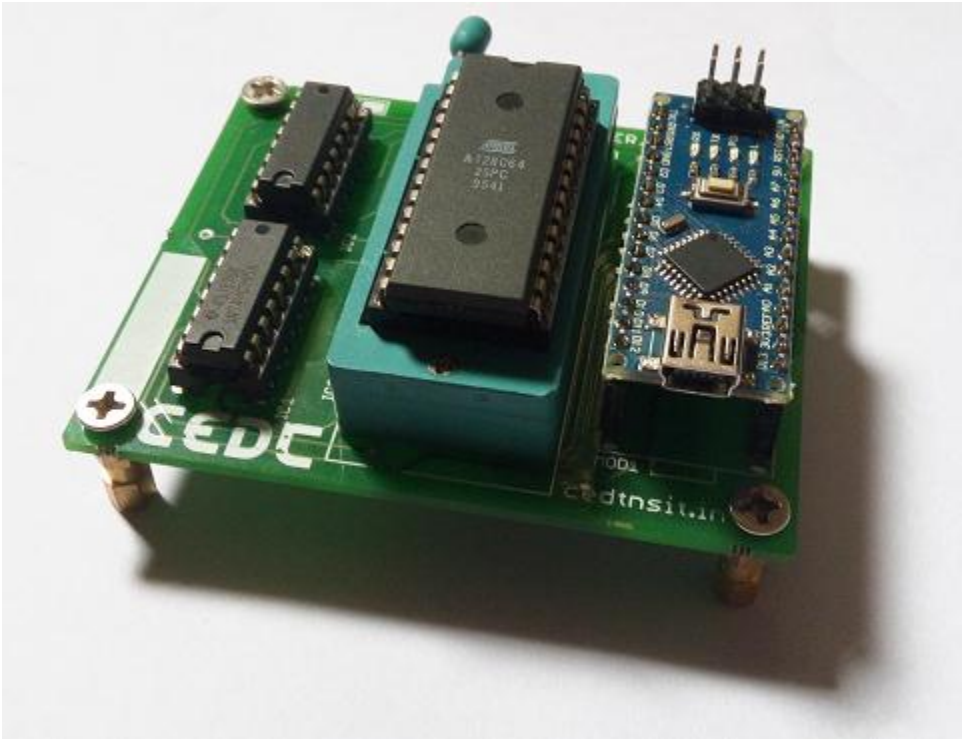
EEPROM

PROGRAMMER



(Quick start guide)

Table of Contents

Introduction

The EEPROM Programmer is an Arduino Nano based circuit that has the capability to load data onto Parallel EEPROMs AT28C256 (32KB) and AT28C64 (8KB). This was developed at the Centre for Electronics Design and Technology, NSIT under the guidance of Prof. Dhananjay V. Gadre and Nikhilesh Prasannakumar as an aid for the students of the course EC-316 in Netaji Subhas Institute of Technology, Delhi.

This document guides the user through the process of setting up the tool and using it.

All queries can be directed to Anshuman Mishra (cedt.eepromprogrammer@gmail.com)

Kit Contents

The kit provided to the user should have the following:

- 1 x EEPROM Programmer PCB
- 2 x IC Socket 16-pin DIP pitch 2.54mm, row spacing 0.3” (7.62mm)
  (http://www.digikey.com/product-detail/en/assmann-wsw-components/A16-LC-TT/AE9992-ND/821746)
- 1 x IC Socket 28-pin DIP pitch 2.54mm, row spacing 0.6” (15.24mm)
  (http://www.digikey.com/product-detail/en/assmann-wsw-components/A28-LC-TT/AE10004-ND/821758)
- 3 x 0.1µF 0805 SMD Capacitor (Ceramic)
  (http://www.digikey.com/product-detail/en/samsung-electro-mechanics-america-inc/CL21F104ZBCNNNC/1276-1007-2-ND/3886665)
- 2 x Female Socket 40 Pins Single Row Straight 2.54mm Pitch

Additional items required and not provided in the kit:

- 1 x Arduino Nano
- 2 x 74HC595* (16-PDIP) Shift Register
  (http://www.digikey.com/product-detail/en/texas-instruments/CD74HC595EE4/CD74HC595EE4-ND/1507397)
- 1 x Parallel EEPROM (AT28C256 or AT28C64)

Note: The user can replace the 28-pin 0.6” IC socket with a 28-pin 0.6” ZIF socket. The ZIF socket is not provided in the kit and will have to be arranged by the user himself if he so wishes to.

*Recommended. Other families may also be compatible.

Soldering

The kit provided to you needs to be soldered properly in order to be useful. Proper soldering would play a major role in the operation of the Programmer later. Hence special care should be taken at this stage to avoid troubleshooting issues later.

What tools are needed for soldering the board?

- The kit, containing the PCB and the required components
- A soldering station
- Soldering sponge
- Water spray
- Tweezers
- Solder

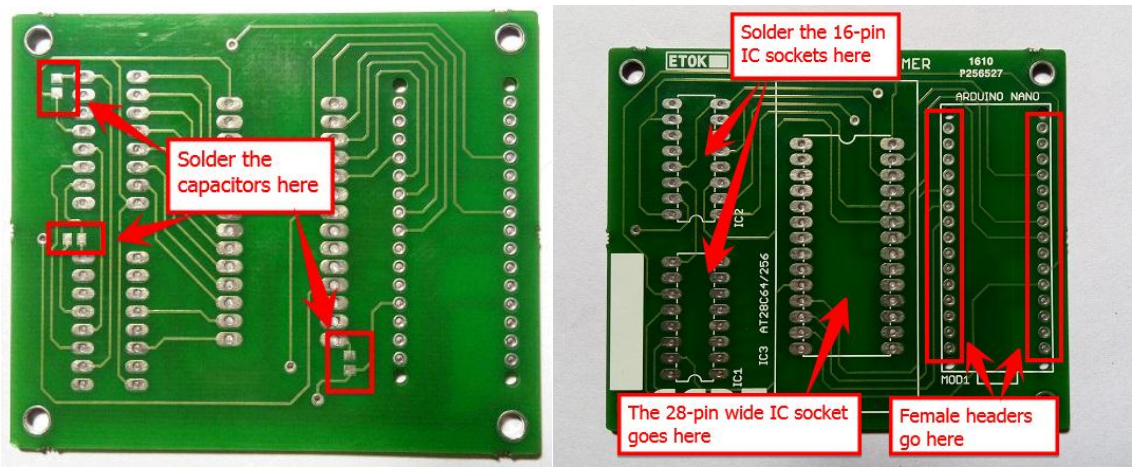
What precautions are needed before starting soldering?

a) The workspace should be cleared of all unnecessary material
b) There should be no entanglement of wires
c) The soldering sponge should be just wet. Neither too much nor too little.
d) Gather all the required tools and materials within easy reach

Soldering the components

The positions of various components have been marked on the PCB. The order of soldering the components should be

1. 0.1µF Capacitors (SMD)
2. 28-pin Wide IC socket (for the AT28C256 or AT28C64)
3. 16-pin IC sockets (for the shift registers)
4. 2 strips of Female headers (for the Arduino Nano)



Finished look

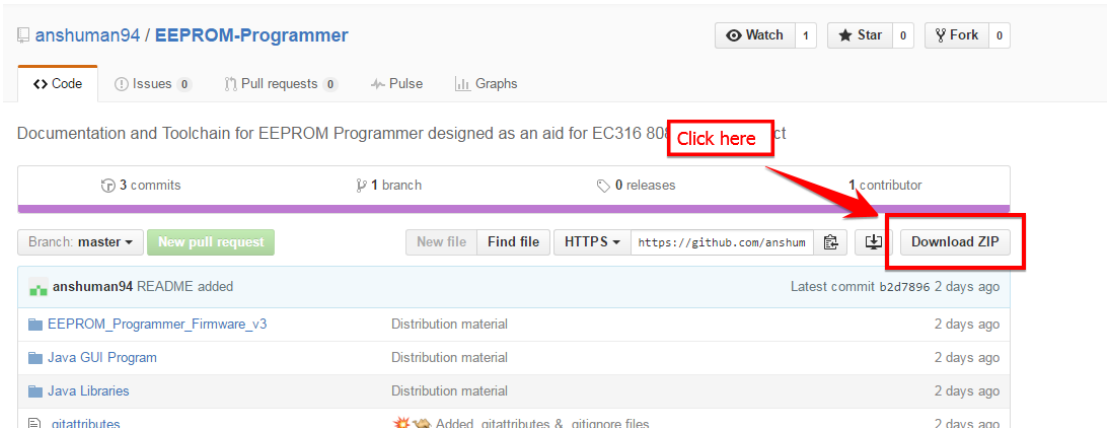
Software Toolchain

Pre requisites

- Java JRE 1.8.0 or later
- 8085 Simulator IDE

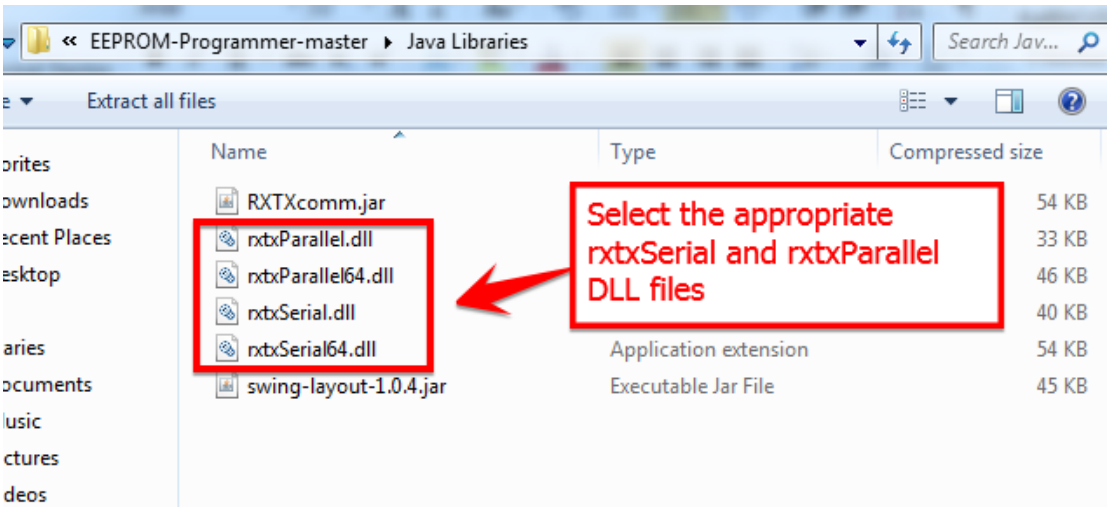
Steps

1. Go to www.github.com/anshuman94/EEPROM-Programmer and Click on the **Download ZIP** button

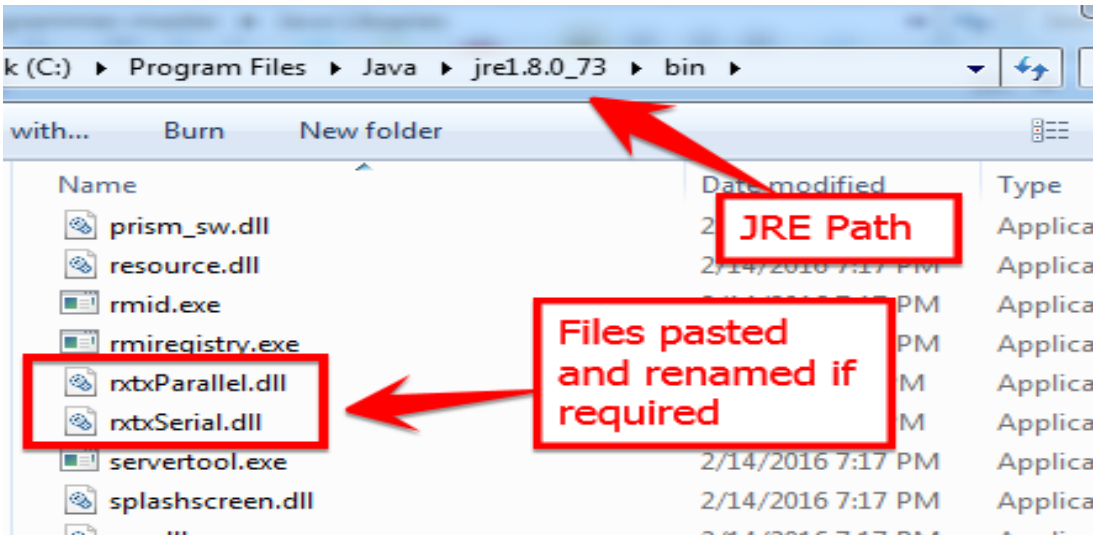


This will download a zip file containing all the requisite files.

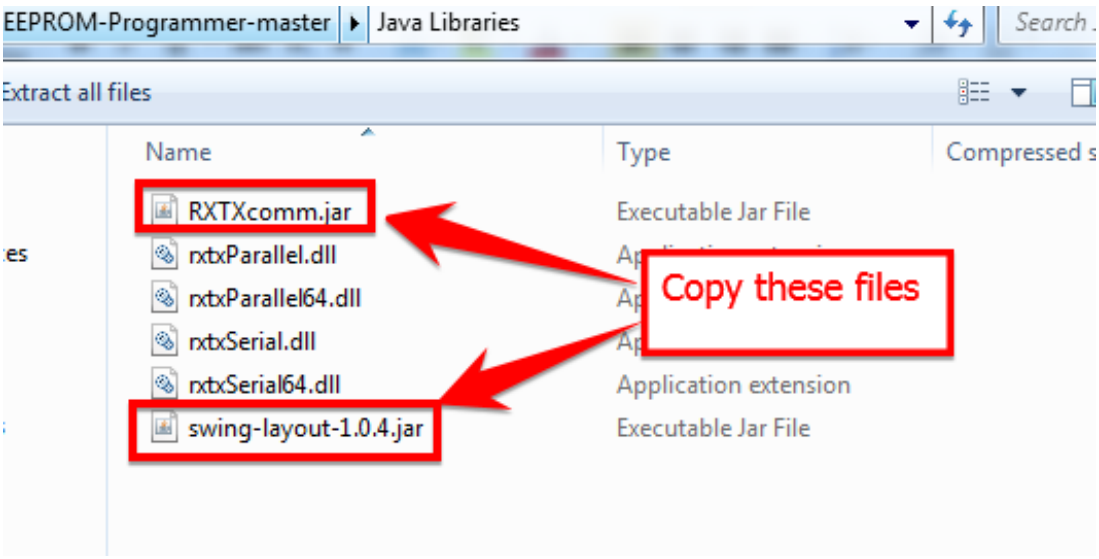
2. Go to the **JAVA Libraries** folder and copy the **rxtxSerial.dll** and **rxtxParallel.dll** files (If you are on a 64-bit machine, copy the **rxtxSerial64.dll** and **rxtxParallel64.dll** files and remove the 64 suffix from the file names)



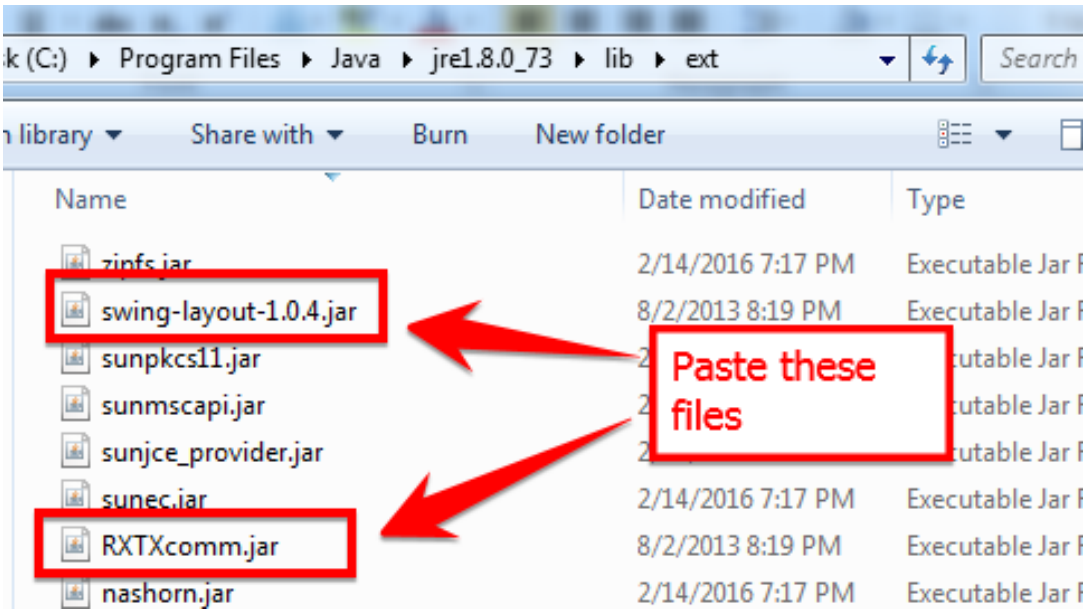
3. Navigate to your JRE directory (default: C:/Program Files/Java/jre<your jre version>). Open the **bin** folder therein and paste the files copied in step 2



4. Go back to the JAVA Libraries folder and copy the rxtxComm.jar and swing-layout-1.0.4.jar

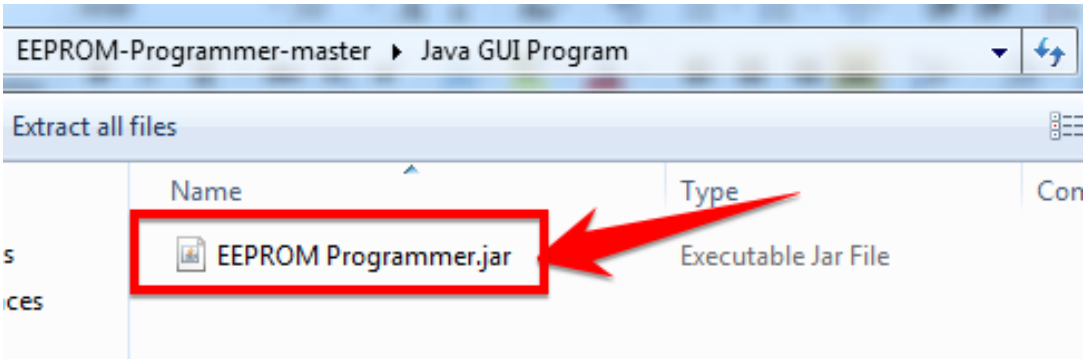


5. Navigate to your JRE directory (default: C:/Program Files/Java/jre<your jre version>). Open the **ext** sub folder in **lib** folder and paste the files copied in step 4



Now your System is ready to interface with the hardware.

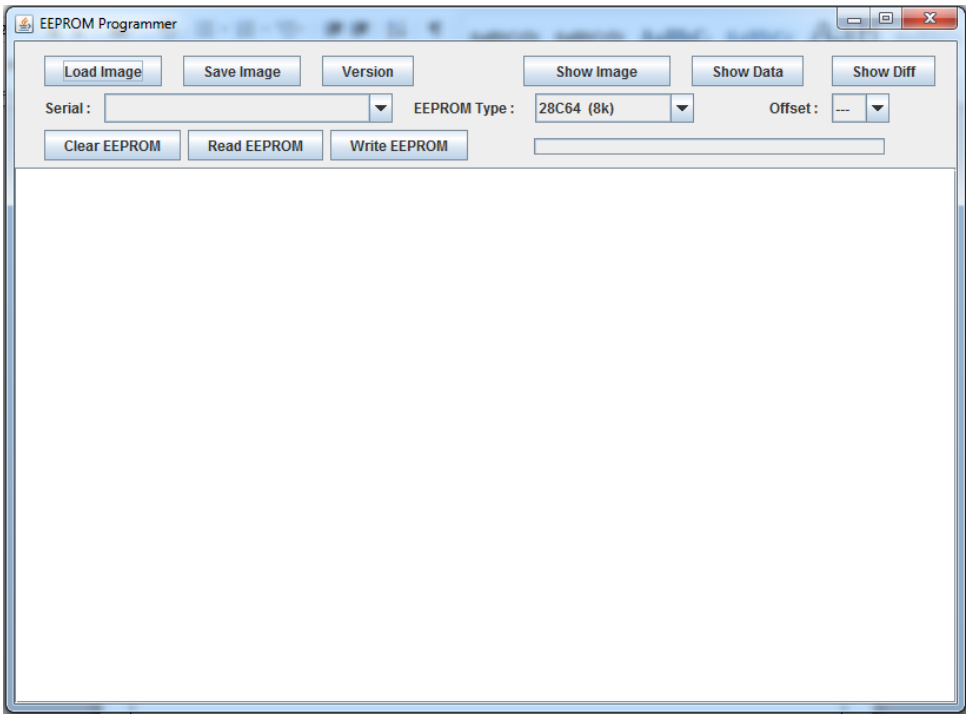
6. Go back to the downloaded folder and open the **EEPROM Programmer.jar** in the **JAVA GUI Program** folder



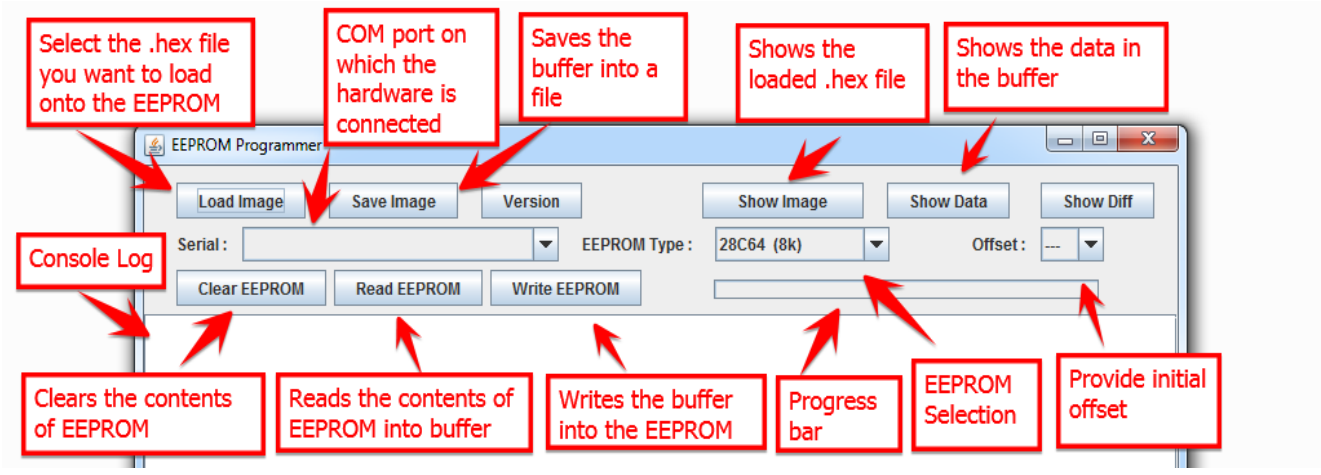
This will open up Graphical Interface to be used to burn your programs onto the EEPROM.

Introduction to the User Interface

When the interface is opened, the following window appears:



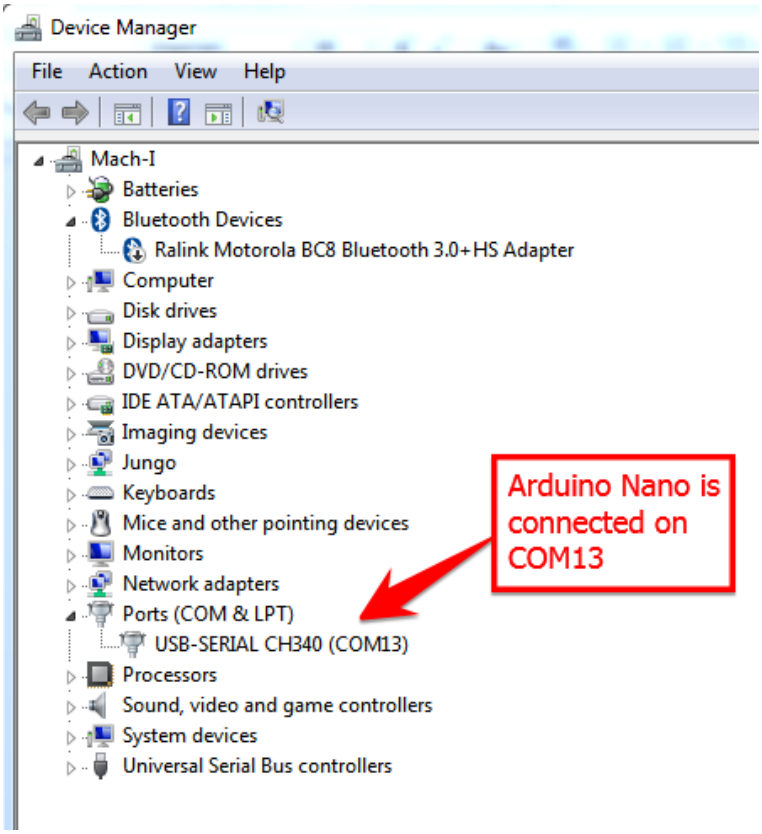
The role of various buttons are explained as follows:



Operating Instructions

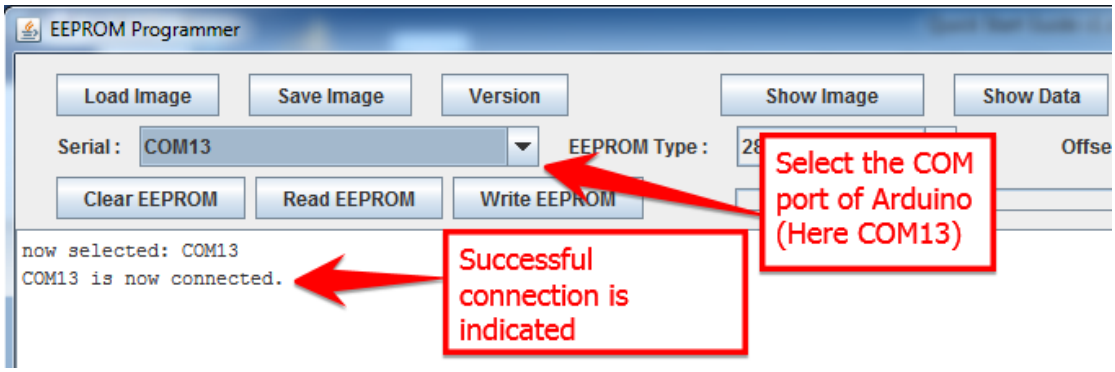
First Steps

1. Connect your Arduino Nano to your computer and wait for the driver to install (if required)
2. Open Device Manager and Note down the COM port on which your Arduino is connected.



3. Go to the downloaded **EEPROM-Programmer-master/EEPROM_Programmer_Firmware_v3** folder
4. Open the sketch in Arduino IDE and upload the code to the connected Arduino Nano.
5. Open the **EEPROM Programmer.jar** in the **JAVA GUI Program** folder.

6. From the list of shown COM ports, Select the one on which Arduino Nano is connected (Important)



7. Select the type of EEPROM you want to program (Important)

Now you are ready to burn your data onto the EEPROM. You can verify that you are set to go by clicking the Verify button. You should get the following output in console log:

```
EEPROM Programmer - Revision :  CEDT v2.0, Date: 16/2/2016 11:12 PM
EEPROM Programmer CEDT v2.0
```

8. **<u>Clearing the EEPROM</u>**

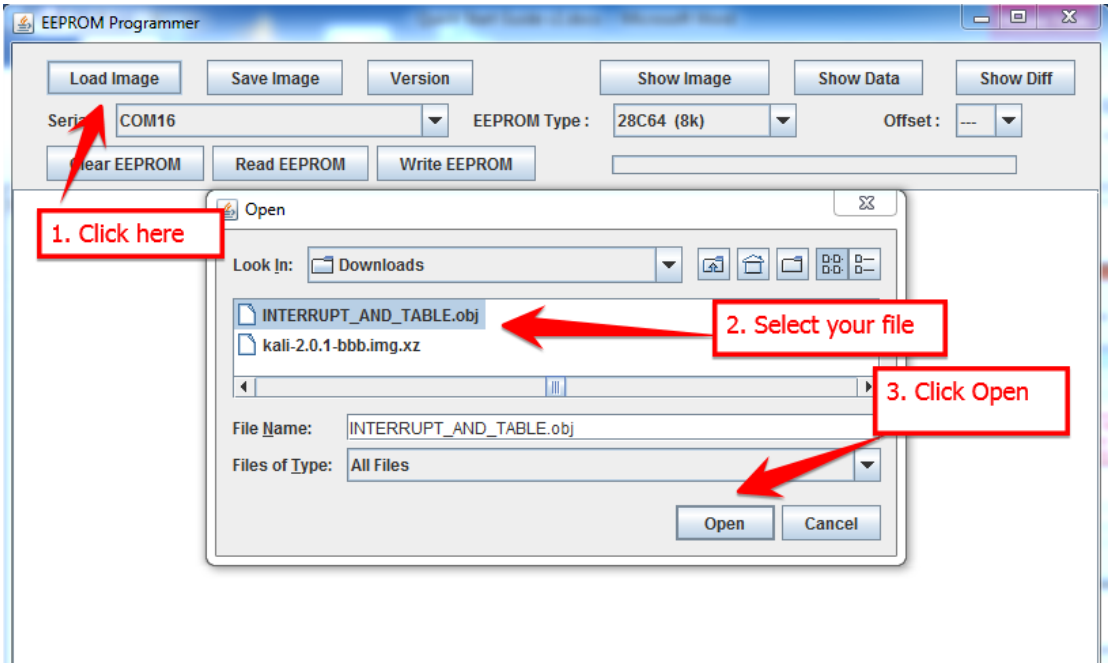
In order to clear the EEPROM; make sure that the appropriate COM Port and the EEPROM is selected.

Once that is done, Press the Clear EEPROM button and wait till the process is completed (It will be shown in the Console Log)

```
Clearing EEPROM. setting 32768 bytes to 0xff
sending command.
wrote data from 0x0000 to 0x03FF
wrote data from 0x0400 to 0x07FF
wrote data from 0x0800 to 0x0BFF
wrote data from 0x0C00 to 0x0FFF
wrote data from 0x1000 to 0x13FF
wrote data from 0x1400 to 0x17FF
wrote data from 0x1800 to 0x1BFF
wrote data from 0x1C00 to 0x1FFF
wrote data from 0x2000 to 0x23FF
wrote data from 0x2400 to 0x27FF
wrote data from 0x2800 to 0x2BFF
wrote data from 0x2C00 to 0x2FFF
wrote data from 0x3000 to 0x33FF
wrote data from 0x3400 to 0x37FF
wrote data from 0x3800 to 0x3BFF
wrote data from 0x3C00 to 0x3FFF
wrote data from 0x4000 to 0x43FF
wrote data from 0x4400 to 0x47FF
wrote data from 0x4800 to 0x4BFF
wrote data from 0x4C00 to 0x4FFF
wrote data from 0x5000 to 0x53FF
wrote data from 0x5400 to 0x57FF
wrote data from 0x5800 to 0x5BFF
wrote data from 0x5C00 to 0x5FFF
wrote data from 0x6000 to 0x63FF
wrote data from 0x6400 to 0x67FF
wrote data from 0x6800 to 0x6BFF
wrote data from 0x6C00 to 0x6FFF
wrote data from 0x7000 to 0x73FF
wrote data from 0x7400 to 0x77FF
wrote data from 0x7800 to 0x7BFF
wrote data from 0x7C00 to 0x7FFF
data sent.
wrote 32768 bytes from 0x0000 to 0x7FFF in 196.736 seconds
```

9. **Writing a program to the EEPROM**

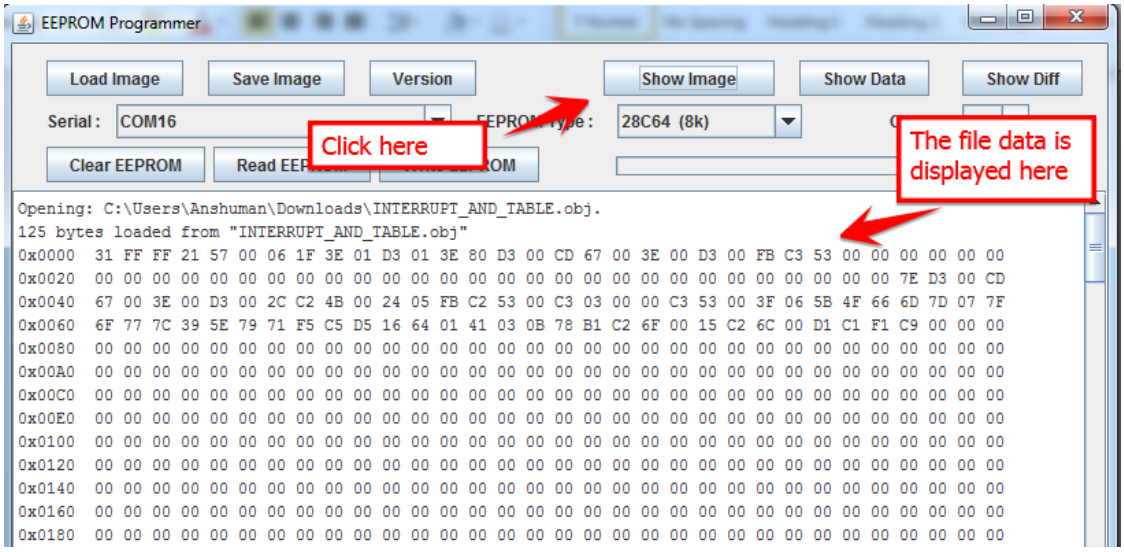
   a. Select the file you want to load onto the EEPROM by clicking the **Load Image** button
   b. Select your file and Click Open



c. You should get a message in the Console Log as shown:

```
Opening: C:\Users\Anshuman\Downloads\INTERRUPT_AND_TABLE.obj.
125 bytes loaded from "INTERRUPT_AND_TABLE.obj"
```

d. You can verify the code loaded into the buffer by clicking **Show Image**



e. Now all that's left is to burn this image to the EEPROM. You can do this by clicking the **Write EEPROM** button.

```
sending command.
wrote data from 0x0000 to 0x03FF
data sent.
wrote 146 bytes from 0x0000 to 0x0091 in 6.17 seconds
```

10. **Read Data from an EEPROM**

Sometimes, you might want to read the EEPROM to view its contents or to back up the contents. You can do this by the following steps:

a. Make sure the EEPROM Programmer is connected, COM port is selected and the correct EEPROM is selected.
b. Click the **Read EEPROM** button and wait till the operation is completed. (You will be notified in the console)

```
sending command.
command sent.
trying to read.
32768 bytes read in 0.674 seconds
```

c. You can view the contents by clicking the **Show Data** button.

```
0x0000  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 37 0D 0A 3A 31 30 30
0x0020  30 30 30 30 30 32 31 35 30 43 30 37 45 32 46 33 32 35 32 43 30 32 33 37 45 32 46 33 32 35 33 43
0x0040  30 37 36 30 30 34 33 0D 0A 3A 31 30 43 30 35 30 30 30 38 35 35 34 30 30 30 30 30 30 30 30 30 30
0x0060  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 37 0D 0A 3A 31 30 30 30 30 30 30 30 32
0x0080  31 35 30 43 30 37 45 32 46 33 32 35 32 43 30 32 33 37 45 32 46 33 32 35 33 43 30 37 36 30 30 34
0x00A0  33 0D 0A 3A 31 30 43 30 35 30 30 30 38 35 35 34 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30
0x00C0  30 30 30 30 30 30 30 30 30 30 30 30 30 37 0D 0A 3A 31 30 30 30 30 30 30 30 32 31 35 30 43 30 37
0x00E0  45 32 46 33 32 35 32 43 30 32 33 37 45 32 46 33 32 35 33 43 30 37 36 30 30 34 33 0D 0A 3A 31 30
0x0100  43 30 35 30 30 30 38 35 35 34 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30
0x0120  30 30 30 30 30 30 30 37 0D 0A 3A 31 30 30 30 30 30 30 30 32 31 35 30 43 30 37 45 32 46 33 32 35
0x0140  32 43 30 32 33 37 45 32 46 33 32 35 33 43 30 37 36 30 30 34 33 0D 0A 3A 31 30 43 30 35 30 30 30
0x0160  38 35 35 34 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30
0x0180  30 37 0D 0A 3A 31 30 30 30 30 30 30 30 32 31 35 30 43 30 37 45 32 46 33 32 35 32 43 30 32 33 37
0x01A0  45 32 46 33 32 35 33 43 30 37 36 30 30 34 33 0D 0A 3A 31 30 43 30 35 30 30 30 38 35 35 34 30 30
0x01C0  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 37 0D 0A 3A 31
0x01E0  30 30 30 30 30 30 30 32 31 35 30 43 30 37 45 32 46 33 32 35 32 43 30 32 33 37 45 32 46 33 32 35
0x0200  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 37 0D 0A 3A 31 30 30
0x0220  30 30 30 30 30 32 31 35 30 43 30 37 45 32 46 33 32 35 32 43 30 32 33 37 45 32 46 33 32 35 33 43
0x0240  30 37 36 30 30 34 33 0D 0A 3A 31 30 43 30 35 30 30 30 38 35 35 34 30 30 30 30 30 30 30 30 30 30
0x0260  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 37 0D 0A 3A 31 30 30 30 30 30 30 30 32
0x0280  31 35 30 43 30 37 45 32 46 33 32 35 32 43 30 32 33 37 45 32 46 33 32 35 33 43 30 37 36 30 30 34
0x02A0  33 0D 0A 3A 31 30 43 30 35 30 30 30 38 35 35 34 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30
0x02C0  30 30 30 30 30 30 30 30 30 30 30 30 30 37 0D 0A 3A 31 30 30 30 30 30 30 30 32 31 35 30 43 30 37
```

Troubleshooting

Q1. I cannot find JRE folder. What do I do?

It might very well be possible that JRE is not installed on your system. You can verify it by going to Command Prompt (Windows) and type java –version and press Enter. If you get an error, your system does not have the JRE, so you need to download and install the Java Runtime Environment (SE).

Q2. I tried to execute the program but it doesn't seem to run?

It is possible that you copied the wrong libraries (as in used 32-bit files on a 64-bit machine or vice versa) Do check and try to execute. Make sure you have renamed the DLL files (rxtxSerial and rxtxParallel) in case of 64-bit machines.

You can also try to check for errors by executing the program via the command prompt using the command java jar <Name of Program>

If you still face a problem, drop a mail at cedt.eepromprogrammer@gmail.com

Q3. I was able to open the program but it does not detect my hardware?

- Disconnect the hardware
- Close the program
- Connect the hardware and wait till it shows up in the Device Manager
- Open the program now
- Select the COM port specifically.
- Now make sure you specifically select the EEPROM you are programming (especially if you are programming the AT28C256 32KB EEPROM)
- You are good to go. You can verify by clicking the **Version** button.

Q4. Data didn't change when I burned my program onto the EEPROM?

Make sure you read the EEPROM again and only then Show the data to get the updated values.

Any unanswered issues can be directed at Anshuman Mishra
(cedt.eepromprogrammer@gmail.com)