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
  – Total of 4 individually-controllable channels
  – (the 4 blue LEDs are controlled by one digital line)
• Each button release: increment the counter
Part 2

• Generate tone with the speaker
  – Different tone for each counter state (lower frequencies for higher values)

• 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

Modular code

• Implement a separate function: 
  DisplayCounter()

  Translates the current counter value into the LED state
Project Administrivia

Due in 12 days (Feb 26th)

• Demonstrate to me, Di Wang, or Dan Flippo

• Documented code: hand-in on D2L
  – One copy per 2-person group

• Personal report: distribution of work

• Lab appointments:
  – Check the appointments page
  – Send email to es@cs.ou.edu
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 unix 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
Project Menu: New Project

1. Select the project type.
2. Specify the location.
3. Enter the project name.
4. Proceed to finish the project.
Project Menu: Configuration Options

1. Use External Makefile

2. Device: atmega328
   Frequency: 16 MHz
   Optimization: -00

7.1.2007 at 23:17:33
mcu=attiny861 -Wall -gdwarf-2 -00 -MD -MP -MT firstproject.o -MF dep/firstproject.d -c ..;/firstproject.c

c1:10: `malloc': No such file or directory
```c
#include "oulib.h"

int main(void)
{
    DDRB = 7;
    while(1) {
        PORTB = 1;
        delay_ms(500);
        PORTB = 0;
    }
}
```
Add oulib.h to the Header Files. Note that it must be in the local directory (same as firstproject.c)
1: add "." to the path

2: select

3: select

4: click OK
Now for the code...

```c
#include "oulib.h"

int main(void)
{
    DDRB = 7;

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

```c
{  
  DDRB = 7;
  
  while(1) {
    PORTB = 1;
    delay_us(500);
    PORTB = 0;
    delay_us(500);
  }
}
```
You should get this
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
```c
int main(void)
{
    DDRB = 7;

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

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

```
6 bytes (39.4% Full)
+ .bootloader)

6 bytes (1.6% Full)
  .noinit)
```
Select your program
(a .hex file in the
“default” subfolder)
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)