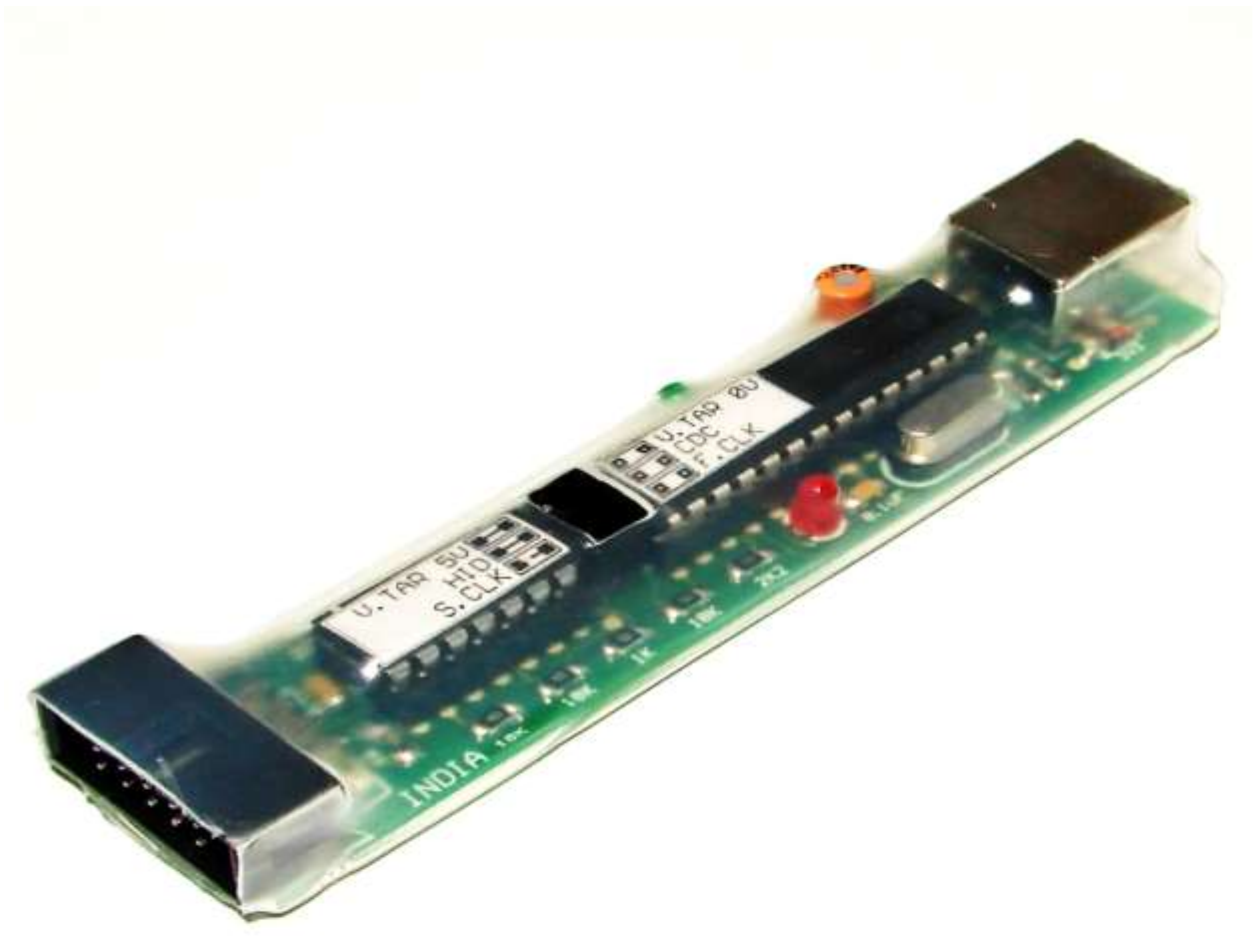
NEX AVR USB ISP STK500V2

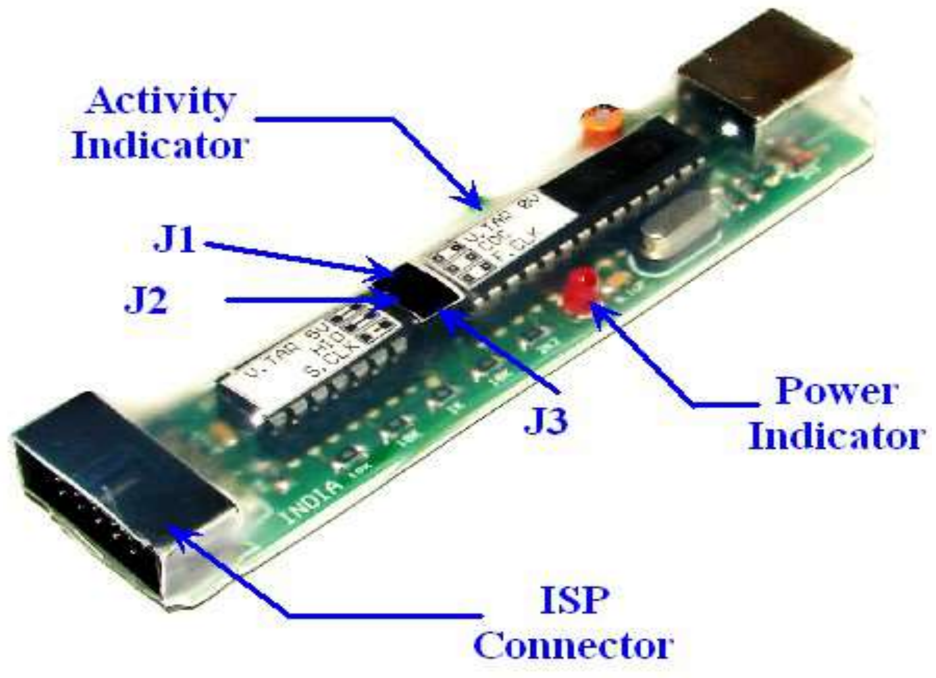


1.1 Introduction

NEX AVR USB ISP STK500V2 is a high-speed USB powered STK500V2 compatible In-System USB programmer for AVR family of microcontrollers. It can be used with AVR Studio on Win XP platforms. For Windows7 it can be used in HID mode with **Avrdude** command prompt as programming interface. Its adjustable clock speed allows programming of microcontrollers with lower clock speeds. The programmer is powered directly from a USB port which eliminates the need for an external power supply. The programmer can also power the target board from a USB port with a limited supply current of up to 100mA.

Note: The USB port of PC provides 5V DC. For 3.3V microcontrollers, please use appropriate voltage regulators.

NEX AVR USB ISP STK500V2 Overview



Jumper Description

J1: If inserted, provides 5V at VTG (pin no.2) of ISP connector. If removed 0V at VTG (pin no.2) of ISP connector. **In default mode, this jumper is not inserted.**

J2: If inserted, enables UBS HID mode. If removed enables USB CDC mode. **In default mode, this jumper is not inserted.**

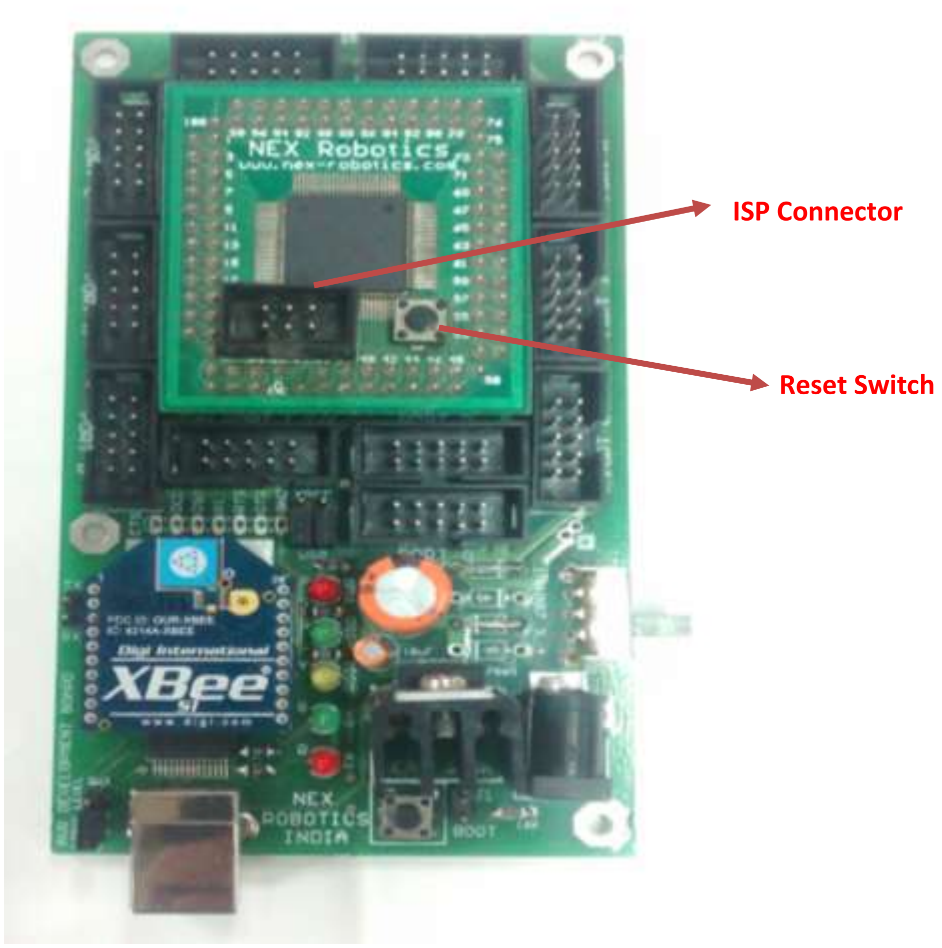
J3: If inserted, enables slow clock speed (for 32 KHz to 1MHz speed microcontrollers). If removed enables normal clock speed. **In default mode, this jumper is not inserted.**

No Jumper: This is the default mode of STK500v2; programming through Linux will be done in default mode.

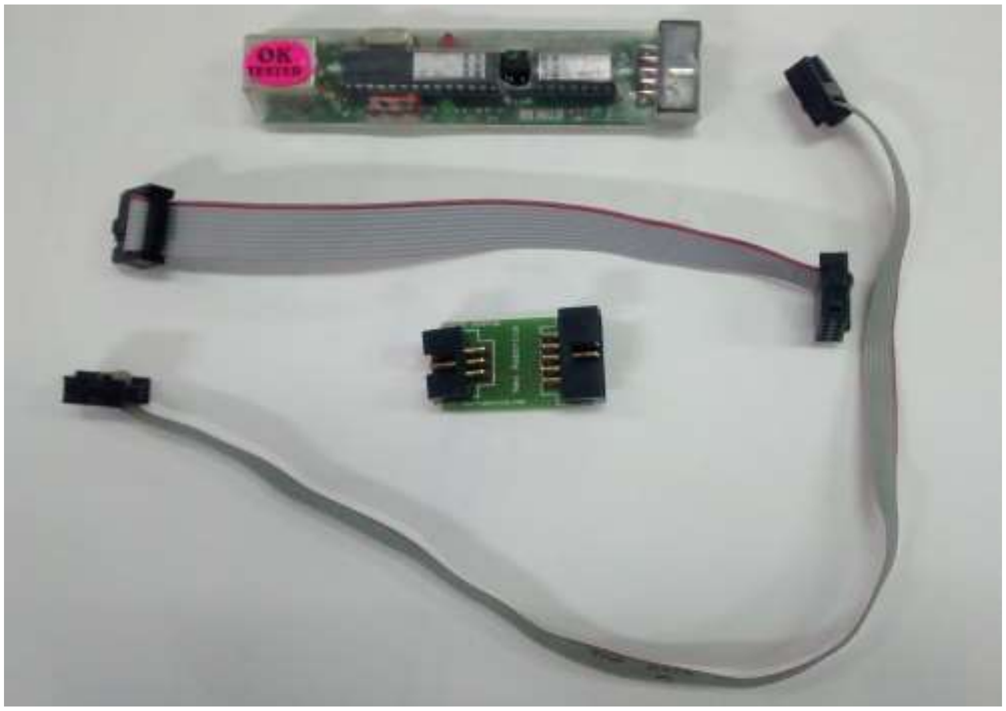
Connections between STK500v2 and ATmega 2560

Please follow the steps to connect STK500v2 and ATmega 2560

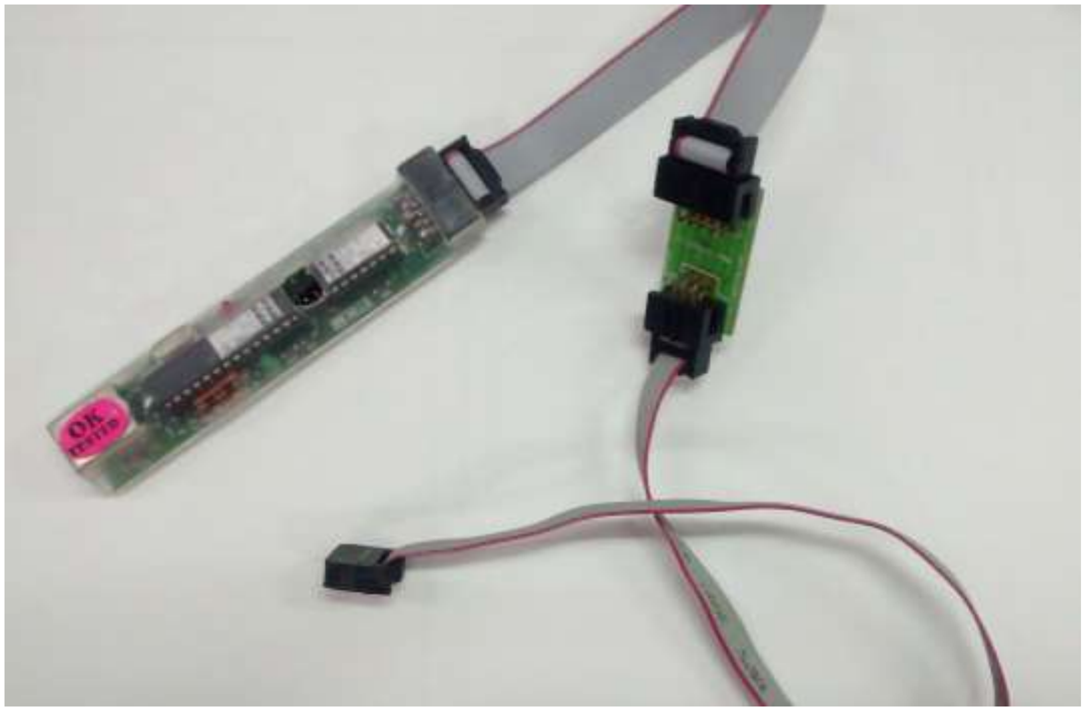
- Locate the ISP connector in the ATmega 2560 microcontroller as shown in the figure:



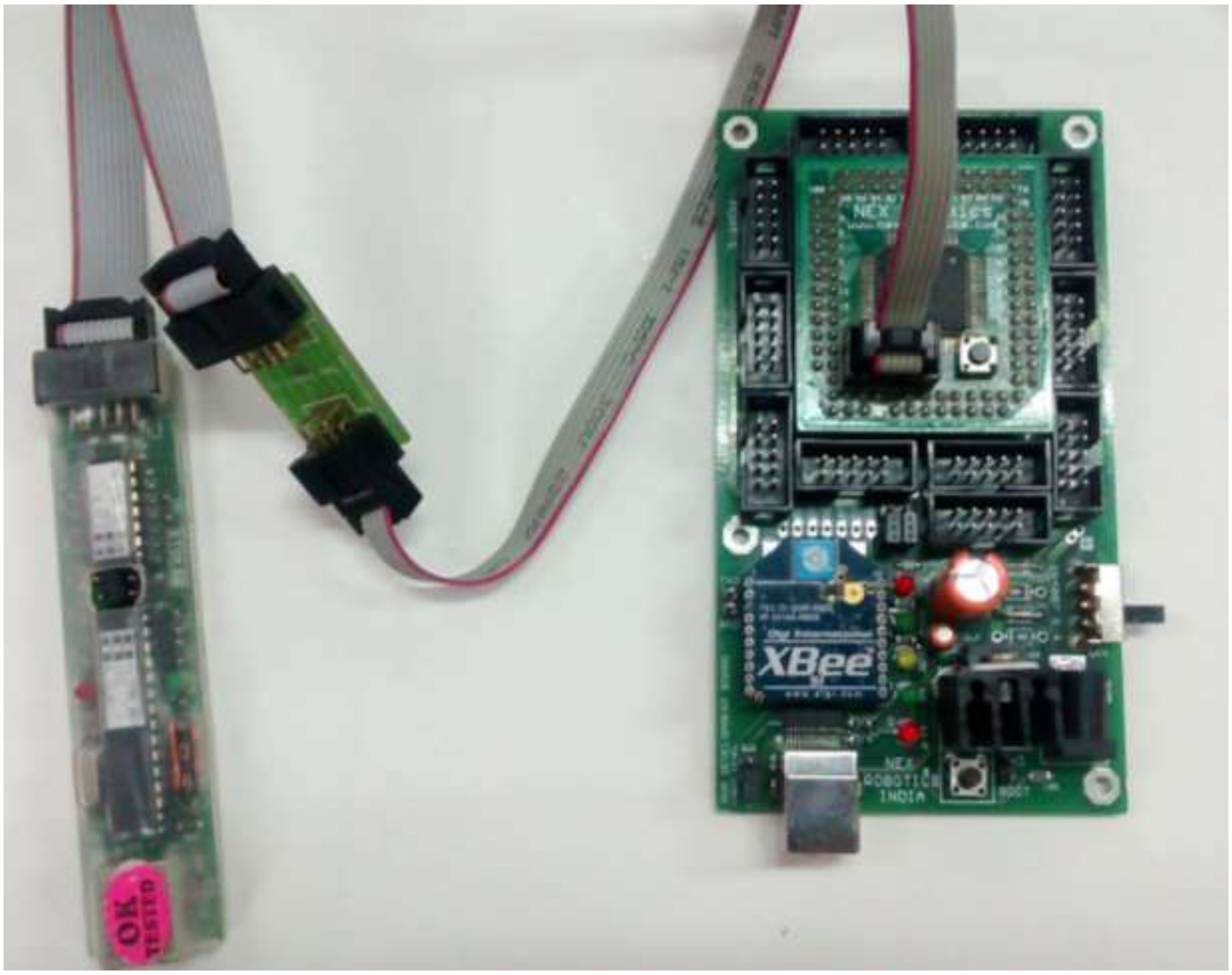
- Ensure that you have the following components in your kit as shown in figure:
    - STK500V2
    - 8 pin connector wire
    - 6 pin connector wire
    - 8-pin to 6-pin converter



- Connect STK500V2 to the converter using 8 pin connector wire and connect the 6-pin connector wire to the other end of the converter as shown below:



- Connect the other end of the 6-pin connector wire to the ISP connector of the ATmega 2560 as shown below:



- Connect power supply to the jack of the microcontroller and switch on the microcontroller.
- Connect your USB cable to the STK500V2 and follow the instructions given below to burn hex file to ATmega 2560 using STK500V2.

Installing drivers for AVR programming:

We will first install all the required packages. This includes

1. binutils - Tools like the assembler, linker.
2. gcc-avr - The GNU C cross compiler for AVR microcontrollers.
3. avr-libc - Package for the AVR C library, containing many utility functions.
4. uisp – Tool for in system programming of AVR microcontrollers.
5. avrdude – Utility to program AVR microcontroller. It supports the STK500v2 programmer

6. flex – Lexical analyser.
7. bison – Parser generator for C.
8. byacc – Another enhanced Parser generator for C. Modified version of Bison parser.

Development Environment (IDE) for writing C-program and compiling the program to generate the .hex file which can be loaded on the microcontroller.

These packages will be installed through the terminal. Open Linux terminal and type command one by one in the terminal to install the packages.

```
sudo apt-get install binutils gcc-avr avr-libc uisp avrdude flex bison byacc
```

Step:2 Creating Project with Arduino

Arduino is an IDE which is used for writing code for AVR and compile it to generate the .hex file. Refer Introduction_to_Arduino.pdf to create project

Step:3 Loading hex file on microcontroller memory

Step3.1

Download and save the script file ("avrdude_script.sh") supplied with this document on your desktop or any desired location. This script file has a bunch of commands which will help us load the .hex file on the microcontroller. We have to execute this script file with a .hex file as one argument. Following steps will describe the process in detail.

Step3.2

In Linux, we need to set file privileges and permissions for the user. A user can read a file, write to file or execute a file. Script file has to be in executable privilege.

Open Linux terminal and **" cd"** to the folder where script file is saved.

We have to first set an execute privilege to the script file. This is done by following Linux command

```
sudo chmod u+x avrdude script.sh
```

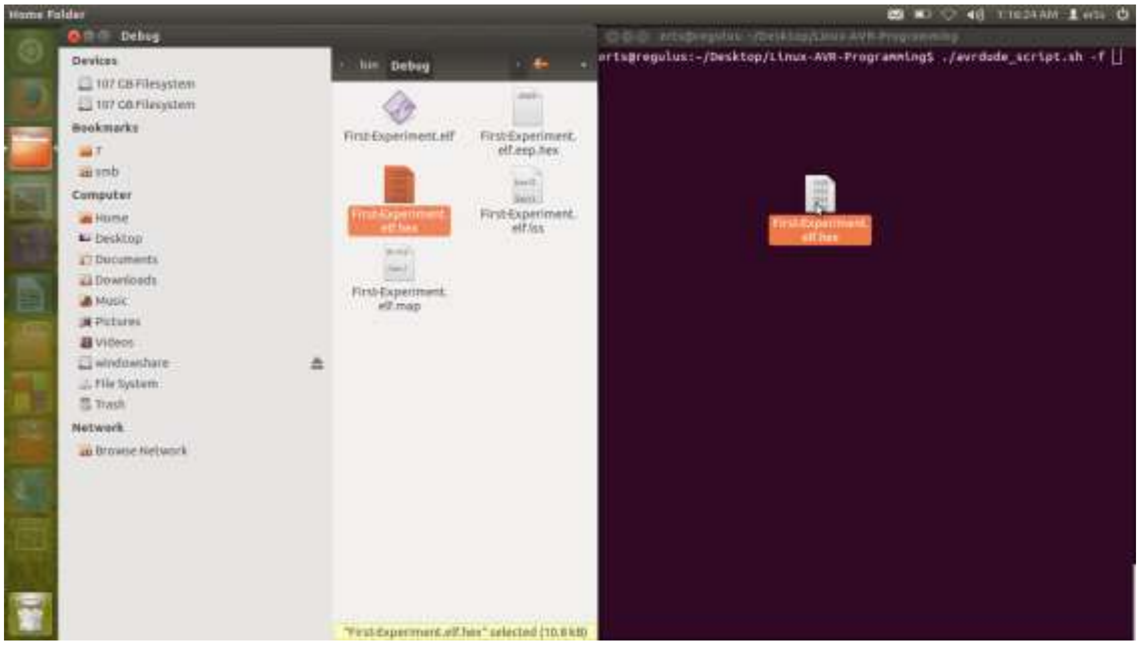
Note: File permission has to be set only once on a machine.

Step3.3

In terminal, use the following command to load the hex file. Note that avrdude_script.sh should be present in your parent directory.

```
./avrdude_script.sh –f <filename.hex>
```

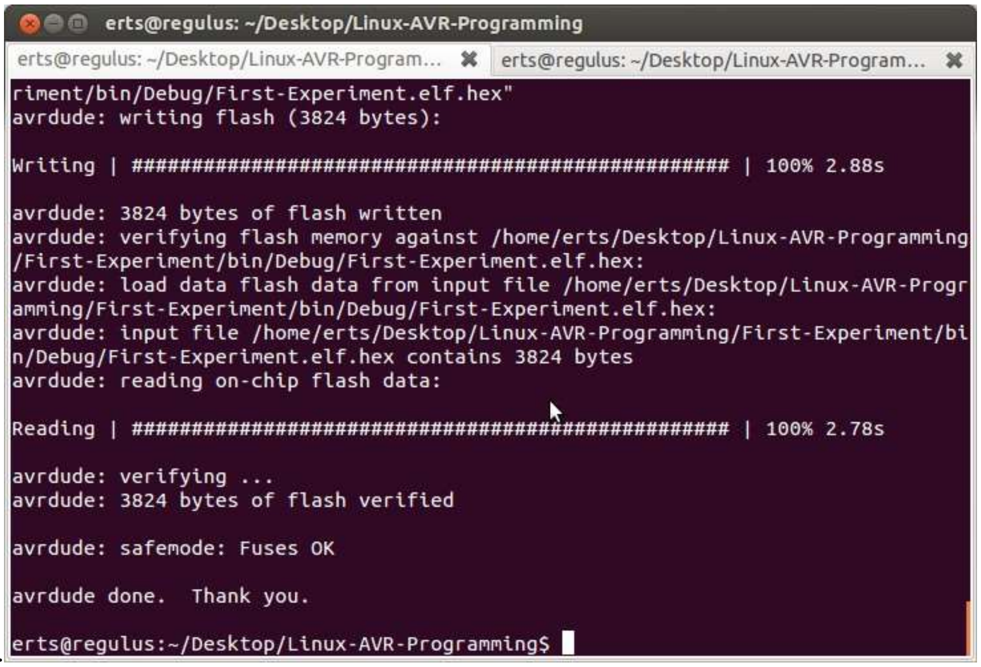
Note: Filename will be the .hex file which needs to be loaded on the microcontroller. Give the complete path of the Hex file. Easier way of doing this is to keep the terminal and directory window beside each other and drag the .hex file from directory and drop it into the terminal after **"./avrdude_script.sh –f".**



```
erts@regulus:~/Desktop/Linux-AVR-Programming$ ./avrdude_script.sh -f
```

Figure 1.2: Drag and Drop file in terminal

Program will be loaded on microcontroller and terminal will show the status as shown in figure 1.3 below



```
riment/bin/Debug/First-Experiment.elf.hex"
avrdude: writing flash (3824 bytes):

Writing | ################################################## | 100% 2.88s

avrdude: 3824 bytes of flash written
avrdude: verifying flash memory against /home/erts/Desktop/Linux-AVR-Programming
/First-Experiment/bin/Debug/First-Experiment.elf.hex:
avrdude: load data flash data from input file /home/erts/Desktop/Linux-AVR-Progr
amming/First-Experiment/bin/Debug/First-Experiment.elf.hex:
avrdude: input file /home/erts/Desktop/Linux-AVR-Programming/First-Experiment/bi
n/Debug/First-Experiment.elf.hex contains 3824 bytes
avrdude: reading on-chip flash data:

Reading | ################################################## | 100% 2.78s

avrdude: verifying ...
avrdude: 3824 bytes of flash verified

avrdude: safemode: Fuses OK

avrdude done.  Thank you.

erts@regulus:~/Desktop/Linux-AVR-Programming$
```

Figure 1.3: program loaded to controller

References:

1. File Permission in Linux. https://help.ubuntu.com/community/FilePermissions