atmel.com/tools/atm ♥ ♥ ♥ ♥ ♥ ♥	≡
Software Description Image: Software Atmel Studio 6.2 sp2 (build 1563) Installer (560 MB, updated February 2015) This installer contains Atmel Studio 6.2 service pack 2 with Atmel Software Framework 3.21 and Atmel Toolchain. Install this if you have previously installed Atmel Studio or have access to	^
Internet when installing. Atmel Studio 6.2sp2 build 1563 resolves an installation issue that was present in build 1548. MD5: 0310b42235f2545d4adba0c423c04ab1 SHA: 672223aa65af7714369e3dc4f677eb07635351fc	

Run the downloaded program to install the Atmel Studio IDE.

Opening Atmel Studio

Go to the *Start* menu and open Atmel Studio.



Creating the first project

1. Go to the *File* menu. Choose *New* and then *Project*.



- 2. In the opened dialog,
 - a. Choose C/C++.
 - b. Select GCC C Executable Project.
 - c. Name the project as toggleProject.
 - d. Choose the path where you like to save the project by clicking on the *Browse* button.
 - e. Press OK.



3. In the *Device Selection* dialog

- a. Select *megaAVR* as the *Device family*.
- b. Choose ATmega32 (or any other Chips you want to use)
- c. Select **OK**.

Device Family:	megaAVR, 8-bit	1		Search for device \$
Name	All megaAVR. 8-bit	bytes) Data Memory (bytes)	EEPROM (b	Device Info:
ATmega169PA ATmega16A ATmega16HVA ATmega16HVB ATmega16HVBrev ATmega16M1 ATmega16U2 ATmega16U4 ATmega2560	tinyAVR, 8-bit AVR XMEGA, 8-bit AVR UC3, 32-bit SAM3, 32-bit SAM4, 32-bit SAM D, 32-bit SAM G, 32-bit SAM R, 32-bit 16	1024 1024 512 1024 1024 1024 512 1280 8192	512 * 512 256 512 512 512 512 512 512 512 512 512 512	Device Name: ATmega32 Speed: 0 Vcc: 2.7/5.5 Family: megaAVR Datasheets Supported Tools AtmeLICE
ATmega2561 ATmega2564RFR2 ATmega256RFR2 ATmega324A ATmega324A ATmega324P ATmega324PA ATmega325	256 256 256 256 32 32 32 32 32 32 32 32	8192 32768 32768 2048 2048 2048 2048 2048 2048 2048	4096 8192 8192 1024 1024 1024 1024 1024	AVR Dragon AVRISP mkli AVR ONE! JTAGICE3 JTAGICE mkli Simulator
ATmega3250	32	2048	1024	STK500 STK600

The compiler automatically makes the *toggleProject* and adds a C file to it.

🗣 toggleProject - AtmelStudio		
Elle Edit View VAssisty ASF Project Build Debug Iools Window Help		
1 3 • 1 1 • 2 2 2 4 2 4 • • • - 2 • 2 • 3 1 1 • • W Debug • 2	remainedTimeVal +	
●□\$	3003 0. 1086	
toggieProject.c ×	 Solution Explorer 	- 4 ×
🗢 toggleProject.c 🔹 🖶 🌳 El\AVRprojects\toggleProject\toggleProject\toggleProject\toggleProject.c	• Co Go	
* Created: 4/2/2015 11:04:16 AM * Author: Zomorrod */	 toggleProjec Depende Output Fi AS VA 	ct ncies les -
<pre>#include <avr io.h=""> Description Desc</avr></pre>	Properties	- + × -
Output		- # ×
Show output from: XDK Packaging		
* Error List 🔟 Output 🗮 Find Results 1		
Matches] Ln 6	Col5 Ch5	INS a

Writing the first C program

Type the following program.

```
#include <avr/io.h>
#define F_CPU 800000UL
#include "util/delay.h"
int main(void)
{
    DDRA = 0xFF; //make port A as output port
    while(1)
    {
        PORTA = 0xFF; //make all pins HIGH
        _delay_ms(1000); //wait 1 sec
        PORTA = 0x00; //make all pins LOW
        _delay_ms(1000); //wait 1 sec
    }
}
```

Note: Auto Complete

The editor knows the available variables and functions. For example, if you type the first few letters of #include, e.g. #in, the following list appears and you can choose **#include** from the list, by pressing the **Enter** button instead of typing **#include**.



While editing the program, if you press *Ctrl+Space*, the auto complete list appears, and you can choose between the available choices.

Compiling

Press **F7** to compile, or choose **Build Solution** from the **Build** menu. The results of compile are shown in the **Output** window.

条 toggleProject - AtmelStudio				• ×
File Edit View VAssistX ASF Project Build Debug	Tools Window	Help		
🗄 🕶 🖷 🖼 🖬 🖉 🔜 🖓 🗛 🛍 👘 🖉 - 🖂	- 🖳 🔚 🍳 🛛		Debug 🔹 🔯 📜 💷 💷	: ' <u>=</u> <mark>:</mark>
i 🗑 📴 🞇 🍋 🔓 🍯 💁 🔬 🛍 📮 i 👀 💷 🌩 🗤) 60 SI (I S		Hex 🛛 🖬 🕶 🚽 👯 🐼 💷	÷:÷:÷
toggleProject.c ×		<u> </u>	Solution Explorer	- ₽ ×
🕈 toggleProject.c 🔹 🗧 🌩 E:\AVRprojects\tog	gleProject\toggle	- 🛟 Go		
#define F_CPU 8000000L #include "util/delay.h"		\$	→ togaleProject	1 - E
			🔍 AS 🔎 VA 🟫 VA	💐 Sol
□int main(void) 100 %		* }	Properties	- ₽×
Output	000000000000000000000000000000000000000			₹₽×
Show output from: Build	-	a) 🚑	1) 🙀 I	
Done building project "toggleProject.cproj".				-
Build succeeded.				
========= Build: 1 succeeded or up-to-date, @) failed, 0 ski	pped ==		-
۲. III.				-
🚼 Error List 📃 Output 🛒 Find Results 1				
Build succeeded	Ln 23	Col 1	Ch 1	INS!

Debugging

- 1. To start debugging, press *Alt+F5* or choose *Start Debugging* from the *Debug* menu.
- 2. The following Dialog appears and asks you to select the debugging tool. Press *Continue*.



3. In the following window, choose *Simulator* as the debugger and then close it by pressing the x next to the *toggleProject*.

🕭 toggleProject - AtmelStudio	
File Edit View VAssistX ASF Project Build Debug Tools Window Help	
🗄 🔂 🕶 🖼 🖉 🛃 🍠 🕺 🖓 🖓 🖓 🦿 🖓 🖓 🖓 🖓 🖓 🖓 🖓 🖓 🖓 👘 🖓 👘 🖓 👘 🖓 👘 🖓	emainedTimeVal 🔹 🖓 🙄
: 🗒 🖾 🖓 🍋 🔓 😘 🔬 🗳 - : Mi 🖬 수 비 🕨 🐼 9월 (과 한 대 한 Hex) 🗃 - : : 💭	🖂 🗊 📮 📓 🖕 l 🖄 🚟 🕹 🖕 l 🙄
toggleProject × toggleProject.c	Solution Explorer 🔹 🕂 🗙
2	
Configuration: N/A *	Solution toggleProject (I project)
Build Events	Dependencies
Toolchain	Dutput Files
Device A	toggleProject.c
Selected debugger/programmer	
Advanced Simulator	AS VA E VA Sol
AVRISP mkII	Properties
	• 4 ×
Done building project "toggleProject.cproj".	
Build succeeded	
====== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ========	C
*m	
📸 Error List 🔳 Output 🖳 Find Results 1	
Ready	

Note: Simulator vs. debugger

Using the simulator you can execute the instructions, and watch the registers and variables. If you have a debugger, e.g. AVRISP mkII or Atmel-ICE, you can connect a trainer board to your computer. In the case, the microcontroller of the board executes the same instructions, when you trace the program. This facilitates you to check the hardware while monitoring the variables in the IDE.

- 4. Press *Alt+F5* again. Two *Memory* Watch windows appear which show the contents of the Flash memory. Close the windows.
- 5. Now a yellow cursor is on the first line of the main program and the IDE is ready to debug the program.

秦 toggleProject (Debugging) - AtmelStudio					• • ×
File Edit View VAssistX ASF Project B	uild Debug Tools Wind	low Help			
👸 - 🕮 🗃 - 😂 🛃 🎒 🐰 🛍 🛍 🖻	• (* • 🚚 • 🖳 🔛 🍳	🔥 🕪 🕅 Debug	- 🧭 remainedT	imeVal 🔹	🛛 🖓 🖓
i 🔁 📴 🖓 🍋 🔓 🍯 💁 🍒 🗳 🚽 💷	■ � ♦ ▲ 63 5⊒ 🕻	🗏 🔄 📲 🚹 He	x 🗳 📲 📮 i 🐼 🚥 💷	🐺 🖳 🚽 🛗 🛗	
toggleProject.c ×	Step into	R	Processor		- ₽ ×
🕈 main 🔹 🗧 🌩 int n	ain(void) (F11)	Step Out cu	rsor o Name	Value	
	Step ove	r	Program Counte	r 0x0000036	<u> </u>
	(F10)		Stack Pointer	0x085D	
<pre>#include <avr io.h=""> #include <avr io.h=""></avr></avr></pre>			X Register	0x0000	=
#include "util/delay.h"			Y Register	0x085F	
wincide deli, delby m			Z Register	0x0000	
⊡int main(void)			Status Register	UTHSVNZC	
{			- Cycle Counter	13	
DDRA = 0XFF; //make port A as outp	it port		Frequency	1.000 MHz	
while(1)			Stop watch	13.00 µs	
{			🖃 Registers		
PORTA = 0xFF; //make all pin	HIGH		R00	0x00	
PORTA = 0x00: //wait i set	LOW		R01	0x00	
_delay_ms(1000); //wait 1 sec			- R02	0x00	-
83 % 🔹 🔹			🕨 🔍 ASF E 🔤 I	Proce 🟹 Soluti 督	Prope
Watch 2	▼ ╄ >	< Breakpoints			- ₽ ×
Name Value	Туре	New - × }	🍋 🐹 🗞 🥹 🖅 🖏	Columns -	
		Name Labels	Condition Hit Count		
	-	·			
🖼 Autos 👼 Locals 👼 Watch 1 🛃 Watch	2	🗔 Breakpoi	🚰 Call Stack 🛛 🔲 Comm	an 🖅 Immediat 📃	Output
Stopped		Ln 16	Col 5	Ch 2	INS 🔡

6. To execute the instructions line by line press *F10* or click on the *Step over* icon.

Step Into vs. Step Over

Both *F10 (Step over)* and *F11 (Step into)* execute one instruction and go to the next instruction. But they work differently when the cursor is on a function call. If the cursor is on the function call, *Step into* goes into the first instruction of the function, but *Step Over* executes the whole function and goes to the next instruction.

Step Out

If the execution is in a function, you can execute the function to the end by pressing the *Step Out*.

Run to Cursor

You can put the cursor on an instruction and then press the Run to Cursor button. In the case, the program runs until it reaches the instruction which the cursor is on it.

To monitor the peripherals, including the I/O ports, click on the *Debug* menu, choose *Windows* and then *I/O view*.

3 · 11 · 1	3 3 3 A 2 3 10 -1		Windows 2		G	Breekpoints	Alt+F9	1
toggleProject.c)	880-241-14-24 0	日間中	Start Debugging and Break Stop Debugging Start Without Debugging	Alt+F5 Ctrl+3bift+F5 Ctrl+Alt+F5	19 19	Data Breakpoints Processor View &O View	8	•
main.while	• 22		Disable debug/MRE and Close		9.	Live Watch		
winclude kan wdefine P_CO winclude "af	vr/io.hx Nu debecedu. til/delay.h"	マウヤ	Continue Execute Stimulifile Set Stimulifile	15	() (a u	Output Parallel Tasks Parallel Stacks	Ctrl+Shift+D, K Ctrl+Shift+D, S	
<pre>int main(ve) { DOWA = (while(i) { POR() del </pre>	Ld) BNFF: //make port 4 as output pc) fa = 0xFF: //make all pins HEI lay_st(1000); //make all pins LON Lay_st(1000); //make all pins LON Lay_st(1000); //make 11 pins LON Lay_st(1000); //make 10 pins LON	1 = 3 5 5 5 F	Restart Break All QuickWatch Step Dver Step Out Hun To Camor Reset	Chri+75 Shift+79 F11 F10 Shift+F11 Chri+730 Shift+F5	5 0 4 5 1 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5	Wetch Autos Locais Immediate Cell Stock Threads Modules Processes	Cbi+Ah+V, A Ah+4 Chi+Ah+1 Ah+7 Cbi+Ah+H Chi+Ah+U Chi+Ah+U	
Vatch 1			Percepio Trace	,		Memory	,	
Name	Value		Toggle Breakpoint New Breakpoint		1	Disassembly Registers	Alt=8 Alt=5	
🖼 Autos 📑 Lo	celt 👩 Watch 1 📑 Wetch 2	2	Oner All DetaTigs Toport DetaTigs Import DetaTigs	, Curro Spotto La	ack	Command	🗩 immediate 🖪	Out

8. The *I/O view* tab appears on the right hand side which shows the peripherals of the microcontroller, including the I/O ports. Select *PORTA*. The values of the related registers (PINA, DDRA, and PORTA) will be shown below.

toggleProject (De File Edit View	ebugging) - AtmelStudio VAmistX ASF Project Build Debu	g Tools Window He	φ.		
	9 00 00 00 00 00 00 00 00 00 00 00 00 00	**************************************	U Debug 🚽 🥭 rema	inedTimeVal • 🔍	A Strength
toggleProject.c ×				10 View	• ‡ ×
💠 main.while	• 🕂 🌩 while(1)		• (*Go	EREC .	-10"
<pre>einclude covu define 5_CM =include "ut! Eint suin(vold {</pre>	<pre>//o.hs idedeeeuL il/delay.h* 0) eFf J/meke port A as output port k = 00f#j //wake all plot HIGH by_cs(1000); //wait 1 Sec A = R000; //wait 1 plot Com by_cs(1000); //wait 1 sec</pre>		÷	Name AD_CONVERTER ADALOG_COMPARAT BOOT_LOAD CPU EEPROM CPU Neme Address Valu Name Address Valu DPNA 0-39 000 DPNA 0-38 000 DPNA 0-38 000 DPNA 0-38 000 DPNA 0-38 000 DPNA 0-38 000 DPNA 0-38 000 DPNA 0-38 000 DPNA 0-38 000 DPNA 0-38 000 DPNA 0-38 000 DPNA 0-38 00 DPNA 0-3	Value OR Bits OCCCC OCCCC
83% • 1		- 4 -	P	■ 10 • A5 ■ Pro	Sol Pro
Name	Volue	Туре	New- X 9 15 10 16	Columns • Hit Count	
Autos 🕼 Loc	as 22 Watch 1 97 Watch 2		📑 Breakpoints 🍶 Call Stact	k 🗖 Command. 🛛 👼 İmme	diate. 🖬 Output
Ready	A A A A A A A A A A A A A A A A A A A				and a second

9. Press F10 (Step Over) a few times and see the PORTA register changes in the I/O View.

Using Breakpoints

If you want to debug a portion of a program, add a breakpoint to the beginning of this part of the code and press the run button. The IDE runs the program and when it reaches the breakpoint, it stops running and the yellow cursor is shown on the breakpoint line. Below, you see the steps in detail.

1. Right click on the "PORTA = 0×00 ;" instruction. A pop-up menu appears. Choose **Breakpoint** and then Insert Breakpoint. A red bullet appears on the hand side of the "PORTA = 0×00 ;" instruction.

toggieProp File Edit V		Relactor (VA) Surround With (VA)	- mai - a -	s Window Help	
03.0	1. 1.	Insert Snippet Surround With	Cbi+K, Cbi+X Cbi+K, Cbi+S	19233TH 8-, 03333, 05 -, -	Cagamita a
toggfefroject		Sreakpoint	•	Add Databased paint Chil-Shift+R Table To Chil	
 main.schi stoclud sdefine 	2 3	Add Watch QuickWatch Pin To Source	Shift+P9	Insert Taxagoint 2 Name Insert Taxagoint 8 & AO_CONVERTER * ANALOG_CONVERTER	Value
vint sai	や日の間	Show Next Statement Run To Currer Set Next Statement Go To Disamembly	Alt+Num* Cel+FLD Cel+Shift+FLD	ERENCIA E EERICIM E CETERNAL INTERRUPT	
uhi t	× 4	Cut Copy	Ctel+X Ctel+C	Name Address Value	Bts 000000-
83 %	_del	1 = 0000; //weit 1 ; #/_ms(1000); //weit 1 ;r	das Lõe E	* PORTA 0x38 0x00 C	Sol_ Pro.
Watch I		100000		- I × Breakpoints	• 4 3
Name		Value		Type - New * X + 3 4 3 4 9 - 3 Columna *	
🖿 Autos 🌘	Lo	cels 📰 Watch 1 🐙 Wat	ich 2	🖬 🖬 Breakpoints 🥻 Cal Stack 🖿 Convinand. 🛛 Ammediate	. 🔳 Output
lopped				1042 C819 (C613	140

2. Press *F5* or the *Run* button. The IDE runs program until it reaches the Breakpoint. Now, you can continue debugging from the breakpoint using the *Step into* and *Step over* buttons.

ile Edit View V	AssistX ASP Project Build De 3 3 4 4 5 0 0 0 0 3 0, 1 0 0 1 0	ag Taols Window Help St- Black of Mindow St-	ie remined ine	Ma •19 31.088.0	
main while	Part of the second	Step Into	- 100	Filter	
<pre>#include cent/ edefine #_CPU = int min(vold) { DORA = 0of while(1) {</pre>	<pre>10.Rs seccoset. /delay.N* F; //make port A as output part = BoFF; //make all pins HJDH me(laee); //make i pins HJDH me(laee); //make i pins HJDH pm(laee); //make i pins HJDH</pre>	Step over (F10) breakpoint		Name AD_CONVERTER ANALOG_COMPARATOR BOOT_LOAD CPU EEPROM EXTERNAL_INTERRUPT ICITIA ame Address Value PDRA 0x30 0x00 DDRA 0x3A 0x00 PORTA 0x38 0x00	Value
Natris 1		* 0 X Benkrou	1 10	A PALADA DEPROVO	• 3 SOL. 📑 PYO
Name	Value	Type - New+	X 95 69 51	Columns *	
		Name 12.0	toggleProject.c, line 71 charac	Labels Condition ter 1 (ne condition	Hit Count on) break always (
🖽 Autos 📑 Local	i 🖉 Watch 1 🛒 Watch 2	G Break	points 🦣 Call Stack 🛄 Co	mmand 💼 Immedia	ete 🖬 Output

Disassembly

The Compiler converts C programs to machine instructions. To see the assembly equivalent of your program:

3 · 41] · 2	1 1 1 × 1 13 9 -		Windows 2	N NE		Breakpoints	Att+F9
Conservative × 10 Address: main	99 04 ⊈ 43 ÷ == ⊿ 9916Noject.c	0.12	Start Debugging and Treak Stop Debugging Start Without Debugging Disable debugWDRE and Close	Alt+15 Chd+Shift+F5 Chd+Alt+F5	N IN ST	Data Breakpoints Processor View D/O View Liver Watch	
economic Option economic Option economic Out economic Out economic Out economic Out	N.21,0000 Class Line 0x35,R29 Out to 1/0 0x30,K28 Out to 1/0 0x30,K28 Out to 1/0	~ ヤヤ	Centinue Execute Stimulifile Set Stimulifile	P5	10 10 11	Output Parallel Tasks Parallel Stacks	Chil+Shift+D, K Chil+Shift+D, S
00000032 399 00000024 399 E:\AWEro { DDRA = 8x 00000030 559 00000035 501 PORTA 00000035 001 *	brodonobbit Jump brodonobbit Jump jectaltoggleProjectitoggl FFJ //make port A as outp R24 Set Register DulA,R24 Out to I/ = 0xFFJ //make all pis Out to I/ m	0 = 3 7 3 7 7 F	Restart Error AB QuickWetch Step Into Step Quit Step Quit Run To Carson Paret	Con+75 Swit+79 F11 F10 Swit+F11 Con+F10 Swit+F1	11日本を自当日	Watch Autos Locals Immediate Call Stock Threads Modules	Cbi+AR+V, A AR+4 Cbi+AR+1 AB+7 Cbi+AR+H Cbi+AR+H Cbi+AR+U
Watch 1			Percepio Trace		-	Processes	Ctrl+Shift+Alt+P
Name	Value	0.0	Toggle Breakpoint New Breakpoint Delate AB Breakpoints Disable AB Breakpoints Clear AR Tata Tipe toward Preactor	19 Chil+Shift+P9		Diassembly Register columnour ; r R Command-	Ah-8 Ah-5 Ino consision prese av

1. Click on the *Debug* menu, choose *Window* and then *Disassembly*.

- 2. The following window appears.
 - a. The black texts show the C program instructions. The assembly equivalent of the C instruction is shown below, in gray.
 - b. The first column shows the memory address where the assembly instruction is located.
 - c. The second column shows the assembly instruction.

isasambly X	toggleProject.c			+ NView	
Lddress main				• 3 a fibs	et -
dooloogg 0 dooloogg 0	01 K29,0000 Lian Immuniste DT 0525,029 Qut to 1/0 location DT 0520,010 Qut to 1/0 location ALL Ar000000055 Call subroutine MP Hu00000040 Jump MP Au0000000 Jump			AD_CO ANALO BOOT CPU BEPRO	DWERTER DG_COMPARATOR LOAD M NAL INTERNIPT
DDRA	MARP; //make port A as output port EN R24 Set Register UT BulA,R34 Out to I/O location The sasembly Instruction The sasembly Instruction	Debug//./toggl	eProject.c	Name Mame Month Marken DDRA Month DDRA	A Address Value Bits 0x39 0x00 000000000000000000000000000000
1 DDRA - 00000035 5 00000035 0 POR 00000035 0	Marke port A as output port _ Burke port A as output port _ EN R2A Set Register UT Bulk_N2A Out to I/O location The assembly instruction	Debug//./togg]	eProject.c uction	Name Mare PINA BODRA PORTA	Address Value Bits 0x39 0x00 000000000000000000000000000000