**Programming the FabISP**

# Computer Setup & Necessary Software

**Programming the FabISP**

**Computer/Software Setup**

Log on to Ubuntu. Open Terminal and then type the following in the command line:

sudo apt-get install flex byacc bison gcc libusb-dev avrdude Next:

sudo apt-get install gcc-avr Then:

sudo apt-get install avr-libc

Final:
sudo apt-get install libc6-dev

Download the FabISP firmware from here (firmware.zip file): <http://academy.cba.mit.edu/classes/electronics_production/index.html>

Save the file on your computer: Desktop> fab\_isp.

Unzip the file (there are two folders- choose the one named "fabISP\_mac.0.8.2\_firmware" and rename it "firmware", then erase everything else in the "fab\_isp" folder). The firmware folder contains several files including a "Makefile" which should have no file extension.

# Programming

Connect FabISP to computer with USB cable Connect Fab ISP to computer via AVR programmer

A green light will come up indicating that the board is getting power

## Navigate to the Firmware Folder

Use the terminal to program your board. When programming, it is necessary for

terminal to have access to the firmware files. Direct the terminal to the location of these files.

In terminal, type:

cd ~/Documents/fab\_isp/firmware

## Check the Makefile

The Makefile is in the firmware folder. By default, it is set up to use the the AVRISP2 programmer. Edit the appropriate line in the Makefile if using another programmer.

To check the makefile, type the following into terminal: nano Makefile

If all looks correct, use "Ctrl x" to exit the makefile

## Clean the Folder

Delete all the extra files in the firmware folder.

Type: make clean

Three of the files were erased and only have "main.c", "Makefile", and "usbconfig.h" files left, along with the "usbdrv" folder.

## Make the Hex File

Compile files in order to create a hex file

Type: make hex

## Set the Fuses

Set the fuses on your microcontroller chip. The fuses define basic parameters of the chip, including things like its operational speed and voltage.

Type:

sudo make fuse

If successful, there should get a message similar to this one:

avrdude -c avrisp2 -P usb -p attiny44 -U hfuse:w:0xDF:m -U lfuse:w:0xFF:m

avrdude: safemode: Verify error - unable to read hfuse properly. Programmer may not be reliable.

avrdude: safemode: Fuses OK (H:FF, E:DF, L:FF) avrdude done. Thank you.

## Make the Program

Upload the program to the chip, so that this board can be used as an ISP.

Type:

sudo make program

Get a message similar to this:

avrdude -c avrisp2 -P usb -p attiny44 -U flash:w:main.hex:i avrdude: AVR device initialized and ready to accept instructions Reading | ##############################

avrdude: Device signature = 0x1e9207

avrdude: NOTE: "flash" memory has been specified, an erase cycle will be performed To disable this feature, specify the -D option.

avrdude: erasing chip

avrdude: reading input file "main.hex" avrdude: writing flash (2002 bytes):

Writing | ##############################

avrdude: 2002 bytes of flash written

avrdude: verifying flash memory against main.hex: avrdude: load data flash data from input file main.hex:

avrdude: input file main.hex contains 2002 bytes avrdude: reading on-chip flash data:

Reading | ##############################

avrdude: verifying ...

avrdude: 2002 bytes of flash verified

avrdude: safemode: Fuses OK (H:FF, E:DF, L:FF) avrdude done. Thank you.

avrdude -c avrisp2 -P usb -p attiny44 -U hfuse:w:0xDF:m -U lfuse:w:0xFF:m

# Verifying the FabISP is Working

Once successfully completed all the previous steps. Keep the terminal window open and type the following:

lsusb

This will provide a list of USB devices connected to the computer. The FabISP will look something like this:

Bus 001 Device 036: ID 1781:0c9f Multiple Vendors USBtiny

If you see this line, the ISP is programmed!

# Using the FabISP as a Programmer

Now that the board is programmed, use it to program other boards.

Remove the two jumpers on the board- the 0 ohm resistor and the solder jumper.

Now the board is ready to be used as a programmer