You will need

Parts: (this should be in you fablab inventory):

FabFTDI is a fabbale usb to serial port converter. It works in all operating system. Windows need a driver. However, Unix and mac doesn’t need any driver. It uses Atmega16U2. The amega has inbilt USB controller and UART which is used to implement the USB to serial port converter. The USB is implemented using LUFA library (link here). Is based on USBtoSerial example of LUFA library.

Littlebit about the circuit. The schematic, the board.

Making the board.

Downlaod the PNG file.

Note if you want to edit the file, here are the original eagle file. Please make sure that when exporing PNG you export with minium 2000dip. Use the fabmodules and the PNG to mill board. De-burr it if required to remove any loose connection. Use a tweezer to remove the excess copper from the USB PCB connector. Note that this is very important, failing to do so can result in a board which won’t work. OK so milled the board. Debureed it. Removed thetip. You will ned following compmontent atmega microcontroller, ressitors, isp and ftdi connectior. Stuff the board. The board is also powered by USB.

Note that resistor, crystal and capacitor has no polarity. However, Zener diode, LED and microcontroller has orientation. The micro-controller should be solder in right orientation by alging the notch (or dot) with the dot shown in board design.

You need some help with stuffing the atmega microcontroller. Best tis to make a little mess using escessive soldering and the using copper grade to remove the excess solter. For final touch touch the hot soldering ion to individual pads to melt and reflow the excess solder and distrubiut it uniformly. While brading be gentle as the pads are only 14mil wide and could just come-off. Picutre of milled board, picture of removing the ecess copper form USB pcb connector, picture of escess solder in atmega chip picture of removing using copper grade and picture of final soldered component. First suggest to solder microcontroller then solder didoes. Can have picture of the board with diode to suggest the right orientation. Then stuff all other parts. Stuff led and other thing. (in the code modify the led file to trun on led while getting connected and disconnected).

Idea sequence is first USB connector, then micro-controller and the small component and finally big connectors.

Order female FTDI conector part for FabFTDI. Ok the board is stuffed. Now

Prgraming the board.

Edit the makefile to add programing commands also.

Rename make file to fabftdi.make and use command

make –f fabftdi.make build

make –f fabftdi.make program

check the programing of fuse and code.

You will need FabISP, and your board. Connect the FabFTDI board to USB. Connect the ISP connector, run the command to read the device ID. Here the assumption is you have already installed avrdude if not follow the instructions from here.

Read the device ID (picture of terminal reading device ID). If you can read the.

Flash the fuses (may be defaut work).

Create a makefile which will flash both fuses and the code.

So you need to first tell them to download the make file and hex file, and INF, and term.py file file for the windows driver + PNG for milling. Then use make program or something for programing the board.

Once the board is programmed you should be able to see it appear as

In Unix

ttyUSBACM0

In MAC

In Windows

Browse to the driver and install it and it should appear as FabFTDI in widnows device manager. (image of FabFTDI in windows device manager). If someone is using FabFTDI in windows then they will also use it for programing so mention the extrcution of programing in windows using AVRdude and usb-tiny. If you have any other programmer than you can change that in the make-file for example for avrisp2 you can use this for the “” for atmel-ice you can use this.

Testing the Board

Once the board is connected if the LED blink then it indicate that the device is successfully connected. You should be able to see the device is /dev/ or windows device manager.

Part list: 16MHz crystal, 18pF capacitor, female FTDI connector, jumper.

Navigate do download folder. Now type python term.py port (which you found in /dev/) plus 9600

This will open a window. Connect the jumper and type something in the terminal if you see things printing back then your fabftdi is working and ready for use.

DFU Programming

Fuse Bits for DFU programing

Set HIGH.BOOTSZ to set bootloader size and starting location.

Set HIG.BOOTRST so that microcontroller go into the bootloader mode after RST.

Editing/Improving FabFTDI

Here is the original eagle file that you can download and edit it. For milling it make sure to export the png with 2000dpi.

Next version:

May not need the D- pulling up for USB controller.