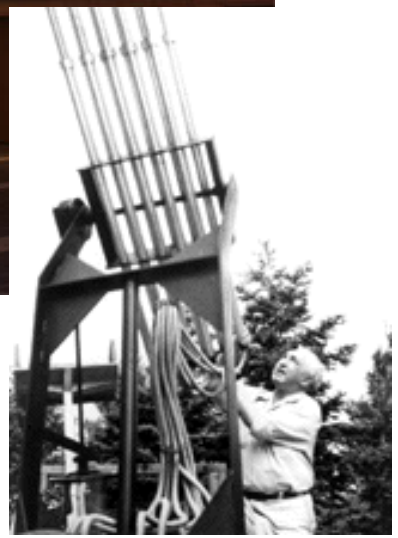
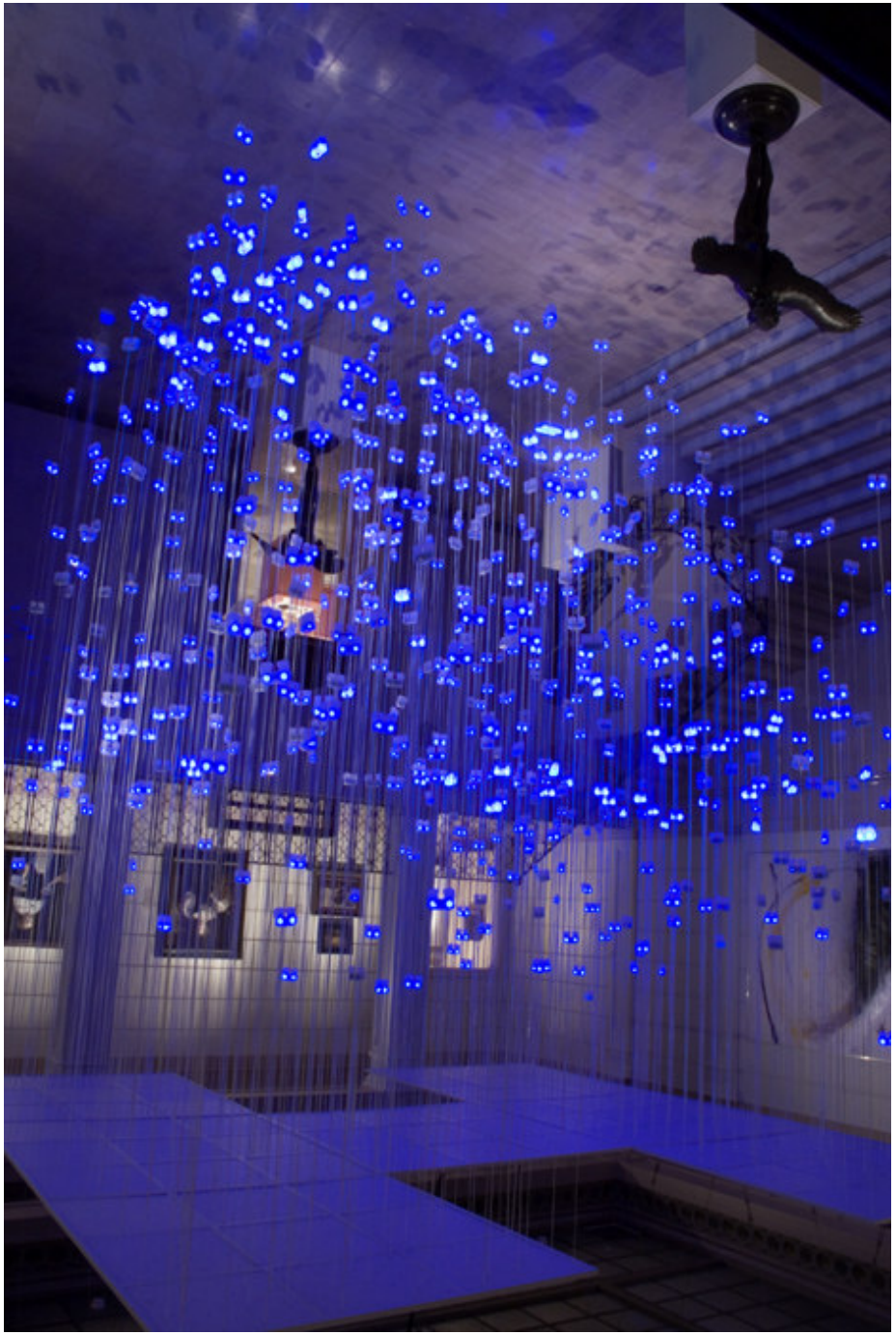


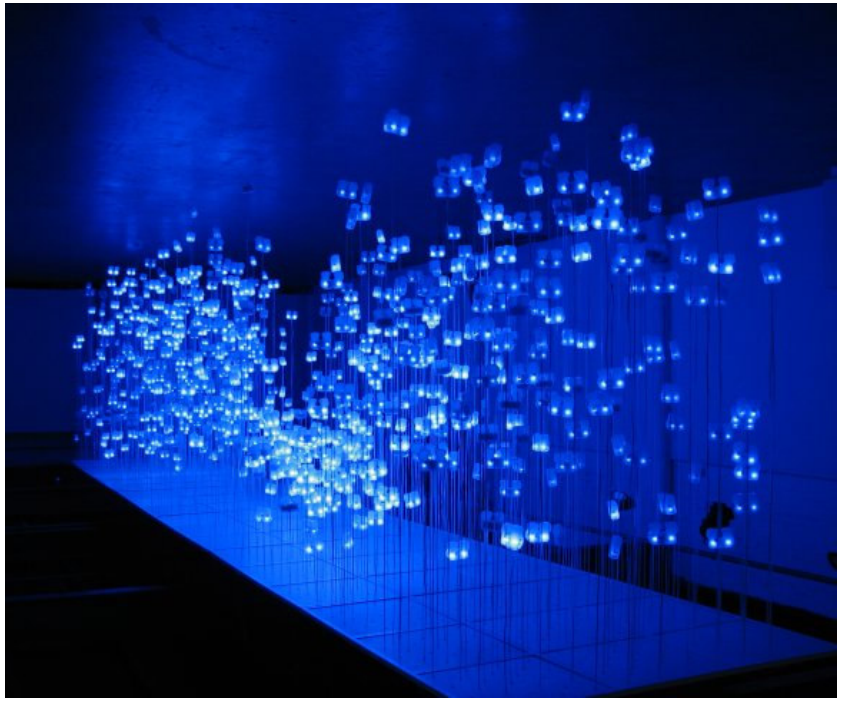
Bion

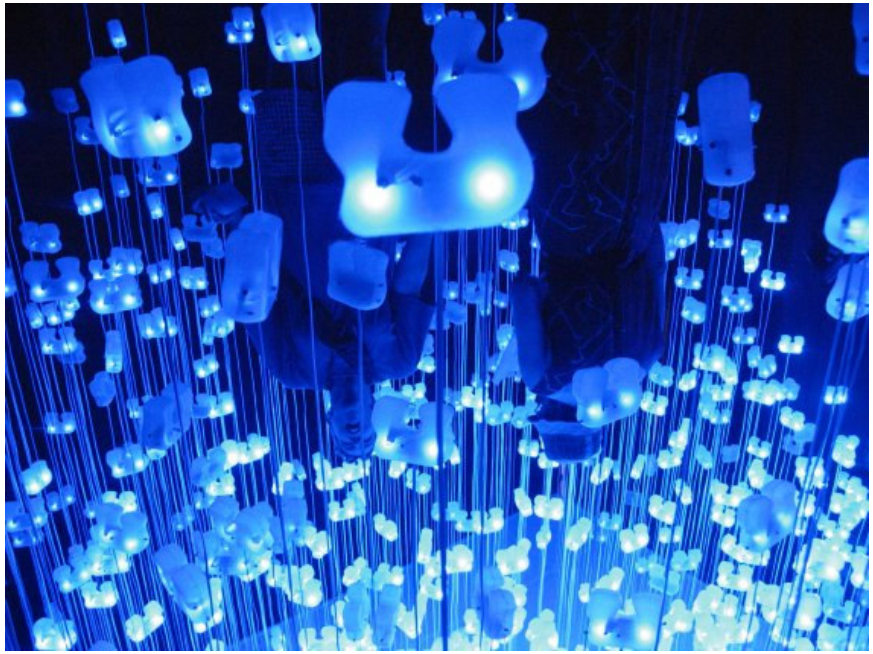
Sensor network:

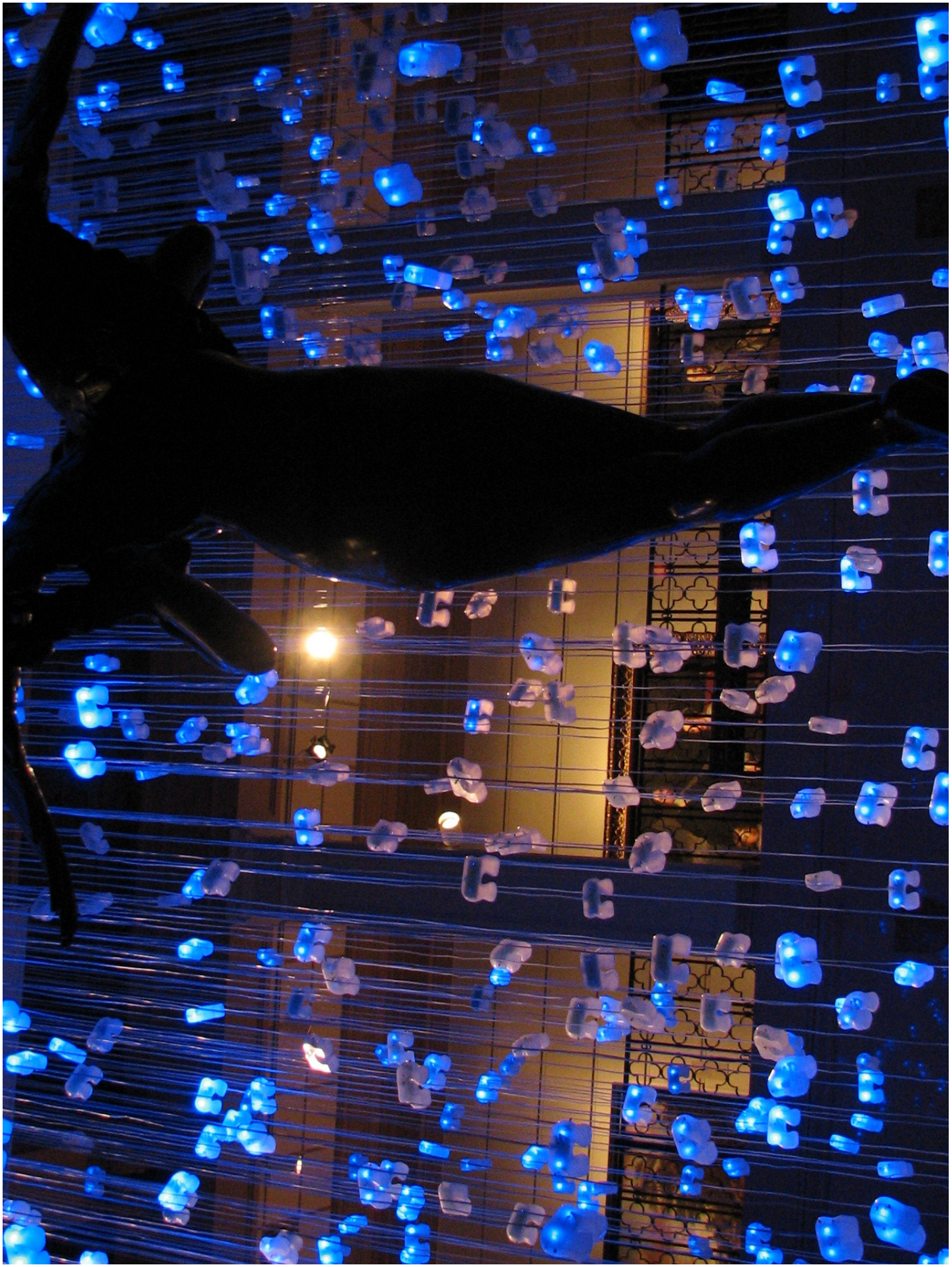
- 1000 sensor nodes
- 3 miles of telephone cable

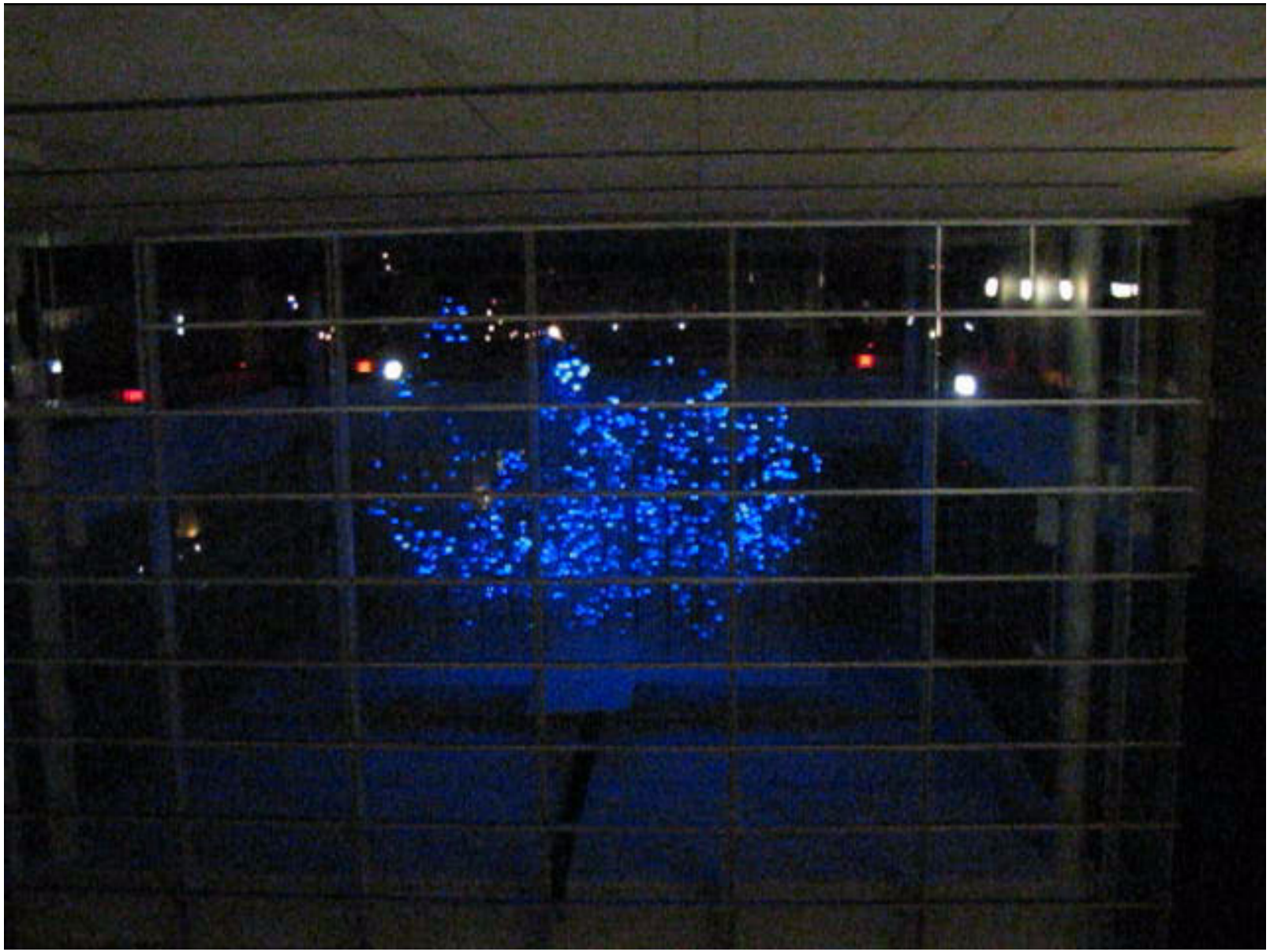
Wilhelm Reich

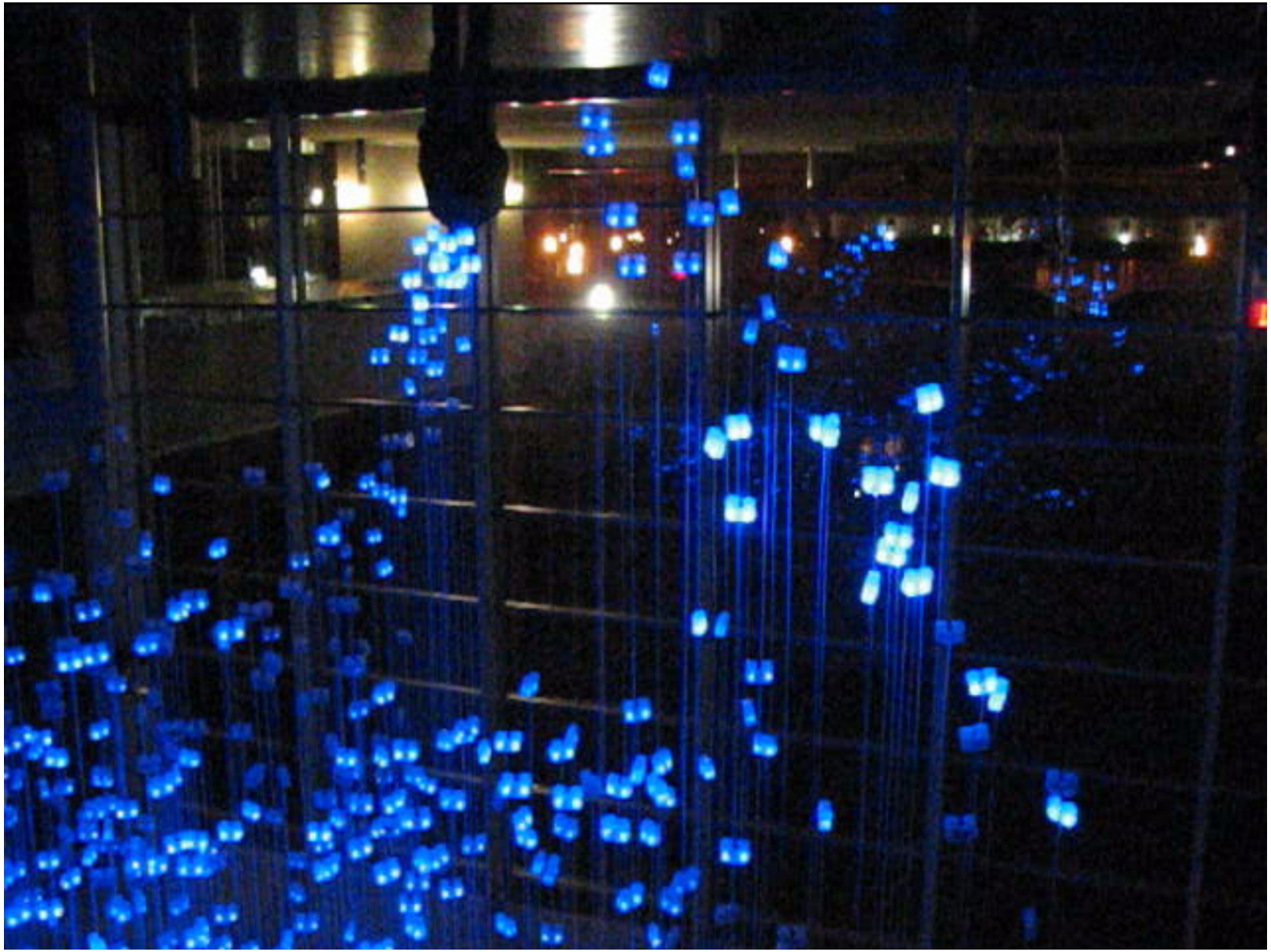






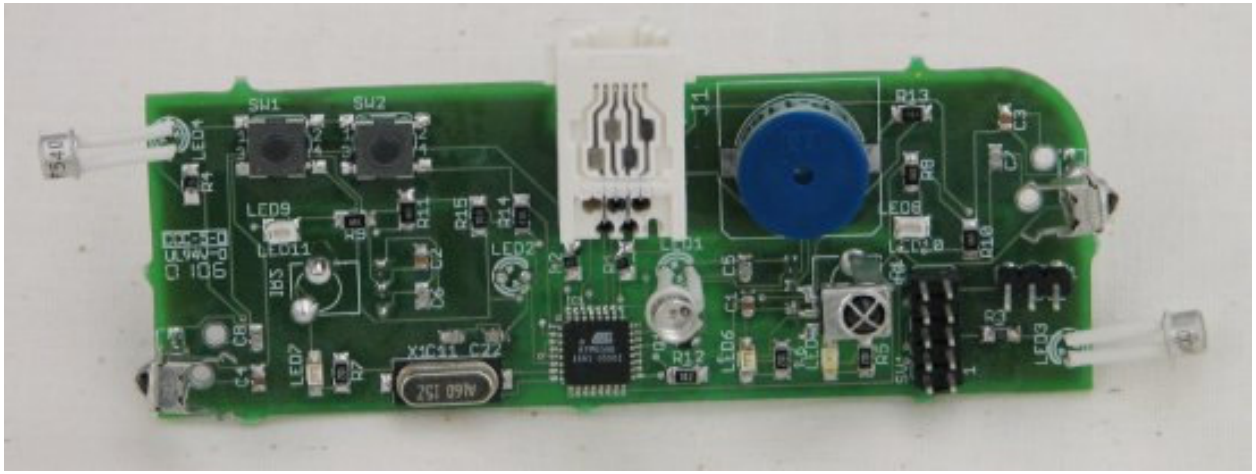






Project 1: Digital I/O and Timing

- Control of LEDs and Speaker
- Respond to button presses



Part 1

- Internal 4-bit (software) counter
- Counter state is reflected by the LEDs
  - Bit 0: Blue
  - Bit 1: Yellow
  - Bit 2: Red
  - Bit 3: Green
- Each button press: increment counter
  - Update LED display

Part 2

- Generate tone with the speaker
  - Different tone for each counter state (higher tones for higher values)
  - Produce tone as long as the button is being pressed
- Speaker is controlled by a digital I/O line
  - So: in one of two states
  - Tones are produced by generating a “square wave” at a given frequency

Project Administrivia

Due in one week (Feb 27th)

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

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

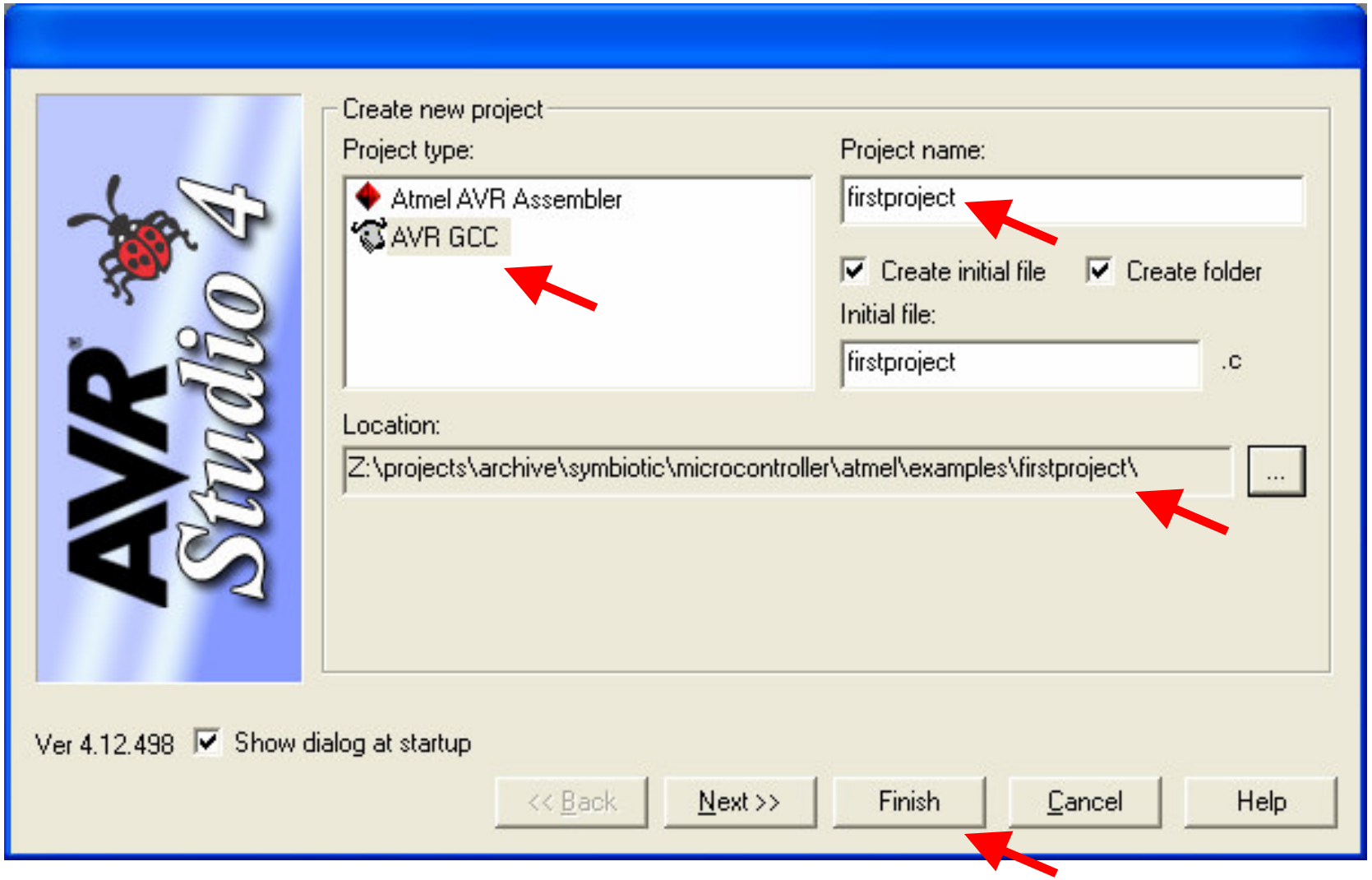
Getting Started

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

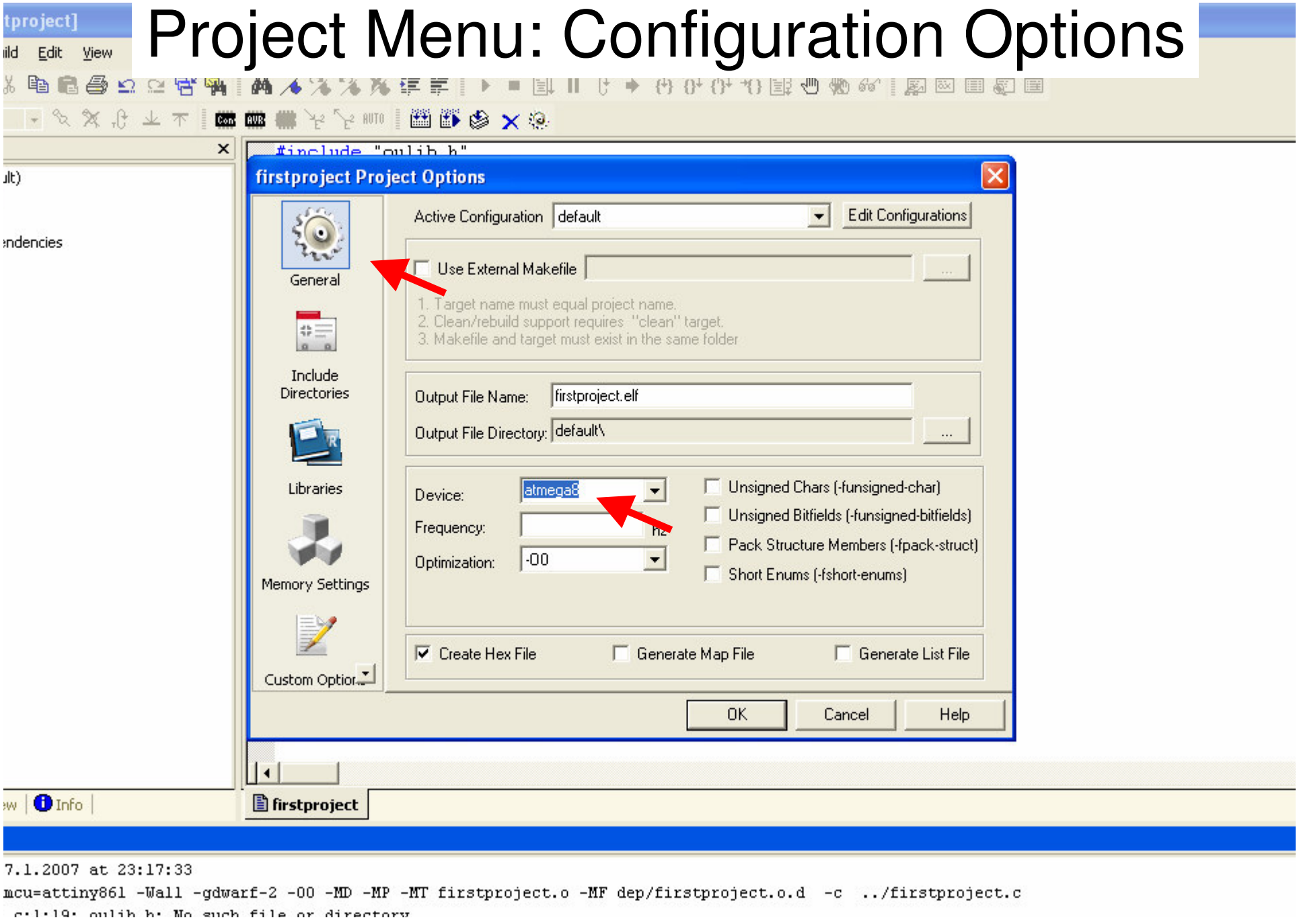
Summary:

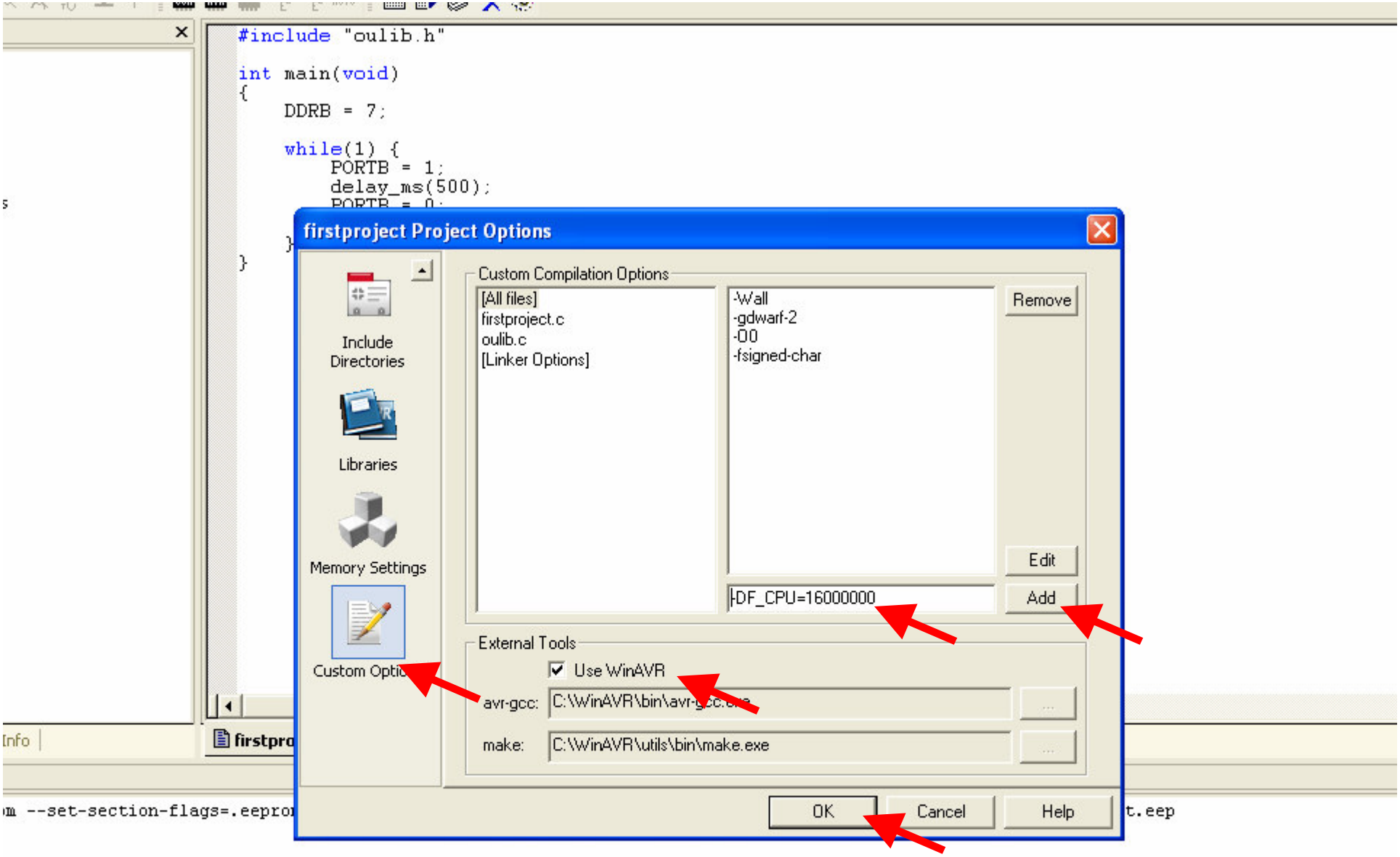
- (perhaps) Install AVRstudio
- (perhaps) Install WinAVR
- Plug the programmer into your computer
- Plug the programmer into the bion
- Plug the power into the bion
- Create a program

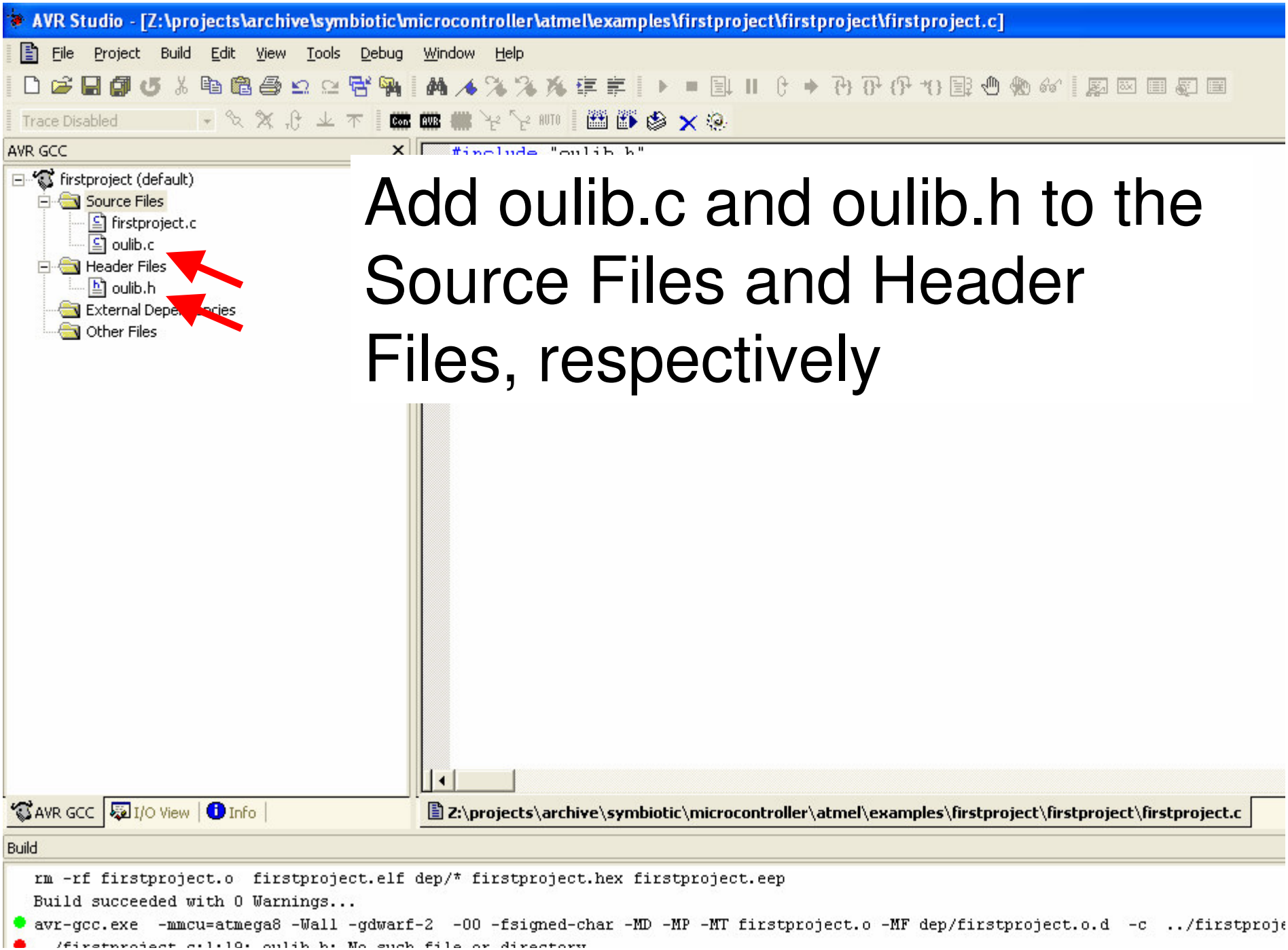
Project Menu: New Project



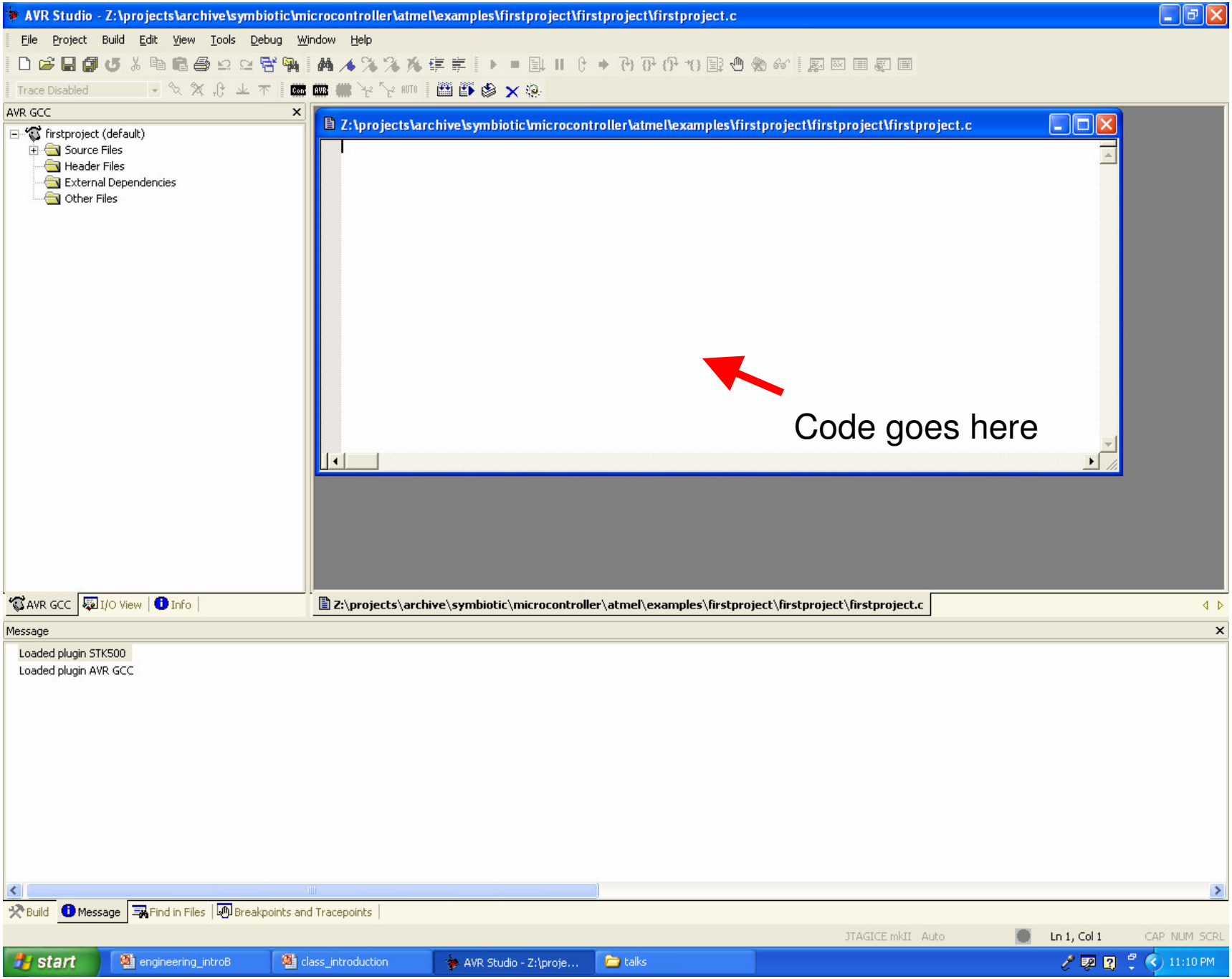
Project Menu: Configuration Options




#include "oulib.h"
int main(void)
{
DDRB = 7;
while(1) {
PORTB = 1;
delay_ms(500);
}
}
firstproject Project Options
Include Directories
Libraries
Memory Settings
Custom Optio
Custom Compilation Options
[All files]
firstproject.c
oulib.c
[Linker Options]
-Wall
-gdwarf-2
-O0
-fsigned-char
Remove
Edit
-DF_CPU=16000000
Add
External Tools
Use WinAVR
avr-gcc:
C:\WinAVR\bin\avr-gcc.exe
make:
C:\WinAVR\utils\bin\make.exe
OK
Cancel
Help



Add oulib.c and oulib.h to the Source Files and Header Files, respectively

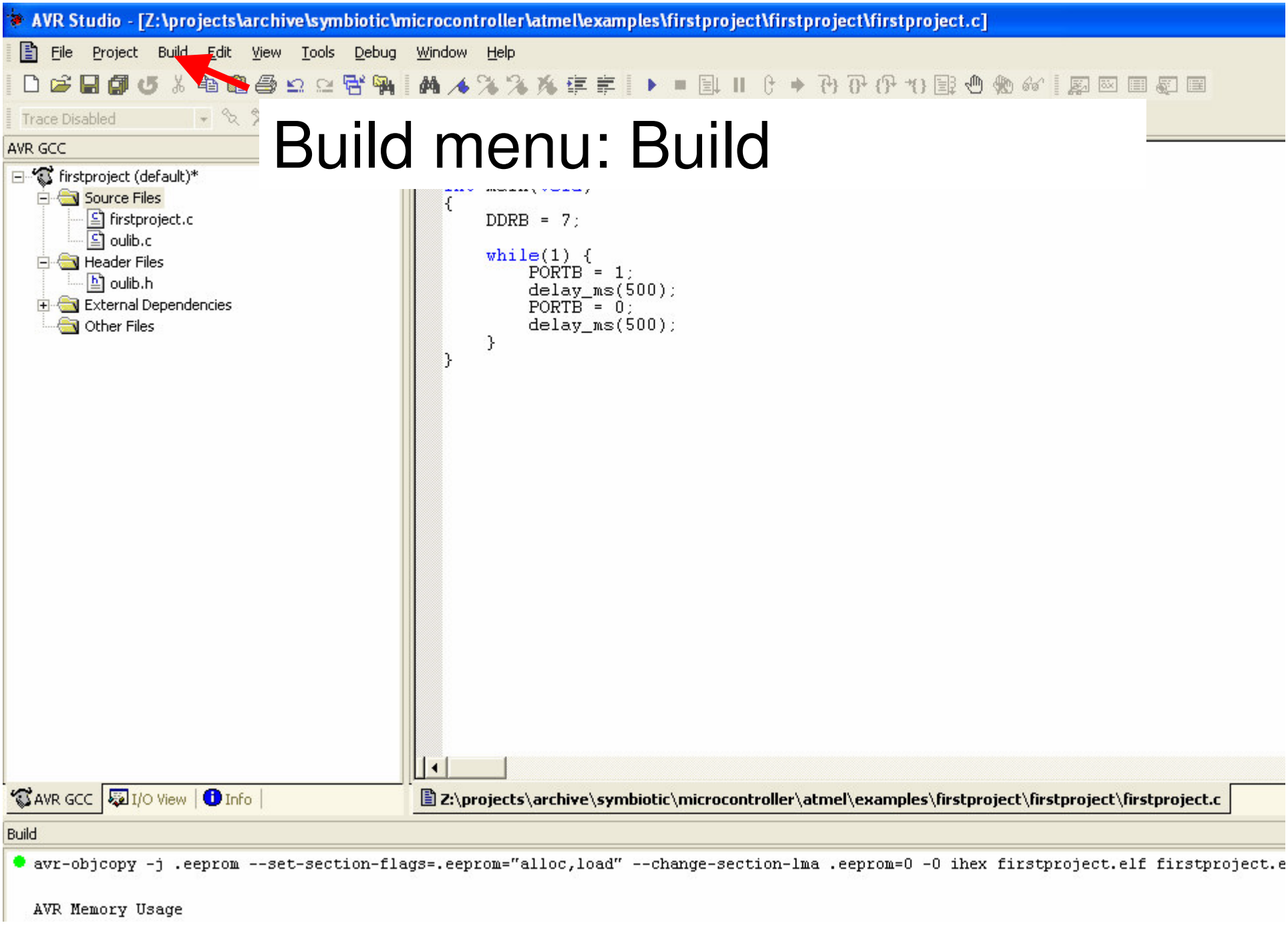

AVR Studio - Z:\projects\archive\symbiotic\microcontroller\atmel\examples\firstproject\firstproject\firstproject.c
File Project Build Edit View Tools Debug Window Help
Trace Disabled
AVR GCC
firstproject (default)
Source Files
Header Files
External Dependencies
Other Files
Z:\projects\archive\symbiotic\microcontroller\atmel\examples\firstproject\firstproject\firstproject.c
Code goes here
AVR GCC I/O View Info
Z:\projects\archive\symbiotic\microcontroller\atmel\examples\firstproject\firstproject\firstproject.c
Message
Loaded plugin STK500
Loaded plugin AVR GCC
Build Message Find in Files Breakpoints and Tracepoints
JTAGICE mkII Auto
Ln 1, Col 1
CAP NUM SCRL
start
engineering_introB
class_introduction
AVR Studio - Z:\proje...
talks
11:10 PM

Now for the code…

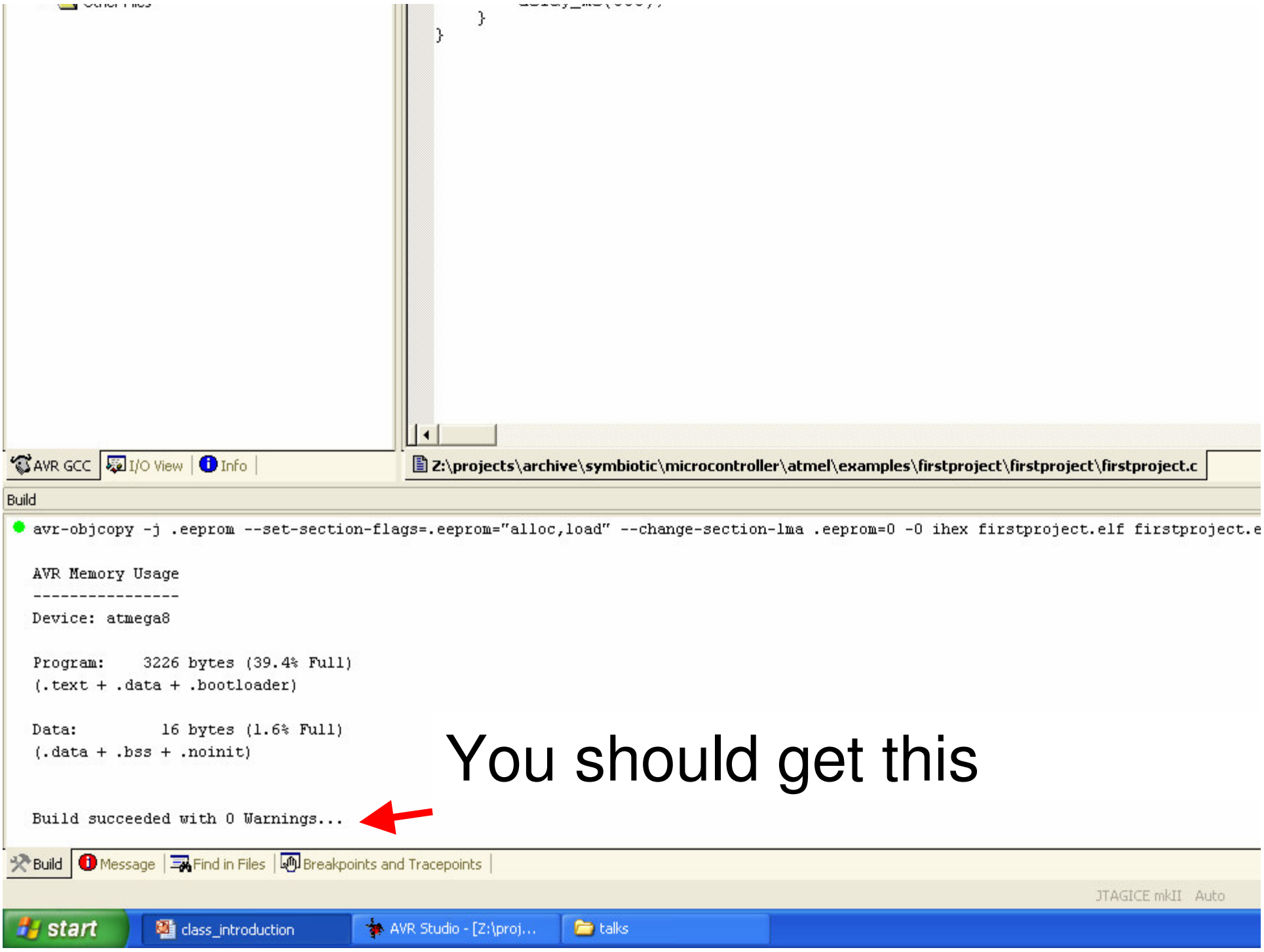
```
#include "oulib.h"

int main(void)
{
   DDRB = 7;

   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

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)

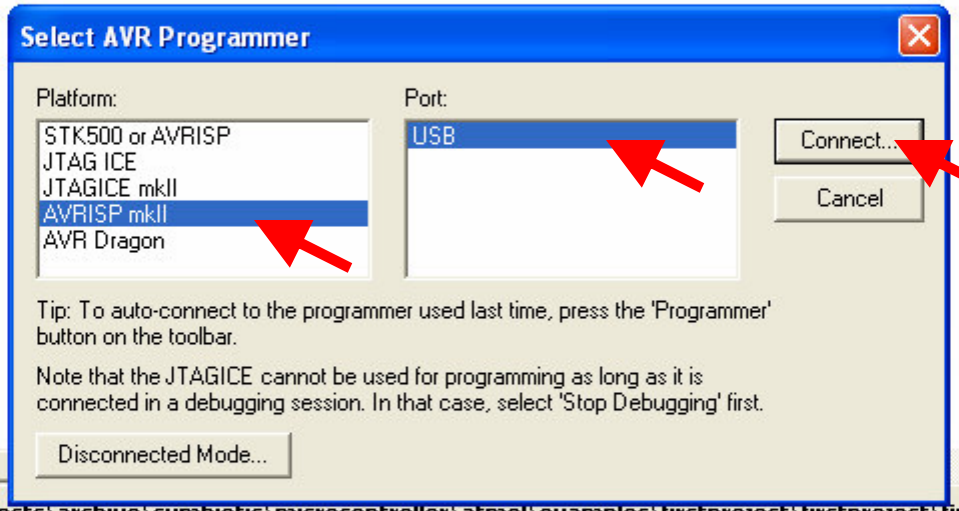
Be careful when you plug your circuit in (check before powering)

AVR ISPs are Cranky

- When things are plugged in and powered, you should see two green LEDs on the ISP
- One red: usually means that your circuit is not powered
- 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

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

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



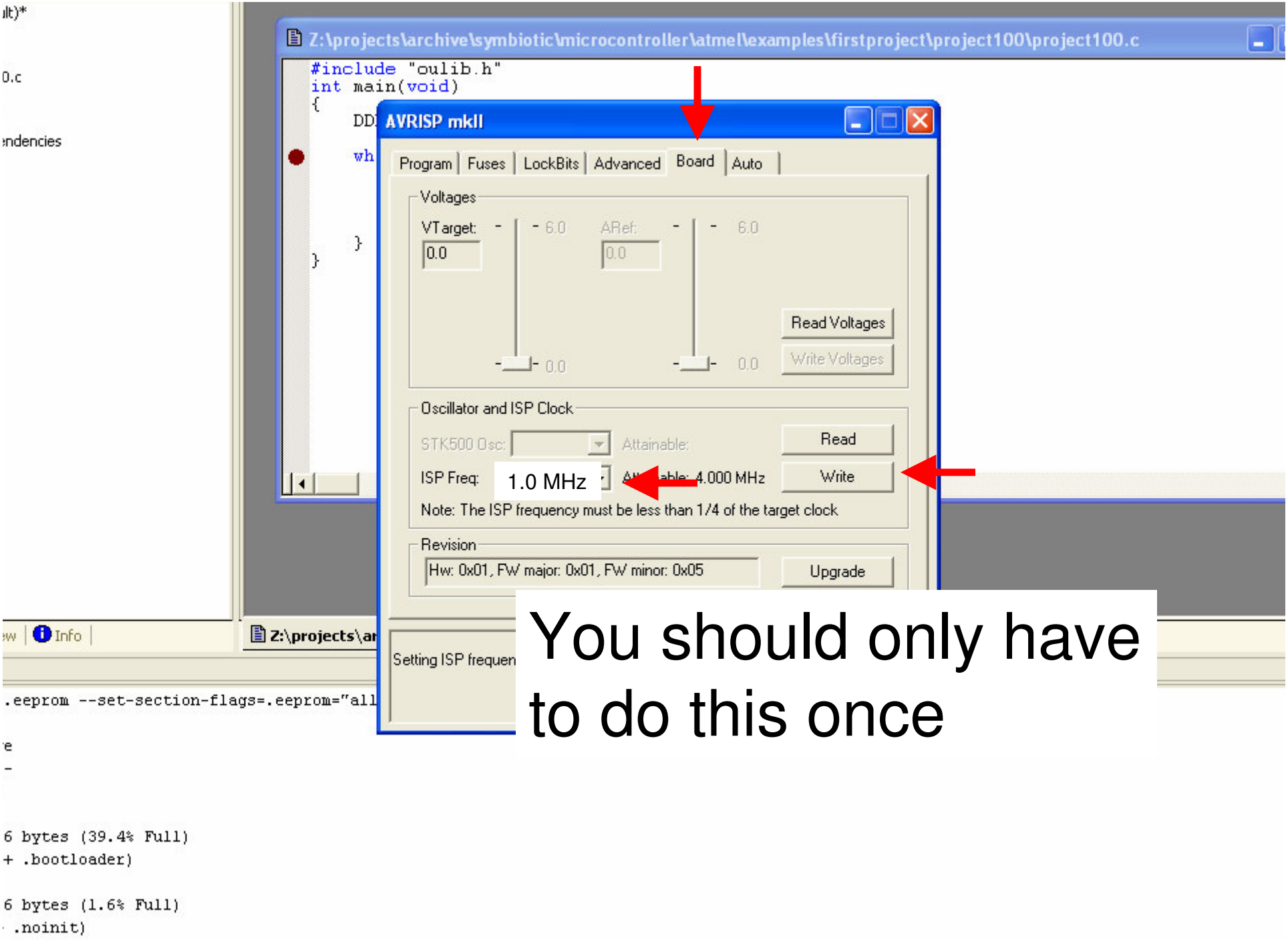
Info

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

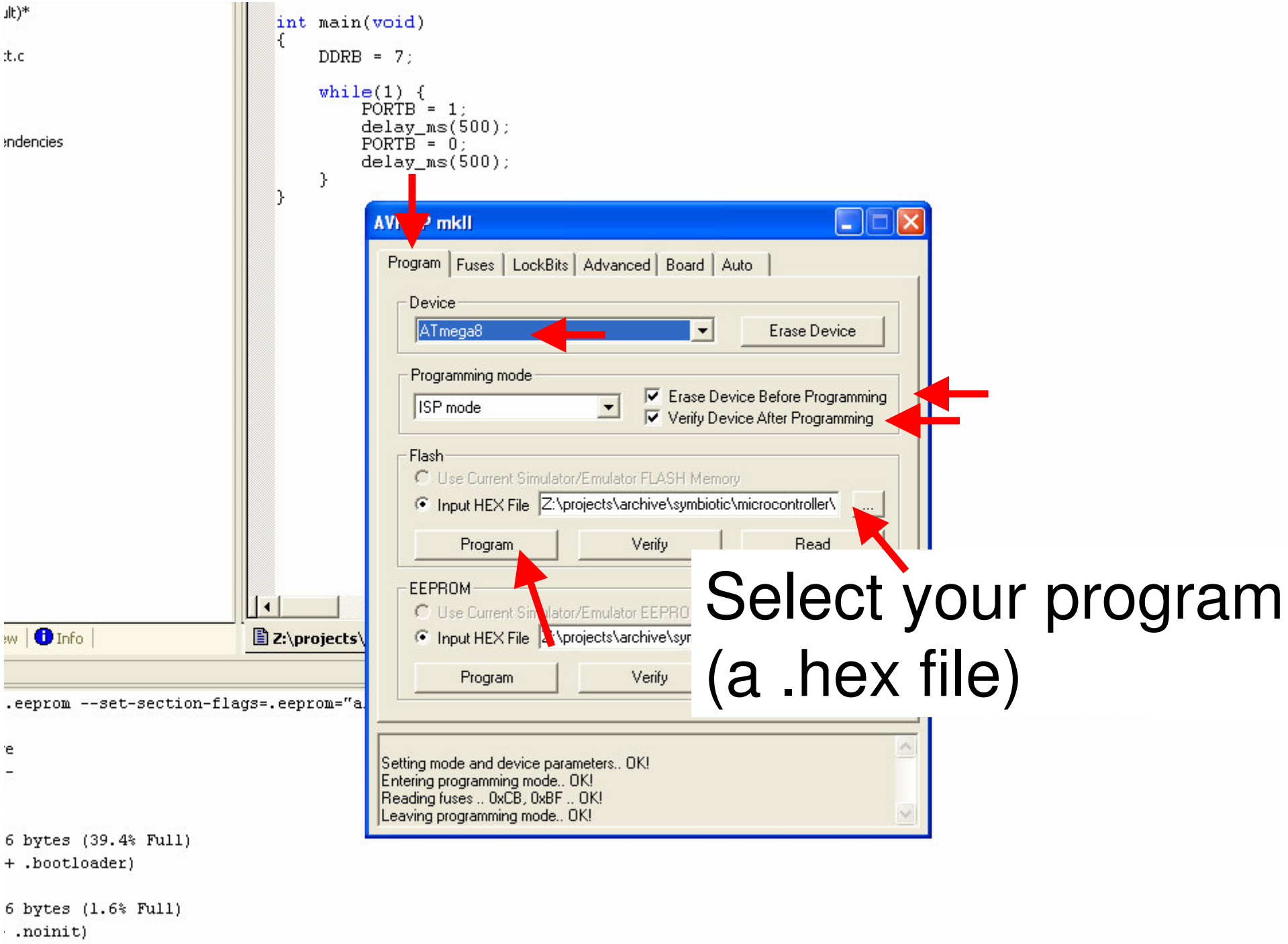
```
.eeprom --set-section-flags=.eeprom="alloc,load" --change-section-lma .eeprom=0 -O ihex firstproject.elf firstproject.eep

6 bytes (39.4% Full)
+ .bootloader)

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



You should only have to do this once


int main(void)
{
DDRB = 7;
while(1) {
PORTB = 1;
delay_ms(500);
PORTB = 0;
delay_ms(500);
}
}
mkII
Program
Fuses
LockBits
Advanced
Board
Auto
Device
ATmega8
Erase Device
Programming mode
ISP mode
Erase Device Before Programming
Verify Device After Programming
Flash
Use Current Simulator/Emulator FLASH Memory
Input HEX File
Z:\projects\archive\symbiotic\microcontroller\
Program
Verify
Read
EEPROM
Input HEX File
Program
Verify
Setting mode and device parameters.. OK!
Entering programming mode.. OK!
Reading fuses .. 0xCB, 0xBF .. OK!
Leaving programming mode.. OK!
Info
.eeprom --set-section-flags=.eeprom="a
6 bytes (39.4% Full)
+ .bootloader)
6 bytes (1.6% Full)
.noinit)
Select your program
(a .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)