# Assembly Programming in Atmel Studio 6.2 Step by Step Tutorial



Atmel

Powered by Visual Studio

Sepehr Naimi BIHE University 4/3/2015

# Contents

Introduction	2
Downloading and Installing Atmel Studio	3
Opening Atmel Studio	3
Creating the first project	4
Writing the first Assembly program	6
Building	6
Debugging	7
Using Breakpoints	.10

## 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 × +	<b>x</b>
	≡
Software       Description         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 Assembler.
  - b. Name the project as *toggleProject*.
  - c. Choose the path where you like to save the project by clicking on the *Browse* button.
  - d. Press OK.

New Project						×
Recent Templates		Sort by: Default			Search Installed Templates	P
Installed Templates		AVR Assembler P	Project	Accembler	Type: Assembler Creates an AVR 8-bit Assembler project	. E
Name Location Solution name	toggleProject E1\AVRprojects\ toggleProject	2			Browse3	
				_	OK Cencel	1

### 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	P
Name	All megaAVR. 8-bit	bytes)	Data Memory (bytes)	EEPROM (b	oʻ.	Device Info:	
ATmega169PA ATmega16A ATmega16HVA ATmega16HVB ATmega16HVBrev ATmega16M1 ATmega16U2 ATmega16U4	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	512 512 256 512 512 512 512 512 512 512		Device Name: ATmega32 Speed: 0 Vcc: 2.7/5.5 Family: megaAVR Datasheets Supported Tools	
ATmega2560 ATmega2561 ATmega2564RFR2 ATmega256RFR2 ATmega324A ATmega324P ATmega324P ATmega325 ATmega3250	256 256 256 256 32 32 32 32 32 32 32 32 32		8192 8192 32768 32768 2048 2048 2048 2048 2048 2048	4096 4096 8192 8192 1024 1024 1024 1024 1024 1024		Atmel-ICE     AVR Dragon     AVRISP mkli     AVR ONE!     JTAGICE3     JTAGICE mkli     Simulator     SIT/E00	
AT 2250	32 22 11	;	2048	1024	٣	STK500 STK600	

The compiler automatically makes the *toggleProject* and adds an assembly file to it.

秦 toggleProject - AtmelStudio				x
File Edit View VAssistX ASF Project Build Debug Tools Window Help				
: 🛅 ▾ 🕀 🗁 💕 🚽 🍠   🔏 💫 🟝 😩 🕫 ㅜ 🔍 ㅜ 📮 ㅜ 🖳   🔚 🍳   ♦ 🚧 Debug	- 🖄		- 🛛 🖓 🚰 🛃	] 🚆
[[[[[[] [[] [[] [[] [[] [[] [[] [[] [[]	- <u>-</u> i 🖉 🖂 🗉	I 🐺 🖪 🚽 🕮 👑	≛   🍝 📲 🗒	
toggleProject.asm ×	•	Solution Explorer	<b>-</b> ₽	x
/*	÷	🕒   🗗		
* toggleProject.asm *	<u>^</u>	🔊 Solution 'toggle	eProject' (1 project	t)
* Created: 4/3/2015 1:36:03 AM		a 📙 toggleProje	ect	
* Author: Naimi	=	Dependent	encies	
		Dutput	Files	
		🚠 togglePi	roject.asm	
		🔍 AS 🤎 VA	👘 VA 🔁 Sol	I
		Properties	<b>-</b> ₽	×
100 % - 4				•
Find Results 1			<b>-</b> ↓	×
▶   急 ▶   乘   幟				
				*
				-
<			÷	
🙀 Error List 🔳 Output 🙀 Find Results 1				
Ready Ln 1	Col 1	Ch 1	INS	

### Writing the first Assembly program

Type the following program.

```
.INCLUDE "M32DEF.INC"
LDI R16,0xFF
OUT DDRA,R16
L1: OUT PORTA,R16
LDI R20,0
OUT PORTA,R20
RJMP L1
```

### **Building**

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



### 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	
: 🛐 • 🔁 🖼 • 🧉 🛃 🍯 🕺 🖓 🖓 • 🖄 🖓 • 🖓 • 🖓 • 🖏 🔚 🔍 🕨 🕅 Debug 🛛 • 🧭 👘	emainedTimeVal 🔹 🖓 🙄 💱
: 🔁 🐷 🖓 🍋 🔓 😘 💑 🍱 🖕 🧰 🍙 수 비 🕨 🐼 🧐 📜 한 🖬 🏹 👘 🐨 📲 🐙	
toggleProject × toggleProject.c	Solution Explorer 🔹 🕂 🗙
2	
Configuration: N/A *	Solution toggleProject (I project)
Build Events	Dependencies
Toolchain	Dutput Files
Device the second	toggleProject.c
Selected debugger/programmer	
Advanced Simulator	AS VA E VA Sol
AVRISP mkII	Properties
	• # ×
Done building project "toggleProject.cproj".	
Build succeeded.	
====== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ========	C239
<m< td=""><td></td></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. A *Memory* Watch windows appears which shows the contents of the Flash memory. Close the window.
- 5. Now a yellow cursor is on the first line of the main program and the IDE is ready to debug the program.

<ul> <li>toggleProject (Debugging) - AtmelStudio</li> <li>File Edit View VAssistX ASF Project</li> <li>The state of the sta</li></ul>	Build Debug Tools Windo 9 • ( • • • • • • • • • • • • • • • • •	ow Help ↓ ▷ ₩ Debug <sup>1</sup> *3 T Hex © Step Out Run to cursor			• • • ×
<ul> <li>Autnor: Naimi</li> <li>.INCLUDE "M32DEF.INC"</li> <li>LDI R16,0xFF OUT DDRA,R16</li> <li>L1: OUT PORTA,R16</li> <li>LDI R20,0</li> <li>OUT PORTA,R20</li> <li>RIMP L1</li> </ul>	(F10)	Processo	Stack Pointer X Register Y Register Z Register Status Register Cycle Counter Frequency Stop Watch Registers Poo	0x0000 0x0000 0x0000 0x0000 I T H S V N Z C 0 1.000 MHz 0.00 μs	
100.84			* R01	0x00	*
Watch 1	<del>~</del> 4	× Breakpoints	ASF E	roce 🤇 Soluti 😭	• Prope • 4 ×
Name Value	Туре	New • X S K	dition Hit Count	lumns *	"
🖼 Autos 👼 Locals 🙇 Watch 1 👰 Wate	h 2	📑 Breakpoints 🛵	Call Stack 🛛 Command	d 🚈 Immediate 🔳	Output
Stopped		Ln 9	Col 1	Ch1	INSi

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.

#### **Processor Tab**

The Processor tab shows the current values of the CPU registers including R0-R31, SP (Stack Pointer) and PC (Program Counter). You can also change the values of registers by double clicking on their values and typing a new value.

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

3. al a.	3 3 3 1 A D D 1 - 1		Windows 2		13	Breekpoints	Alt=F9	1
00501	9 8 0.4 41 - 1 10 4	86	Start Debugging and Break	Att+15	10	Data Breakpoints		F
oggleProject.c >			Stop Debugging	Ctrl+Shift+F5		Processor View		
main.while	• =	19.	Start Without Dabugging	Chil+Alt+F5	10	UO View		ſ
	the second s		Continue	15	N II	Output		Ľ
winclude cav	r/10.fr	4	Exercite Stimulifile	0.070	131	Parallel Tasks	Chil+Shift+D.K	1
winclude "ut	il/delay.h"	4	Set Stimulifile		2	Parallel Stacks	Ctd+Shift+D, S	L
wint main(voi	¢)	-	Restart			Watch		1
1 DORA - 0	xFP: //make port 4 as output of	н.	Break All	Ctrf+75	-	Autos	CHI+AR+V, A	1
	and the second statement of th	64	QuickWatch	Shift+F9	22	Locais	Alt=4	L
1 (		91	Step Into	F11	1	Immediate	Ctrl+Alt+1	Ŀ
PURT	A = QAFF; //make all pins HII	ÇB	Step Over	F19	63	Cell Stock	Alt-7	I.
PORT	A - 0x00; //make all pins LOw	2	Step Out	Shift+F11	14	Threads	Ctrl+Alt+H	L
_del	#y_ms(1000); //wait 1 sec	-1	Tun To Custor	Ctri+F10	3	Modules	Ctrl+AR+U	L
3 76 - 4		T	Reset	Shift+F5	17	Processes	Ctrl+Shift+Alt+P	
atch 1			Percepio Trace	,		Memory		F
Name	Value		Toggle Breakpoint	10	1	Disastersbly	Alt+8	F
			New Breakpoint		-	Registers	Alt+5	r
		큔	Delete Ad Breakpoints	Chil+Shift+F#	-			÷.,
			Clear All DataTigs					
	and the second se		Teport DataTipe				and the second s	

 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	bugging) - AtmelStudio	e Task Window Male			0.0
			Oebug +	TremainedTimeVal	- <b>10 9 6 4</b> 000
103205	8	1 SI 1 1 1 1 1	T   Hex   🐻 • .	1000000.10	States   and Strength
toggleProject.c ×				- 10 View (1)	* \$ X
main.while	• 🕂 🌩 while(1)			• CGo 🖃 🖃 Filter	- 11
<pre>sinclude carr sdefine s_cru sinclude futi sinclude futi sint main(vold ( 000A + 8x shile(1) ( POTA </pre>	/io.hs soebeedu. L/delay.h* 0 \$F; //make port A ws output port i = 0xf9; //make all plos wide y_ms(1000); //wit 1 sec = Rx00; //wit 1 sec			Name     Name	Inne Value ERTER COMPARATOR AD INTERRUPT ddress Value Brts ba39 0x00 000000 ba34 0x00 000000 ba38 0x00 0000000
Watch I		• 4 X 1	Reskncints	32 Koto	- 1 x
Name	Value	Туре	New- X 993 Name Labels Co	5 🖓 🥹 🖃 🗐 Colum ndtion Hit Count	ni •
Maton 🖬 Loca	ale 22 Watch 1 🕅 Watch 2		📑 Breakpoints 🔝	Call Stack 🔲 Command	🚛 îmmediate. 🛛 🖬 Output
Ready	And the second se				

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 "OUT PORTA, R20" instruction. A pop-up menu appears. Choose **Breakpoint** and then Insert Breakpoint. A red bullet appears on the hand side of the "OUT PORTA, R20" instruction.

. 10	oggleProje		Refactor (VA)	•						6	
Re	Edit Vi		Breakpoint			Addi	atabreakpoint	Ctrl+51	utt+R		
	33	2 2	Add Watch QuickWatch Pin To Source	Shift+F9		Insert	Breakpoint 2 Tracepoint		120		178); 11
•			Show Nest Statement Run To Cursor Set Nest Statement Go To Disessembly Cut Copy Paste Outlining RTA, R28	Alt+Num* Cbri+F10 Cbri+Shift-F10 Cbri+X Cbri+X Cbri+V					Name Program Counter Stack Pointer X Register Y Register Status Register Cycle Counter Frequency Stop Watch Registers R00	Value 9x6000 9x6000 9x6000 0x6000 1011日15100 6012 0 1.000 MHz 9x00 μ8	2
100	5 -11							1.0	ASF E Pr	oxoo moor Soluti	Prepa_
Wate	ch3					• # ×	Breakpoints			194000 10	- # x
14	ame		Value		Type		New: X 2	2 3	Col	kamns =	
	Autos 💽	Loci	ali 🏹 Watch 1 🐙 Watc	h 2		-	Name Labeh	Condition	Hit Count		Cutput
Read	¥.						Ln 14		Col7	564	NS 2

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.

🗣 toggleProject	(Debugging) - AtmelStudia					1.000	
File Edit View	VAssistik ASF Project I	fuld Debug Tools We	ndow Help				
Etal alla	3 H 3 X 3 3 1	- M · 第•131回	A D 340 C	6666): 91	10	- 18	2 - 20 - 2
12320	8304 G.I.M.	101349	(P) a T	Het a-		. dela.	1.
Disassembly	toggleProject	Land			<ul> <li>Processor</li> </ul>		• <del>7</del> ×
· Crea Art ·/ ·INCLOD OUT LI: OUT LI: OUT LDI OUT RDIS RDIS	Debugging Itad: 4/1/2015 1:30:03 hor: Walki K "M32DEF.INC" R16,0xFF DORA,R16 R20,0 PORTA,R16 R20,0 A breakpool 1	(F5) (F11) Step o (F10	ver		Program Counter Stack Pointer X Register Y Register Y Register Z Register Status Register Cycle Counter Frequency Stop Watch Registers R00 R01 Q. ASF ED @ P	Value 3x80000000 0x6000 0x6000 0x6000 0x6000 0x6000 0x6000 0x600 0x60 0x7 0x7 0x7 0x7 0x7 0x7 0x7 0x	C)
Watch 1			9 × Breakpo	ints			- 9 ×
Name	Value	Туре	* New+	XPX	49 2 1 1 Co	durmis *	1
-			Name			Labels Condition	Hit Co
			+ 11.000		· · · · · ·	1. p. 15	3.1.4
La Autos M L	ocals 😹 Watch 1 🗶 Watch	12	📑 fire	ekpoints 🔤 🕄	Il Stack Di Commune	t 🚰 Immediate-	Dutput
Stopped				Ln 14	CALL	Ch1	115 _d

3. Using the *Stop Debugging* icon you can stop debugging whenever you want.