Getting Started with the Atmel Mega2560

Questions?

Quiz

Port-Related Registers

Some of the C-accessible registers for controlling digital I/O:

	Directional control	Writing	Reading
Port B	DDRB	PORTB	PINB
Port C	DDRC	PORTC	PINC
Port D	DDRD	PORTD	PIND

Arduino Mega Board

(see schematic)

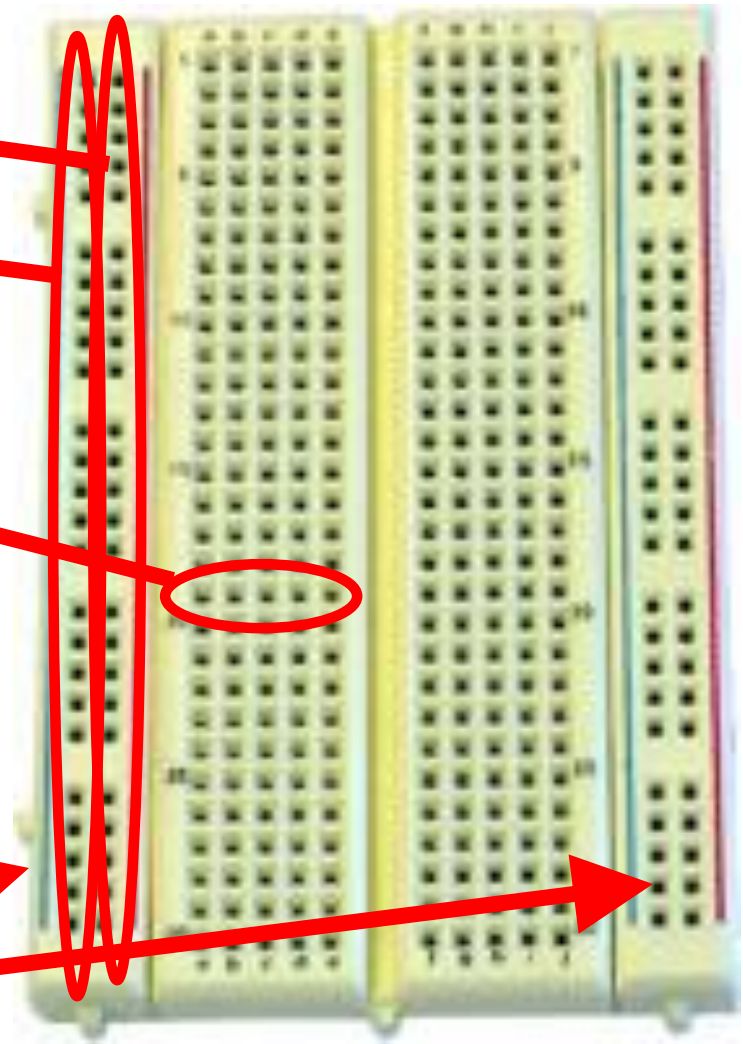
Solderless Breadboards

Power bus (red)

Ground bus (blue)

Component bus

Note that the two sides are not connected



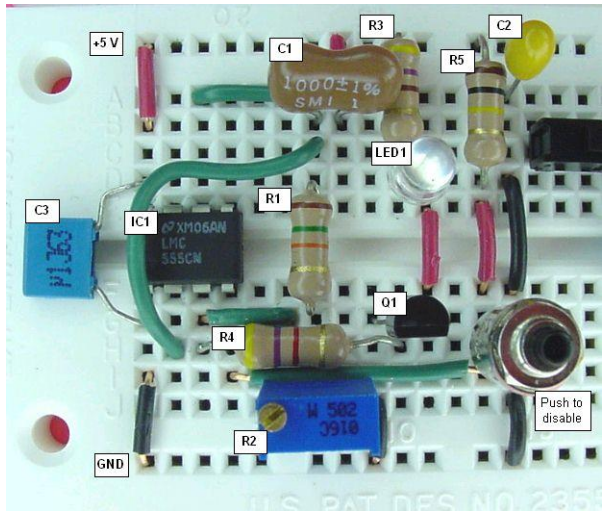
Wiring Standards

When possible, use wire colors for different types of signals:

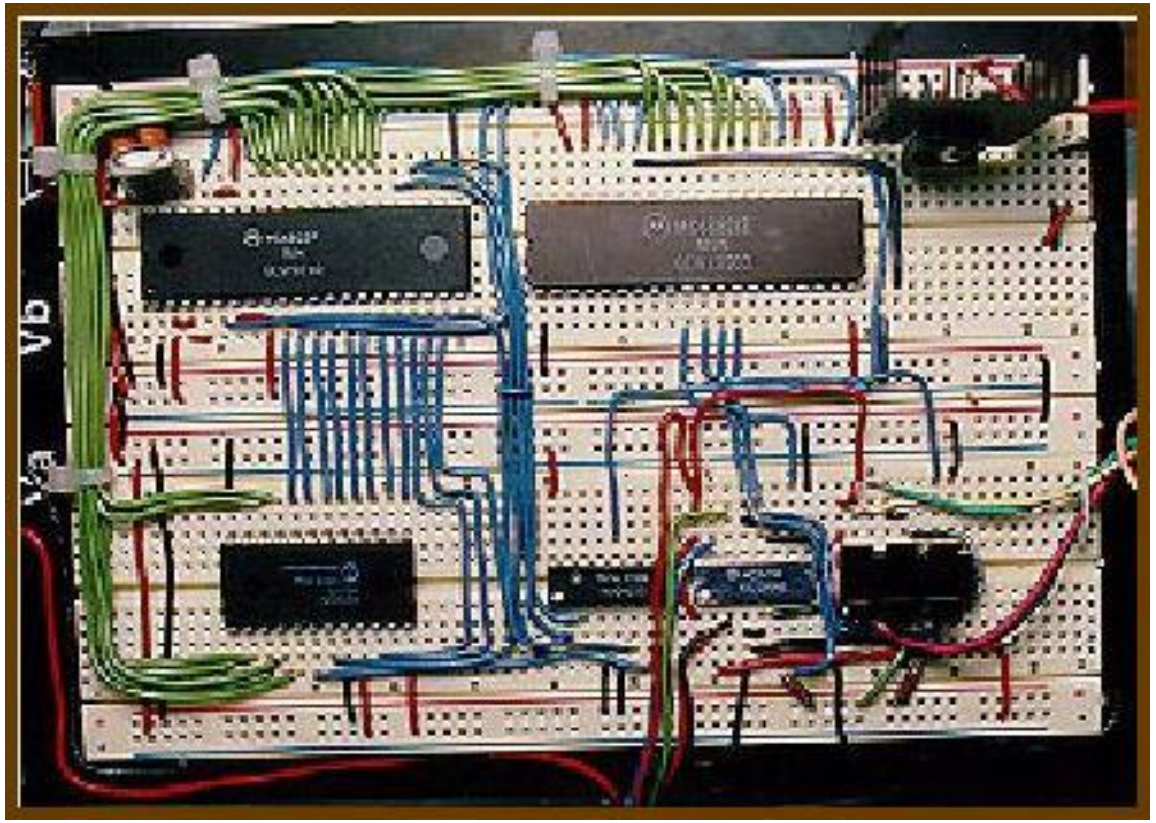
- Black: ground
- Red: power
- Other: various signals

Clean Wiring

A clean breadboard will make debugging easier – and it makes circuits more robust



www.linefollowing.com



tangentsoft.net

Care with Power

- Only insert components and wires into the breadboard when power is disconnected
- “Wire, check-twice, then power”
  - Never reverse power and ground (this is a very common mistake)
- Most chips that we will use expect +5V
  - More can destroy the chips
  - We will use DC/DC converters to step battery voltages down to +5V

Suggested Wiring Procedure

- Power supply
- Power/ground buses
- Insert primary components
- Wire power/ground for components
- Add signals and remaining components
- Test incrementally

Debugging Techniques

- Test incrementally
- Test intermediate sub-circuits

Physical Interface for Programming

AVR ISP

Physical Interface for Programming

AVR ISP

USB
connection to
your laptop

Physical Interface for Programming

AVR ISP

Header connection will connect to your circuit (through an adapter)

Be careful when you plug your circuit in (check before powering)

AVR ISPs are Cranky

- When things are plugged in and powered, you should see two green LEDs on the ISP (on most units)
- One red: usually means that your circuit is not powered
- Flashing orange: connector is backwards!
- Orange: the programmer is confused
  - Could be due to your circuit not being powered at 5V
  - Could be due to other problems
  - Check power and reboot the ISP

Compiling and Downloading Code

Once the chip is programmed, the AVR ISP will automatically reset the processor; starting your program

Hints

- Use LEDs to show status information (e.g., to indicate what part of your code is being executed)
- Remember: on the Arduino boards, there is a LED connected to port B, pin 7
- Have one LED blink in some unique way at the beginning of your program
- Go slow:
  - Implement and test incrementally
  - Insert plenty of pauses into your code (e.g., with delay_ms())

Project 0

- Summary:
  - Write program that flashes the LED attached to PORTB, pin 7 at a chosen (visible) frequency.
  - Connect 4 LEDs and a switch to your Arduino board
  - Write a program that: waits for the switch to close, then displays an interesting LED flashing pattern

- Details are on the class web page

Compiling and Downloading

Preparation:

- Create a class folder to work in: e.g., “ame3623”
- Download libou_atmega2560.a, oulib.h from the Atmel HOWTO page
- Inside of your class folder, create folders “oulib”, “oulib\lib” and “oulib\include”
  - Place libou_atmega2560.a in oulib\lib\
  - Place oulib.h in oulib\include

Compiling and Downloading

Preparation (unix only):

- Create a project folder, e.g. “testproject” in your class folder
- Download makefile into this directory:
  - Modify the “TARGET” line to be the name of the C file that you are about to create
  - Modify the “OULIB_DIR” line as necessary. In this example, it should be “../oulib/”
- Create your C file in testproject

Compiling and Downloading (the Unix way)

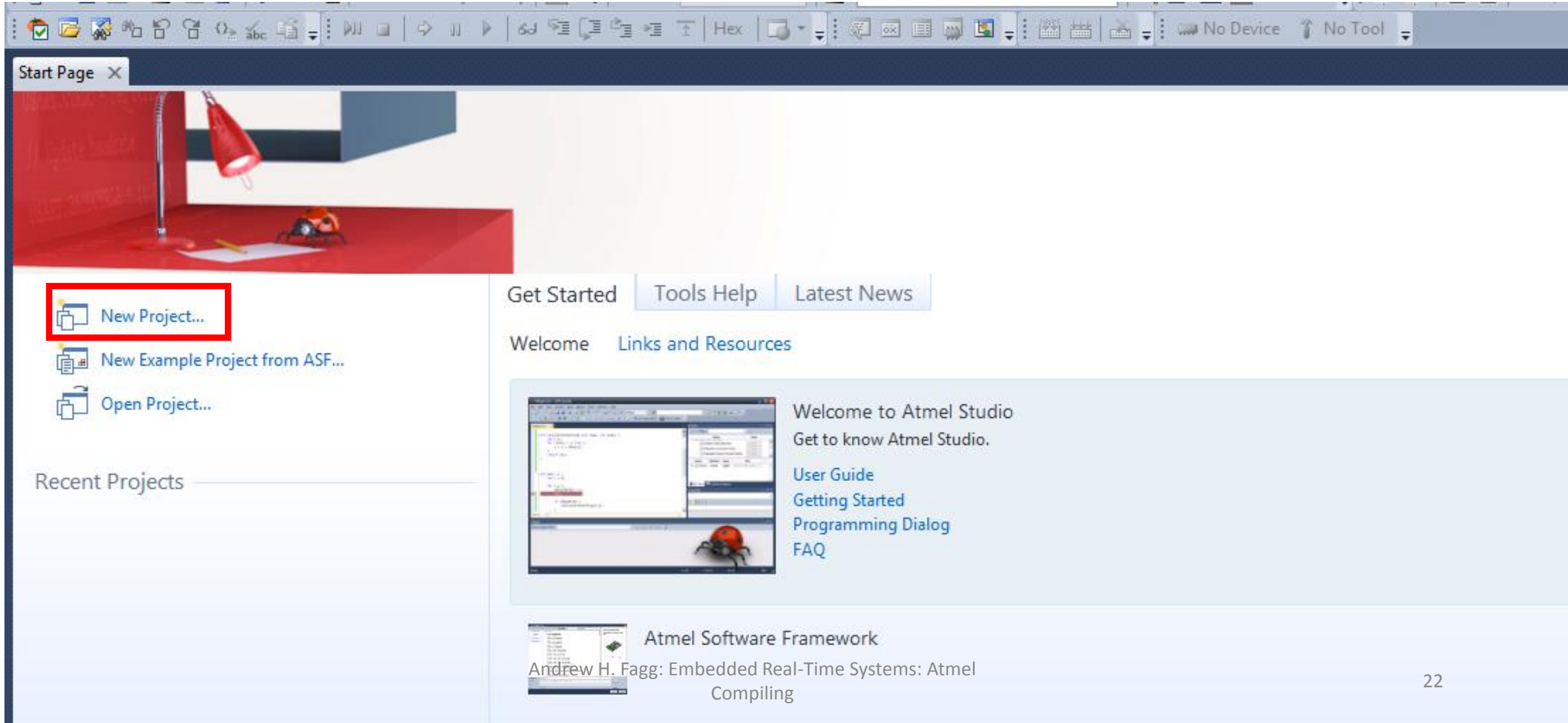
At the command line:

- “cd” to your project directory

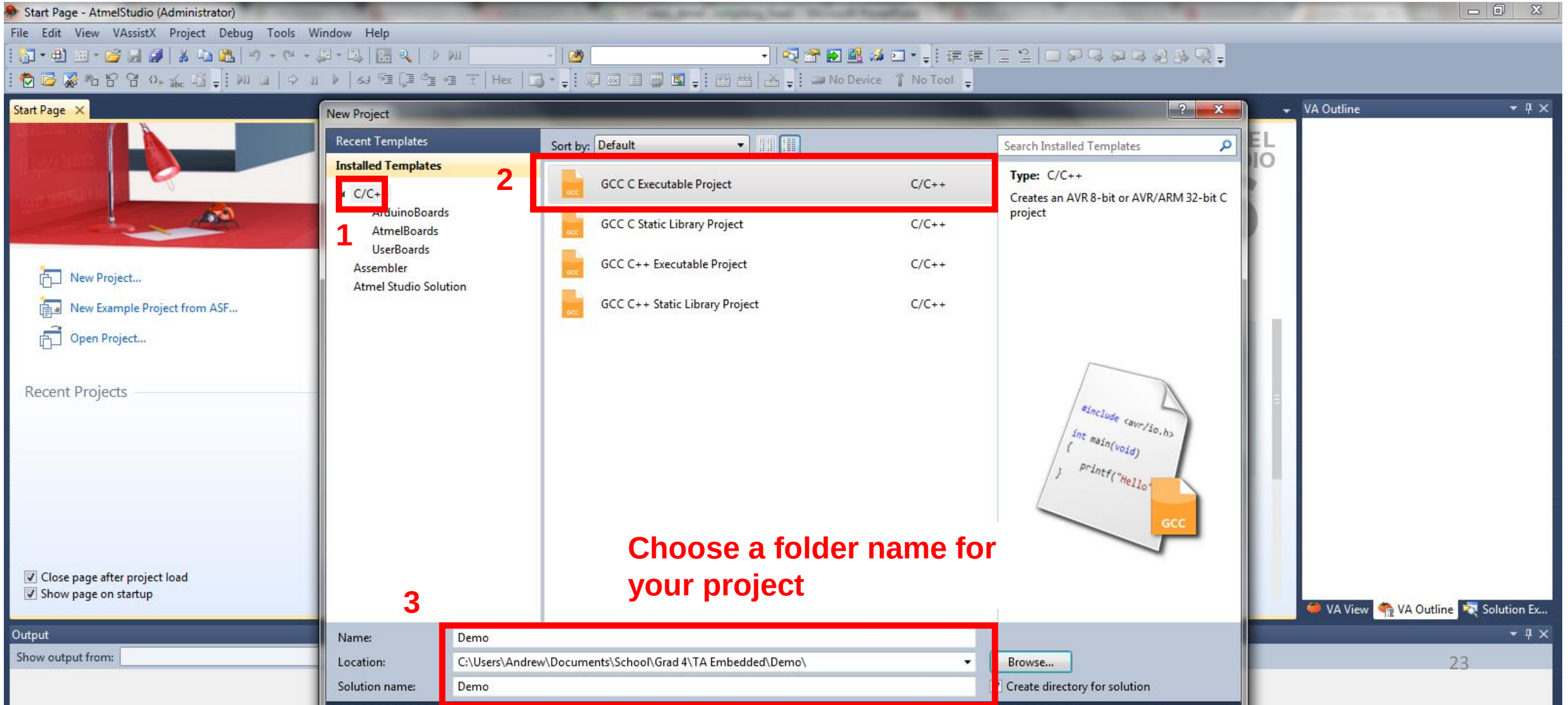
- Type “make”
  - You should see no errors
  - If there are errors, then you must fix them before moving on

- Type “make program”
  - This will download your code to the processor
  - Again, you should see no errors

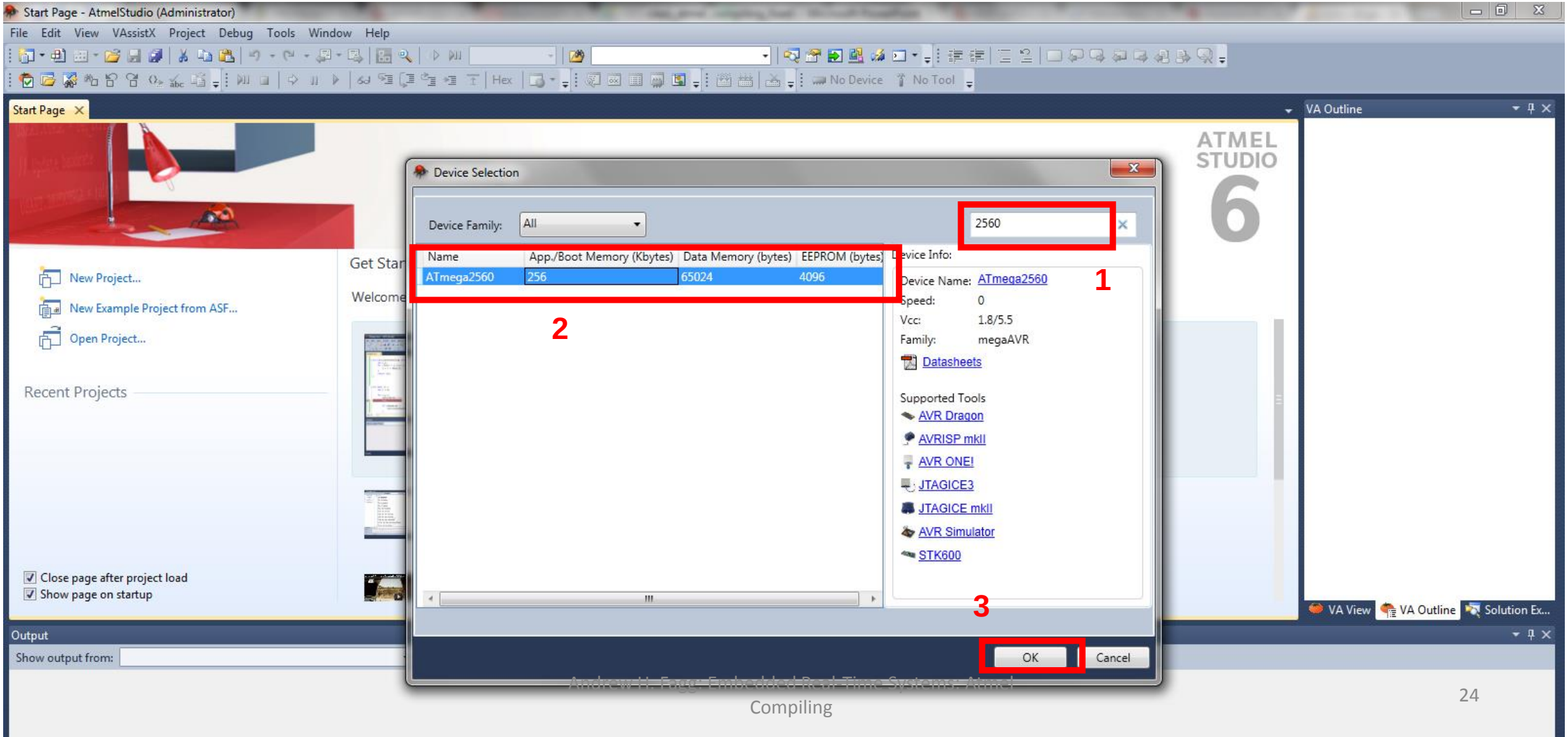
Windows: Getting Started



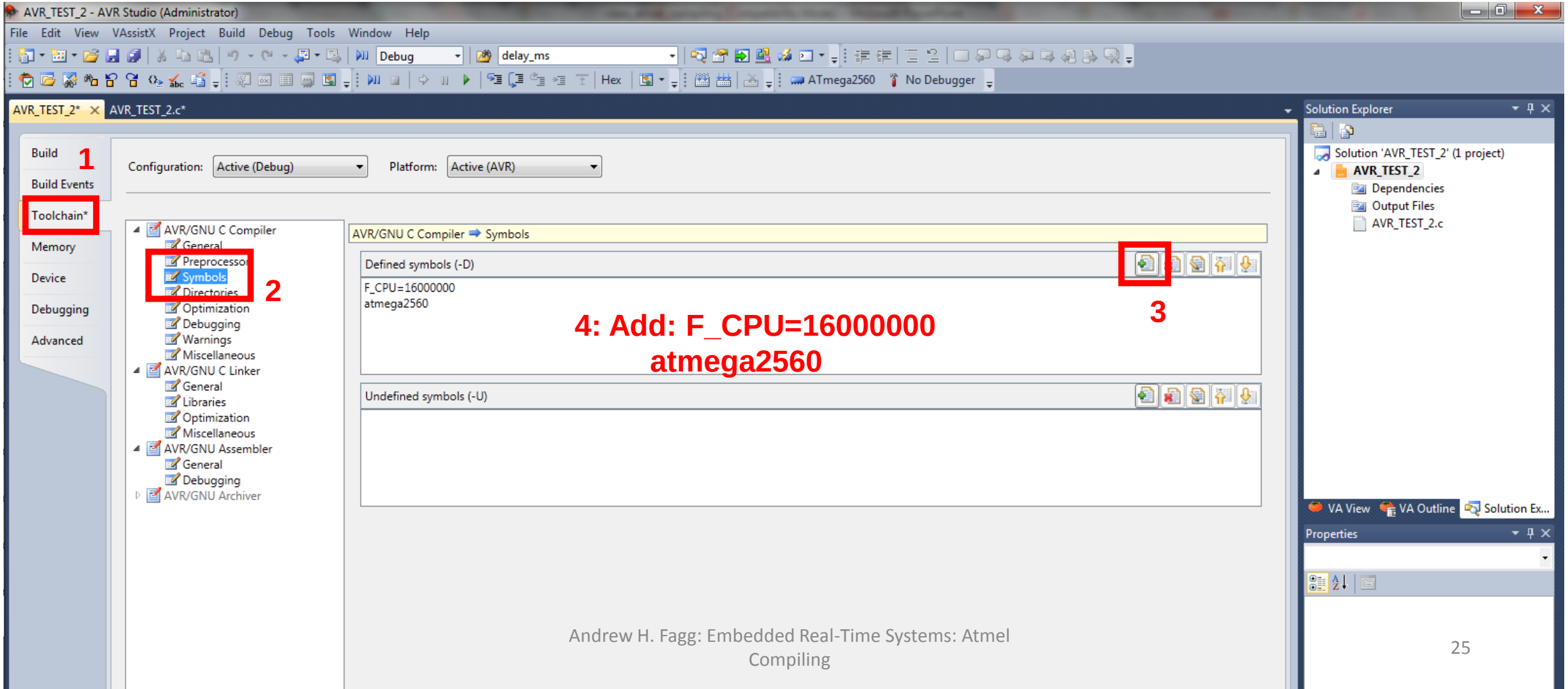
New Project


Start Page - AtmelStudio (Administrator)
File Edit View VAssistX Project Debug Tools Window Help
Start Page
New Project...
New Example Project from ASF...
Open Project...
Recent Projects
Close page after project load
Show page on startup
New Project
Recent Templates
Installed Templates
C/C++
ArduinoBoards
AtmelBoards
UserBoards
Assembler
Atmel Studio Solution
1
2
Sort by: Default
Search Installed Templates
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
}
GCC
Choose a folder name for your project
3
Name: Demo
Location: C:\Users\Andrew\Documents\School\Grad 4\TA Embedded\Demo\
Solution name: Demo
Browse...
Create directory for solution
VA Outline
VA View
VA Outline
Solution Ex...
Output
Show output from:

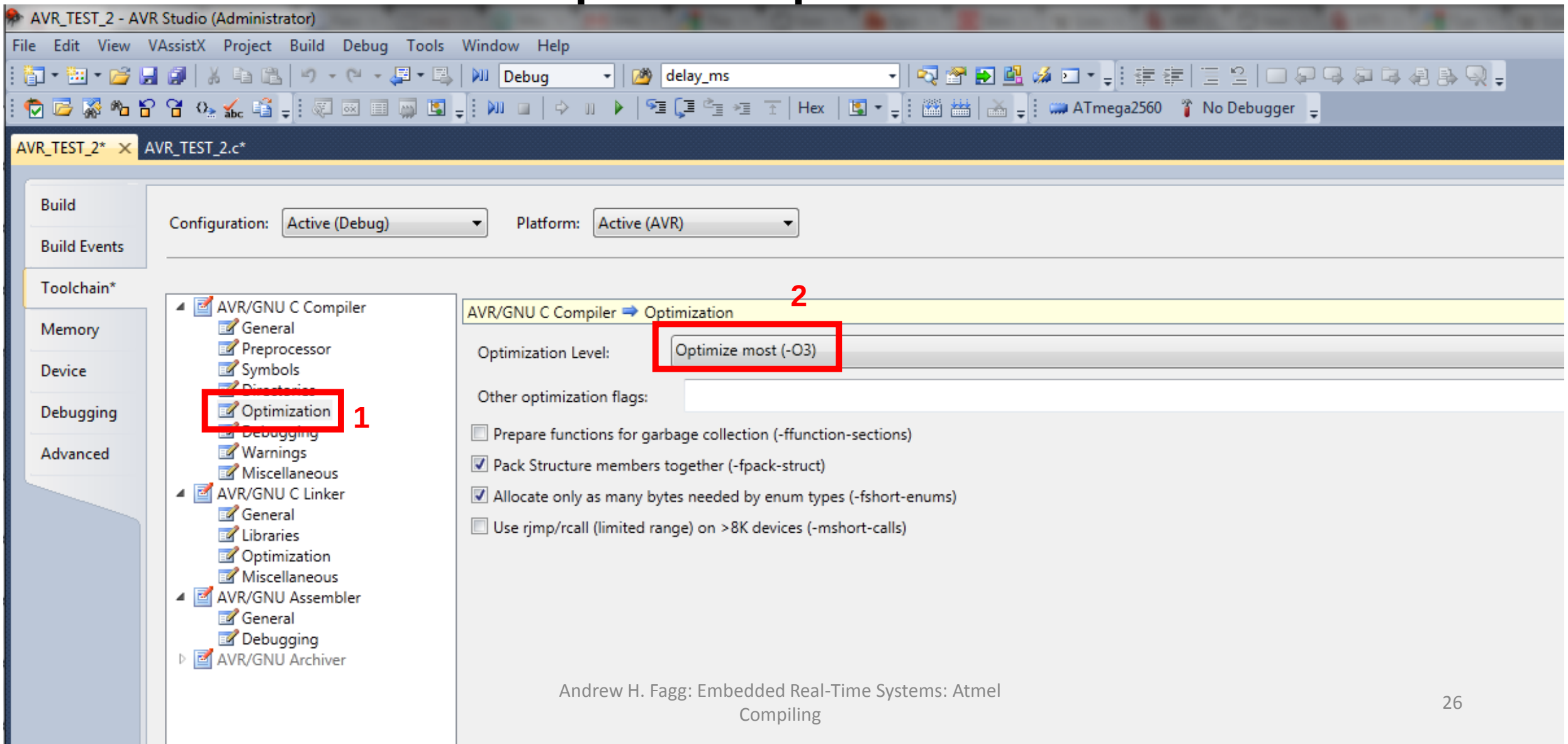
Select the ATmega2560



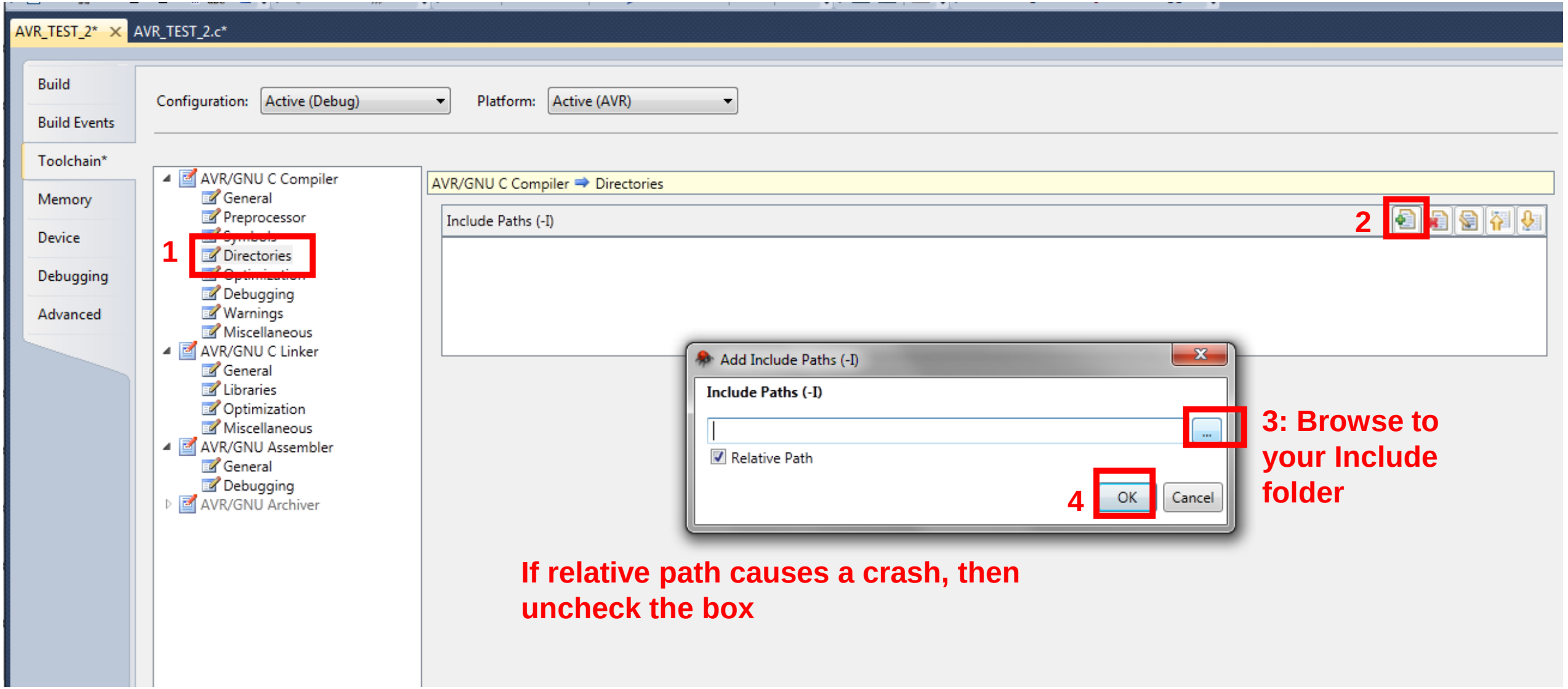
Project➔
<Project Name> Properties (Alt+F7)



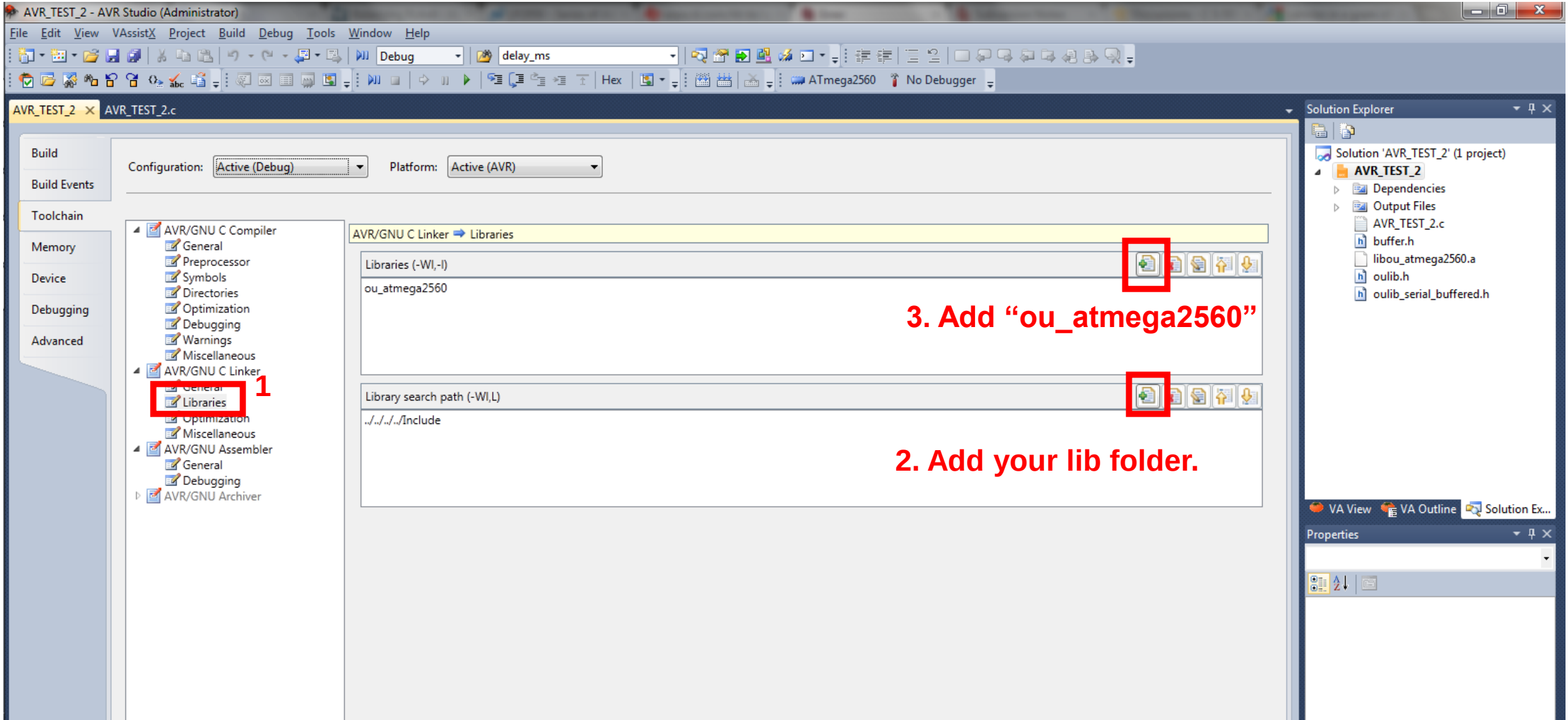
Compiler Optimization



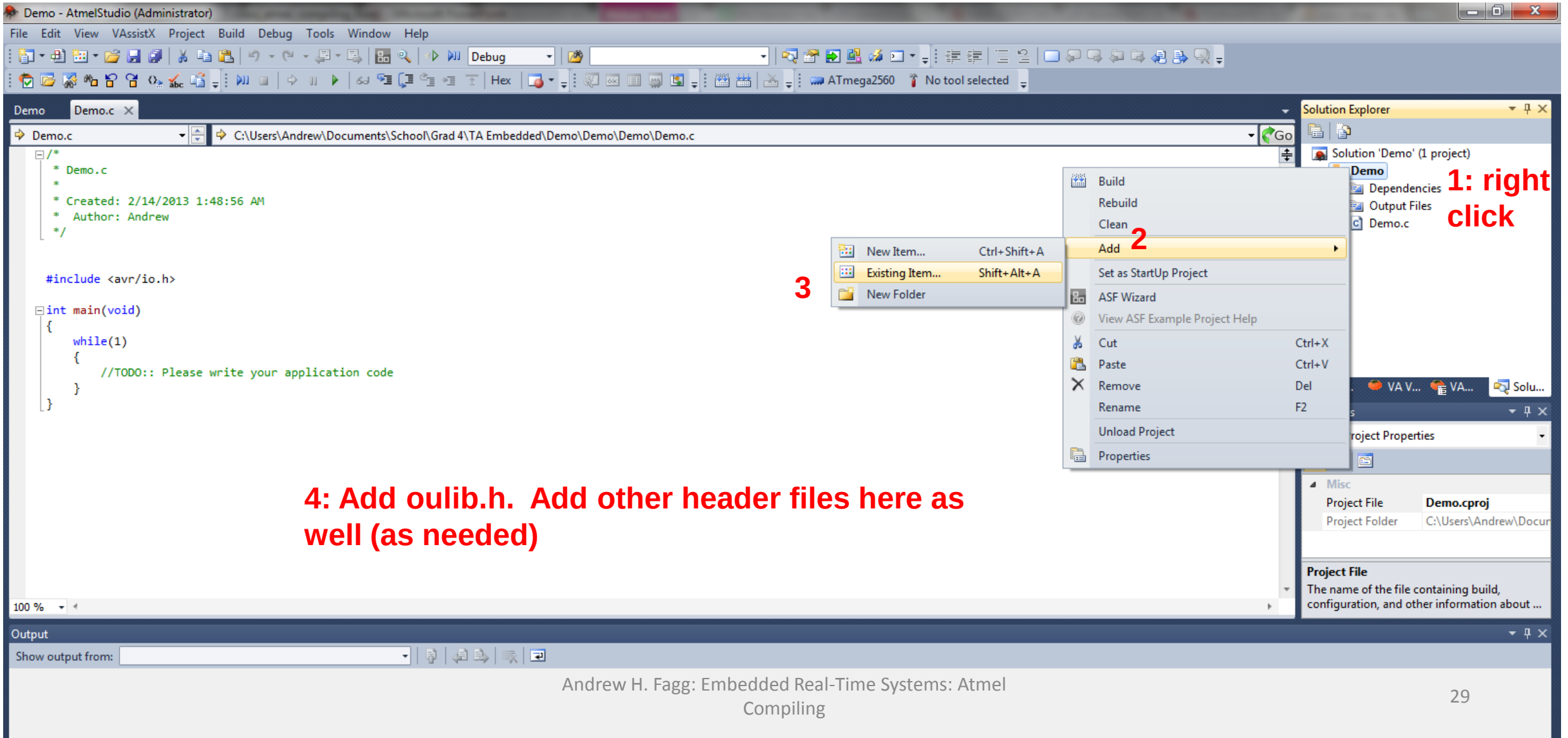
Add Directories



Add Libraries



Add Header Files

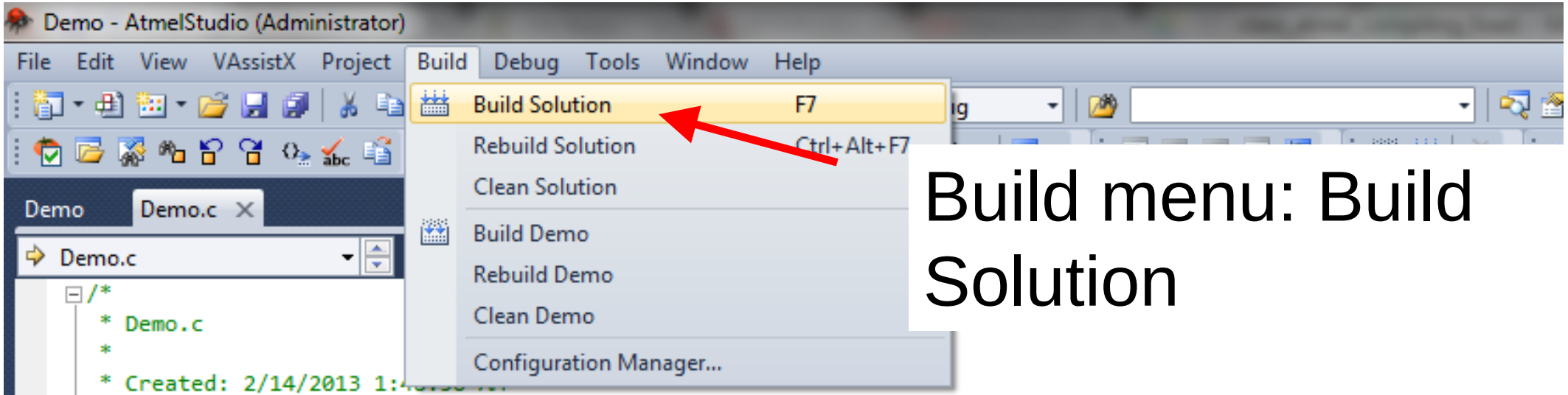


Now for the code…

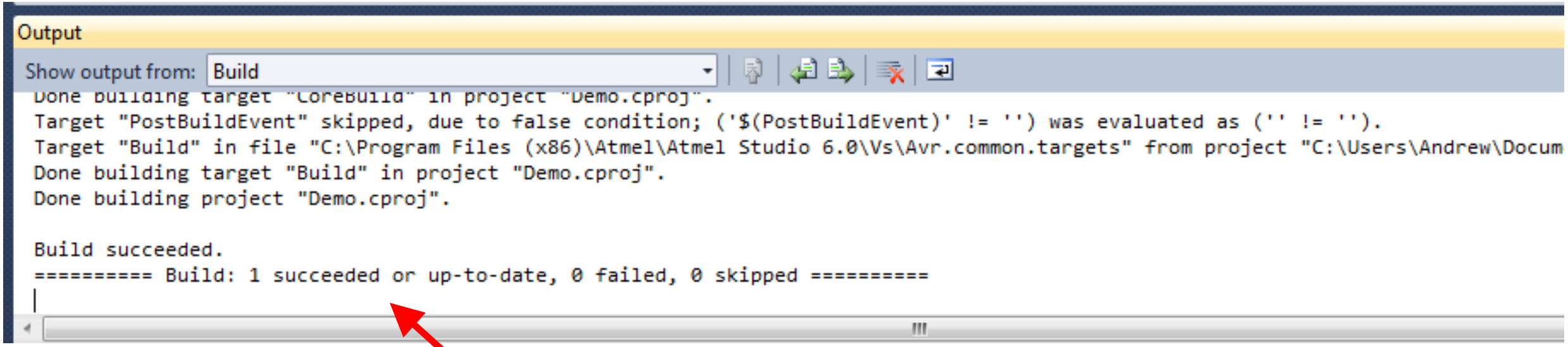
```
#include "oulib.h"

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

   while(1) {
       // Your code here
   }
}
```



Build menu: Build Solution

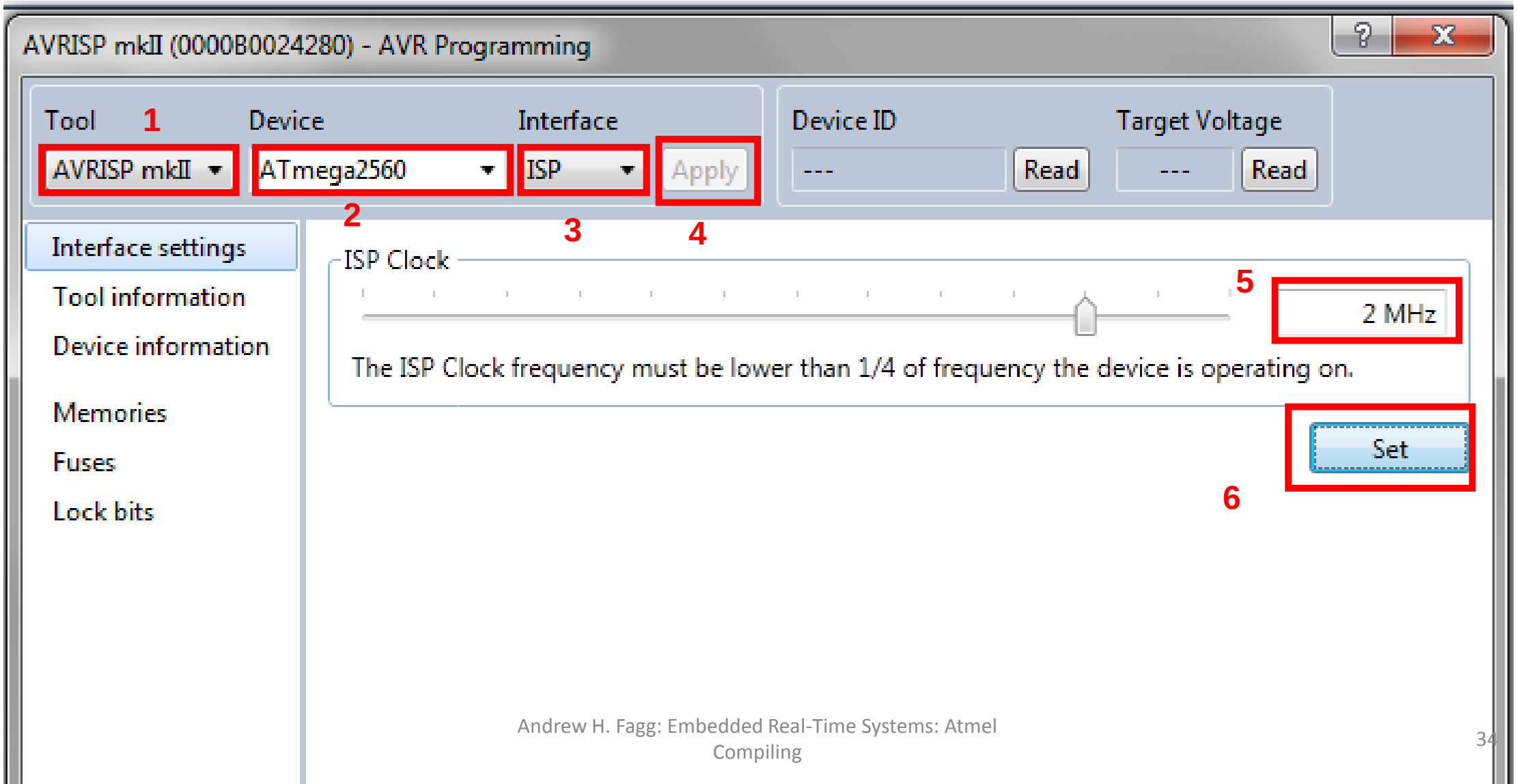


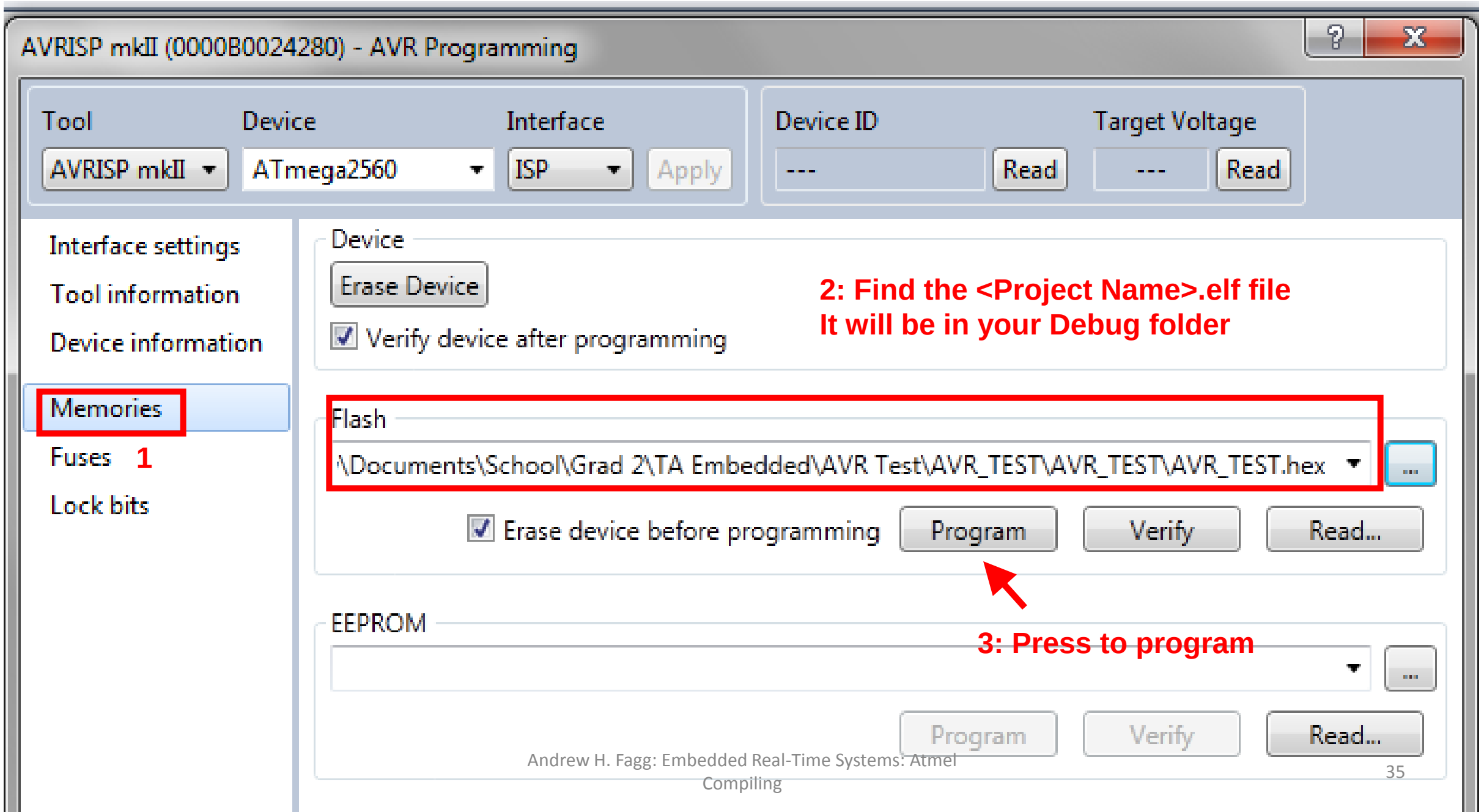
You should get this

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
Tool
AVRISP mkII
Device
ATmega2560
Interface
ISP
Apply
Device ID

Read
Target Voltage

Read
Interface settings
Tool information
Device information
Memories
Fuses
1
Lock bits
Device
Erase Device
Verify device after programming
2: Find the <Project Name>.elf file
It will be in your Debug folder
Flash
\Documents\School\Grad 2\TA Embedded\AVR Test\AVR_TEST\AVR_TEST\AVR_TEST.hex
...
Erase device before programming
Program
Verify
Read...
3: Press to program
EEPROM
...
Program
Verify
Read...

Flashing?

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

Your on-board Light Emitting Diode should be blinking

Next Time

Bit-wise operators for digital input/output

Dr. Sesh Commuri is lecturing