

TECH 3230

Lab #1

First Atmel / Microchip Studio C Program

Purpose: To familiarize students on setting up an Atmel / Microchip studio project, create a C program and run the program on an Arduino Uno board.

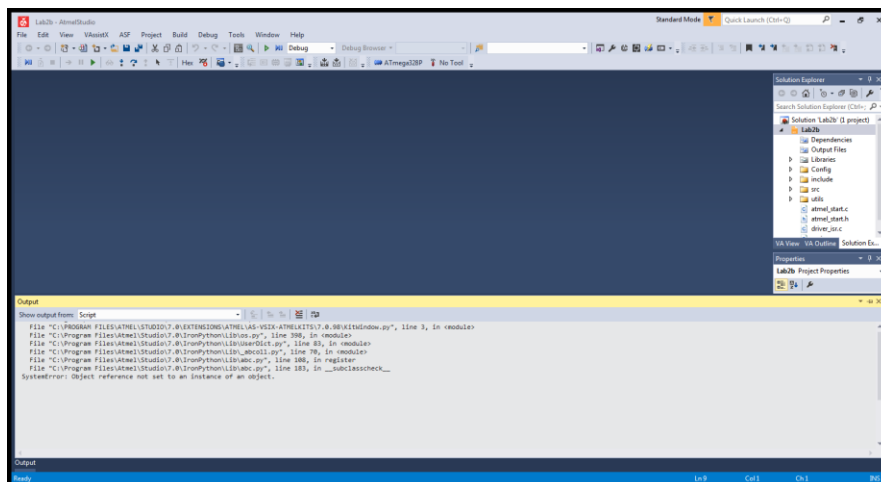
Procedure:

1. Creating the Project

- Open Atmel Studio 7.0 and do FILE | NEW | Atmel Start Project. This will start a project wizard. Select the IC that is on the Arduino Uno Board (atmega328p-pu), and hit the "Create New Project" button. (you can configure some hardware on this page before hitting the button, but for this lab we do not need to do so).
- On the next screen we can do more configuration. We do need to change one item, since the IC defaults to 8MHz and the Arduino board uses a 16MHz Crystal.

To do this click on the Clock Icon . On the screen look for the Input Clock box and hit the settings icon within the box . Now change the value of the "Input Clock Frequency" to 16000000 (16MHz) and click close. Now hit "Generate Project" on the bottom of the window.

- A screen will pop up. Give the project the name "Lab1". If you wish to save the file to your thumb drive, change the location now by browsing to the directory you wish to use. This will create the basic structure of your project within a directory called Lab1. Within that directory will be a file called Lab1.atsh which is the project file.
- You should now have the following screen:



Off to the right you should have a "Solution Explorer" window. If you do not you can click VIEW | Solution Explorer to open it. This shows you all the files associated with the project. The one we are most interested in is within utils and it is called **main.c**. Double click on this to open the file.

2. Write the program that meets the following requirements

- The program should print values 1,2,3,4,5,6,7,9,10 to the terminal using printf.
- A LOOP must be used and the values printed from a variable.
- An IF structure must be used (to skip the 8 or to only print the other values.
- At the end of the program (before the close }) put "while(1);". This is an infinite loop....since embedded systems do NOT have an operating system to return to, we will just put the program in an infinite loop.

3. Compile the program

- Once you are finished typing, you need to compile the C program into something the IC will understand (Machine Language). To do this, go to BUILD | BUILD SOLUTION. The output window should end with a message saying "Build succeeded" if not, check your code.

4. Send the code to the Arduino Uno Board

- Since the code is written for the Arduino, we must send the code over to that IC before we can test it. In industry, IC's like the Atmega328p-pu are programmed out of circuit using a programmer, but since we are using the ATmega on an Arduino board, we are going to use the USB on the Arduino to send the code to it. To do this we must add an external tool to Atmel Studio¹.
- First Plug in the Arduino Uno to the USB of the computer.
- Find the COM port of the device
 - Win10 (opt #1) – right click on Windows Start Icon and select Device Manager and look under Ports (COM & LPT)
 - Win10 (Opt #2 - Device Manager Locked) – search for "PowerShell" and open. Type in "Get-WMIObject Win32_SerialPort" (no quotes) and check results for Arduino Uno (Com __)
 - Win 7 – use Start | Control Panel | Device Manager and look under Ports (COM & LPT)

¹ <http://www.instructables.com/id/How-to-Load-Programs-to-an-Arduino-UNO-From-Atmel-/>

Note which COM port it is (you will need that info in a bit)

- Back in Atmel Studio, click on Tools | External Tools and click the Add button and fill out the form as follows:

- Title: Arduino Uno

- Command*: (click on the  icon and browse to C:\Program Files (x86)\Arduino\ then use the search box to find avrdude.exe and double click on the found file)

- Copy the path up to avr\ (highlight and control C).

- Arguments*: type in the following line:

```
-C "[ctrl-v]etc\avrdude.conf" -p atmega328p -c arduino -P  
COM7 -b 115200 -U  
flash:w:"$(ProjectDir)Debug\$(TargetName).hex":i"
```

Replace [ctrl-v] with the path from the previous step. Replace COM7 with COM Port found in step 7b. (It is typed as one continuous line)

- Uncheck "Close on Exit" and check "Use Output Window"
- Click OK
- Now go back to TOOLS and there should be a new menu item called Arduino Uno. When you click on this, it should send the code to the Arduino board. If successful, the last line in the output window should be "avrdude.exe done. Thank you."

5. Open Putty

- Since the Arduino does not have a screen for output in this lab, we will be using a terminal emulation program called Putty. If it is not already on your computer, it can be downloaded from <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>. Once you open it, you will need to select Serial and change the "Serial Line" to the COM Port found above. Then select open. This will reset the program back to the start and show the output.
- Did you get the correct output. If NO ask for help!
- If it worked **Demo to the instructor**

6. Submit the project in .ZIP format

- Close Atmel Studio (or at minimum close the project) so no files are locked.
- Open File Explorer and navigate to your project folder, e.g.: Documents\Atmel Studio\7.0\Lab1\
- Right-click the project folder itself (not just individual files) and choose:
- Send to → Compressed (zipped) folder (Windows built-in)
- Verify your zip file before submitting:
 - Extract it to a temporary location and confirm it contains:
 1. main.c
 2. The .cproj (or .atsln) project file
 3. The Debug folder containing your .hex file
- Upload to the course submission portal

Do not submit just the main.c file by itself — the instructor needs the full project folder (including the .hex file) to verify your build.