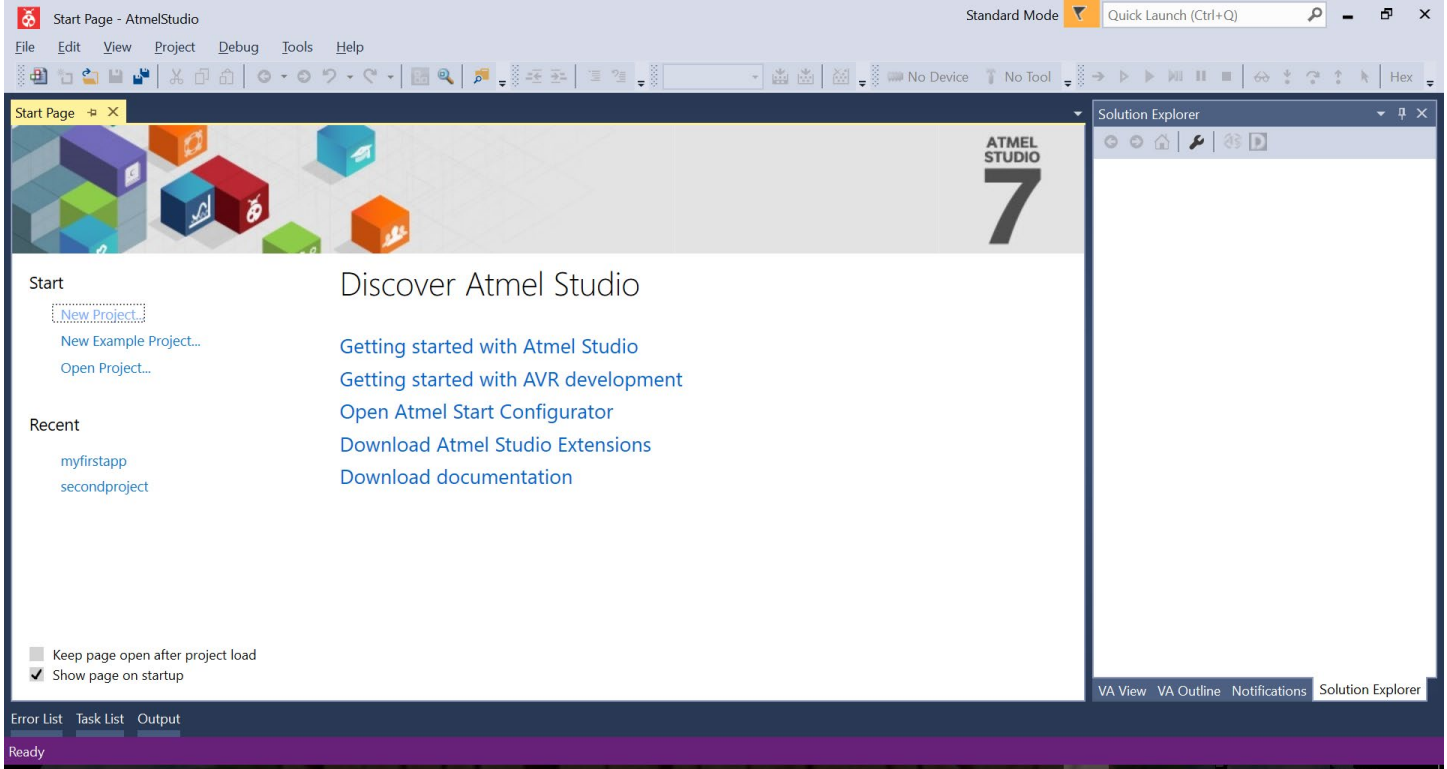


University of Massachusetts Dartmouth
Department of Electrical and Computer Engineering

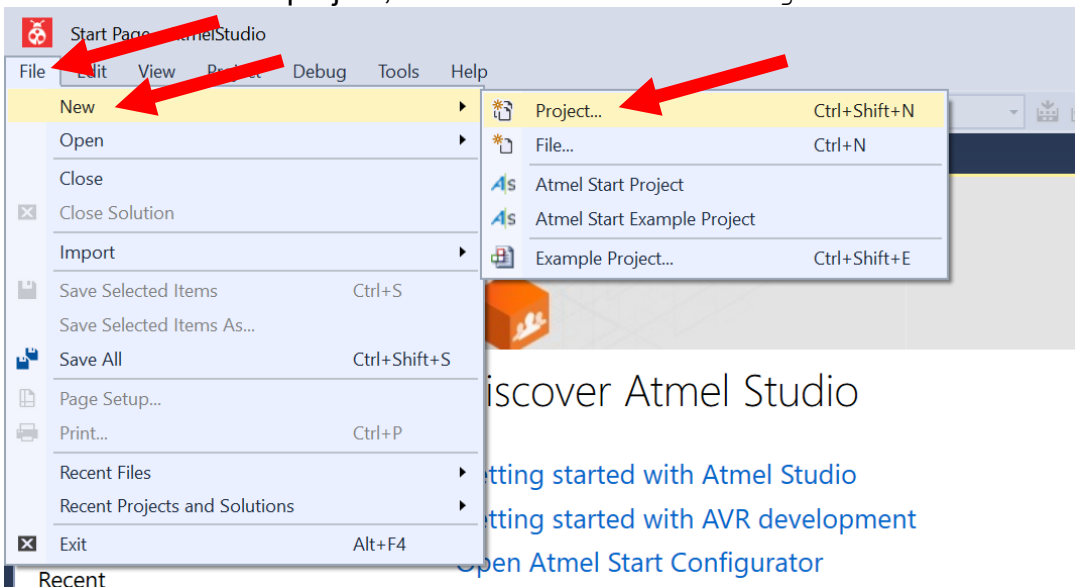
ECE 263 VERSION 2
Atmel Lab 1 – First program in Atmel Studio 7.0

Lab name: led_main.c
due: see <http://ece160.org>

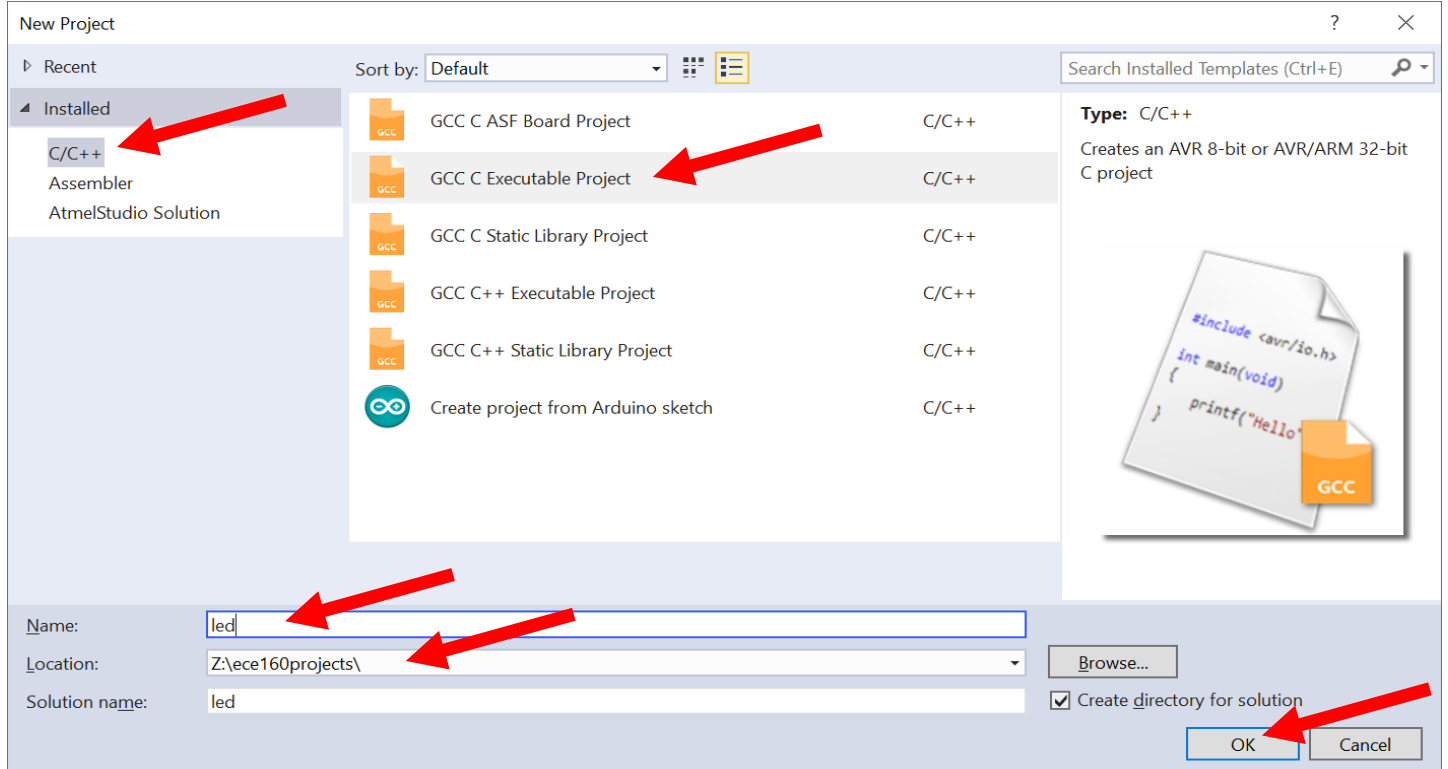
1. Launch Atmel Studio by clicking on the desktop icon. You will see the screen below. It is possible you will receive a dialog box notifying you there are updates. If you do, just cancel out of this box since you do not have sufficient privileges to install updates.



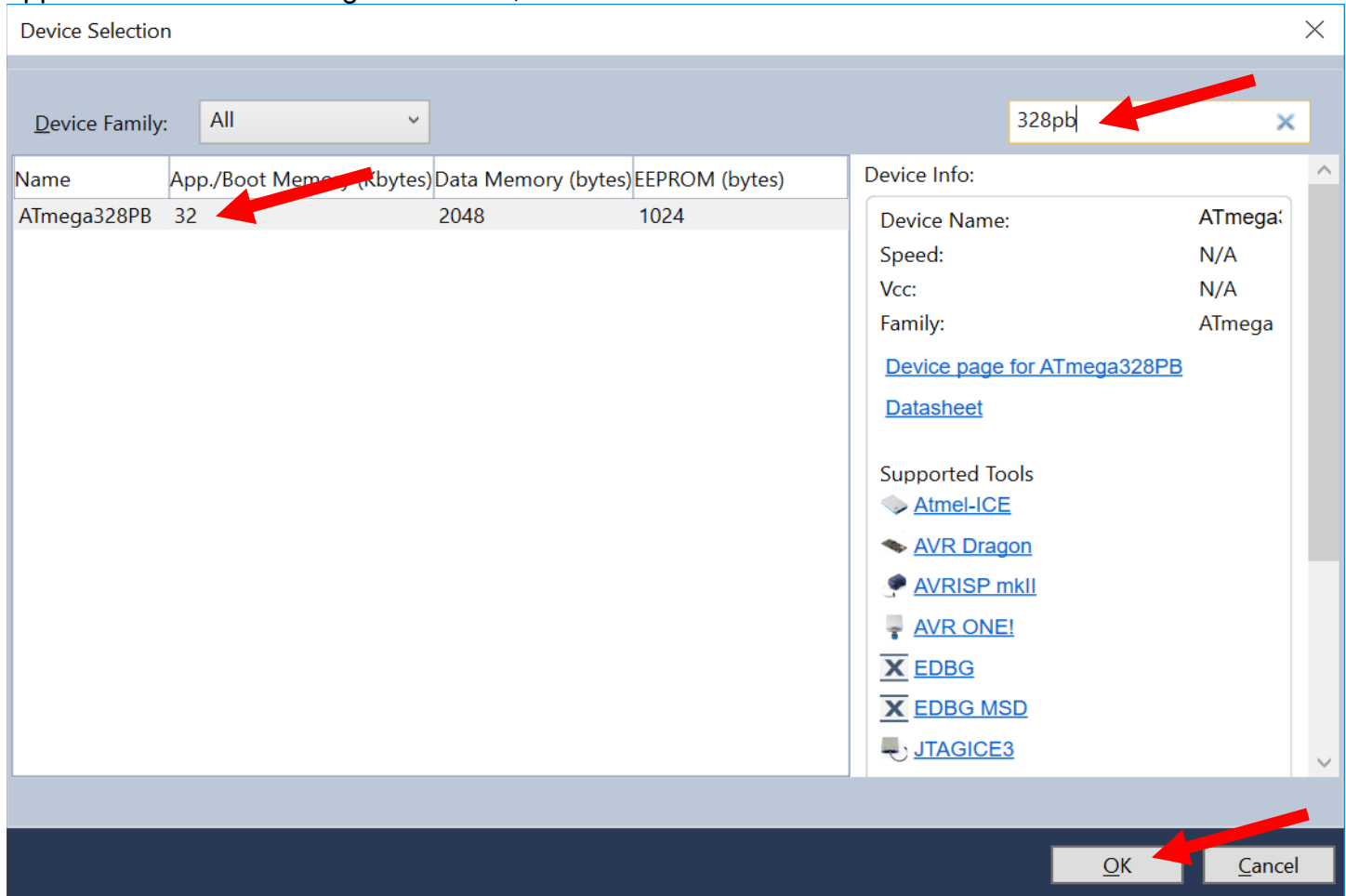
2. To create a new project, click File > New > Project



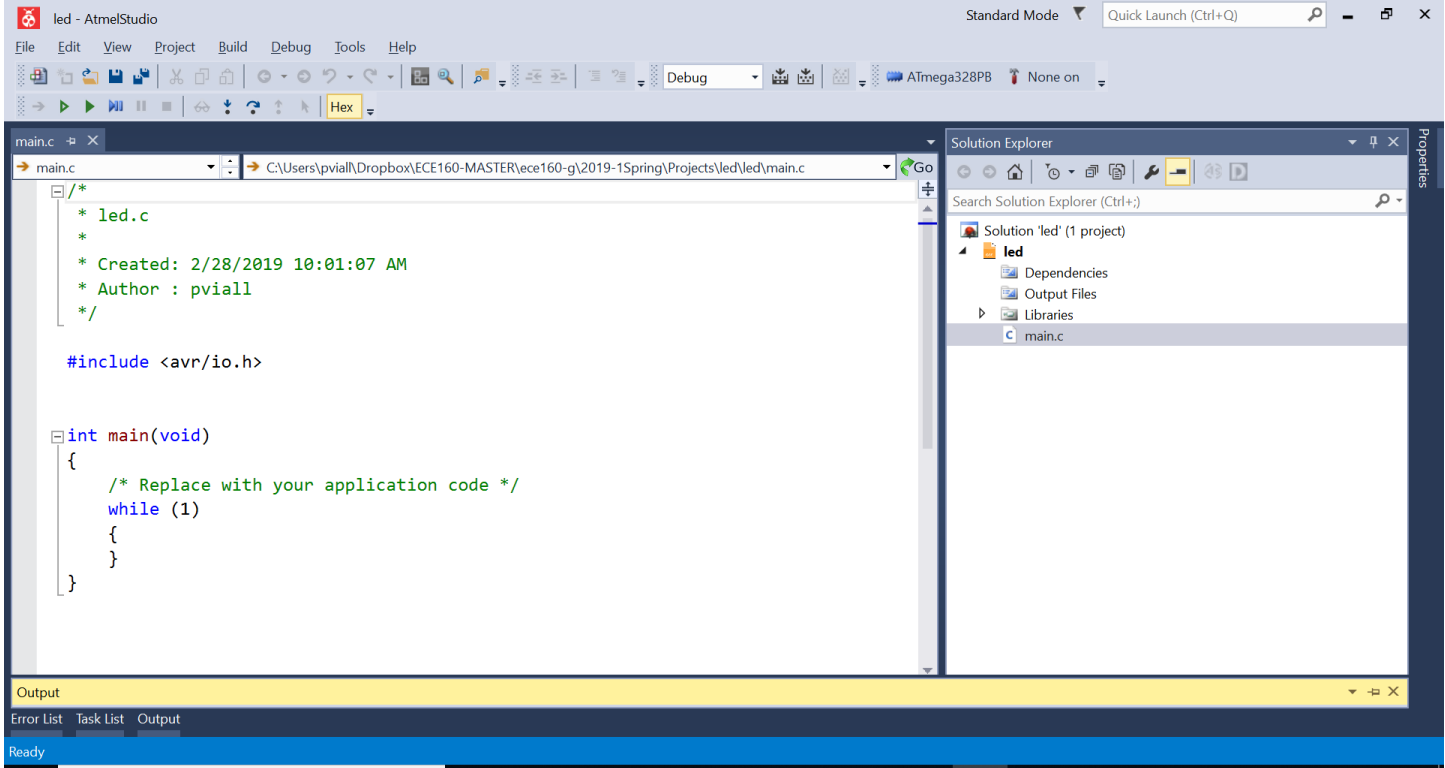
3. In the **New Project** window, click **C/C++** in the left pane. Click **GCC C Executable** in the middle pane. Next change the **Name** to something meaningful (such as **led** for the first lab). If you wish to change from the default **Location**, (suggested: drive **Z:**), you may do that as well.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



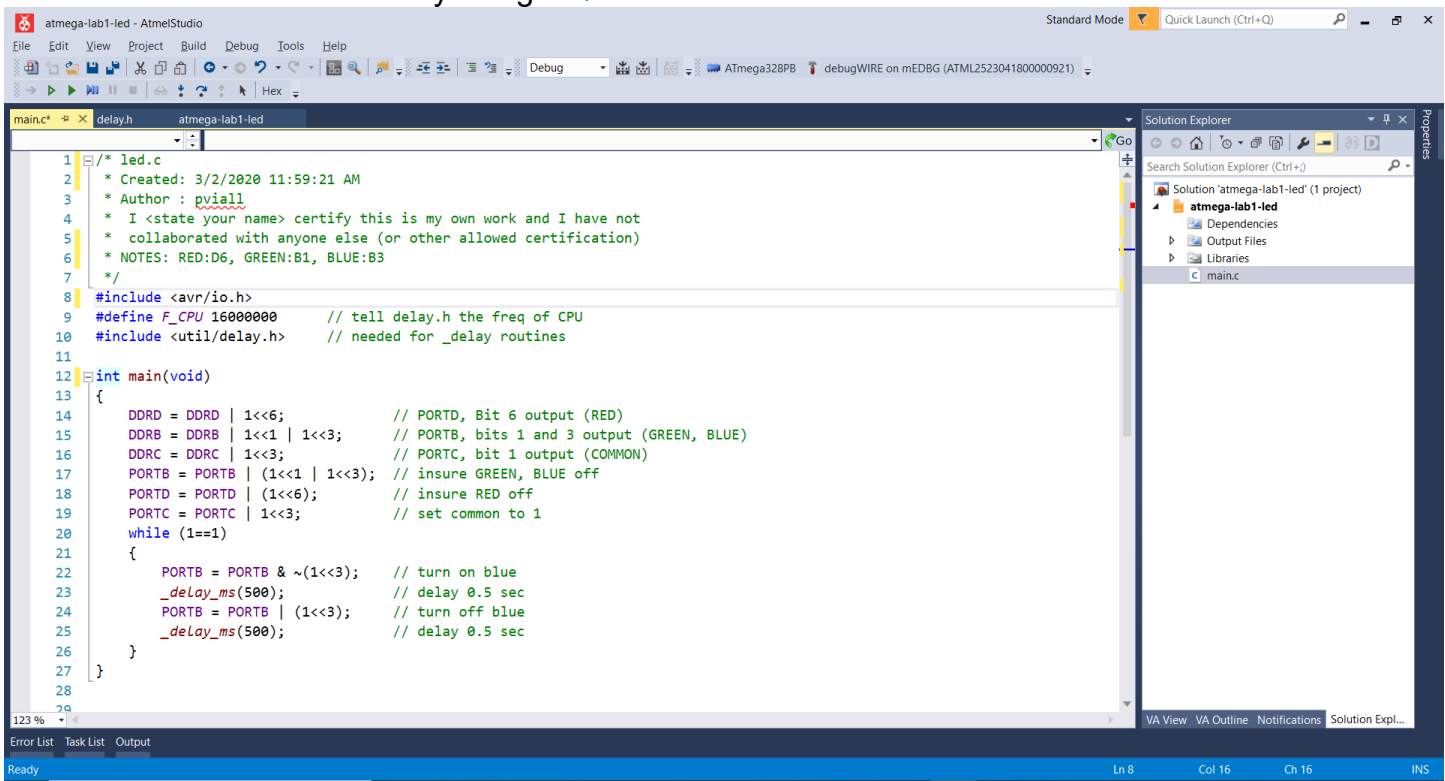
4. In the search box, type **328pb** (upper or lower case doesn't matter). **Atmega328pb** should appear in the device listing. Click on it, then click **OK**.



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.



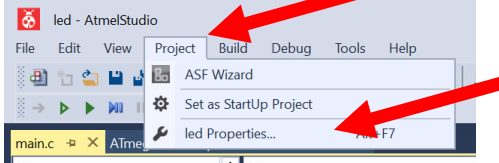
6. Alter the code the Atmel puts in the window, and substitute the code below. You may increase or decrease the size of the font by using ctrl/scroll wheel.



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.

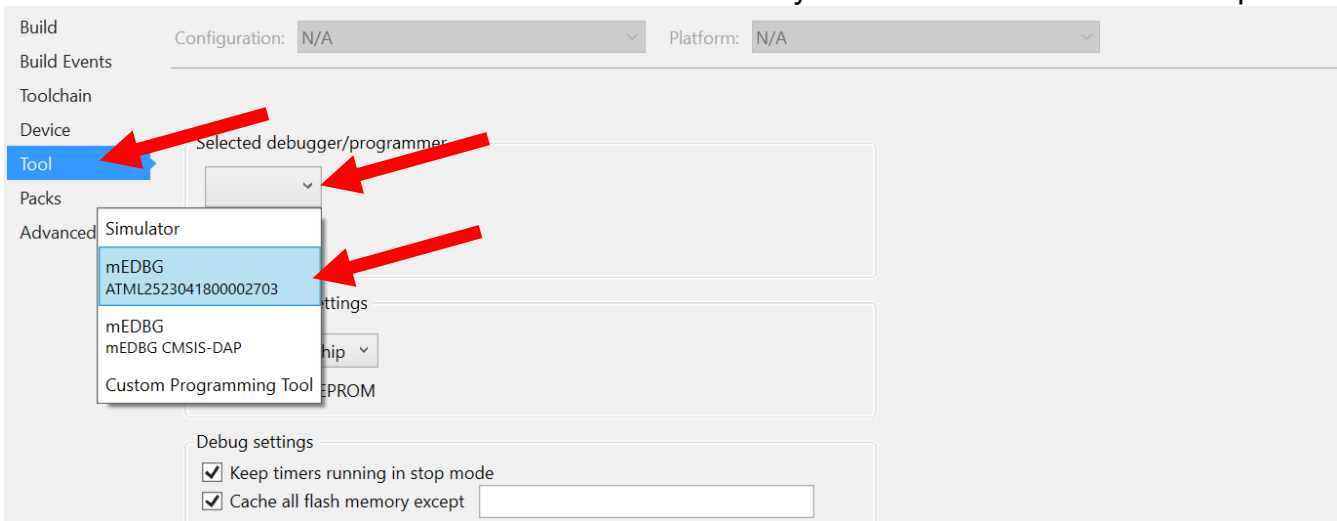
You must have your ATmega328PB board plugged in before doing these steps.

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

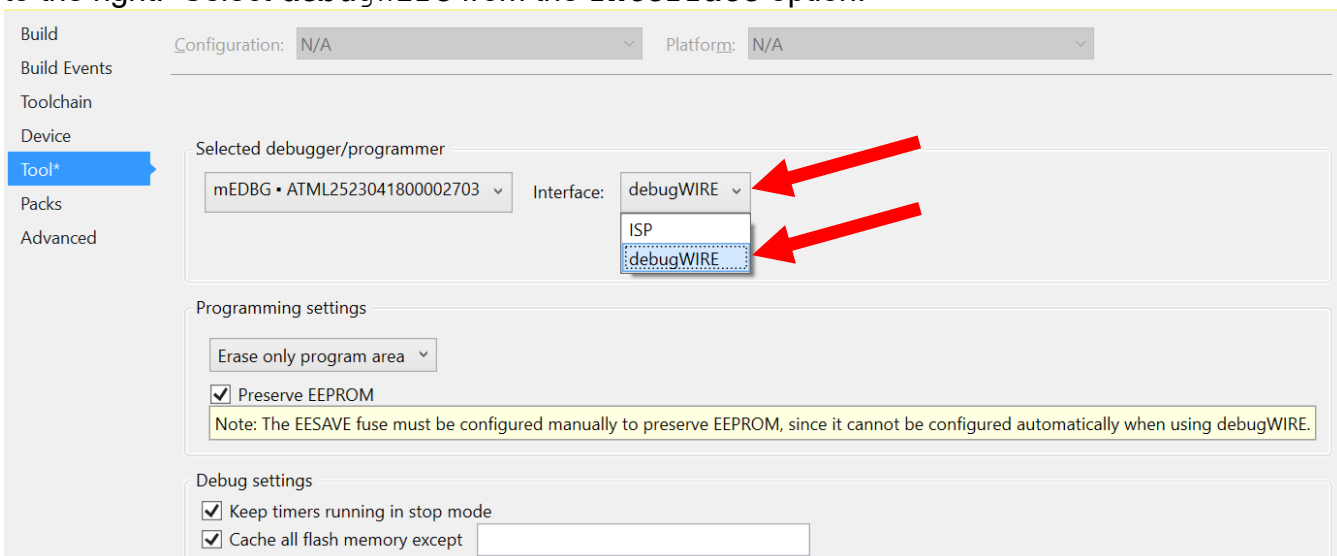



This should take you to the properties dialog box:

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\ATML25230418xxxxxxx, where the xxxxxxxx matches the serial number on the bottom of your Atmel board. Select this option.

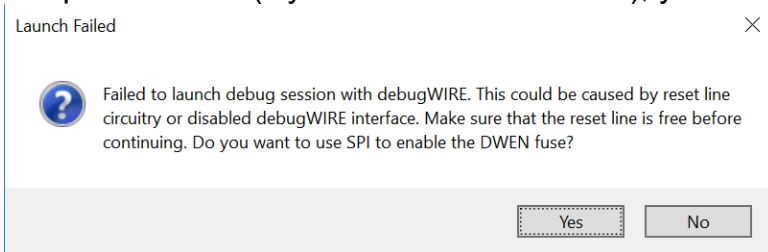


7C. When mEDBG ATML25230418xxxxxxx is selected, an Interface drop down box will appear to the right. Select debugWIRE from the Interface option.



8. To download and run the project, press F5 or click on  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.

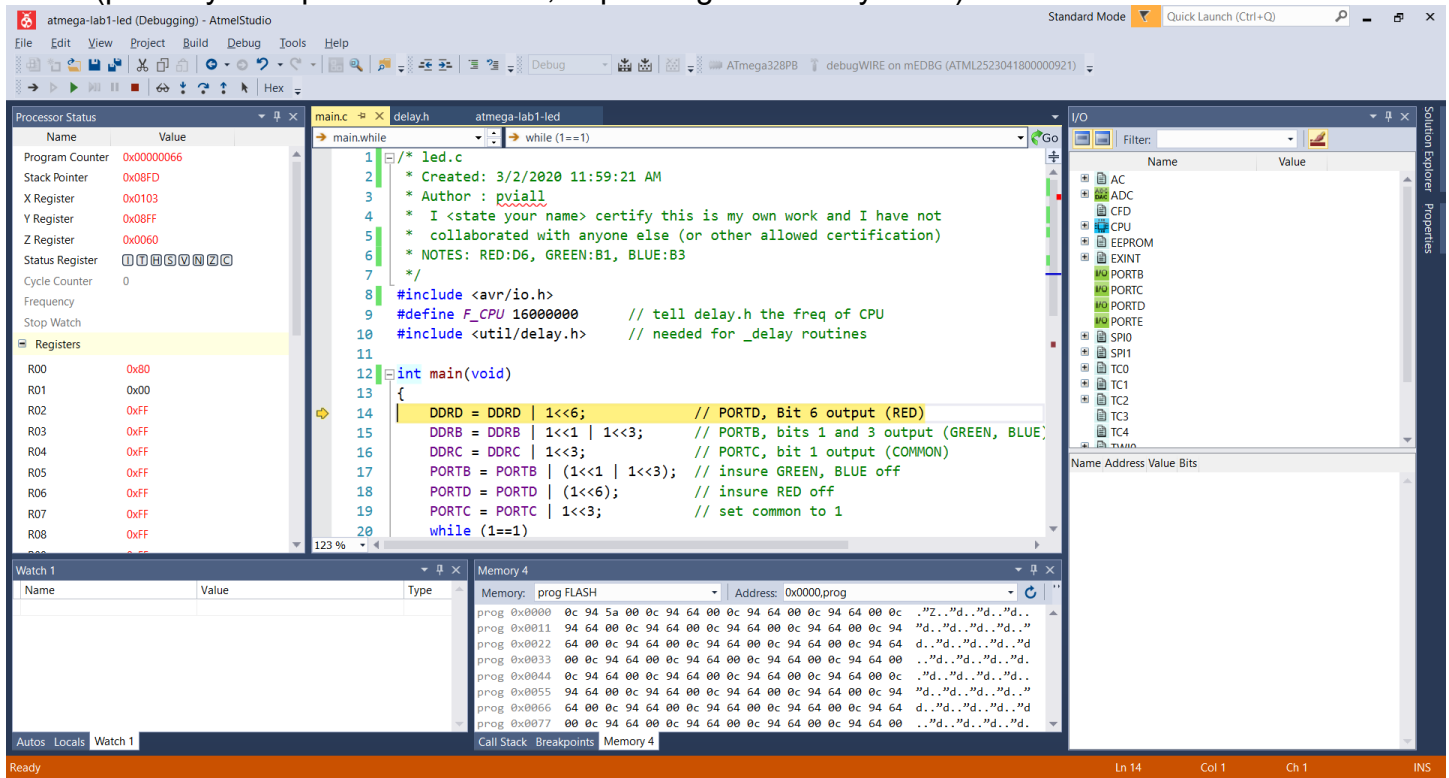
9. If all goes well, you should see the blue LED blinking on the tri-color LED.

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:

```
18 PORTD = PORTD | (1<<6); // insure RED off
19 PORTC = PORTC | 1<<3; // set common to 1
20 while (1==1)
21 {
22     PORTB = PORTB & ~(1<<3); // turn on blue
23     _delay_ms(500); // delay 0.5 sec
24     PORTB = PORTB | (1<<3); // turn off blue
25     _delay_ms(500); // delay 0.5 sec
26 }
27
28
```

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  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.

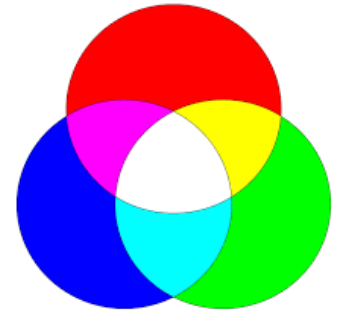
Problem Statement

For this lab, you are to write code which will cycle the LED through various color combinations: Blue → Cyan → Green → Yellow → Red → Magenta → 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. Be sure to show the working program to the TA.

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: _____