Assembly Programming in Atmel Studio 7 Step by Step Tutorial



Sepehr Naimi BIHE University 12/1/2017

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 7.

Downloading and Installing Atmel Studio

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

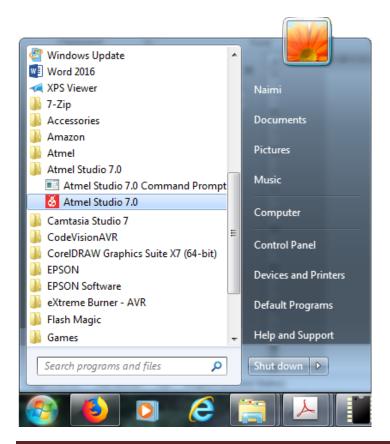
http://www.microchip.com/avr-support/atmel-studio-7

🔹 Atmel Studio 7 Micro	chip Tecl X +		
\leftrightarrow > C \textcircled{a}	① www.microchip.com/≈ 🛛 🗷 🔞 🗰 🐨 🛇 🍳 Search	👱 III\	Ξ Ξ
	Downloads Documentation		•
	Atmel Studio 7		
	Title Date Size Published	D/L	
	Windows (x86/x64)		
	Atmel Studio 7.0 (build 1645) web installer (recommended) - This installer contains Atmel Studio 7.0 with Atmel October Software Framework 3.35.0 and Atmel Toolchains. It is 2017 2.4 M recommended to use this installer if you have internet access while installing. 2017	в 戦	
	Atmel Studio 7.0 (build 1645) offline installer - This installer contains Atmel Studio 7.0 with Atmel Software Framework 3.35.0 and Atmel Toolchains. Use this installer if you do not have internet access while installing. Quita SHA1: b7da45b0a5c594d7298640f4237eb1d84a3a5834	18 戦	E

Run the downloaded program to install the Atmel Studio IDE.

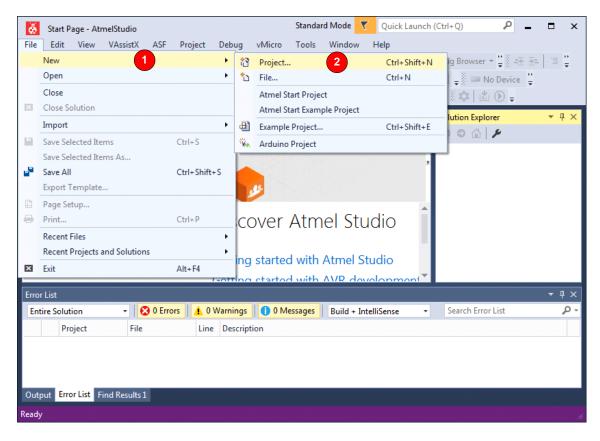
Opening Atmel Studio

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

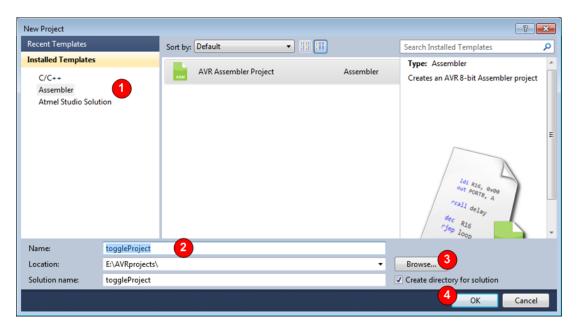


Creating the first project

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.



3. In the *Device Selection* dialog

- a. Select *megaAVR* as the *Device family*.
- b. Choose ATmega328 (or any other Chips you want to use)
- c. Select **OK**.

Device Family:	ATmega	•			Searc	h for device
Name	All ATautomotive	rtes) Data Memory	(bytes)EEPROM (b	ytes	Device Info:	
ATmega32 ATmega324A ATmega324P ATmega324PB ATmega325 ATmega3250 ATmega3250A ATmega3250P ATmega3250PA ATmega325PA ATmega325P ATmega328P ATmega328P ATmega328P ATmega329 ATmega329 ATmega3290 ATmega3290	ATmega 1 ATtiny XMEGAA XMEGAB XMEGAC XMEGAD XMEGAD XMEGAE 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	* 	Device Name: Speed: Vcc: Family: Datasheet (Summa Device Page Supported Tools Atmel-ICE AVR Dragon AVR Dragon AVR SP mkll AVR ONE! X EDBG X EDBG MSD J JAGICE3	ATmega328P N/A N/A ATmega

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

👸 toggleProject - AtmelStudio St	tandard Mode 🔻	Quick Launch	(Ctrl+Q)	- • ×
File Edit View VAssistX ASF Project Build Det © • ○ ☆ • ⊕ ☆ • ☆ ₩ ₩ ₩ ↓ ⊕ ☆ ♡ • № ☆ = → !! ▶ ↔ ☆ ☆ ♪ ▶ ☆ ↓ ☆ Hex %	९ - 🔠 🔍	Debug	 Debug Browse 	
Arduino 1.6/1.8 • Arduino/Genuino Uno				
<pre>main.asm* +> X ASF Wizard ; toggleProject.asm ; ; Created: 12/07/2017 03:26:35 ; Author : Naimi ; ; Replace with your application code start: inc r16 rjmp start 100 % •</pre>		÷ 	Solution Explorer Search Solution Explore Search Solution Explored Search Solution Explored	r (Ctrl+;) P · t cies es
Output				- ₽×
Show output from: Output Error List Find Results 1	• C ₂ 1		ي ب	
Ready	Ln 4	Col 31	Ch 31	INS "

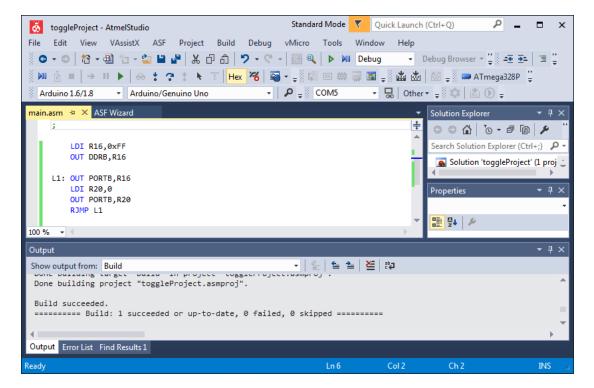
Writing the first Assembly program

Type the following program.

```
;
; toggleProject.asm
;
LDI R16,0xFF
OUT DDRB,R16
L1: OUT PORTB,R16
LDI R20,0
OUT PORTB,R20
RJMP L1
```

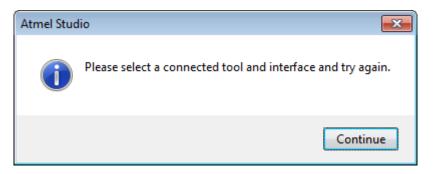
Building

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 and Break* 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 - A	tmelStudio	Standard Mode	Quick Launch	(Ctrl+Q)	_ = ×
		Debug vMicro Tools \	Window Help		
G • 🛛 🔤 🔁 • 🗄	*1 - 21 ■ ■ ↓ □ 1 ? •1 - 21 ■ ■ ■ ↓ □ 1 ? •1 - 21 ↔ • ↑ ↓	- ° - 🖪 🔍 🕨 🕅	Debug 🗸 🗸	🛛 🛃 🚽 🗰 ATmega	328P 🚆
toggleProject*2×	main.asm ASF Wizard		-	Solution Explorer	- ₽ ×
Build Build Events Toolchain Device Tool* Components Advanced	Configuration: N/A Platform: N/A Selected debugger/programmer Simulator 1 Custom Programming Tool Programming settings	v v		Search Solution Expl Search Solution Expl Search Solution Expl Search Solution Expl Search Solution Expl Depende D	erct Ancies
Output					- ₽ ×
Show output from: G	eneral	- <u></u>	¥ :		
] Please select a connected too				
Ready		Ln 1	Col 1	Ch 1	INS

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. Now a yellow cursor is on the first line of the main program and the IDE is ready to debug the program.

toggleProject (Debugging) - AtmelStudi	o Standard Mod	e 🔻 Quick Laund	:h (Ctrl+Q)	- □ ×
	ject Build Debug vMicro Tools	Window Help		
G • O 🕾 • 🕘 🐀 • 🚔 💾 🔐	ኤርባስ 🎾 - ୯ - 🔝 🔍 🕨 ነ	🛛 Debug 👻	Debug Browser 👻 🚆 🛓	€ <u>₹</u> = '≣ 🚆
🕺 🕅 🝈 ■ 🛛 → 🗉 🕨 🔂 🛟 🛟 🚺	🖎 🛨 🕂 Hex 🄏 📓 - 📮 🛒 📼 🛱) 📷 🖪 📮 📇 🛃	🖌 🔯 📮 🕅 ATmega3	28P 🚆
Arduino 1.6/1.8 • Arduino/Genuino			er • 🝦 🗱 🗼 🖲 🖲	
toggleProject <mark>main.asm 😕 🗙</mark>	•	Processor Status		- ₽ ×
3	+	Name	Value	
LDI R16,0xFF	^	Program Counter Stack Pointer	0x00000000 0x08FF	
OUT DDRB,R16		X Register	0x0000	
L1: OUT PORTB,R16		Y Register	0x0000	
LDI R20,0		Z Register Status Register	0x0000 [] [] H S V N Z C	
OUT PORTB,R20 RJMP L1		Cycle Counter		
		Frequency	16.000 MHz	
		Stop Watch	0.00 µs	_
		Registers		
		R00 R01	0x00 0x00	
	-	R02	0_00	-
100 % 👻 4		Processor Status 1/	O Solution Explorer Pro	perties
Watch 1	- † ×	Memory 4		- ₽ ×
Name Value	Туре	Memory: prog F	LASH	-
			f ef 04 b9 05 b9 40	
			0 45 b9 fc cf ff ff f ff ff ff ff ff ff	
Autos Locals Watch 1 Watch 2 Find Resul	× 1	ppog 0y0015 +	* ** ** ** ** ** **	
			Comm Immed C	
Stopped	Ln 15	i Col 5	Ch 5	INS 📑

5. 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 RO-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.

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

toggleProject (Debugging) - AtmelStudio	(1		Standard Mode 🔻	(Juick Launch (Ctrl+Q)	₽ = ¤ ×	
File Edit View VAssistX ASF Project Build	Deb	ug vMicro Tools Window	Help				
G • O 🕄 • 🕘 🏷 • 🖕 💾 🖉 🗶 🗗 🖒 '		Windows 2	•		Breakpoints	Alt+F9	
🕅 🕅 ■ → II ▶ 60 🕇 ? ↑ ト 🛨 Hex	M	Start Debugging and Break	Alt+F5	5	Data Breakpoints		
Arduino 1.6/1.8 • Arduino/Genuino Uno		Attach to Target			Processor Status		
		Stop Debugging	Ctrl+Shift+F5	1	I/O	3	
toggleProject main.asm ⊕ ×	⊳	Start Without Debugging	Ctrl+Alt+F5		Live Watch	-	
,		Disable debugWIRE and Close		Θ	Program Counter Trace		
DI R16,0xFF	•	Continue	F5	∍	Output		
OUT DDRB,R16	4	Execute Stimulifile			Watch	•	
L1: OUT PORTB,R16	4	Set Stimulifile		्र (%)	Autos	Ctrl+Alt+V, A	
LDI R20,0 OUT PORTB,R20	ð	Restart			Locals	Alt+4	
RJMP L1	П	Break All	Ctrl+F5	Þ	Immediate	Ctrl+Alt+I	
	⇔	QuickWatch	Shift+F9	Œ	Call Stack	Alt+7	
	*	Step Into	F11	10.	Threads	Ctrl+Alt+H	
	3	Step Over	F10		Modules	Ctrl+Alt+U	
	:	Step Out	Shift+F11	d ^{ar}	Processes	Ctrl+Shift+Alt+P	
	h.	Run To Cursor	Ctrl+F10		Memory	•	
100 % -	Ť	Reset	Shift+F5	G	· · · · · · · · · · · · · · · · · · ·	Alt+8	
Watch 1		Toggle Breakpoint	F9	88	,	Alt+5	
Name Value		New Breakpoint	•				
Value	89	Delete All Breakpoints	Ctrl+Shift+F9	-	LASH • fef04b905b940e0	45 .ï@àE 🔺	
		Clear All DataTips			9 fc cf ff ff ff ff ff		
		Export DataTips		ff			
Autos Locals Watch1 Watch 2 Find Results1		Import DataTips			f ff ff ff ff ff ff ff ff oi Comman Immediat.		
Stopped		Save Dumn As					

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

Arduino 1.6/1.8 Arduin	ASF Project Build Debug vMi ■ ■ & ① ① ♡ - ♥ ♥ @ t ⑦ t N T Hex % ■ - = no/Genuino Uno • P	🥵 📼 🛱 🙀	Window Help Debug → Debug Browser + "
OUT DDRB,R16 L1: OUT PORTB,R16 LDI R20,0 OUT PORTB,R20 RJMP L1			CPU Registers (CPU) EEPROM (EEPROM) External Interrupts (EXINT) Wo I/O Port (PORTB) Z Wo I/O Port (PORTC) Name Address Value Bits Wo PINB 0x23 0x00 DRB 0x24 0xFF Wo PORTB 0x25 0x00 Processor Status I/O Solution Explorer Properties
Watch 1		- ₽ ×	Memory 4 → 平×
Name	Value	Type	Memory: prog FLASH • prog 0x0000 0f ef 04 b9 05 b9 40 e0 .i00 prog 0x0008 45 b9 fc cf ff ff ff ff ff E.u_u prog 0x0018 ff ff ff ff ff ff ff ff ff .uuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu
Autos Locals Watch1 Watch2 Stopped	Find Results 1	Ln 1	Call Sta Breakp Comm Immed Output Memo 11 Col 1 Ch 1 INS

8. Press *F10 (Step Over)* a few times and see the PORTB register changes in the *I/O* window.

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 left side of the "OUT PORTB, R20" instruction.

秦 te	oggleProjec		Refactor (VA)	•	1								- 0	×
File	Edit Vie		Breakpoint 1	•		Add [atabreakp	oint C	trl+Sł	hift+R				
: 6] • 曲 洒	61	Add Watch		۲	Insert	Breakpoint	2				-	🗟 🚰	2
1 🔁) 📴 🐺 🕯	61	QuickWatch	Shift+F9		Insert	Tracepoint				🐺 🖪 :	- 🗄 🛗 🔒	5 - 1 ₽	
Dis	assembly		Pin To Source						•	Processor			-	Ψ×
	*	⇔	Show Next Statement	Alt+Num *					÷	Name		Value		
	* Cr	*3	Run To Cursor	Ctrl+F10					*	Program Co		0x00000000		*
	*/ A	•	Set Next Statement	Ctrl+Shift+F10						Stack Point X Register		0x0000 0x0000		=
Ι.		tā,	Go To Disassembly							Y Register		0x0000		
	.INCL	ằ	Cut	Ctrl+X					Ξ	Z Register		0x0000		
	OU		Сору	Ctrl+C						Status Regi Cycle Coun		ITHSVE 0	IZC	
	L1: 0U	2	Paste	Ctrl+V						Frequency		1.000 MHz		
			Outlining	•						Stop Watch		0.00 µs		
			RTA,R20							Registers	;			
	KJ	MP L:	1							R00 R01		0x00 0x00		
100	% - 4							•			Pro	ce 🟹 Soluti	🚰 Pro	ppe
Wat	ch 1				-	Ψ×	Breakpoir	ats						Ψ×
	lame		Value		Туре				10	0	Colur	mns •		**
										n Hit Count				- 1
							i tonic	Contraction Contraction	ancion	Count				- 1
	Autos 🚟	Loca	ıls 📺 Watch 1 👼 Wa	tch 2		Ŧ	Break	moints 🛵	Call S	tack 🗖 Cor	nmand	📻 Immediate	a 🗖 Oi	itout
		LOCA	a an watch i an wa	ich z			Lag break		call 3					
Read	y							Ln 14		Col 7	Ch	14	INS	S!

2. Press *F5* or the *Continue* 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) -		Standard Mode	Quick Launch (Ci	trl+Q) 🔑 🗕	□ ×
File Edit View VAssist) St		/Micro Tools Wi	ndow Help		
G • O 13 • ⊕ 12 •	(F10) 米 凸 台 フ ・ペ・	🔐 🔍 🕨 🕅 De	ebug 👻 Debug Br	rowser * 🍟 🖛 🚈 🕴 🗉	2≣ ∵
₩ 🛉 🔳 🗲 🖬 🕨 🕁	🔹 🖓 🏦 k 🝸 Hex 76 🖓 -	- 📮 🖳 📾 🙀 🙀	🖪 📮 🏙 🏙 🔛 📮	🗰 ATmega328P 🥤 Sir	nulator 🚽
Arch 10 1.6/1.8 - Arduj	p/Genrie	P _ COM5	- 🔜 Other - 🚽	× • • • =	
Stop Continue	Step Out		1/0		• 4 ×
Debugging (Rull) in	to have been a second se	÷			• # ×
; Auchi III (F1	1)	T	Filter:	- 4	
			Name	Value	
LDI R16,0xFF			🗉 🧰 CPU Registers (Cl		^
OUT DDRB,R16			EEPROM (EEPROI External Interrupt		
L1: OUT PORTB,R16			External Interrupt I/O Port (PORTB)		
LDI R20,0			VO Port (PORTC)		-
OUT PORTB, R20	A breakpoint				
RJMP L1			Name Address		
			PINB 0x23	0x00 0xFF	
			PORTB 0x25		
100 % -		-	December Status I/O C	olution Explorer Propertie	× .
				olution explorer Propertie	
Watch 1			Memory 4		- ₽ ×
Name	Value		Memory: prog FLASH	-	
				04 b9 05 b9 40 e0 .ï	
				fc cf ff ff ff ff E. Ff ff ff ff ff ff	
			nrog ØvØØ18 ff ff f	** ** ** ** ** **	
Autos Locals Watch 1 Watch 2	Find Results 1		Call Sta Breakp Cor	mm Immed Output	Memo
Stopped		Ln 13	Col 1	Ch 1	INS 📑

3. Using Stop Debugging, you can stop debugging whenever you want.