A printed circuit board (PCB) constructs a conductive and insulating layer. It connects the electronics components internally as well as soldering them to drive the system mechanically. The connection is made over a copper (conductive) sheet laminated onto a non-conductive subpath. Nowadays, PCB is widely used due to its significant impacts compared to its size, making a massive difference in the production world. The following steps show how we fabricate a simple board that allows us to program the upcoming projects:

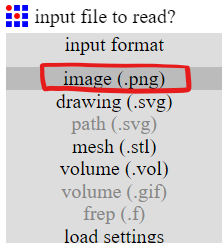
1-Designing part:

In this board, we have used a ready design to print it using CNC milling machine RMS-20 (the one used in UAE FabLab). There are two design parts: the trace and the outline. The traces one is responsible for the internal connection, while the outline is for cutting the outer board. I followed the following steps:

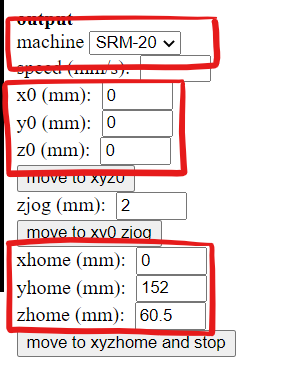
1- Download the design files from <http://fab.cba.mit.edu/classes/863.16/doc/projects/ftsmin/fts_mini_traces.png> and <http://fab.cba.mit.edu/classes/863.16/doc/projects/ftsmin/fts_mini_cut.png> as a PNG image

2- Go to <http://fabmodules.org/> to adjust the setting for each image (trace and outline).

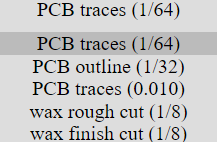
3- I started with the traces part by choosing the input format (image.png)



4- Select the output format that our CNC milling machine can read (Ronald. rml) and keep all the axis at the origin point (0,0,0).

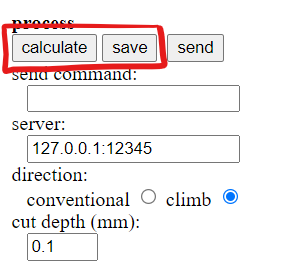
 

5- Moving to the process section >> traces (1/64), where 64 is the size of the used bit (pin) in the machine

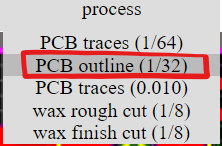


6- Check that all the axis are still at the origin point

7- Press calculate and then save; now you're done!!



8- Do the same steps for the outline, except for the 5th step, choose outline (1/32) instead of traces (1/64).



2-Production Part:

Moving to the fabrication part and printing our PCB using Roland milling machine RMS-20.

1- Stick the copper sheet over an MDF wood to ensure its stabilization and to engrave both traces and outline parts well.



2- Place the milling bit in the collet and tighten it down gently to avoid slipping out and breaking the tip. These are the two used pin sizes.





3- Open Roland software to adjust the cutting and milling position.

4- Start with milling the traces part (1/64) always since it's more accurate, and then repeat the same steps for the board outline (1/32).

5- Manually set up the system coordination by choosing >> user coordinate system



6- Change the cursor step to continue instead of X10



7- Use the xy arrow button to specify the starting point; look to the machine itself to know. Note: the printing will start from down to up from left to right.



8- Click on the x/y button at the right of the page to save the coordination and zero the shown values.



**2**

**1**

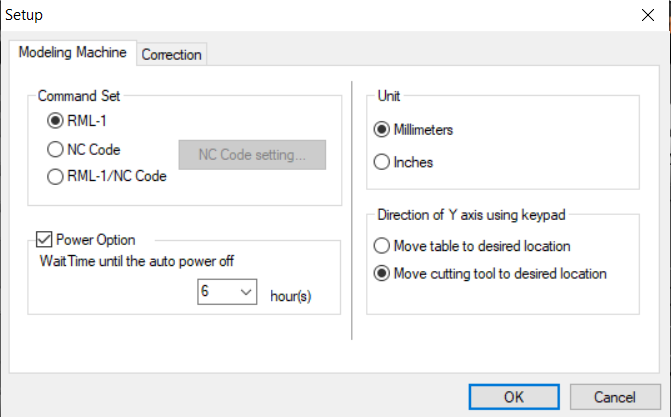
9- Use the z arrow to control the height; once feeling that the pin is near to the copper sheet, continue controlling it by hand to make sure that the tip is touching the sheet and sidestep breaking it. Don't forget to tighten the pin by hand.



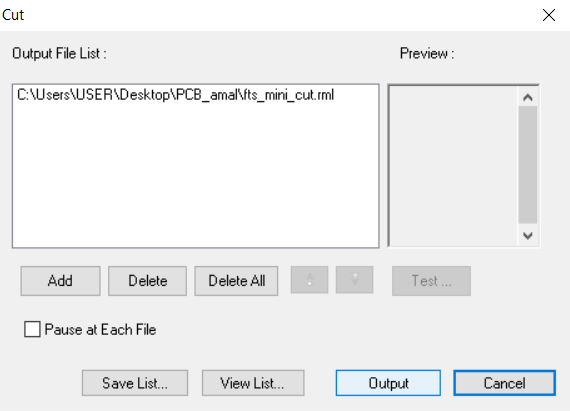
10- Click on the z button at the right of the page to save the coordination and zero the shown values.



11- Press the setup button on the lower right side to check the setting. The command set shall be RML-1.



12- Directly cut the file >> cut>> delete all >> choose the needed file (traces) >> add >> output



**2**

**1**

**3**

13- For the outline, select (outline) not traces.

14- The CNC starts the process!

15- Change the bit (pin) size to (1/32) and repeat the steps for the outline pattern.

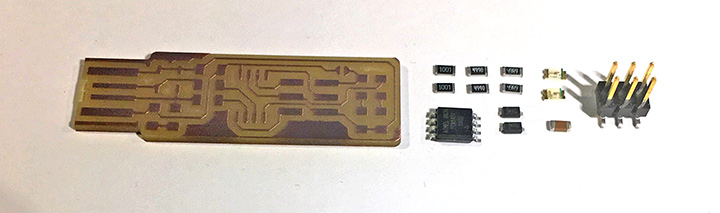
16- Remove the cutted board from the sheet carefully.

17- Remove the extra copper layer because it might burn the chip or shorten the circuit.

18- Check the internal printed connection again; there might be a short circuit.

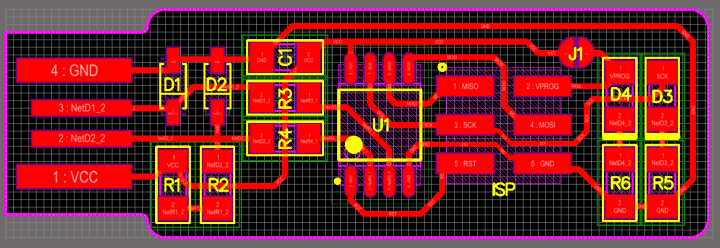
3- Soldering Part:

To build our ISP chip, we need the following components: 2 zener diode of 3.3 v, 1 RED LED, 1 Blue LED, 2R of 1K ohm, 2R of 49 ohm, 2 R of 499 ohm, 2x3 pin holder, 1C of 100nF, 1 of ATtiny85 controller.

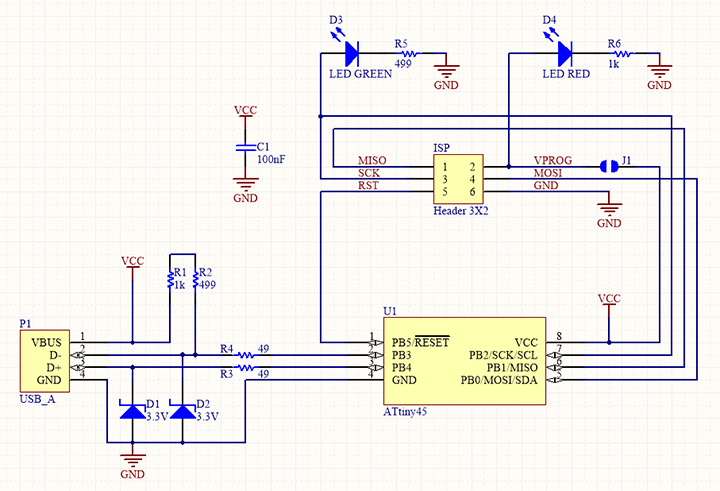


Before soldering any components, we must check their conductivity using a multimeter. First, connect the red probe to the V node and the black probe to the com node. Based on what part is going to be measured, I placed the pointer. In this case, the pointer will be placed beyond the ohm's point for all the parts. Since there is no specific polarity for the resistor and the capacitor, it doesn't matter where to connect the black and red bars. The metric values of the resistors must be in ohm and almost the same as the actual value. For the zener diode, the voltage must be around 0.5 for the forward direction while zero in the reverse order. Note that the cathode side is negative (the light line). The LED will blink in the forward direction for the typical diode (LED), which means it is working correctly. Connect the cathode side to the black probe (negative) and the anode side to the red one (positive). After testing all the elements and ensuring they are not damaged, start soldering them!!

Follow the below schematics to understand how the components are connected to each other internally, and be careful:



Assure the orientation of each component. The ATtiny85 marks pin 1 with a dot laser-etched into the corner of the package. Pin 1 is located at the left-down corner.



There are no specific steps to solder the components, but my own way that I followed is:

1- Heat the element and the place you want to solder

2- Add a solder material to that place

3- Use a tweezer to fix the element at that place

4- Heat the soldering material and add more again

5- Keep holding the element and press on its body till fixing it.

6- Solder the second side of the element in the same manner but ignore the 2nd step here.

After finishing soldering the components, test the circuit conductivity again and guarantee there is no short between any two nodes (Vcc and GND). The following picture shows my own chip:

4-Programming part:

To program the FabTinyISP chip, Windows users require five software to set up a suitable environment. I followed Maha Al Hashimi's way of installing the requirements.

The installed softwares are:

1-  [WinAVR 20100110](https://sourceforge.net/projects/winavr/" \t "_blank)

2- [Zadig](https://zadig.akeo.ie/" \t "_blank)

3- [FTS firmware](http://fab.cba.mit.edu/classes/863.16/doc/projects/ftsmin/fts_firmware_bdm_v1.zip)

4-

5-

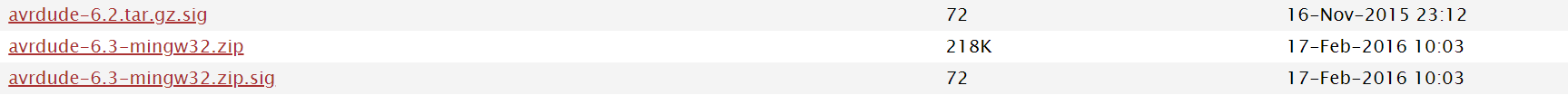
I started with downloading AVR software because it takes a long time.

1- Go to  [WinAVR 20100110](https://sourceforge.net/projects/winavr/" \t "_blank) to install it, and then run it.

2- on your computer go to This PC >> WinAVR folder >> bin

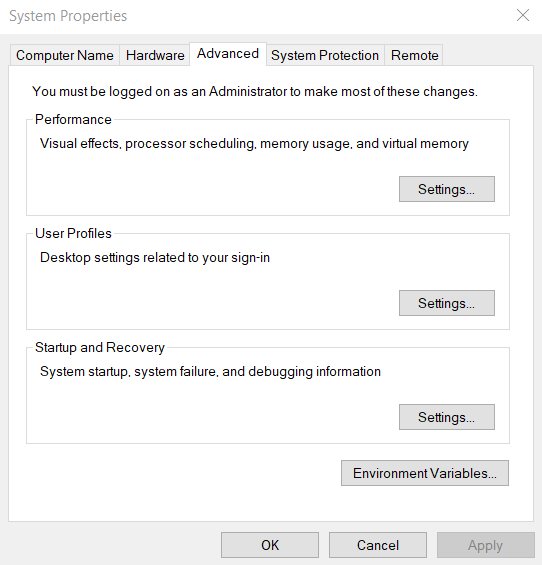
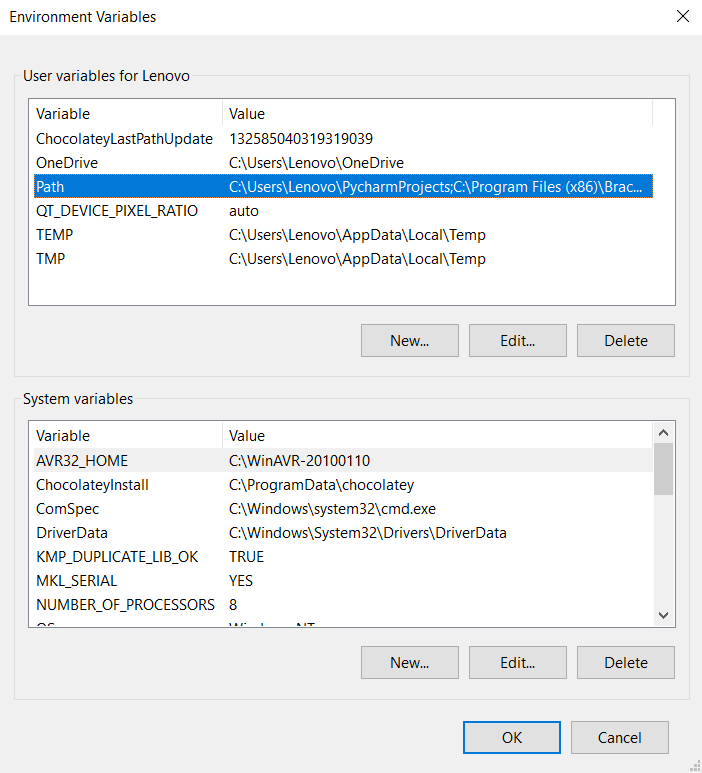
3- Delete both **avrdude.conf** and **avrdude** files

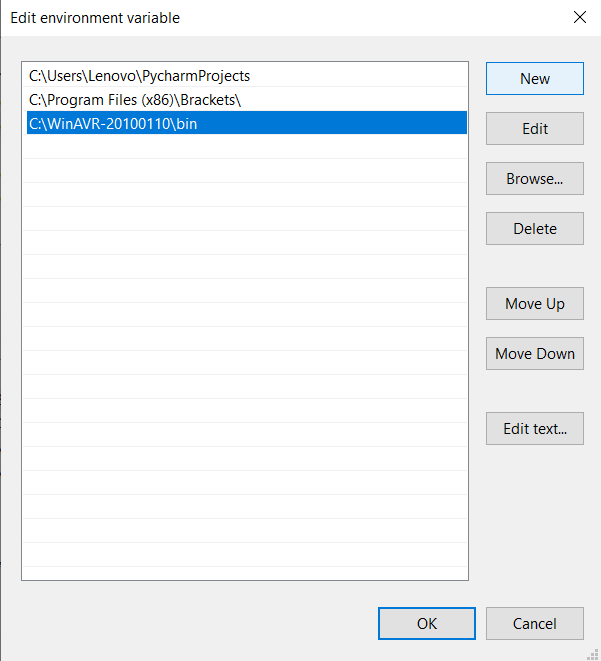
4- Visit [Version 6.3 2016](http://download.savannah.gnu.org/releases/avrdude/) and save **the avrdude-6.3-mingw32** zip file



5- Replace the deleted files with the newer version of them (on bin folder)

6- In control panel >> systems >> advanced system settings >> environment variables >> path >> edit >> delete the old AVR path (if installed) >> add the new path of the bin folder by copping its location (../WinAvr/bin)

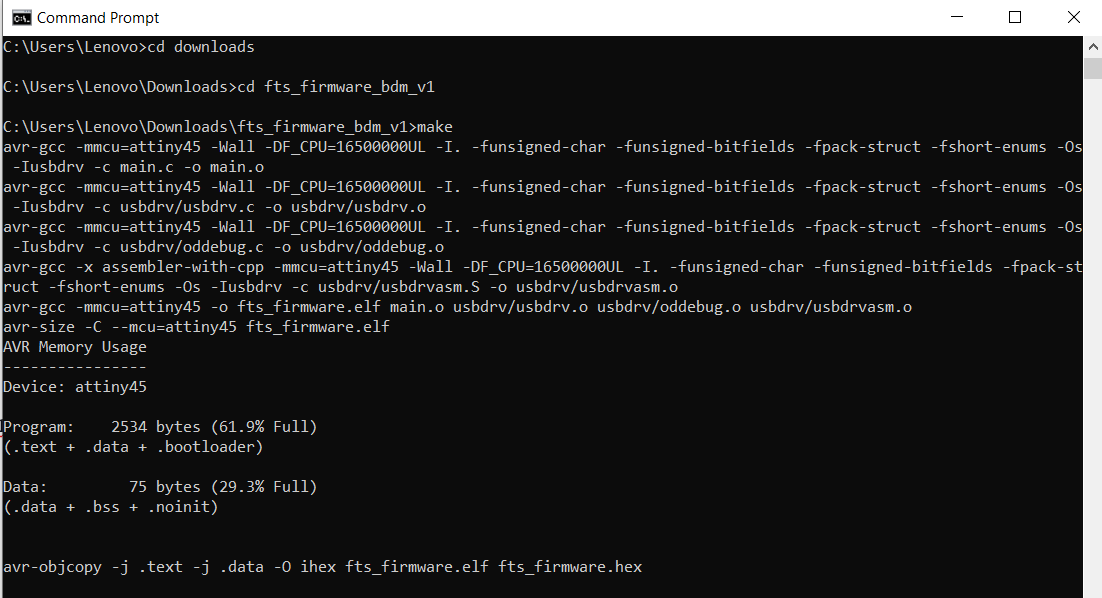




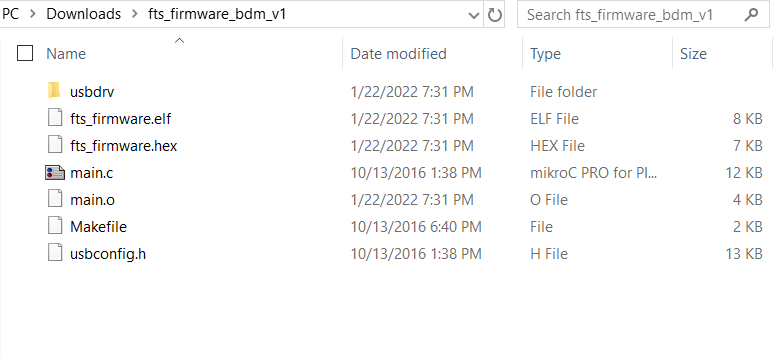
7- Now download [Zadig](https://zadig.akeo.ie/" \t "_blank), a Windows application that installs generic USB drivers, such as [WinUSB](https://docs.microsoft.com/en-us/windows-hardware/drivers/usbcon/winusb" \t "_blank), [libusb-win32/libusb0.sys](https://sourceforge.net/p/libusb-win32/wiki/Home/) or [libusbK](http://libusbk.sourceforge.net/UsbK3/" \t "_blank), to help you access USB devices

8- Install [FTS firmware](http://fab.cba.mit.edu/classes/863.16/doc/projects/ftsmin/fts_firmware_bdm_v1.zip) >> extract the zip file

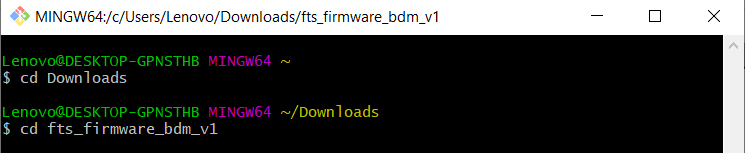
9- Open the terminal window >> open the extracted directory (source code) using cd >> run make command to build a hex file that will get programmed onto the Ttiny85



10- A fts\_firmware.hex file is created on that directory

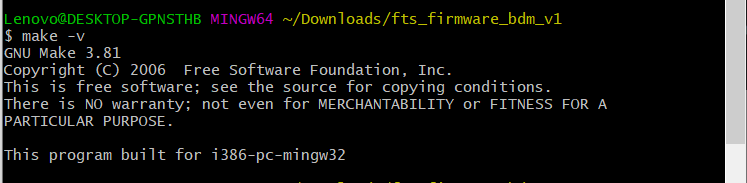


11- Open GitBash and open the fts\_firmware\_bdm\_v1 folder using the cd command

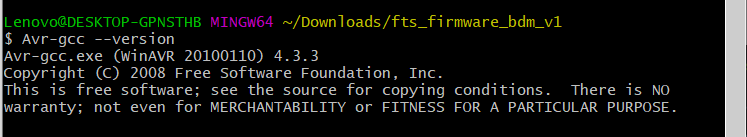


12- Check if all the drivers are correctly installed using the following commands:

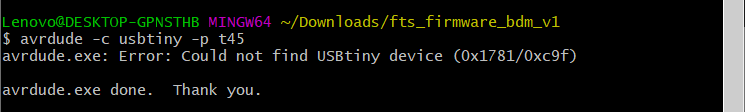
1. Make -v >> to test if the command is work



1. Avr-gcc --version >>  ensure that the appropriate version of the software is installed



1. avrdude -c usbtiny -p t45 >> check for the avrdude file
2. Before connecting the TinyISP



1. After connecting it

11- Connect the programmed ISP (Eng. Hashim ISP) and run zadig

12- Select the programming >> select “libsub-win32 (v1.2.6.0)” >> replace drive

13- Prepare a ribbon cable of 6 wires since our header is of 6 pins

1. Cut a suitable wire length, and connect a female connector on both sides.
2. Place the wire in between the body of the connector and its cover
3. Be careful; both wires' sides shall be connected to the same pin on both connectors, the two connectors must face the same direction.
4. Use a hammer to secure them
5. Connect your ISP chip and the programmed one to the cable

Note: if the blue wire is located at the left side of the first connector, the first left wire of the second connector must be blue also. Look at the below image to understand

14- Program the ISP chip through your terminal window:

1. Open the directory where you saved the .hex file
2. Run make flash command >> to erase the target chip and program its flash memory with the contents of the .hex file you built before. several progress bars will appear while avrdude erases, programs and verifies the chip
3. Run make fuses command >> to set up the fuses
4. Run make rstdisbl >> disable the reset pin to ban program it again

15- Congratulation!! your ISP is ready to program another board!!

From my laptop:

1. downloaded the design files as an images: traces + outline , the outline is responsible for cutting the board while the traces is for the internal connection
2. go to modules website & adjust the setting for each image
3. first we started with the traces part by choosing the output format that our CNC milling machine can read it (Roland mill .rml) and then keeping all the three axis at the origin point (0,0,0)
4. then we went to the outline part and did the same thing

