## University of Massachusetts Dartmouth Department of Electrical and Computer Engineering

ECE 263 VERSION 2 Atmel Lab 1 – First program in Atmel Studio 7.0 Lab name: led\_main.c due: see http://ece160.org

1. Launch Atmel Studio by clicking on the desktop icon. You will see the screen below. It is possible you will receive a dialog box notifying you there are updates. If you do, just cancel out of this box since you do not have sufficient privileges to install updates.



2. To create a new project, click File > New > Project

ð	Start Page AmerStudio					
File	Llit View Price Deb	ug Tools Hel	р			
	New	•	*3	Project	Ctrl+Shift+N	
	Open	•	*)	File	Ctrl+N	
	Close		As	Atmel Start Project		
×	Close Solution		As	Atmel Start Example Proj	ect	
	Import	•	≞	Example Project	Ctrl+Shift+E	
	Save Selected Items	Ctrl+S			~	
	Save Selected Items As			<u>9</u>		
<b>1</b>	Save All	Ctrl+Shift+S				
	Page Setup		ÍSC	cover Atmel	Studio	
	Print	Ctrl+P				
	Recent Files	•	ttir	ng started with Ati	mel Studio	
	Recent Projects and Solutions	•	ttir	on started with AV	'R development	
×	Exit	Alt+F4		Atmal Start Confi	aurator	
F	Recent	<u> </u>	Pen	Auner start Conn	gurator	

3. In the New Project window, click C/C++ in the left pane. Click GCC C Executable in the middle pane. Next change the Name to something meaningful (such as led for the first lab). If you wish to change from the default Location, (suggested: drive Z:), you may do that as well. Click OK.

New Project					? ×
▷ Recent		Sort by	Default 🔹 🏭		Search Installed Templates (Ctrl+E)
▲ Installed	Installed     Image: GCC C ASF Board Project       C/C++     GCC C ASF Board Project       Assembler     GCC C Executable Project       AtmelStudio Solution     GCC C Static Library Project		GCC C ASF Board Project	C/C++	Type: C/C++ Creates an AVR 8-bit or AVR/ARM 32-bit
Assembler			GCC C Executable Project	C/C++	C project
AtmelStudio Solut			C/C++		
		GCC	GCC C++ Executable Project	C/C++	
		GCC C++ Static Library Project		C/C++	int main(void)
		<b></b>	Create project from Arduino sketch	C/C++	3 Printf("Hello" GCC
<u>N</u> ame:	led				
Location:	Z:\ece160project	ts\		•	<u>Browse</u>
Solution name:	led				✓ Create directory for solution
					OK Cancel

4. In the search box, type 328pb (upper or lower case doesn't matter). Atmega328pb should appear in the device listing. Click on it, then click OK.

Device Selection	n										$\times$
<u>D</u> evice Family	: [	All	~					328pb		×	
Name	Арр	./Boot Memo	, (Kbytes)	Data Mem	ory (bytes)	EEPROM (bytes)	Device Info:				$\sim$
ATmega328PB	32			2048		1024	Device Name Speed: Vcc: Family: Device page Datasheet Supported To Atmel-ICE AVR Drag AVR Drag AVR ONE X EDBG X EDBG MS UTAGICE	for ATmeg ols ion nkll i1 SD 3	<u>a328PB</u>	ATmega: N/A N/A ATmega	~
								<u>(</u>	<u>o</u> k 🔶	<u>C</u> ancel	

5. You will be presented with a window similar to the following. This should look very familiar to anyone who has used Microsoft Visual Studio.

👸 led - AtmelStudio	Standard Mode 🔻 Quick Launch (Ctrl+Q)	- 8	×					
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>P</u> roject <u>B</u> uild <u>D</u> ebug <u>T</u> ools <u>H</u> elp								
🗃 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1								
⇒ ► ► ₩ II = ⊖ * ? * k Her =	•							
main.c + X	Solution Explorer	<b>-</b> ₽ ×	Prop					
→ main.c → C:\Users\pviall\Dropbox\ECE160-MASTER\ece160-g\2019-1Spring\Projects\led\led\main.c	○ ○ ☆ ĭo - ₫  ○ ○ ☆ Io		pertie					
<b>₽</b> /*	Search Solution Explorer (Ctrl+:)	- م	S.					
* led.c	Solution 'led' (1 project)							
	<ul> <li>Jodalon ica (i project)</li> <li>Ied</li> </ul>							
* Created: 2/28/2019 10:01:07 AM	📴 Dependencies							
* Author : pviali	Output Files							
	Libraries     G main c							
tinclude (avn/io h)	• mane							
□int main(void)								
ſ								
/* Replace with your application code */								
while (1)								
{								
}								
}								
· · · · · · · · · · · · · · · · · · ·								
Output · · · · ×								
Error List Task List Output								
Ready								

6. Alter the code the Atmel puts in the window, and substitute the code below. You may increase or decrease the size of the font by using ctrl/scroll wheel.

Atmega-lab1-led - AtmelStudio         Elle       Édit       View       Project       Build       Debug       Iools       Help         Image: I	Standard Mode	Quick Launch (Ctrl+Q)	P _ 8	X
<pre>number ' ' gudy' gudygenergy</pre>		VA View VA Outline No Col 16	tifications Solution Expl	operties
Ready	Ln 8	Col 16	Cn 16	INS

7. PRIOR to running your code, you must set up Atmel Studio to use the debugger in the ATmega 328PB board, as well as insure the proper include file is referenced. Follow steps 7A-7D to do this.

## You must have your ATmega328PB board plugged in before doing these steps.

7A. Click Project > led Properties... (or type Alt+F7).

🥉 led - AtmelStud	io				
File Edit View	Project Build Debug Tools Help				
) 🕘 🐮 🖕 🖬 🔮	ASF Wizard				
⇒ ► ► № I	Set as StartUp Project				
main.c 🔹 🗙 ATme	F7				

This should take you to the properties dialog box:

7B. In the left pane, click on Tool. Click on the Selected debugger/programmer drop-down box. One of the entries should be of the form mEDBG\ATML252330418xxxxxxx, where the xxxxxxx matches the serial number on the bottom of your Atmel board. Select this option.

Build	Configuration:	N/A v	Platform: N/A	~
Build Events				
Toolchain				
Device	Selected deb	nugger/programmer		
ТооІ		v statistics and stat		
Packs				
Advanced Sim	nulator			
mE ATM	DBG ML2523041800002703	ittings		
mE mEC	DBG DBG CMSIS-DAP	hip Y		
Cus	stom Programming To	PROM		
	Debug settir Keep tin  Cache a	ngs ners running in stop mode Il flash memory except		

7C. When mEDBG ATML25230418xxxxxx is selected, an Interface drop down box will appear to the right. Select debugWire from the Interface option.

Build Build Events	Configuration: N/A · Platform: N/A · ·
Toolchain Device Tool* Packs Advanced	Selected debugger/programmer mEDBG • ATML2523041800002703 v Interface: debugWIRE v ISP
	idebugWIREi         Programming settings         Erase only program area ▼         ✓ Preserve EEPROM         Note: The EESAVE fuse must be configured manually to preserve EEPROM, since it cannot be configured automatically when using debugWIRE.
	Debug settings          Image: Contract of the set of t

8. To download and run the project, press F5 or click on **b** to download and run the project in debugging mode.

It is possible that (if your board is brand new), you will receive a message box of the form:



If you get this message, CAREFULLY unplug the daughter board from the 328PB, clicking "Yes", and download again if needed. This should only need to be done when the board is brand new.

9. If all goes well, you should see the blue LED blinking on the tri-color LED.

10. Atmel studio may also single step your code just as you may with Visual Studio. Set a breakpoint at a line of code by clicking the mouse where the red arrow points:



You should see a red circle appear. This indicates a "breakpoint" is set. The debugger will stop execution of the code when this line is ABOUT to be executed. Again, this is exactly like Visual Studio, so it should be very familiar.

11. You may press the M in the tool bar to begin execution and break at the breakpoint (or Debug/Start debugging and break, which will break at the first statement). You will see a yellow arrow (possibly on top of the red circle, depending on which you do).



The yellow arrow indicates that the DDRD = DDRD | 1 << 6; is about to be executed. Note also the left top pane contains the PC, flags, registers X, Y, and Z (which we have yet to learn about), and R00, R01, etc. Under the code window is the memory window, in which one may select prog FLASH, EEPROM, RAM or other memory to display. In the upper left is a I/O Port pane. Note that PORTD is one of the I/O ports. The pane that is of most interest to C programmers is the Watch 1 window. In this window, you may enter variable names, and see the current value displayed. You may also change the value of a variable.

## **Problem Statement**

For this lab, you are to write code which will cycle the LED through various color combinations: Blue  $\rightarrow$  Cyan  $\rightarrow$  Green  $\rightarrow$  Yellow  $\rightarrow$  Red  $\rightarrow$  Magenta  $\rightarrow$  Blue, and it should repeat as long as power is applied. The tri-color LED is connected to the microcontroller as follows:

Bit	LED
D6	RED
B1	GREEN
B3	BLUE
C3	Common Anode (+)

The LED is common anode, so a "1" must be applied to the anode, and a "0" to the pin(s) of the LED(s) that you wish to turn on.

If you are not familiar with tri color LED's and how multiple segments may be turned on to produce secondary colors, the following color chart may be useful. Turning on all three LED's produce white.



Enter your code into your project. You may initially ignore using \_delay\_ms(), and step through the program (or not, your choice). Prior to demonstrating the program to the TA, you must add delays of about 500ms to your code, so each color may be seen. <u>Be sure to show the working program to the TA</u>.

Submitting the project:

Atmel Studio names the main program main.c (your project is named led, but the c program is named main.c). Submitting each program with this name will cause overwriting of projects. Hence, the procedure to submit is as follows: (1) copy the main.c file into your M:\ECE-160\xxx folder. (2) rename the main.c file to led\_main.c in your M:\ECE-160\xxx folder. DO NOT change the file name in the project. The submit time of the .c file will be considered the time it is turned in.

Prior to (or within 3 school days of) turning the .c file in, you must have a sign off from the TA. The TA will give you a grade for the project, and record that grade. The signoff below is a verification. This signoff does NOT need to be turned in.

NAME:					
TA Signature:					
Grade	Checksum:				