Getting Started

See: http://www.cs.ou.edu/~fagg/classes/general/atmel/

Summary:

- Install compiler
- Download your subversion tree
  - Today: work in "testproject"
- Plug the programmer into your computer
- Plug the programmer into the Arduino board
- Create a program

Subversion

Similar to “Dropbox”: allows you to easily share a folder across multiple computers

Subversion

Key commands:

- **Checkout**: get initial copy of the shared folder
- **Add**: mark a file or a folder as shared
  - Only share necessary files: .c, .h, makefile, .ppt, .pptx, .avrsln, .avrsuo
  - Don't share: .o, .hex
- **Update**: copy changes to the folder down to your computer
- **Commit**: copy your changes to the folder up to the server

Subversion

When you sit down to work:

- It is best if you are the only one editing a particular file (so coordinate with your group members)
- Perform an update
- Make your changes (until you are happy)
- Add any new files
- Commit your changes:
  - Always remember to do this when you are done

Subversion

Conflicts occur when two people edit the same file & then try to check in their changes

- The second person to commit will end up with several versions of the file in their folder:
  - A file with the two sets of changes (with changes clearly marked)
  - A file each that corresponds to the changes made by one individual
- The second person must select one, copy it over to the original file name, make any necessary changes, and commit again

Downloads from Atmel HOWTO

Already in your subversion tree:

- lib/libou_atmega2560.a
- include/oulib.h
- include/oulib_serial_buffered.h
- testproject/makefile
- project1/makefile

For Unix users (also in your tree):

- makefile

Compiling and Downloading (the Unix way)

- Makefile:
  - Modify the “TARGET” and “OULIB_DIR” lines for your program
- Type “make”
  - You should see no errors
- Type “make program”
  - This will download your code to the processor
  - Again, you should see no errors

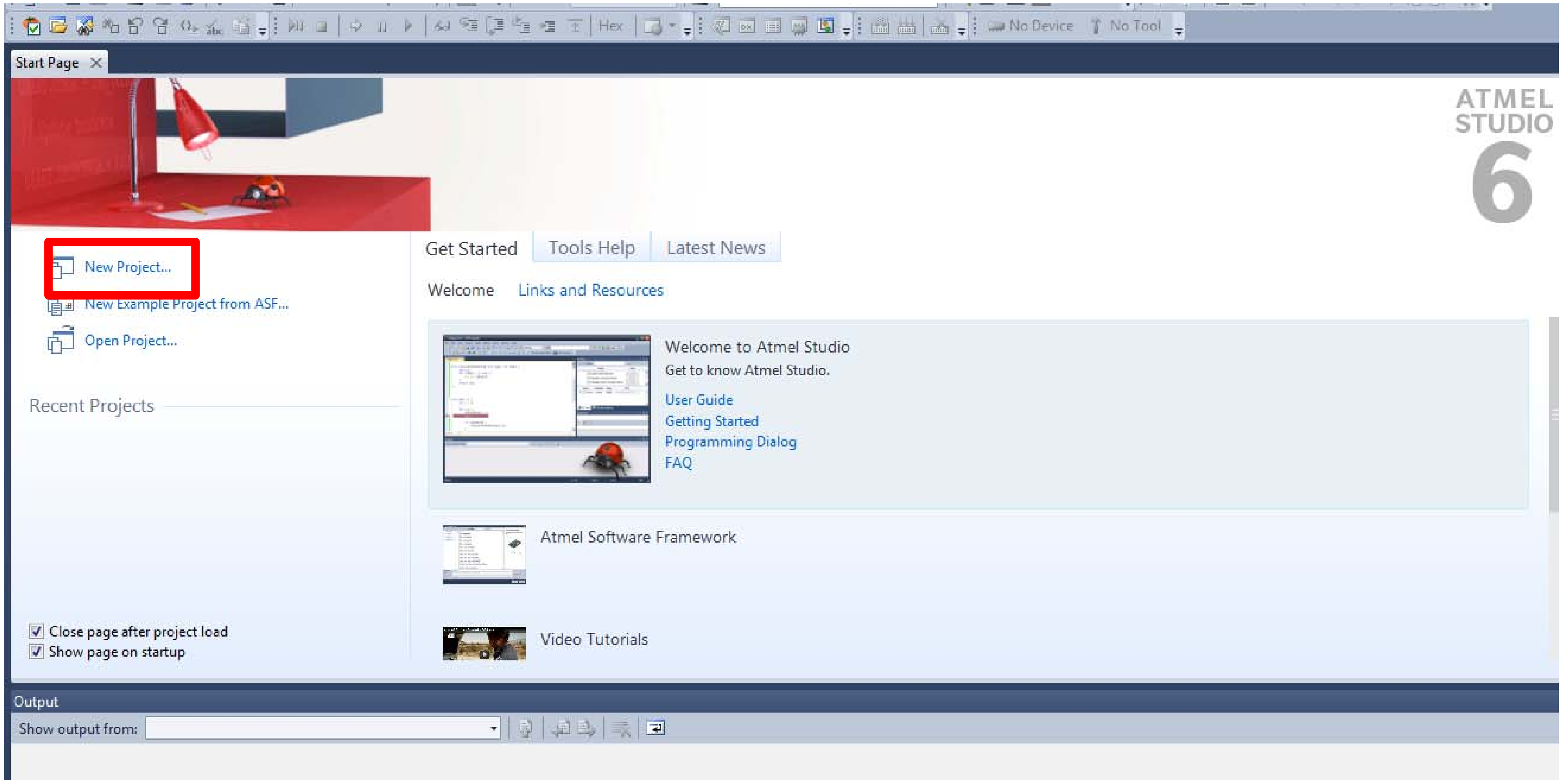
Plan for Today

- Start working through exercise 1
  - All group members must show some form of LED control
  - Groups need to show some wiring of additional LEDs
- Project 1

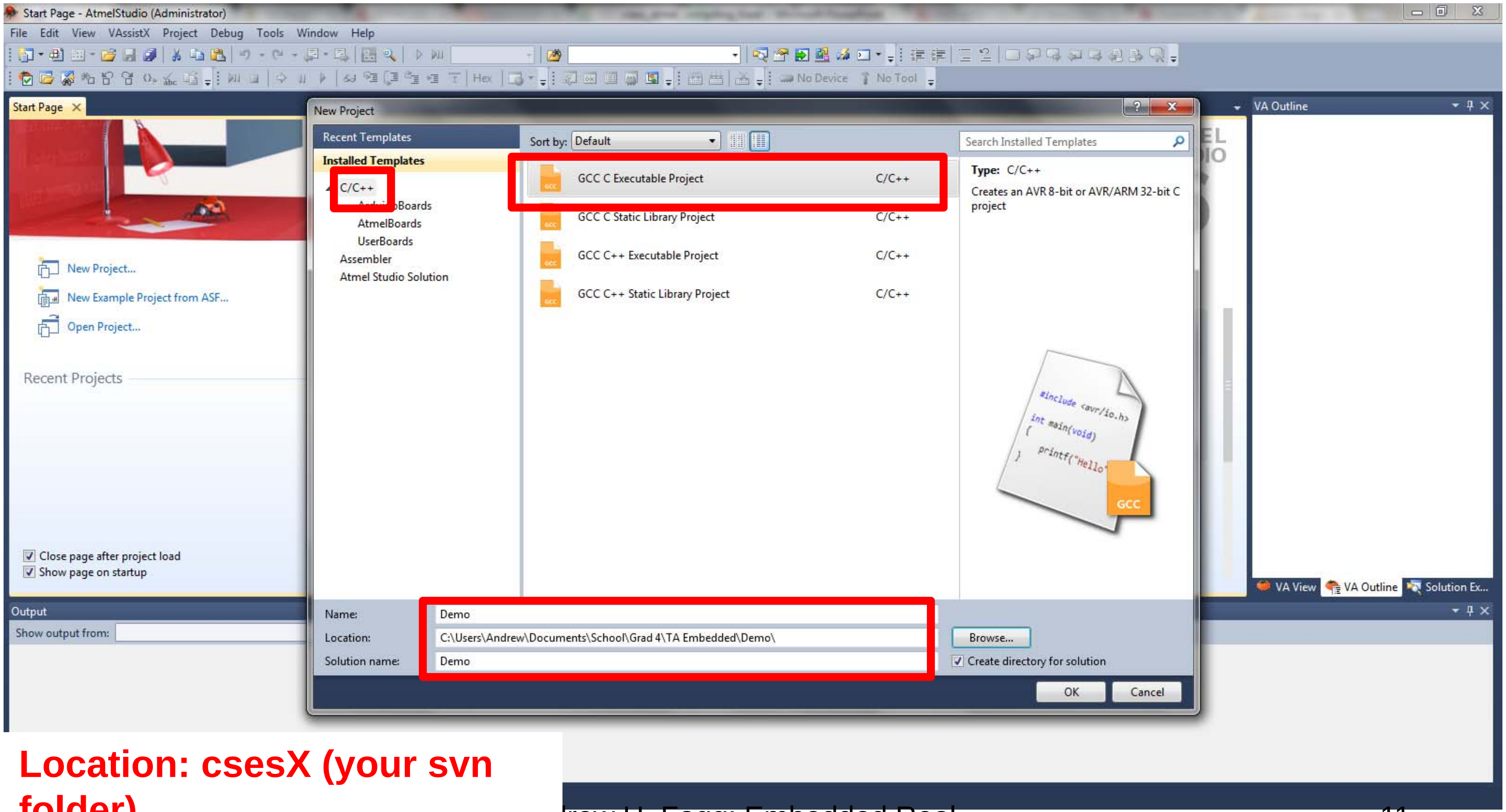
Everyone must demonstrate:

- Svn works
- Compiling/downloading to Atmel works

Windows: Getting Started



New Project



Installed Templates:

- C/C++
  - ArduinoBoards
  - AtmelBoards
  - UserBoards
- Assembler
- Atmel Studio Solution

GCC C Executable Project	C/C++
GCC C Static Library Project	C/C++
GCC C++ Executable Project	C/C++
GCC C++ Static Library Project	C/C++

Type: C/C++

Creates an AVR 8-bit or AVR/ARM 32-bit C project

```
#include <avr/io.h>
int main(void)
{
    printf("Hello"
}
```

Name: Demo

Location: C:\Users\Andrew\Documents\School\Grad 4\TA Embedded\Demo\

Solution name: Demo

[✓] Create directory for solution

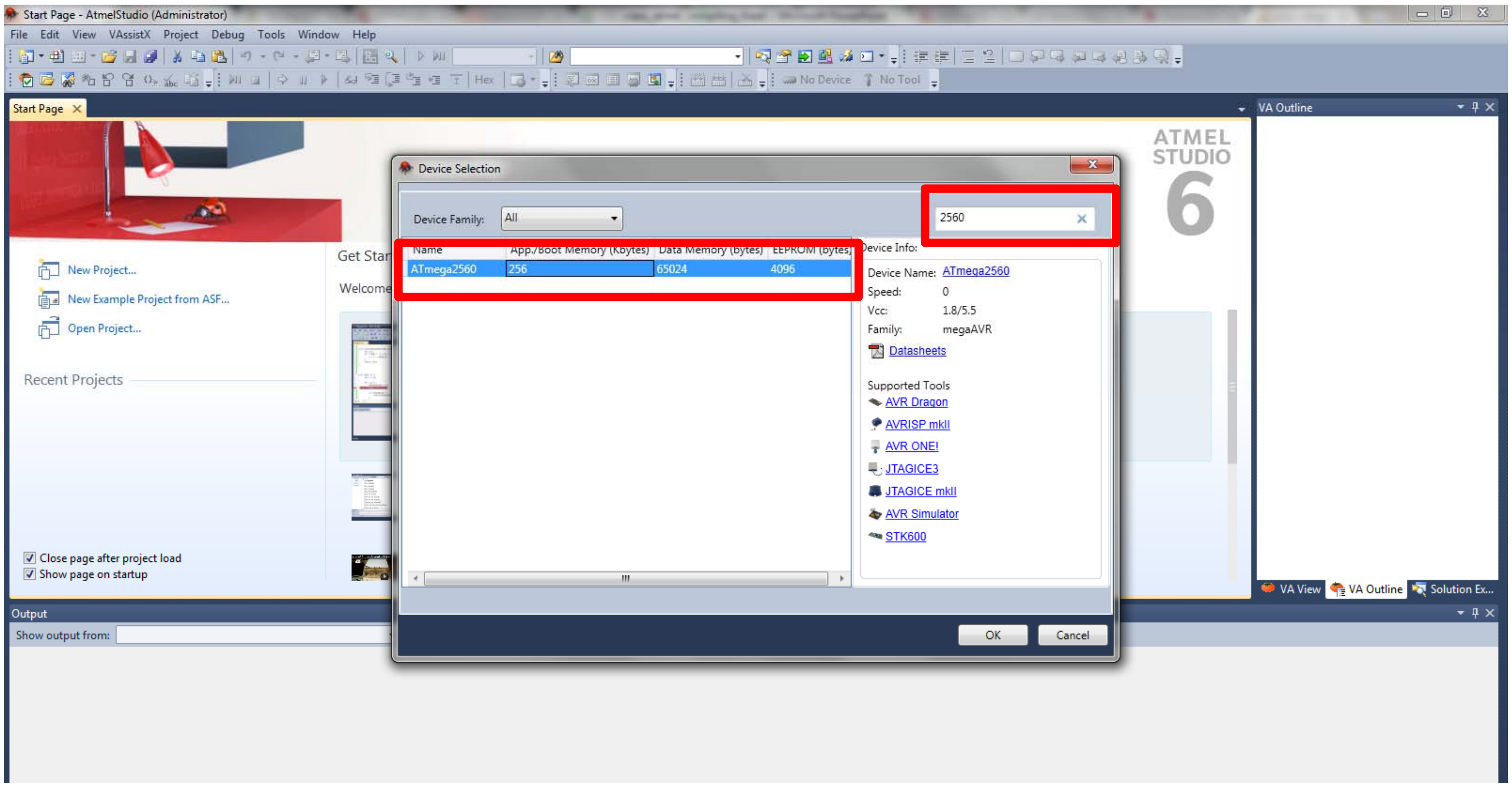
[✓] Close page after project load

[✓] Show page on startup

Location: csesX (your svn folder)

Name: testproject (for today)

Select the ATmega2560



Device Family: All

Search: 2560

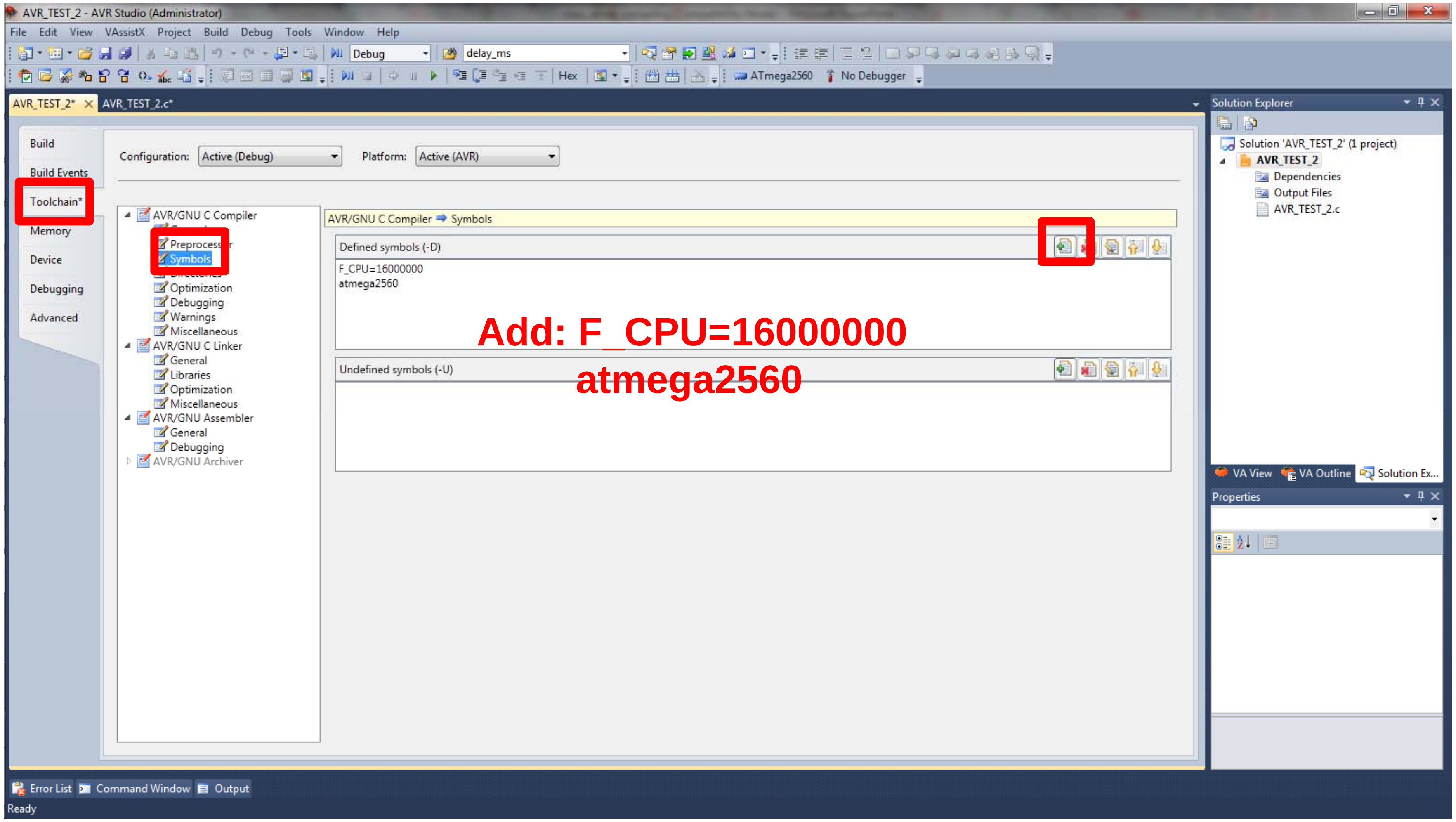
Name	App./Boot Memory (Kbytes)	Data Memory (bytes)	EEPROM (bytes)
ATmega2560	256	65024	4096

Device Info:

Device Name: ATmega2560
Speed: 0
Vcc: 1.8/5.5
Family: megaAVR
Datasheets

Supported Tools
AVR Dragon
AVRISP mkII
AVR ONE!
JTAGICE3
JTAGICE mkII
AVR Simulator
STK600

Project➔ <Project Name> Properties (Alt+F7)



Configuration: Active (Debug)
Platform: Active (AVR)

AVR/GNU C Compiler ➡ Symbols

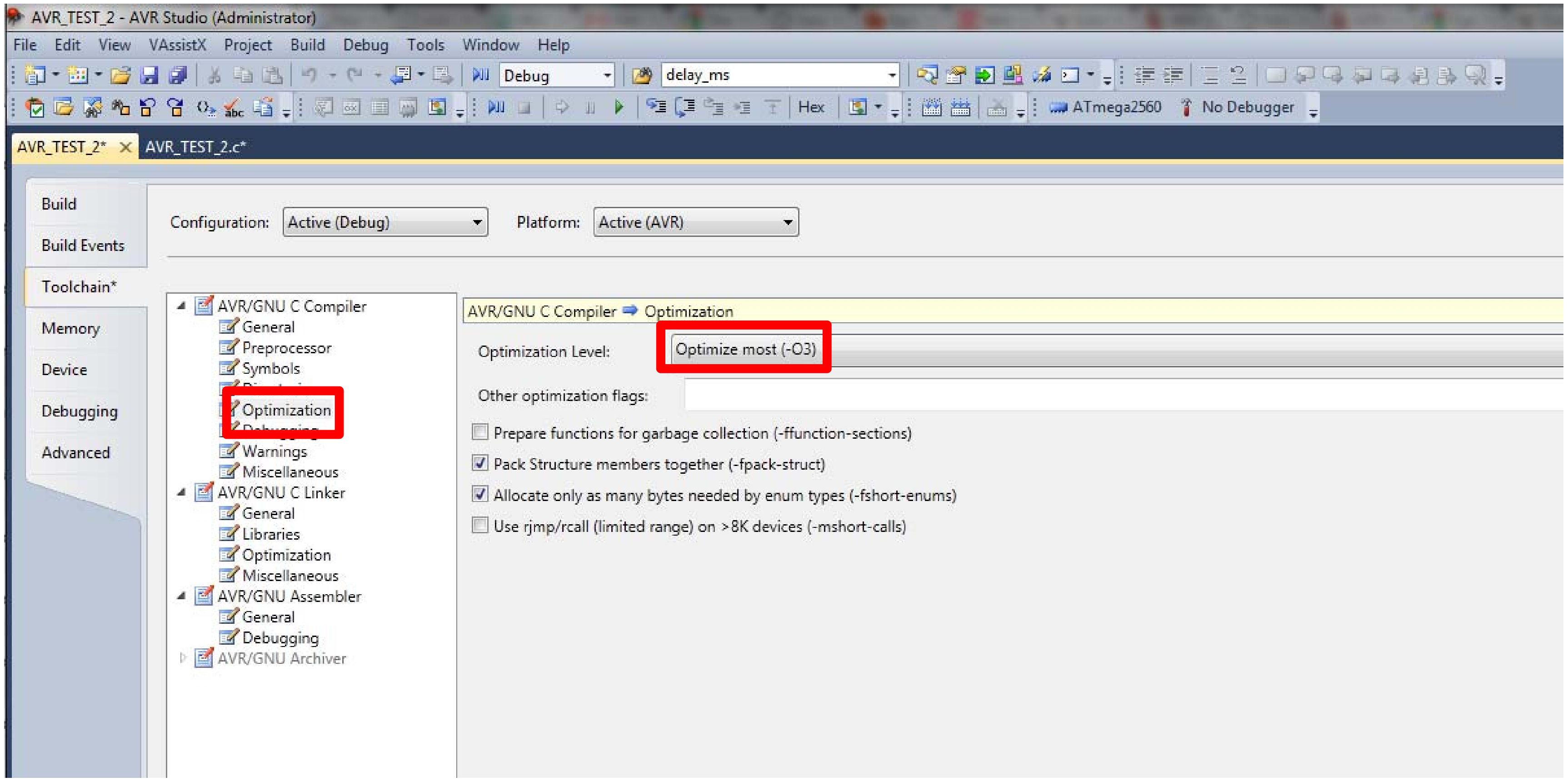
Defined symbols (-D)

```
F_CPU=16000000
atmega2560
```

Undefined symbols (-U)

Add: F_CPU=16000000
atmega2560

Compiler Optimization



Configuration: Active (Debug)   Platform: Active (AVR)

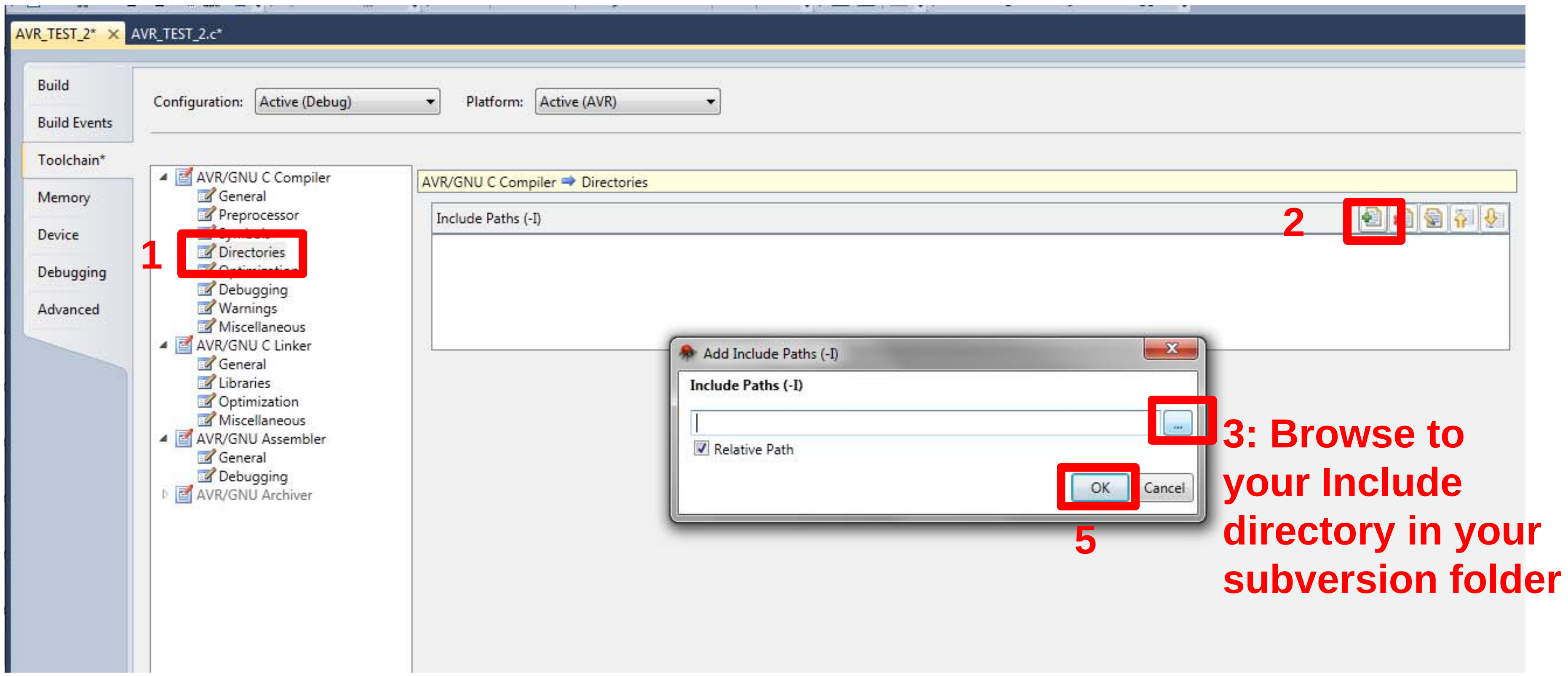
AVR/GNU C Compiler ⇒ Optimization

Optimization Level: Optimize most (-O3)

Other optimization flags:

- [ ] Prepare functions for garbage collection (-ffunction-sections)
- [x] Pack Structure members together (-fpack-struct)
- [x] Allocate only as many bytes needed by enum types (-fshort-enums)
- [ ] Use rjmp/rcall (limited range) on >8K devices (-mshort-calls)

Add Directories



Configuration: Active (Debug) Platform: Active (AVR)

AVR/GNU C Compiler ➡ Directories

Include Paths (-I)

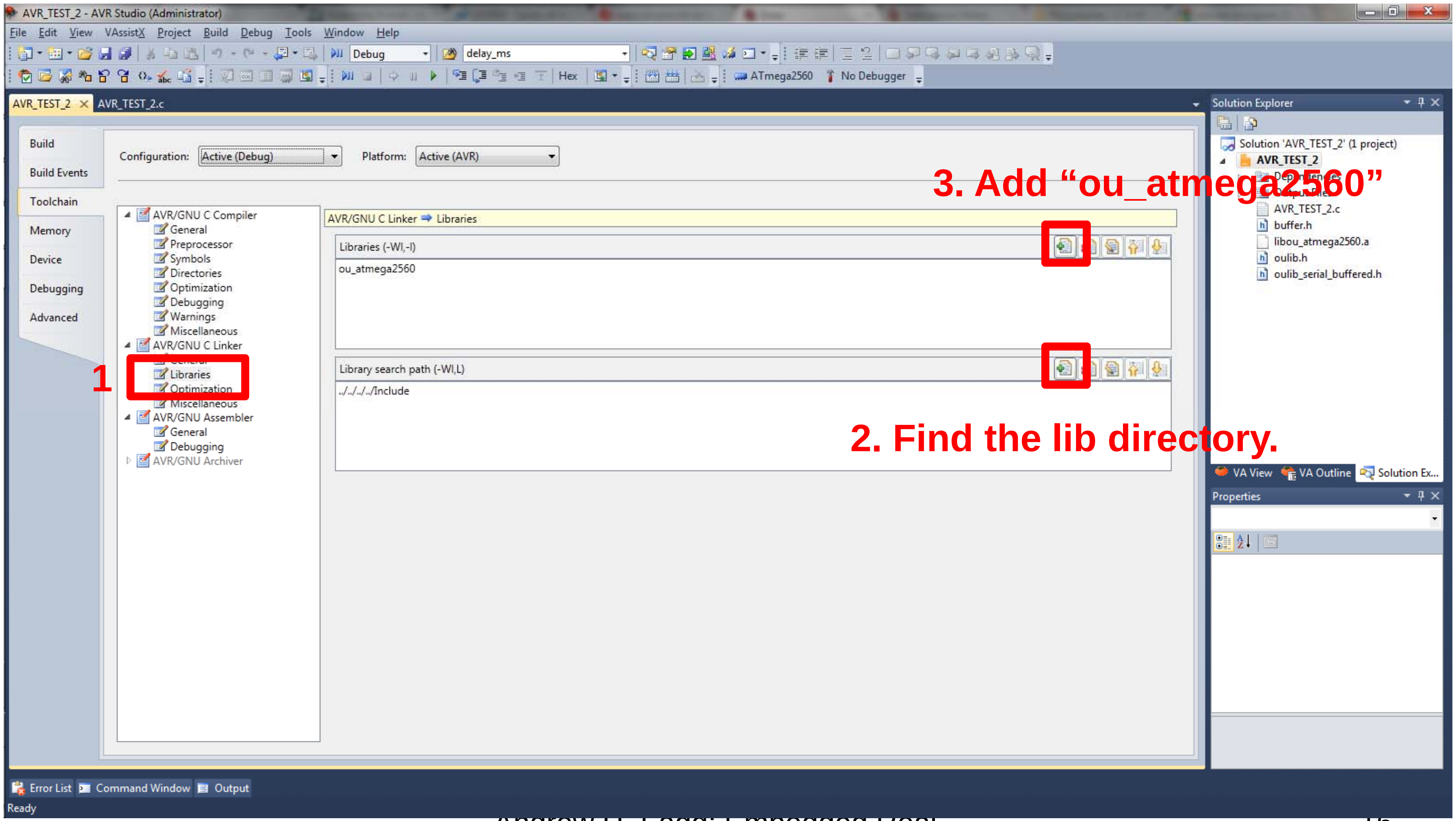
Include Paths (-I)

☑ Relative Path

3: Browse to your Include directory in your subversion folder

If relative path causes a crash, then uncheck the box

Add Libraries



AVR_TEST_2 - AVR Studio (Administrator)

Configuration: Active (Debug)   Platform: Active (AVR)

AVR/GNU C Linker ➡ Libraries

Libraries (-Wl,-l)

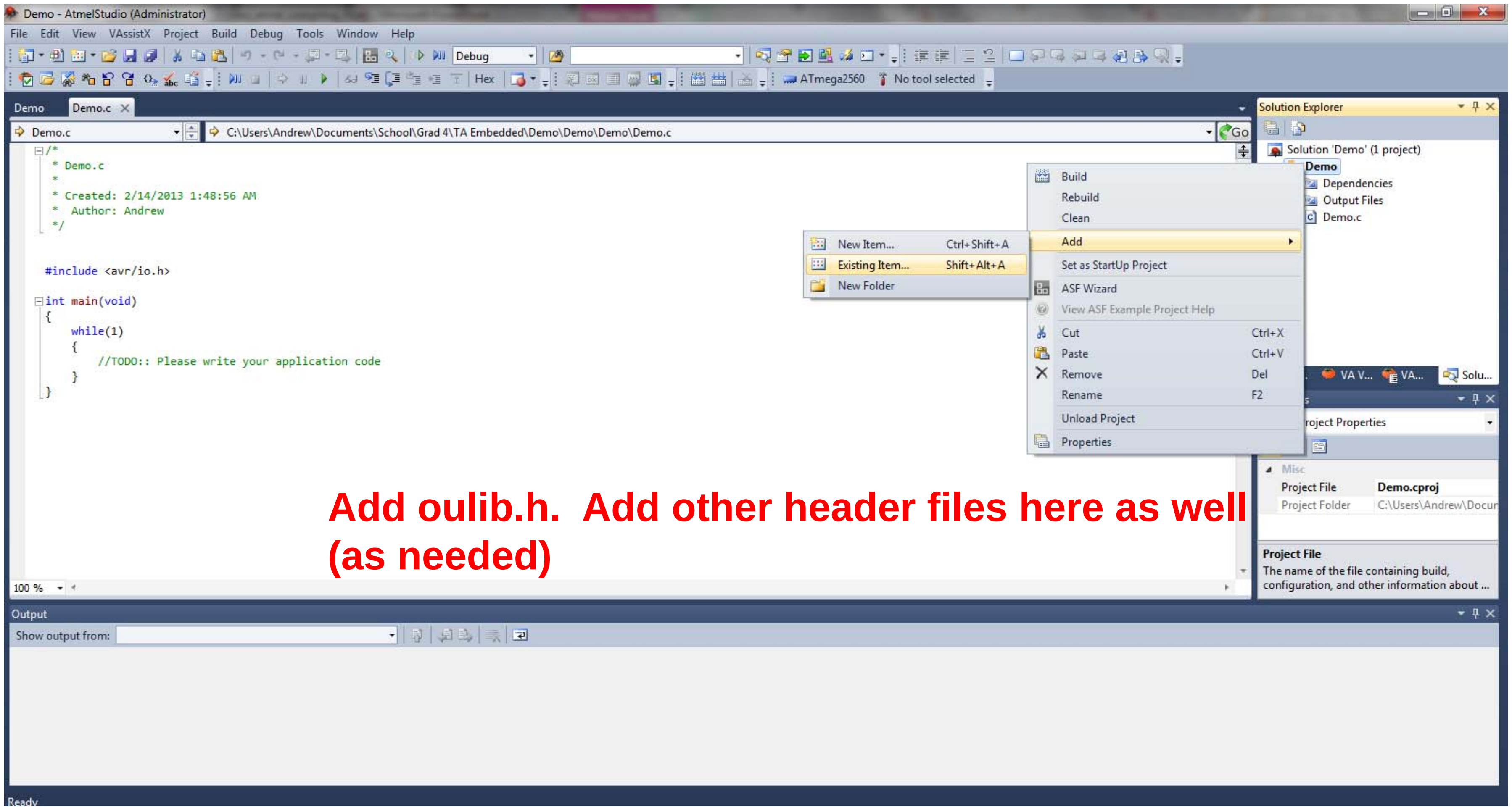
ou_atmega2560

Library search path (-Wl,L)

../../../../Include

Solution 'AVR_TEST_2' (1 project)
- AVR_TEST_2
  - Dependencies
  - AVR_TEST_2.c
  - buffer.h
  - libou_atmega2560.a
  - oulib.h
  - oulib_serial_buffered.h

Add Header Files


Demo - AtmelStudio (Administrator)
File Edit View VAssistX Project Build Debug Tools Window Help
Debug
ATmega2560
No tool selected
Demo
Demo.c
C:\Users\Andrew\Documents\School\Grad 4\TA Embedded\Demo\Demo\Demo\Demo.c
Go
Solution Explorer
Solution 'Demo' (1 project)
Demo
Dependencies
Output Files
Demo.c
Build
Rebuild
Clean
Add
Set as StartUp Project
ASF Wizard
View ASF Example Project Help
Cut Ctrl+X
Paste Ctrl+V
Remove Del
Rename F2
Unload Project
Properties
New Item... Ctrl+Shift+A
Existing Item... Shift+Alt+A
New Folder
VA V... VA... Solu...
roject Properties
Misc
Project File Demo.cproj
Project Folder C:\Users\Andrew\Docur
Project File
The name of the file containing build, configuration, and other information about ...
100 %
Output
Show output from:
Ready

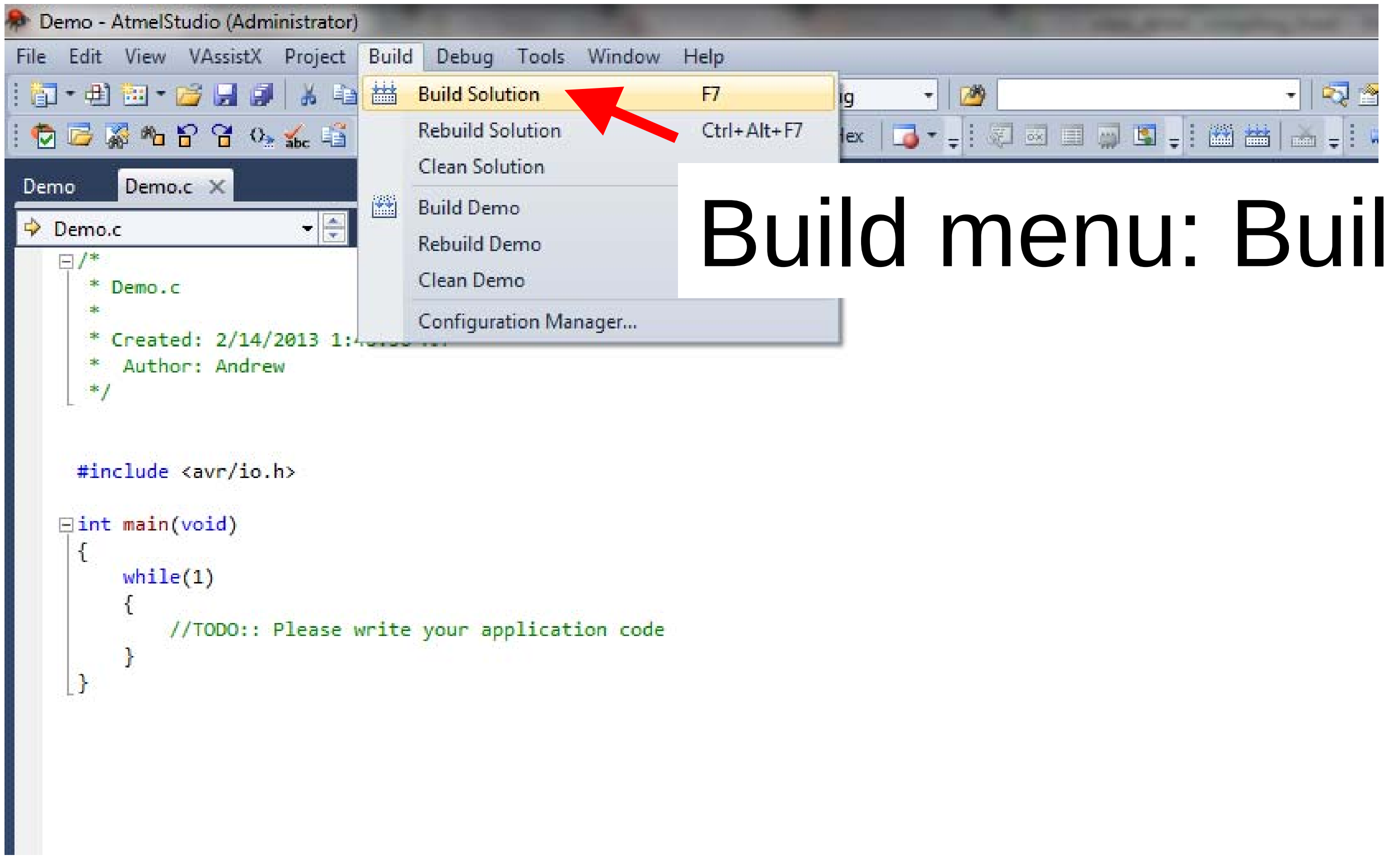
```
/*
 * Demo.c
 *
 * Created: 2/14/2013 1:48:56 AM
 *  Author: Andrew
 */


#include <avr/io.h>

int main(void)
{
    while(1)
    {
        //TODO:: Please write your application code
    }
}
```

Add oulib.h. Add other header files here as well (as needed)

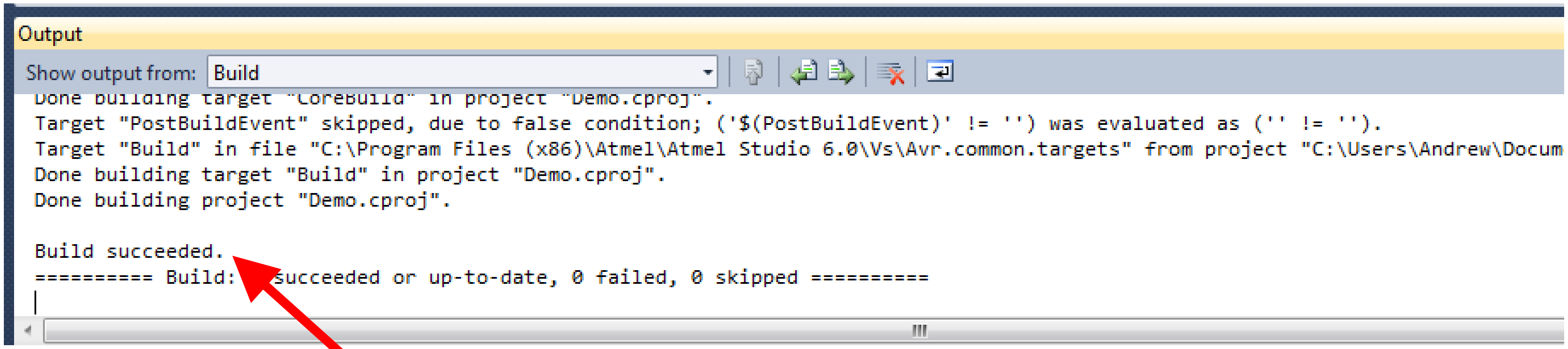
Now for the code…

```
#include "oulib.h"

int main(void)
{
   DDRB = 0x80;        // port B, pin 7

   while(1) {
       PORTB ^= 0x80;
       delay_ms(500);
   }
}
```

Build menu: Build


Demo - AtmelStudio (Administrator)
File Edit View VAssistX Project Build Debug Tools Window Help
Build Solution F7
Rebuild Solution Ctrl+Alt+F7
Clean Solution
Build Demo
Rebuild Demo
Clean Demo
Configuration Manager...
Demo
Demo.c
Demo.c

```
/*
 * Demo.c
 *
 * Created: 2/14/2013 1:
 *  Author: Andrew
 */


#include <avr/io.h>

int main(void)
{
    while(1)
    {
        //TODO:: Please write your application code
    }
}
```



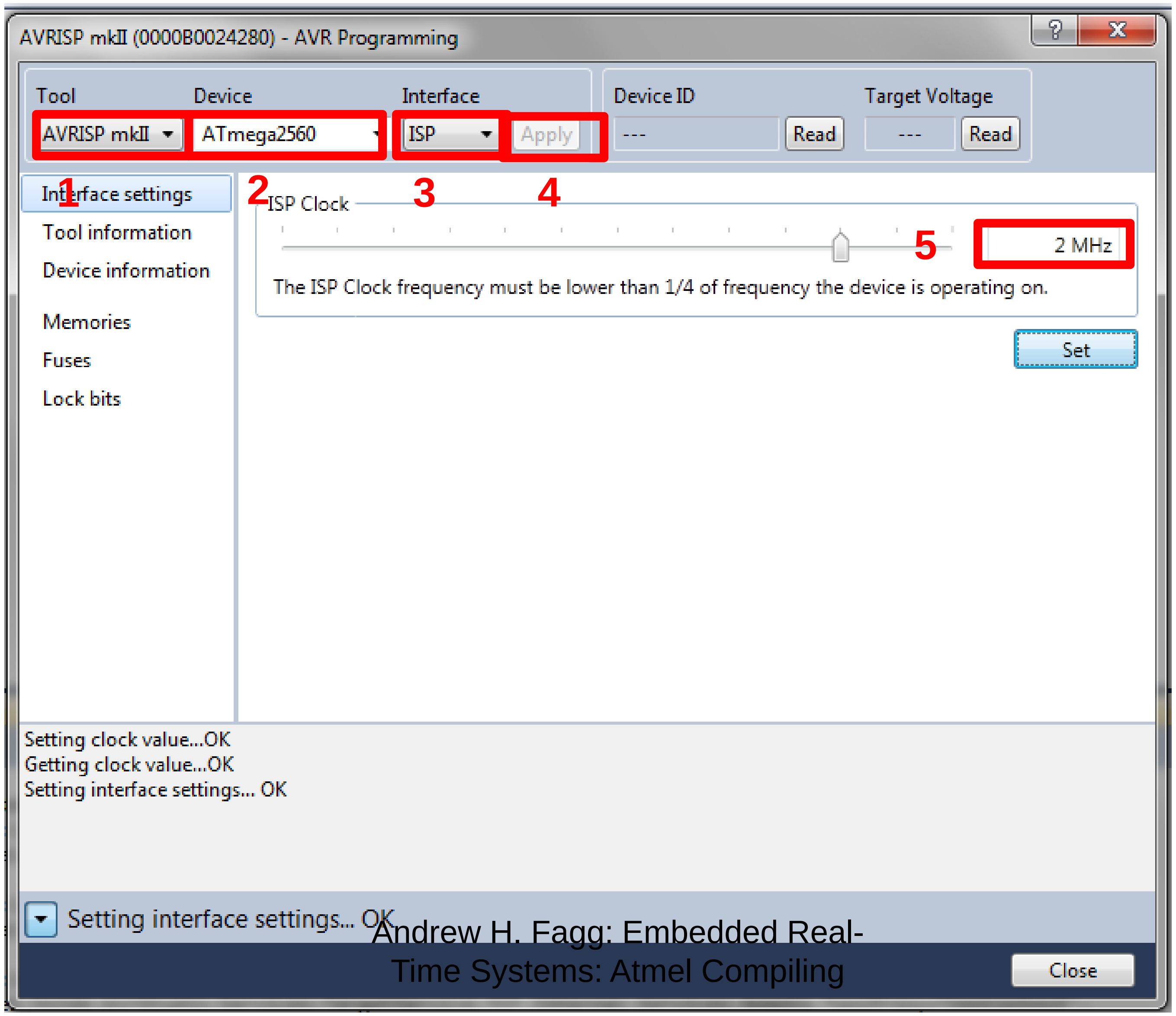
```
Done building target "CoreBuild" in project "Demo.cproj".
Target "PostBuildEvent" skipped, due to false condition; ('$(PostBuildEvent)' != '') was evaluated as ('' != '').
Target "Build" in file "C:\Program Files (x86)\Atmel\Atmel Studio 6.0\Vs\Avr.common.targets" from project "C:\Users\Andrew\Docum
Done building target "Build" in project "Demo.cproj".
Done building project "Demo.cproj".

Build succeeded.
========== Build: succeeded or up-to-date, 0 failed, 0 skipped ==========
```

You should get this

Now We Are Ready…

- Plug the programmer into your computer **and** into the Arduino board (If it is not already)
- Make sure your Arduino board has power
  - Either from USB or batteries
- And download the program…
  - Tools Menu: Device Programming

Select the AVR Mk II

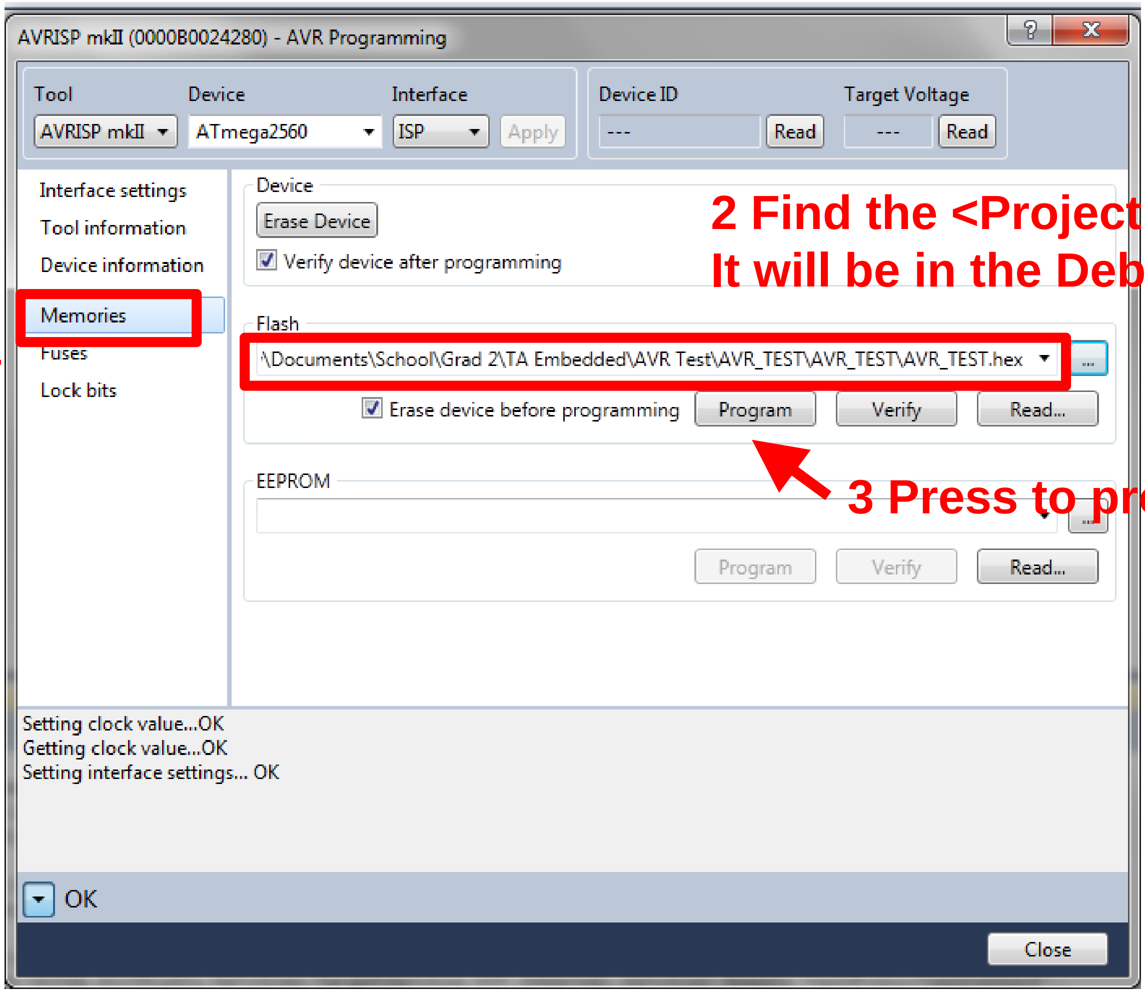


AVRISP mkII (0000B0024280) - AVR Programming

The ISP Clock frequency must be lower than 1/4 of frequency the device is operating on.

```
Setting clock value...OK
Getting clock value...OK
Setting interface settings... OK
```

Setting interface settings... OK



Setting clock value...OK
Getting clock value...OK
Setting interface settings... OK

Flashing?

Your program will start executing as soon as the download is complete …

Your on-board Light Emitting Diode should be blinking at 1 Hertz (once per second)

Next Task

- Add several more LEDs in a line
- Write a program that turns the LEDs on in sequence