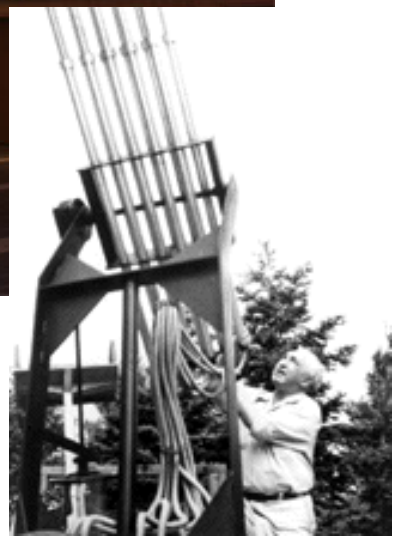
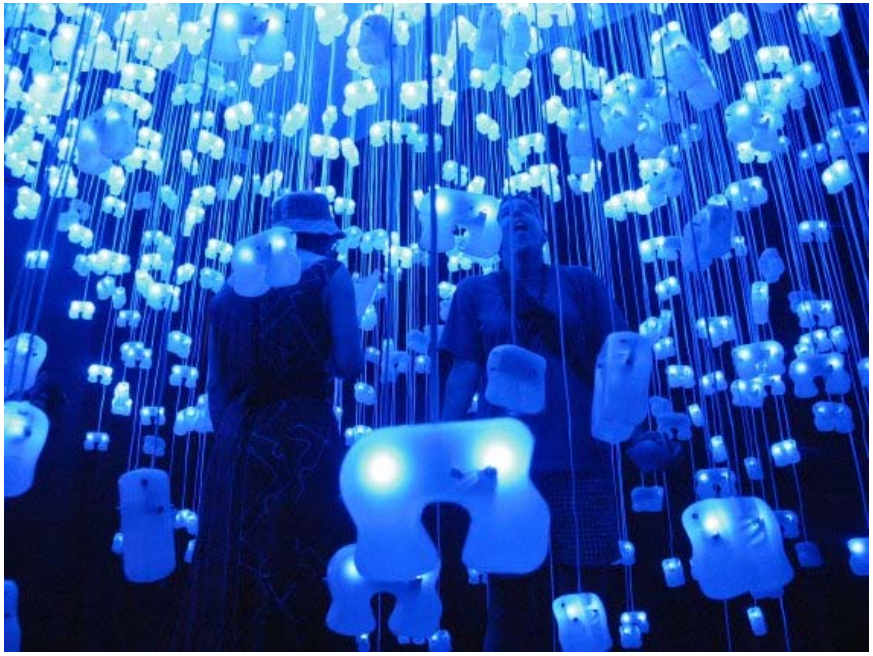


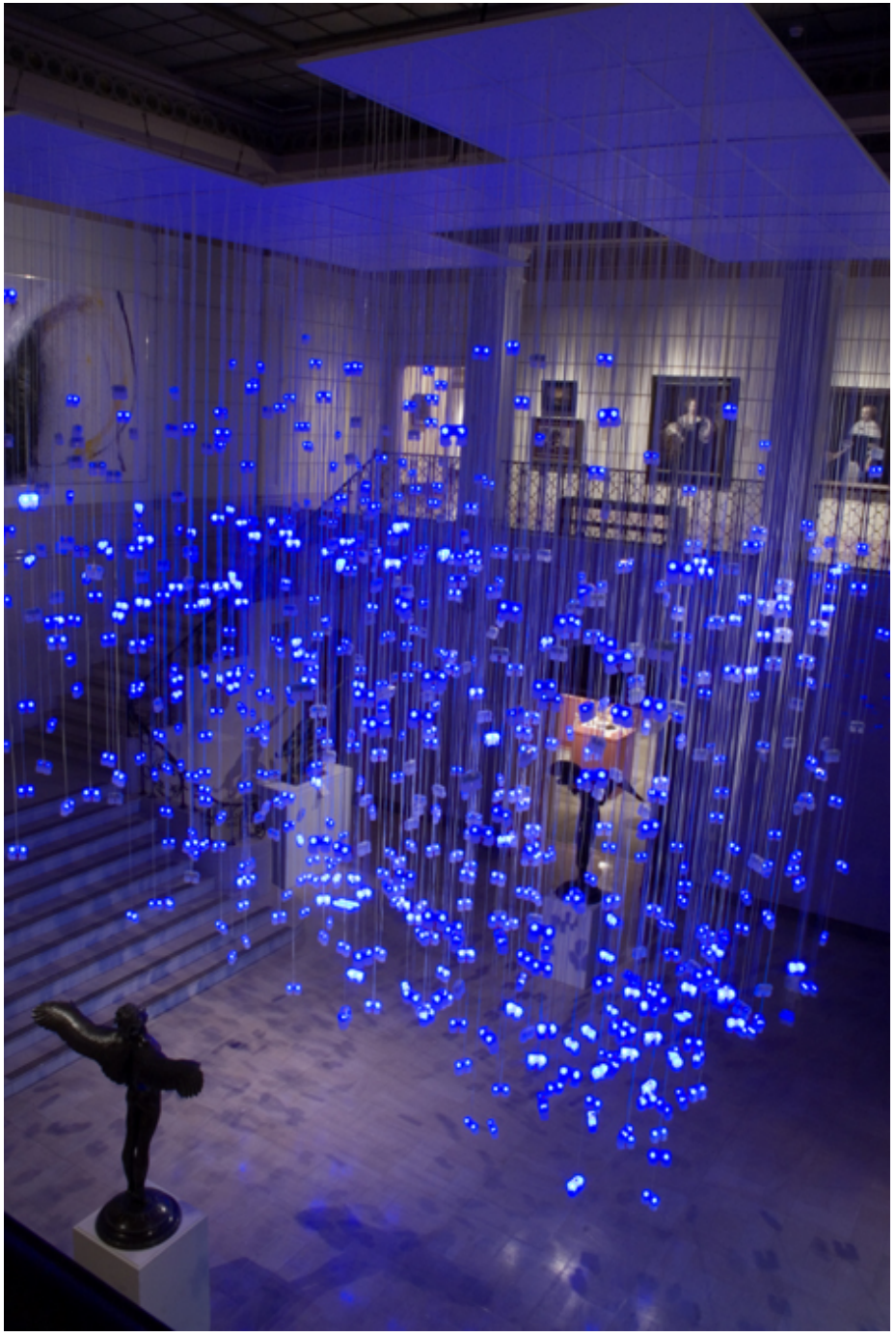
Bion

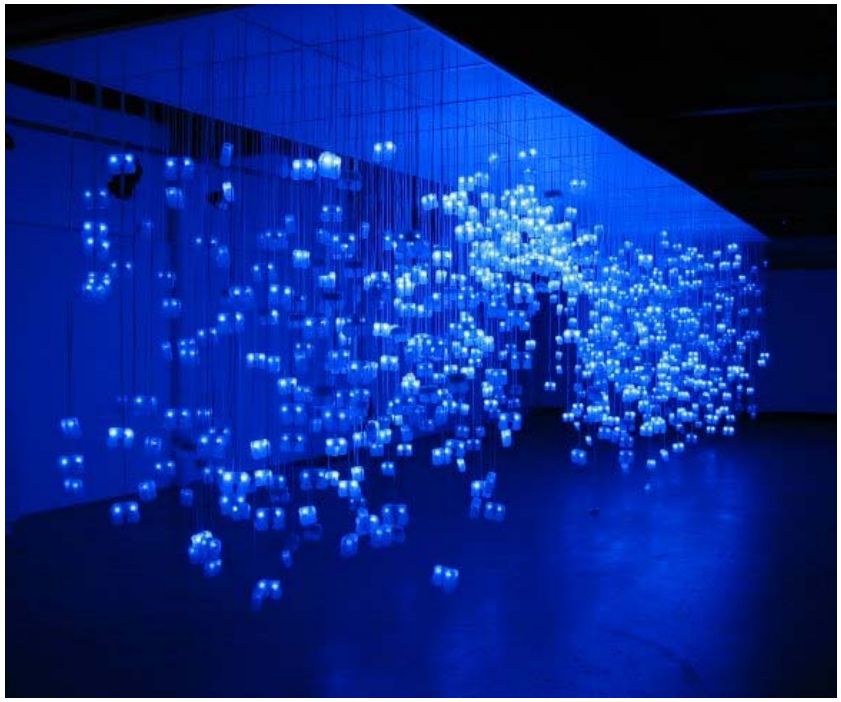
Sensor network:

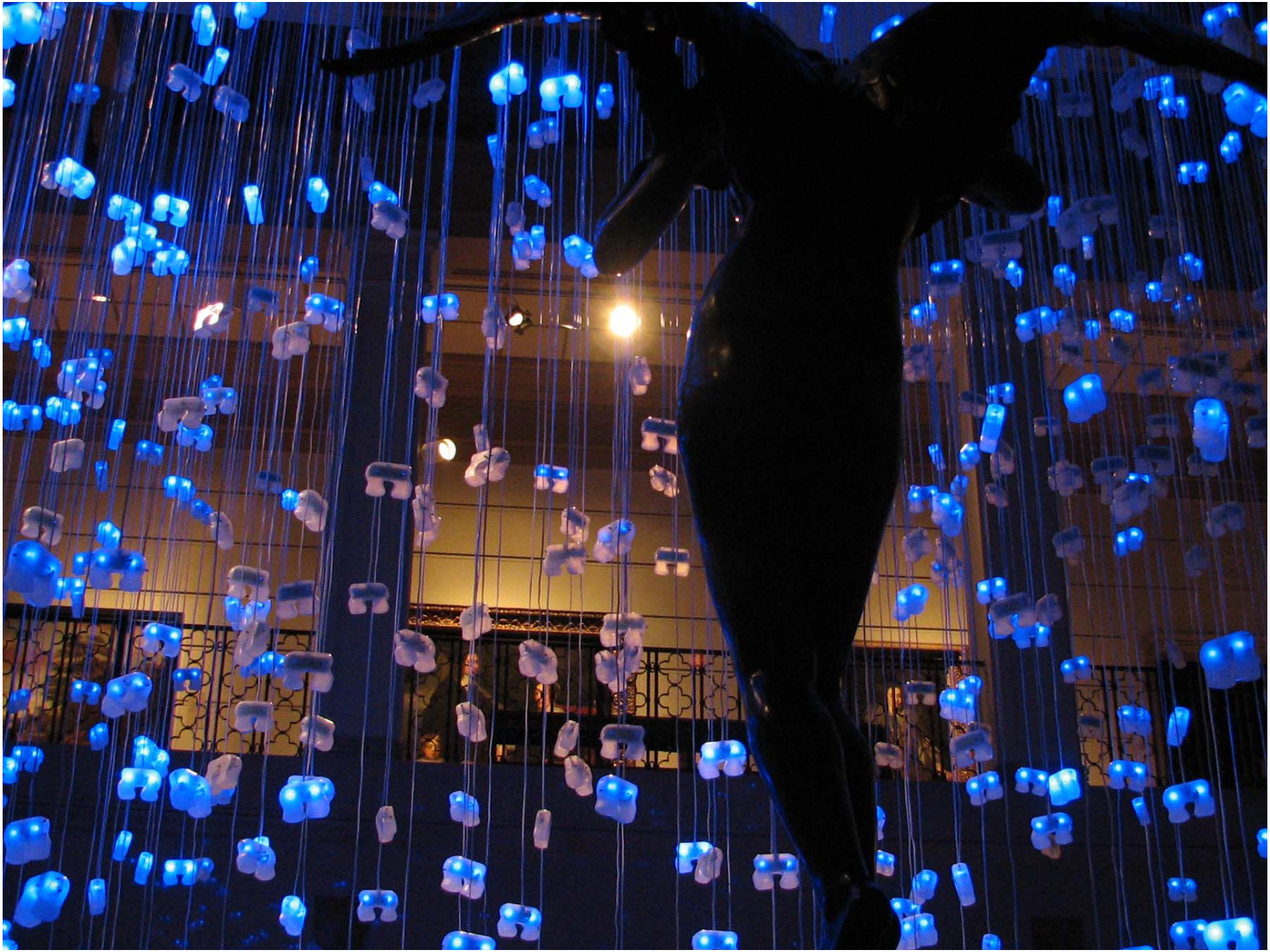
- 1000 sensor nodes
- 3 miles of telephone cable

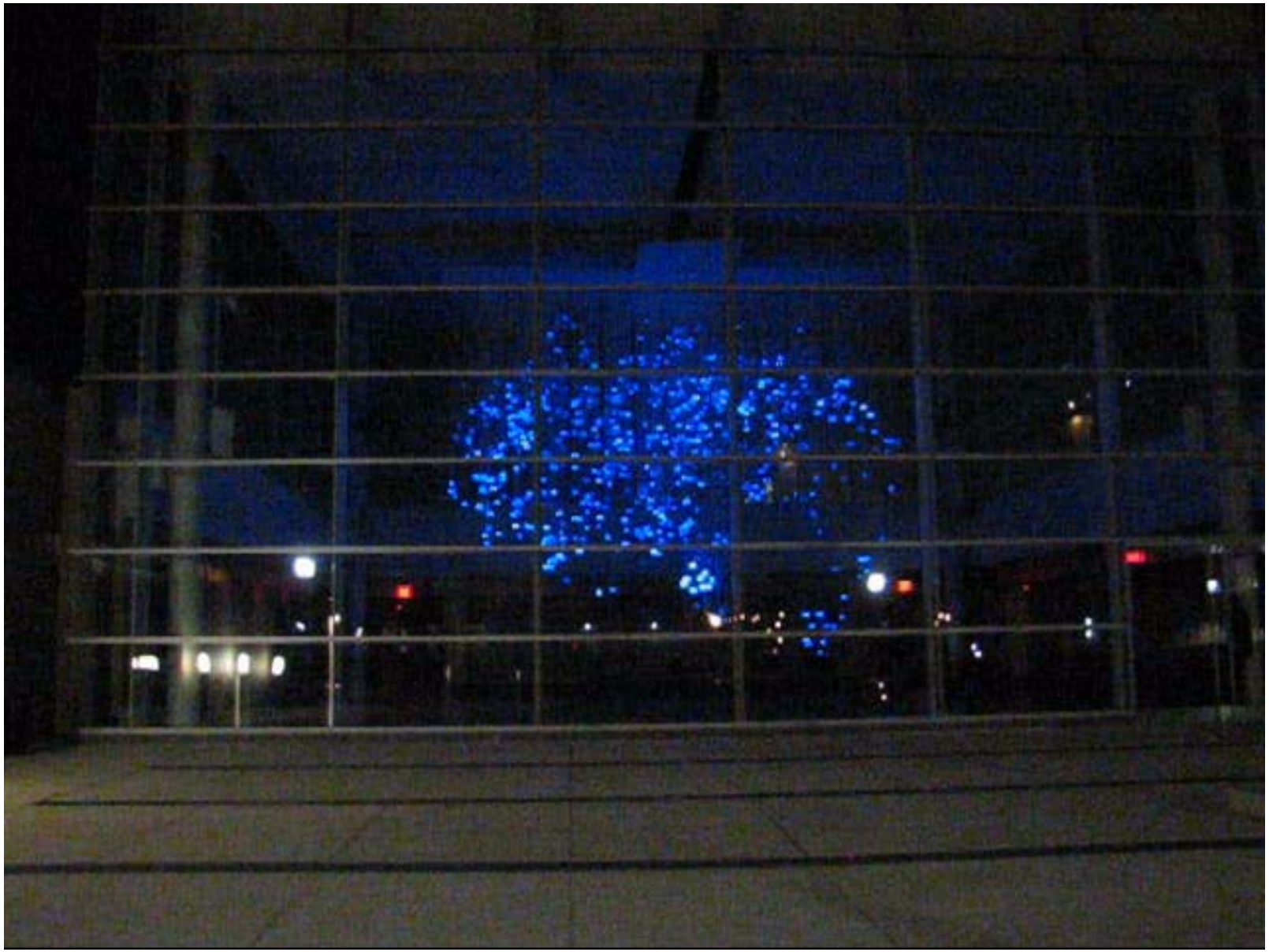
Wilhelm Reich





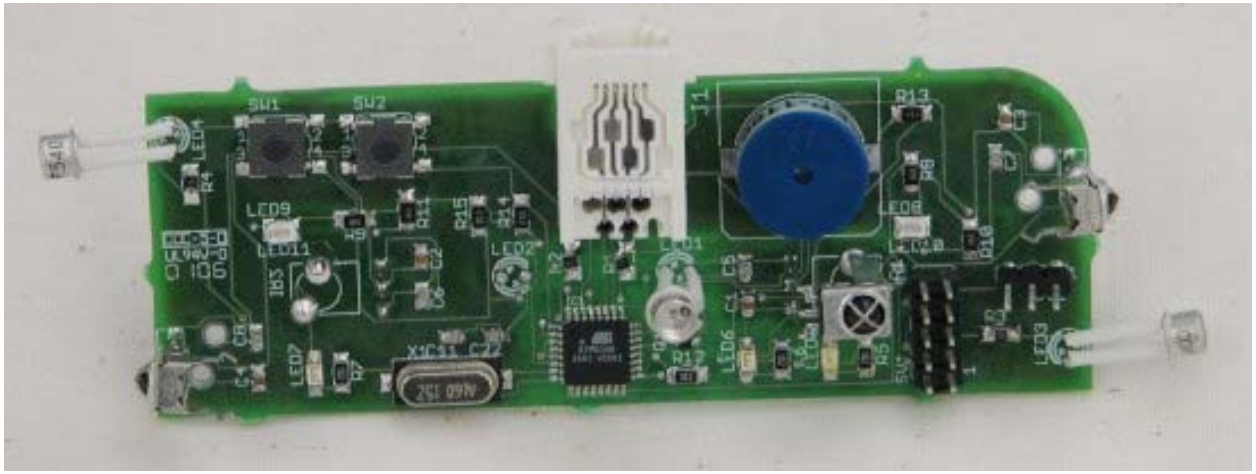






Project 1: Digital I/O and Timing

- Control of LEDs and Speaker
  - Precise timing requires timer use
- Respond to button presses



Part 1

- Internal 4-bit (software) counter
- Counter state is reflected by the LEDs
  - Bit 0 (LSB): Blue
  - Bit 1: Green
  - Bit 2: Red
  - Bit 3: Yellow

Part 1

- Each button release:
  - Increment counter
  - Implement proper debouncing

Part 2

- Generate tone with the speaker
  - Different tone for each counter state (higher frequencies for higher values)
  - This tone should be produced continuously (no pauses)
- Speaker is controlled by a digital I/O line
  - So: in one of two states
  - Tones are produced by producing a “square wave” at a given frequency

Required Components

- Software debounce of the button (switch2)
- Modular code
  - E.g., implement a separate function that translates the current counter value into the LED state

Project Administrivia

Due on March 3rd

- Demonstrate to me, or Di WangDocumented code: hand-in on D2L
  - One copy per 2-person group
- Personal report: distribution of work

Bion Care

- Hold bions on the side of the board (don’t touch the components)
- Minimize the bending of the components
- Don’t let the bion come in contact with metal while it is powered on

- If things get hot: disconnect power immediately and ask for help

Getting Started

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

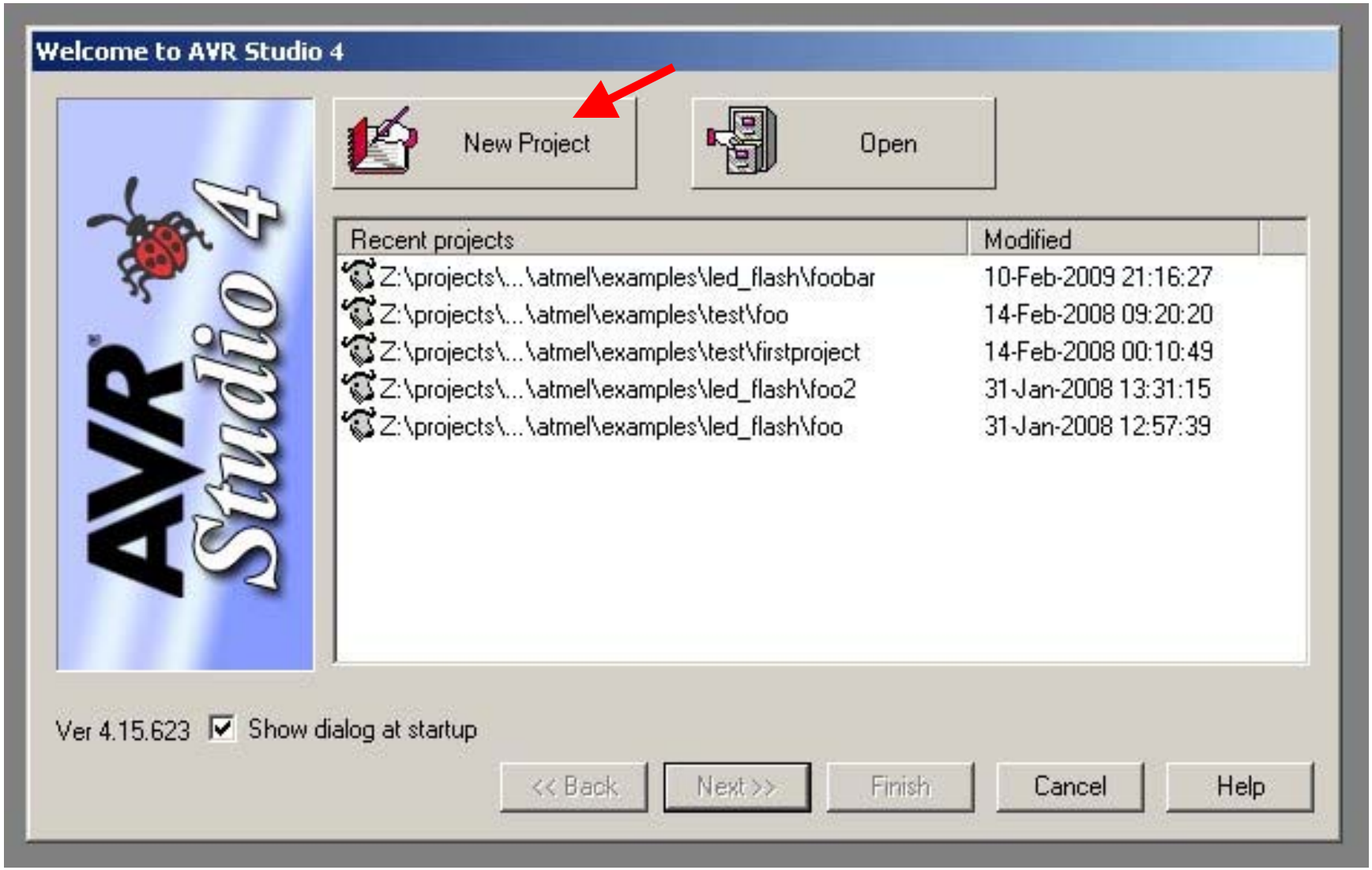
Summary:

- (perhaps) Install AVRstudio
- Install WinAVR
- Plug the programmer into your computer
- Plug the programmer into the bion
- Plug the power into the bion
- Create a program

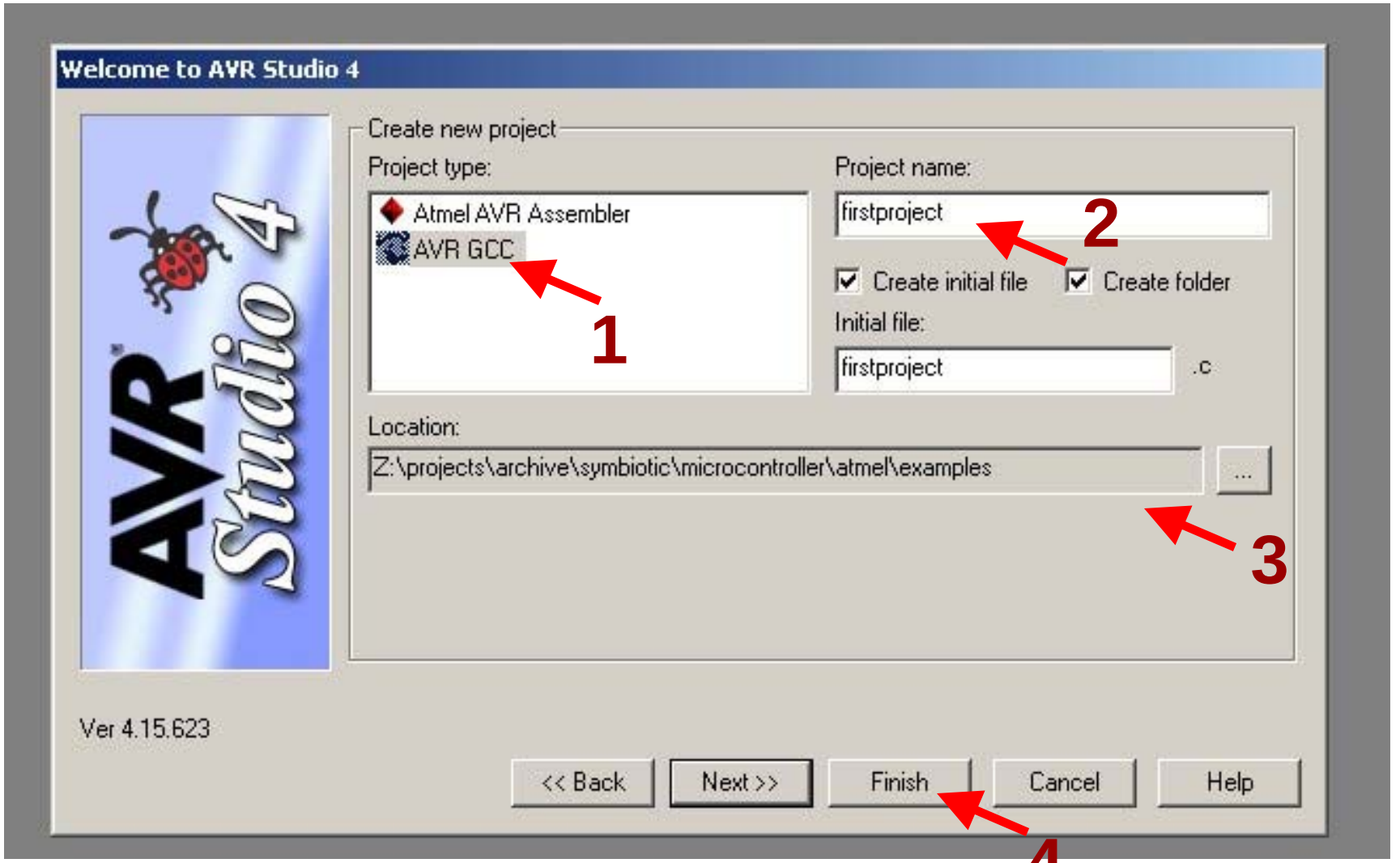
Compiling and Downloading (the easy way)

- Obtain a copy of the “makefile”
  - Modify the “TARGET” line for your program
- Type “make”
  - You should see no errors
- Type “make program”
  - This will download your code to the bion
  - Again, you should see no errors

Getting Started



Project Menu: New Project

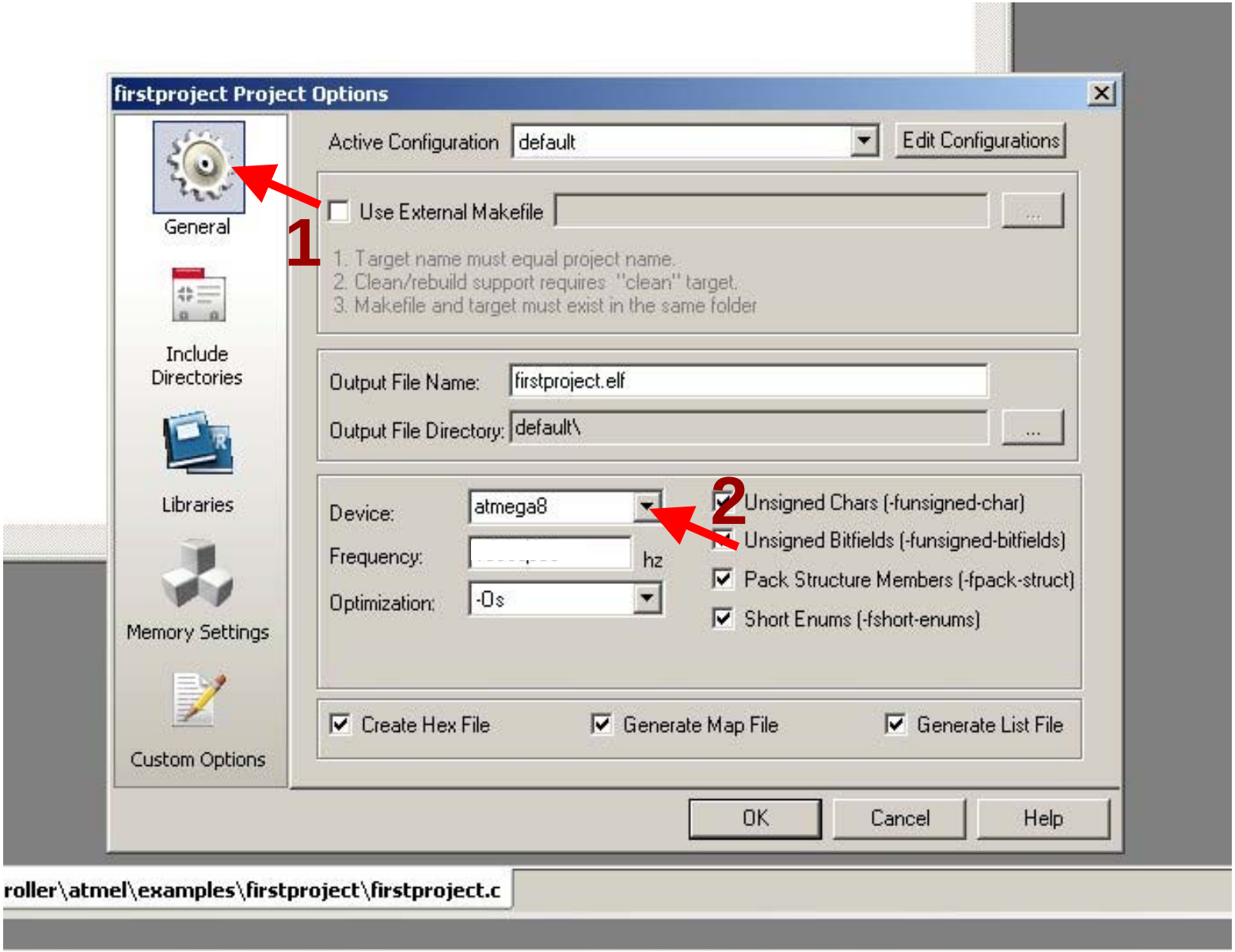


Back to the OS…

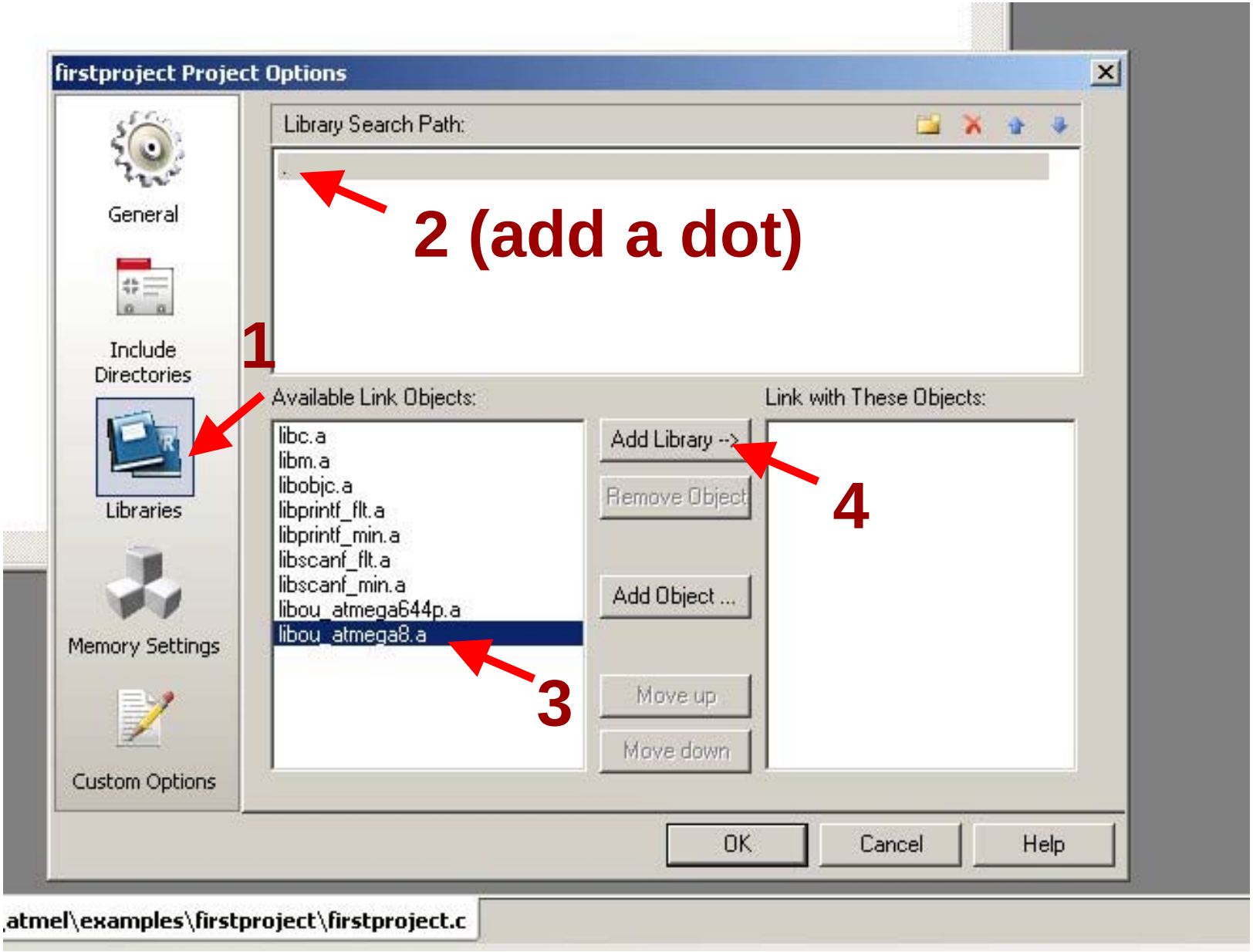
Copy the following to your “firstproject” folder:

- oulib.h
- libou_atmega8.a
- (useful later): oulib_serial_buffered.h

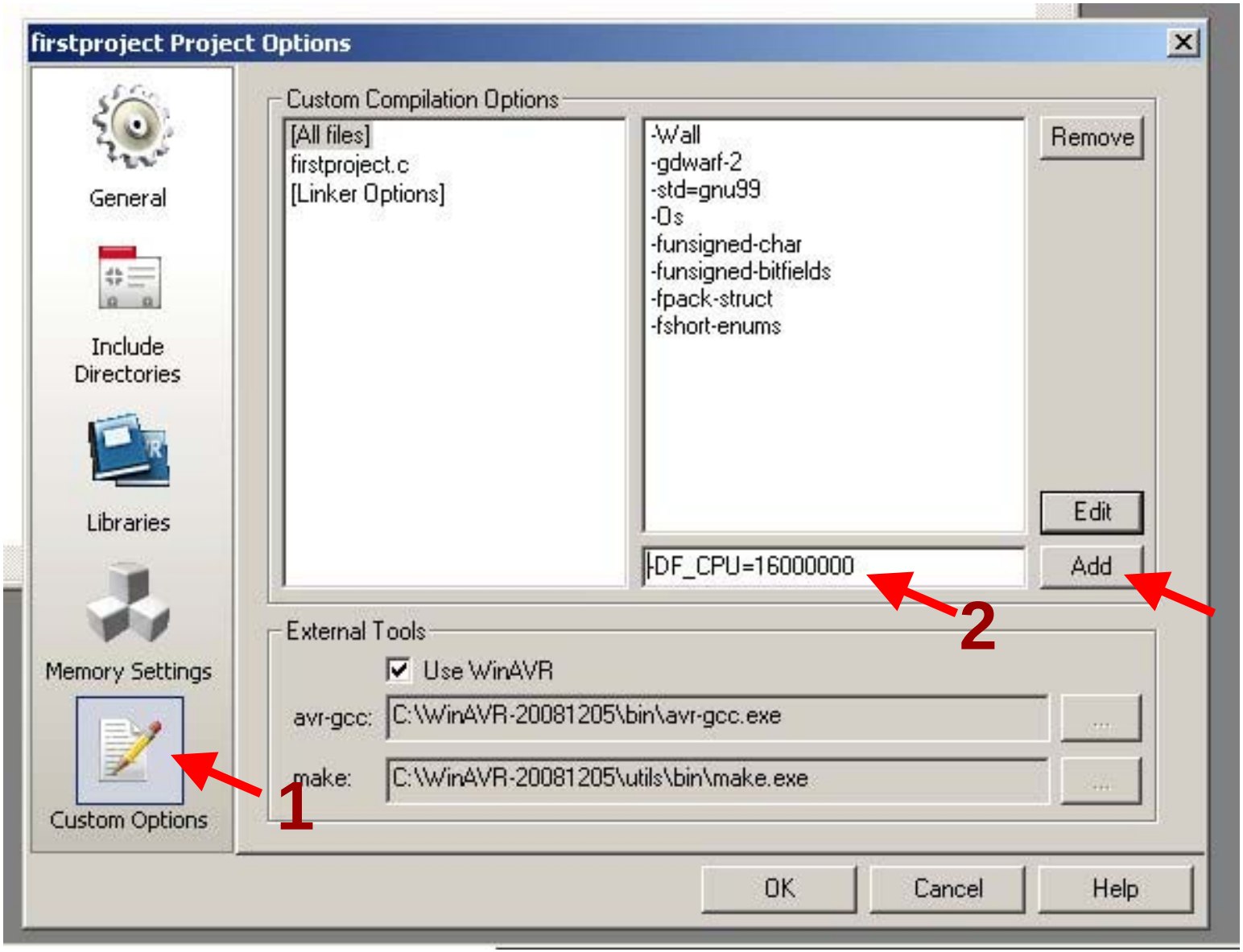
Project Menu: Configuration Options



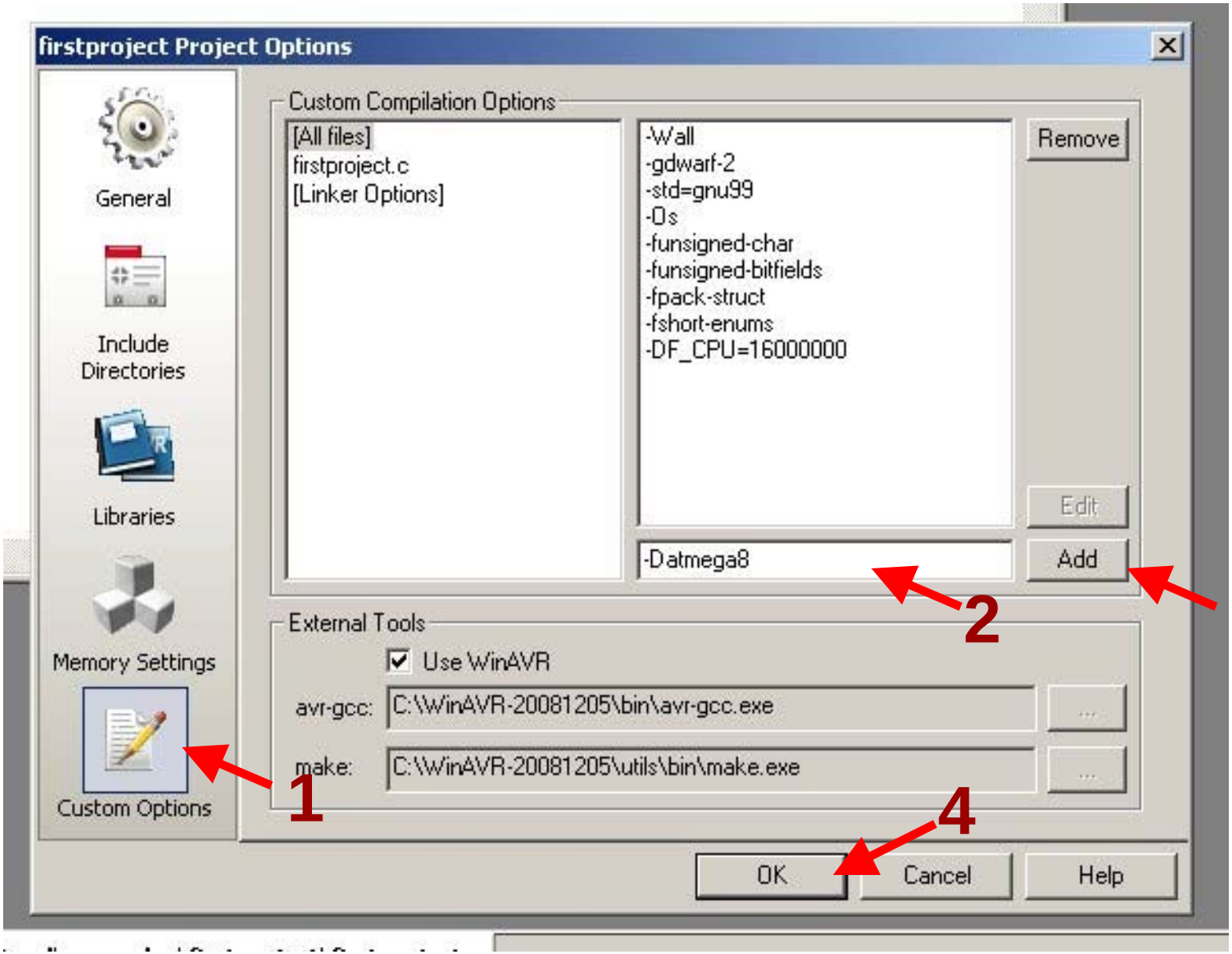
Project Menu: Configuration Options

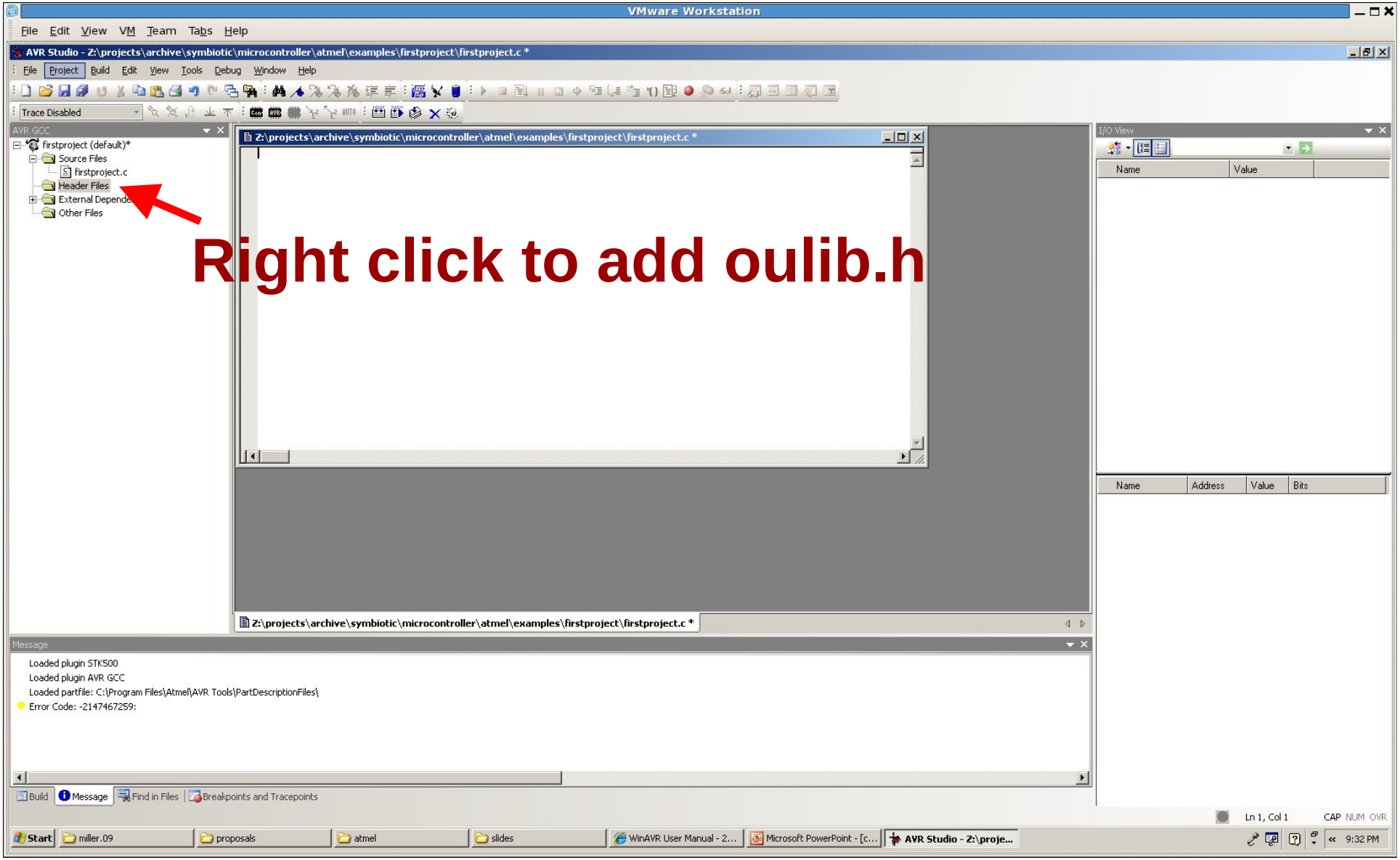


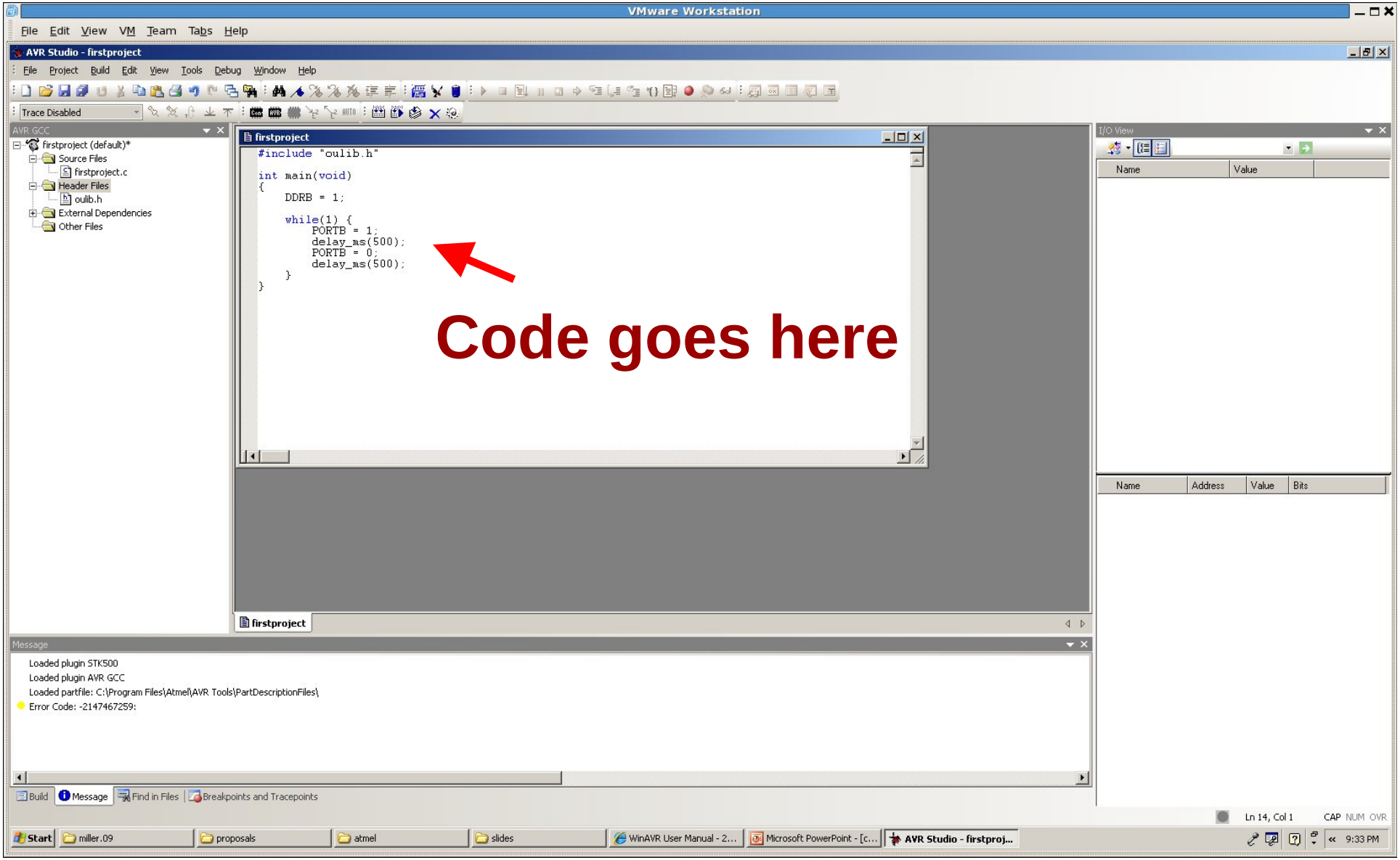
Project Menu: Configuration Options



Project Menu: Configuration Options

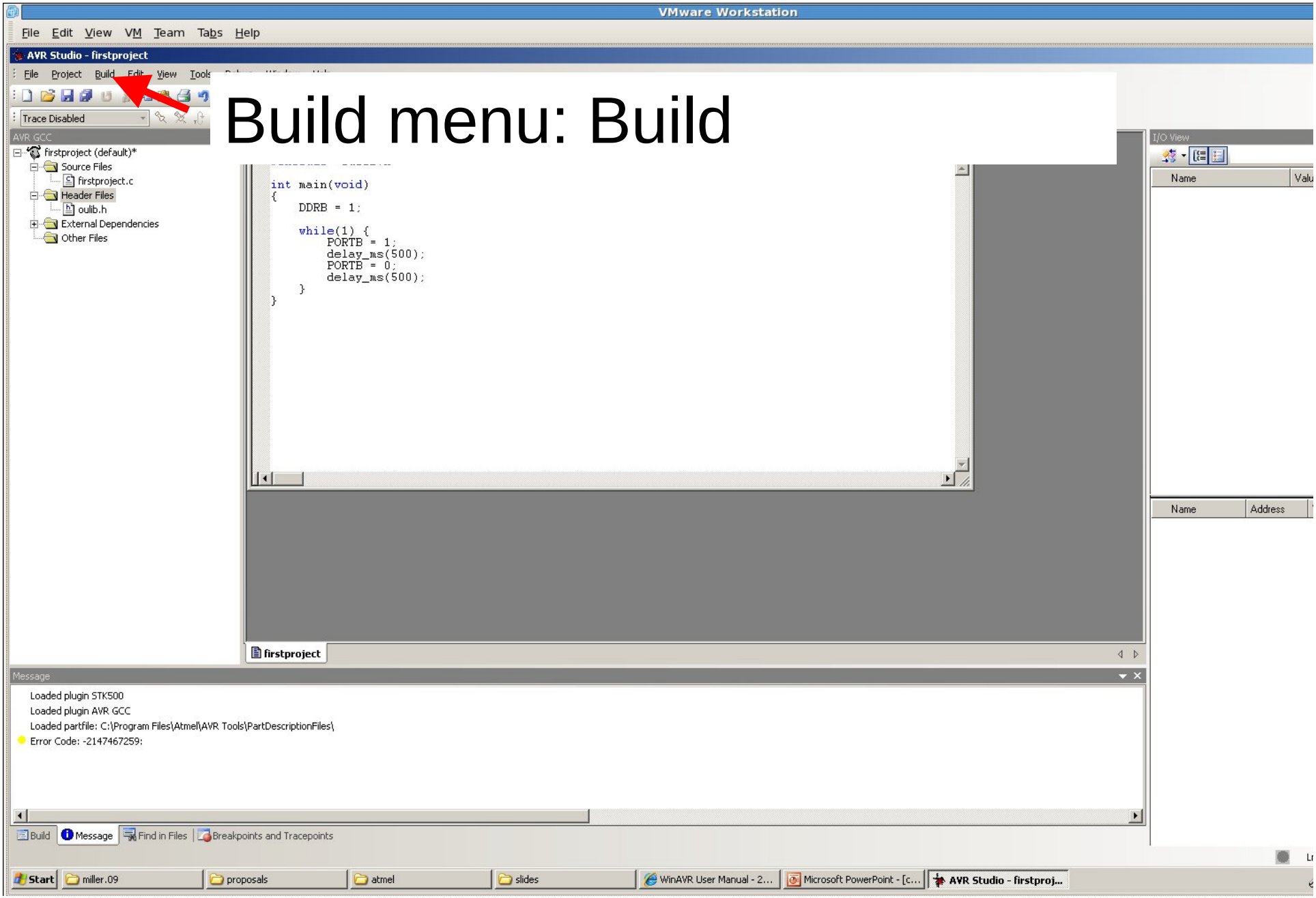



VMware Workstation
File Edit View VM Team Tabs Help
AVR Studio - Z:\projects\archive\symbiotic\microcontroller\atmel\examples\firstproject\firstproject.c *
File Project Build Edit View Tools Debug Window Help
Trace Disabled
AVR GCC
firstproject (default)*
Source Files
firstproject.c
Header Files
External Depende
Other Files
Right click to add oulib.h
Z:\projects\archive\symbiotic\microcontroller\atmel\examples\firstproject\firstproject.c *
I/O View
Name
Value
Name
Address
Value
Bits
Message
Loaded plugin STK500
Loaded plugin AVR GCC
Loaded partfile: C:\Program Files\Atmel\AVR Tools\PartDescriptionFiles\
Error Code: -2147467259:
Build
Message
Find in Files
Breakpoints and Tracepoints
Ln 1, Col 1
CAP NUM OVR
Start
miller.09
proposals
atmel
slides
WinAVR User Manual - 2...
Microsoft PowerPoint - [c...
AVR Studio - Z:\proje...
9:32 PM

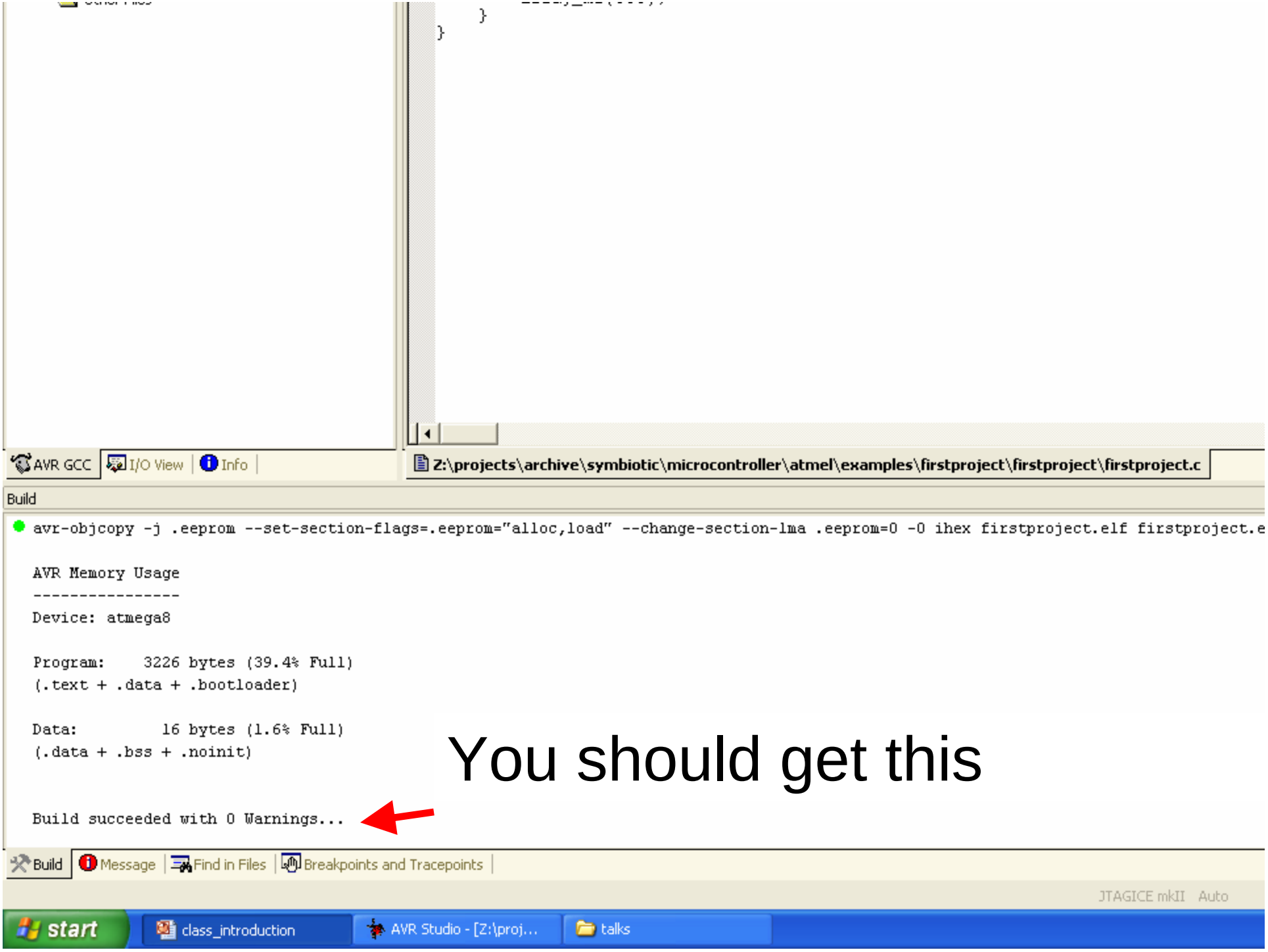

VMware Workstation
File Edit View VM Team Tabs Help
AVR Studio - firstproject
File Project Build Edit View Tools Debug Window Help
Trace Disabled
AVR GCC
firstproject (default)*
Source Files
firstproject.c
Header Files
oulib.h
External Dependencies
Other Files
firstproject
#include "oulib.h"
int main(void)
{
DDRB = 1;
while(1) {
PORTB = 1;
delay_ms(500);
PORTB = 0;
delay_ms(500);
}
}
Code goes here
I/O View
Name
Value
Name
Address
Value
Bits
Message
Loaded plugin STK500
Loaded plugin AVR GCC
Loaded partfile: C:\Program Files\Atmel\AVR Tools\PartDescriptionFiles\
Error Code: -2147467259:
Build
Message
Find in Files
Breakpoints and Tracepoints
Ln 14, Col 1
CAP NUM OVR
Start
miller.09
proposals
atmel
slides
WinAVR User Manual - 2...
Microsoft PowerPoint - [c...
AVR Studio - firstproj...
9:33 PM

Now for the code…

```
#include "oulib.h"

int main(void)
{
   DDRB = 1;

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



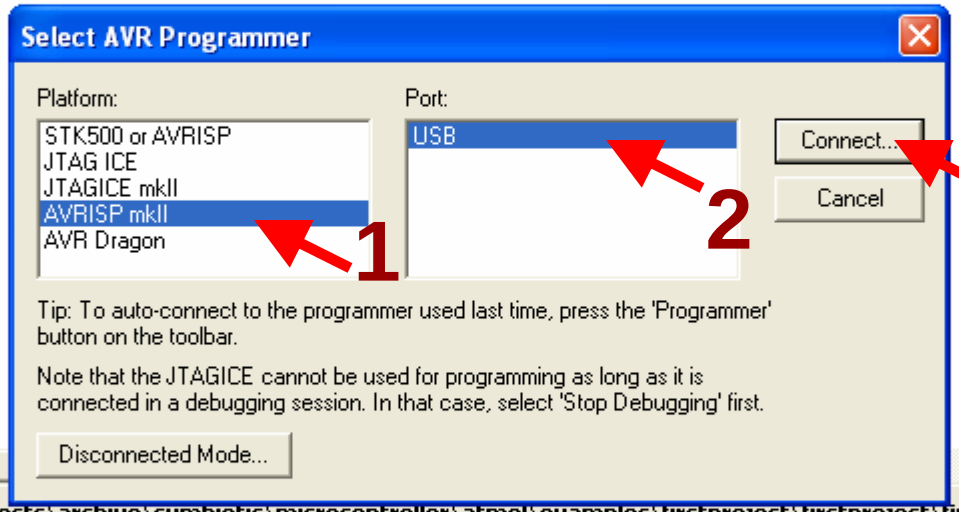
Build menu: Build


}
}
AVR GCC
I/O View
Info
Z:\projects\archive\symbiotic\microcontroller\atmel\examples\firstproject\firstproject\firstproject.c
Build
avr-objcopy -j .eeprom --set-section-flags=.eeprom="alloc,load" --change-section-lma .eeprom=0 -O ihex firstproject.elf firstproject.e
AVR Memory Usage
Device: atmega8
Program: 3226 bytes (39.4% Full)
(.text + .data + .bootloader)
Data: 16 bytes (1.6% Full)
(.data + .bss + .noinit)
Build succeeded with 0 Warnings...
You should get this
Build
Message
Find in Files
Breakpoints and Tracepoints
JTAGICE mkII Auto
start
class_introduction
AVR Studio - [Z:\proj...
talks

Now We Are Ready…

- Plug the programmer into the bion (If it is not already)
- Power up the bion
- And download the program…
  - Tools Menu: AVR: Connect

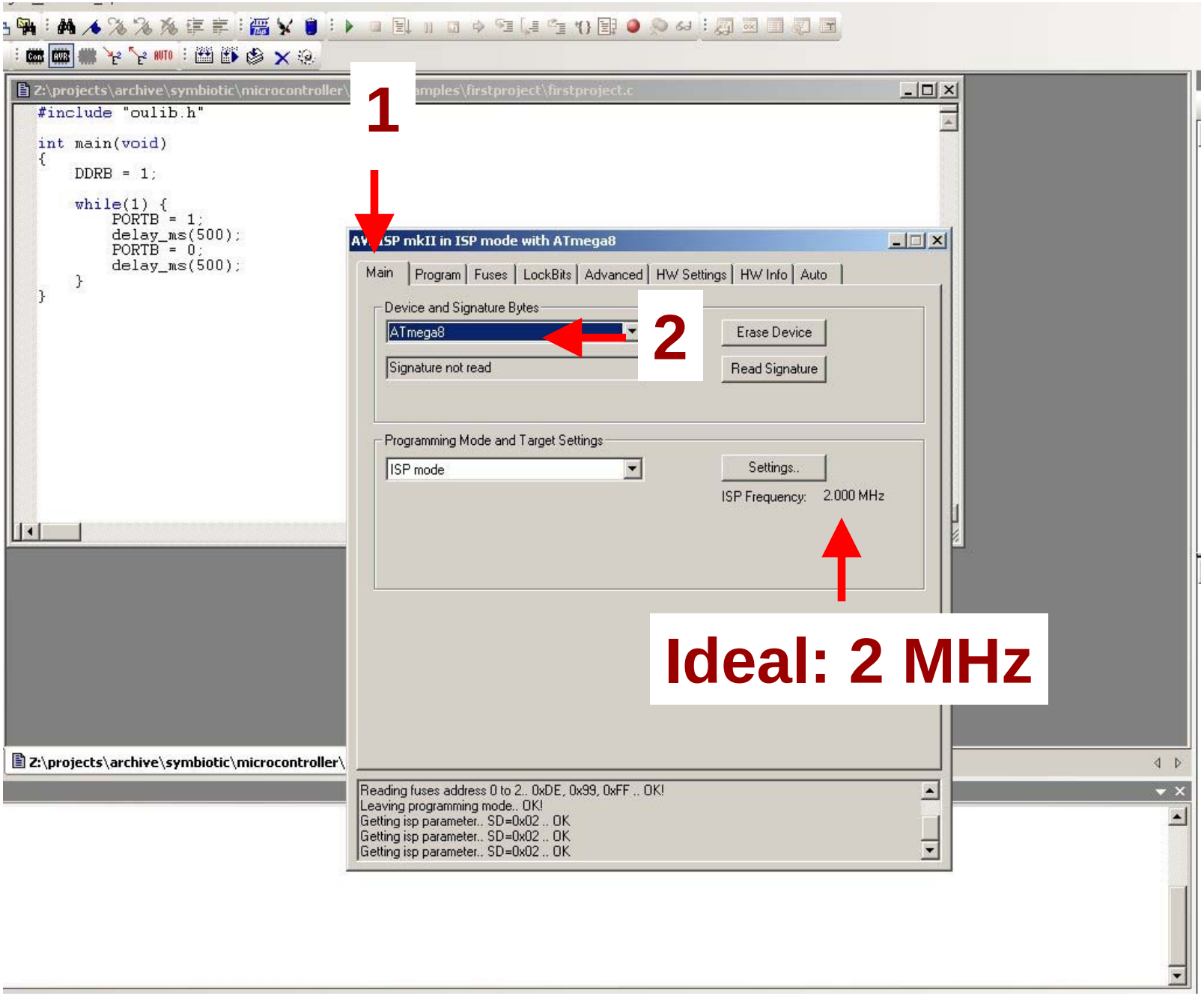
```
int main(void)
{
    DDRB = 7;

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



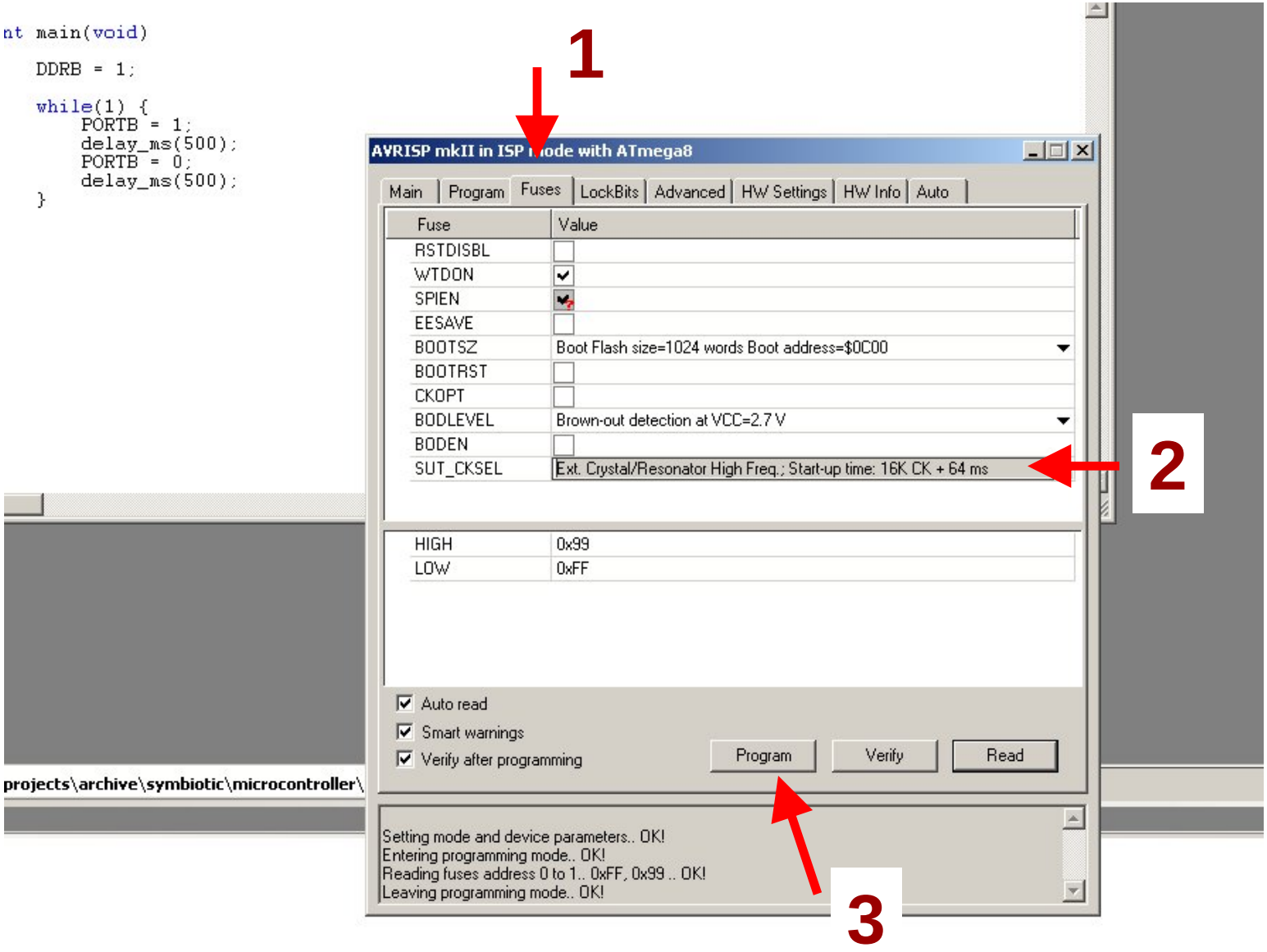
Info

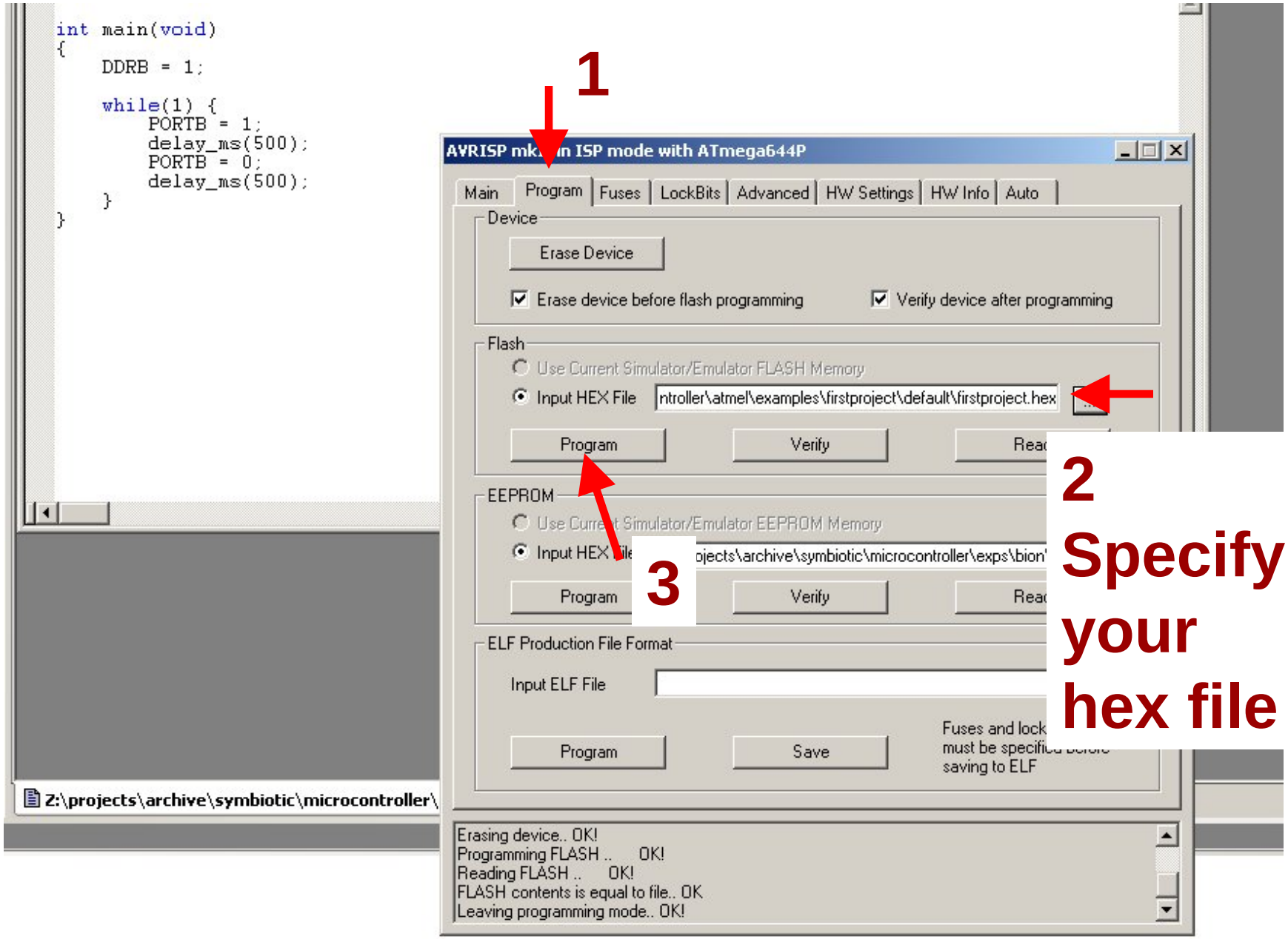
Z:\projects\archive\symbiotic\microcontroller\atmel\examples\firstproject\firstproject\firstproject.c

```
.eeprom --set-section-flags=.eeprom="alloc,load" --change-section-lma .eeprom=0 -O ihex firstproject.elf firstproject.eep

6 bytes (39.4% Full)
+ .bootloader)

6 bytes (1.6% Full)
 .noinit)
```


Z:\projects\archive\symbiotic\microcontroller\
#include "oulib.h"
int main(void)
{
DDRB = 1;
while(1) {
PORTB = 1;
delay_ms(500);
PORTB = 0;
delay_ms(500);
}
}
1
AVR ISP mkII in ISP mode with ATmega8
Main
Program
Fuses
LockBits
Advanced
HW Settings
HW Info
Auto
Device and Signature Bytes
ATmega8
Signature not read
Erase Device
Read Signature
2
Programming Mode and Target Settings
ISP mode
Settings..
ISP Frequency: 2.000 MHz
Ideal: 2 MHz
Reading fuses address 0 to 2.. 0xDE, 0x99, 0xFF .. OK!
Leaving programming mode.. OK!
Getting isp parameter.. SD=0x02 .. OK
Getting isp parameter.. SD=0x02 .. OK
Getting isp parameter.. SD=0x02 .. OK

(should only need to do this once)




int main(void)
{
DDRB = 1;
while(1) {
PORTB = 1;
delay_ms(500);
PORTB = 0;
delay_ms(500);
}
}
1
AVRISP mk in ISP mode with ATmega644P
Main
Program
Fuses
LockBits
Advanced
HW Settings
HW Info
Auto
Device
Erase Device
Erase device before flash programming
Verify device after programming
Flash
Use Current Simulator/Emulator FLASH Memory
Input HEX File
ntroller\atmel\examples\firstproject\default\firstproject.hex
Program
Verify
EEPROM
Use Current Simulator/Emulator EEPROM Memory
Input HEX
ojects\archive\symbiotic\microcontroller\exps\bion
Program
Verify
3
ELF Production File Format
Input ELF File
Program
Save
Fuses and lock
must be specifi
saving to ELF
Erasing device.. OK!
Programming FLASH .. OK!
Reading FLASH .. OK!
FLASH contents is equal to file.. OK
Leaving programming mode.. OK!
Z:\projects\archive\symbiotic\microcontroller\
2
Specify
your
hex file

Flashing?

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

Your green Light Emitting Diode should be blinking at 1 Hertz (once per second)