Procedure for Single Board Heater System virtual labs

Rakhi R
Rupak Rokade
Inderpreet Arora
Kannan M. Moudgalya
Kaushik Venkata Belusonti

IIT Bombay
March 3, 2016

Chapter 1

Using Single Board Heater System, Virtually!

1.1 Introduction to Virtual Labs at IIT Bombay

The concept of virtual laboratory is a brilliant step towards strengthening the education system of an university/college, a metropolitan area or even an entire nation. The idea is to use the ICT i.e. Information and Communications Technology, mainly the Internet for imparting education or exchange of educational information. Virtual Laboratory mainly focuses on providing the laboratory facility, virtually. Various experimental set-ups are hooked up to the internet and made available to use for the external world. Hence, anybody can connect to that equipment over the internet and carry out various experiments pertaining to it. The beauty of this idea is that a college who cannot afford to have some experimental equipments can still provide laboratory support to their students through virtual lab, and all that will cost it is a fair Internet connection! Moreover, the laboratory work does not ends with the college hours, one can always use the virtual lab at any time and at any place assuming the availability of an internet connection.

A virtual laboratory for SBHS is launched at IIT Bombay. Here is the url to access it: **vlabs.iitb.ac.in/sbhs/**. A set of 36 SBHS are made available to use over the internet 24$\times$ 7. These individual kits are made available to the users on hourly basis. We have a slot booking mechanism to achieve this. Since there are 36 SBHS connected with an hours slot for 24 hrs a day, we have 864 one hour slots a day. This means that 864 individual users can access the SBHS in a day for an hour.

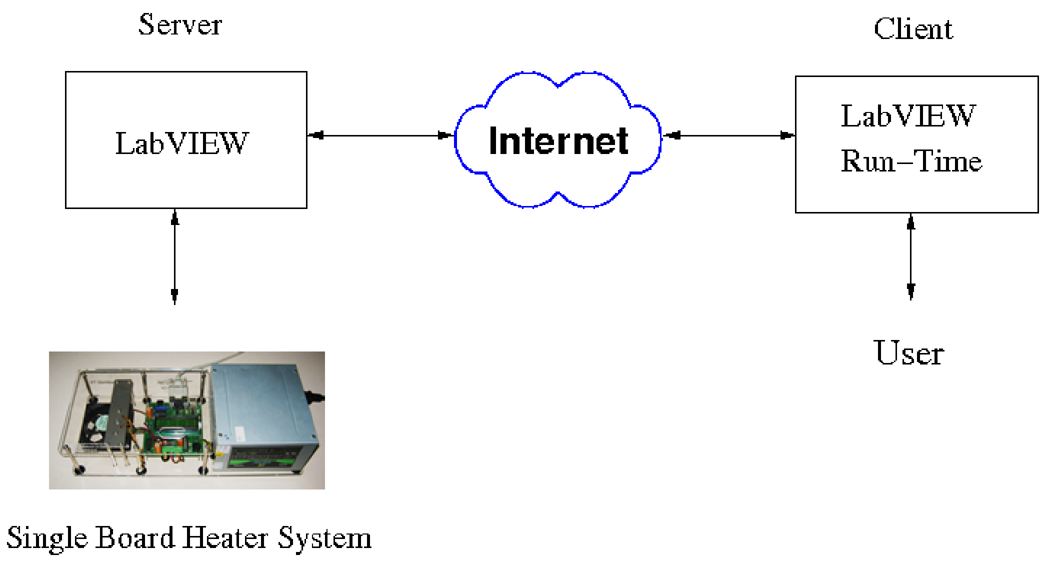


Figure 1.1: SBHS virtual laboratory with remote access using LabVIEW

This also means that up to 6048 users can use the SBHS for an hour in a week and 181440 in a month! A web page is hosted which is the first interface to the user. The user registers/logs in himself/herself here. The user is also supposed to book a slot for accessing the SBHS. A database server maintains a record of the data generated through the web interface. A python script is hosted on the server side and it helps in connecting the user with the corresponding SBHS placed remotely. A free and open source scientific computing Software, Scilab, is used by the user for implementing the experiment on SBHS, in terms of simple Scilab coding.

1.2 Evolution of SBHS virtual labs

In [?], the control algorithm is implemented at the server end and the remote student just keys in the parameters, as shown in Figure 1.1. LabVIEW was used for the implementation of the same. The server end consisted of a computer connected with an SBHS with a full blown copy of LabVIEW installed on it. The client has a LabVIEW run time engine available for free download from the National Instruments website. A few LabVIEW algorithms/experiments were hosted on the server. The client accesses these algorithm/experiment over the Internet using a web browser by entering appropriate parameters.

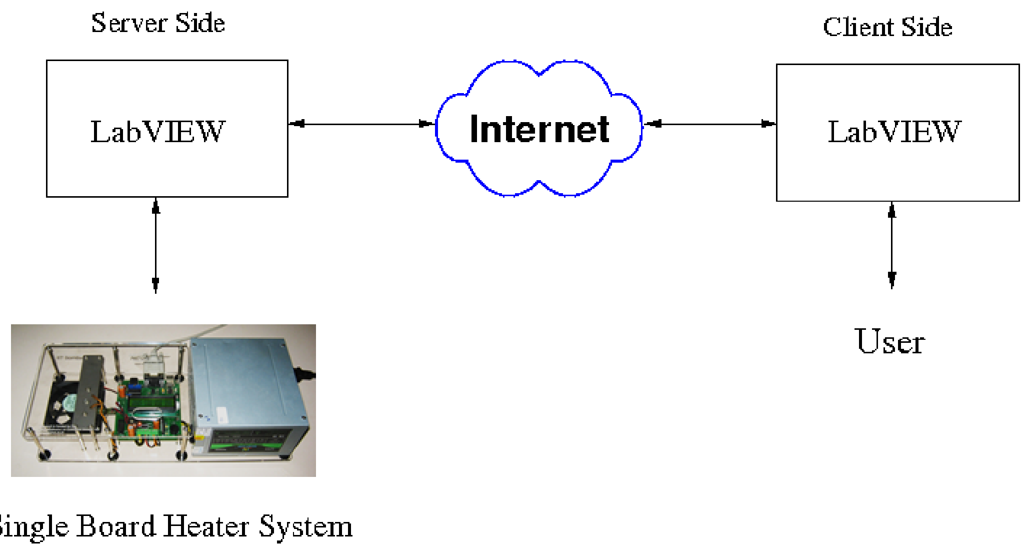


Figure 1.2: SBHS virtual laboratory with remote access and live data sharing using LabVIEW

It was realized that the learning experience is not complete for this structure. This is because the server hosts some pre-built LabVIEW algorithms and a user can only access these few algorithms. The user can in no way change the program and can only input experimental parameters. Hence, we came up with a new architecture as shown in the Figure 1.2 that used full blown copies of LabVIEW at both server and client ends.

This idea uses the DataSocket technology of LabVIEW. Since now the client is having a complete LabVIEW installation on his/her computer she can now implement her own algorithms. Thus this architecture did provide a complete learning experience to the students. There are some shortcomings as well:

- LabVIEW is expensive and students may not be able to afford to buy it. It is also prohibitively expensive for the Government to distribute it.
- We used the LabVIEW version 8.04, which had restricted scripting language. It was tedious to create new control algorithms in it.

This made us shift to free and open source (FOSS) software. We replaced LabVIEW with Java and Scilab as shown in Figure 1.3. Scilab at the server end is

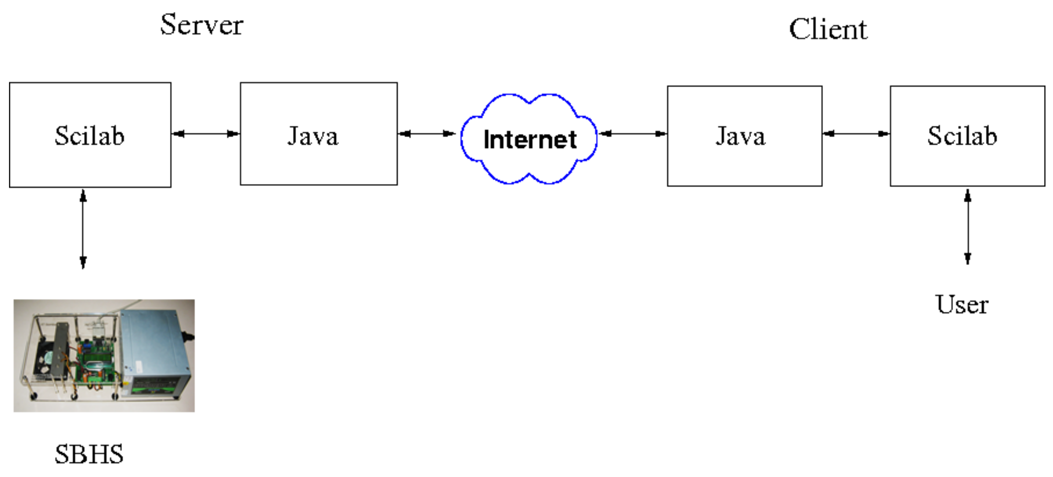


Figure 1.3: SBHS virtual laboratory using open source software

used for communicating with SBHS. Scilab at the client end is used for implementing the algorithms. Java is used at both the server as well as client end for communication over the Internet thereby connecting the client with the server.

For the above solution, we need a dedicated copy of scilab running at the server end for every SBHS. One way to do this is to host it on multiple computers with unique IPs. Hence the number of SBHS we want to host requires as many computer's and public IPs thereby making it expensive. Moreover, it also limits its scalability. The other way to do this is to host multiple java and scilab servers on the same computer. Hosting many copies of Scilab simultaneously requires a powerful computer for the server.

For these reasons we decided to take scilab off the server computer and to use java alone to communicate with the SBHS directly. Java also communicates with the client computer. We connected seven SBHS systems to a USB port through a serial port hub. This architecture was implemented on a Windows Operating System. We faced the following difficulties in this solution.

- When we connected more than one serial hub to a PC, the port ID could not be retrieved correctly. Port ID information is required if we want a student to use the same SBHS for all their experiments during different sessions.

- The experiments required time stamping of the data communicated to and from the server. But this time stamping was not linear and suffered instability.

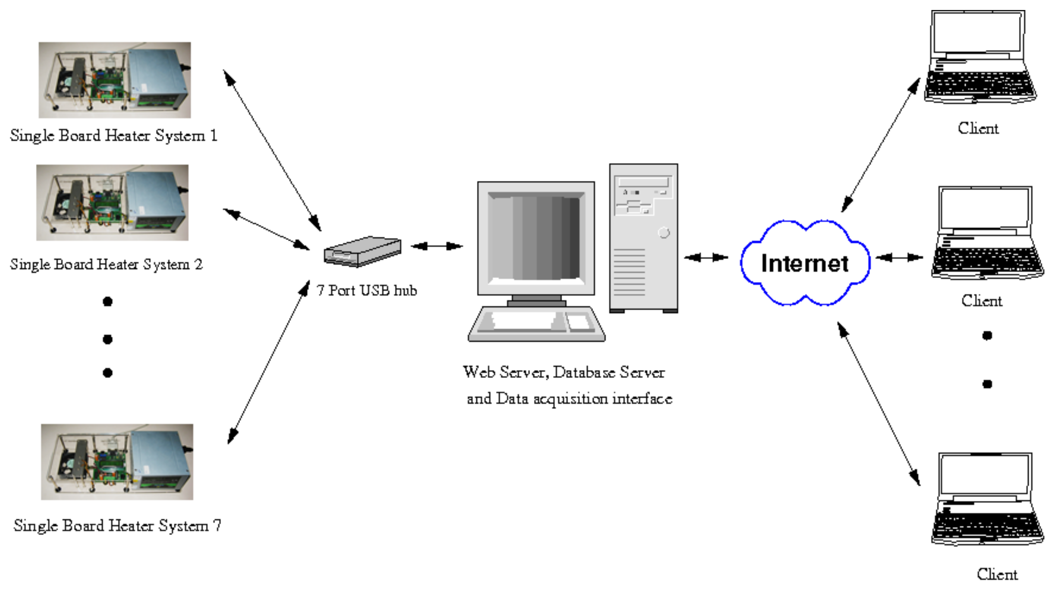
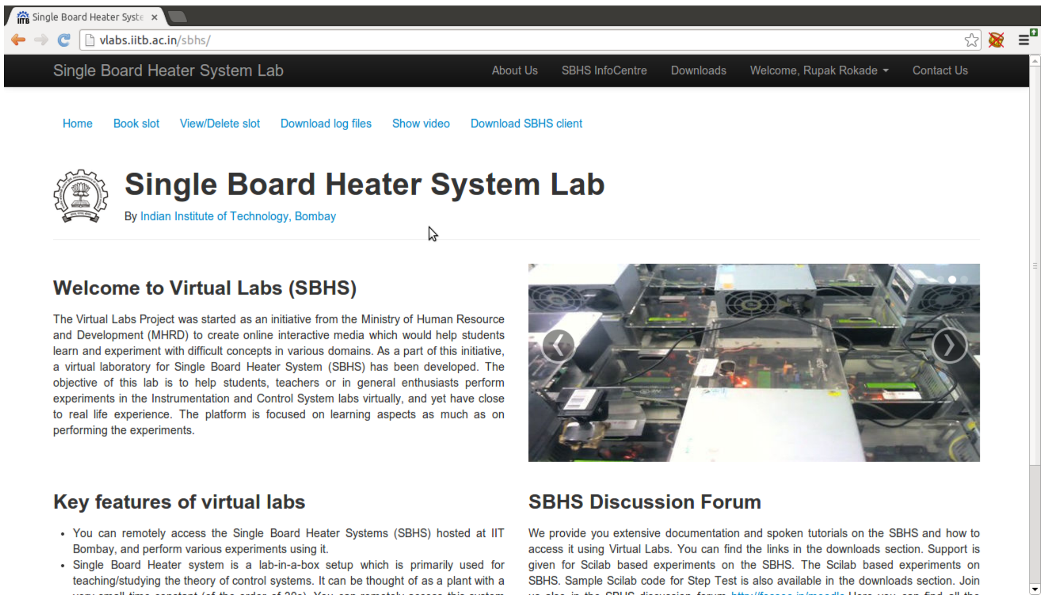

Single Board Heater System 1
Single Board Heater System 2
Single Board Heater System 7
7 Port USB hub
Web Server, Database Server
and Data acquisition interface
Internet
Client
Client
Client

Figure 1.4: Virtual control lab hardware architecture

This made us to completely switch to FOSS with Ubuntu Linux as the OS and is the current structure of the Virtual lab as shown in Figure 1.6

1.3 Current Hardware Architecture

The current hardware architecture of the virtual single-board heater system lab involves 36 single-board heater systems connected to the server via multiple 7 and 10-port USB hubs. The server computer is connected to a high speed inter-network and has enough processing capability to host data acquisition, database, and web servers. It has been successfully tested for the undergraduate Process Control course and the graduate Digital Control and Embedded systems courses conducted at IIT Bombay as well as few workshops over the internet. Currently, this architecture is integrated with a cameras on each SBHS to facilitate live video streaming. This gives the user a feel of remote hands-on.



Single Board Heater System Lab

By Indian Institute of Technology, Bombay

Welcome to Virtual Labs (SBHS)

The Virtual Labs Project was started as an initiative from the Ministry of Human Resource and Development (MHRD) to create online interactive media which would help students learn and experiment with difficult concepts in various domains. As a part of this initiative, a virtual laboratory for Single Board Heater System (SBHS) has been developed. The objective of this lab is to help students, teachers or in general enthusiasts perform experiments in the Instrumentation and Control System labs virtually, and yet have close to real life experience. The platform is focused on learning aspects as much as on performing the experiments.

Key features of virtual labs

- You can remotely access the Single Board Heater Systems (SBHS) hosted at IIT Bombay, and perform various experiments using it.
- Single Board Heater system is a lab-in-a-box setup which is primarily used for teaching/studying the theory of control systems. It can be thought of as a plant with a

SBHS Discussion Forum

We provide you extensive documentation and spoken tutorials on the SBHS and how to access it using Virtual Labs. You can find the links in the downloads section. Support is given for Scilab based experiments on the SBHS. The Scilab based experiments on SBHS. Sample Scilab code for Step Test is also available in the downloads section. Join

Figure 1.5: Home page of SBHS V Labs

1.4 Current Software Architecture

The current software architecture of this virtual SBHS control lab is shown in Figure 1.6. The server computer runs Ubuntu Linux 12.04.2 OS. It hosts a Apache-MySQL server. The SBHS server is based on Python-Django framework and is linked to Apache server using Apache's WSGI module. The MySQL database server has the details of all the registered users, their slot details, authentication keys to allow remote access, etc. As shown in Figure **??**, the Python-Django server has pages for registration, login, slot booking etc. [**?**]. On the client end, control algorithms are running in Scilab and a python based client application communicates with virtual labs server over the Internet.

The steps to be performed before and during each experiment are explained next.

1.5 Conducting experiments using the Virtual lab

This section explains the procedure to use Single Board Heater System remotely using Scilab i.e. when you are accessing SBHS remotly using your computer over

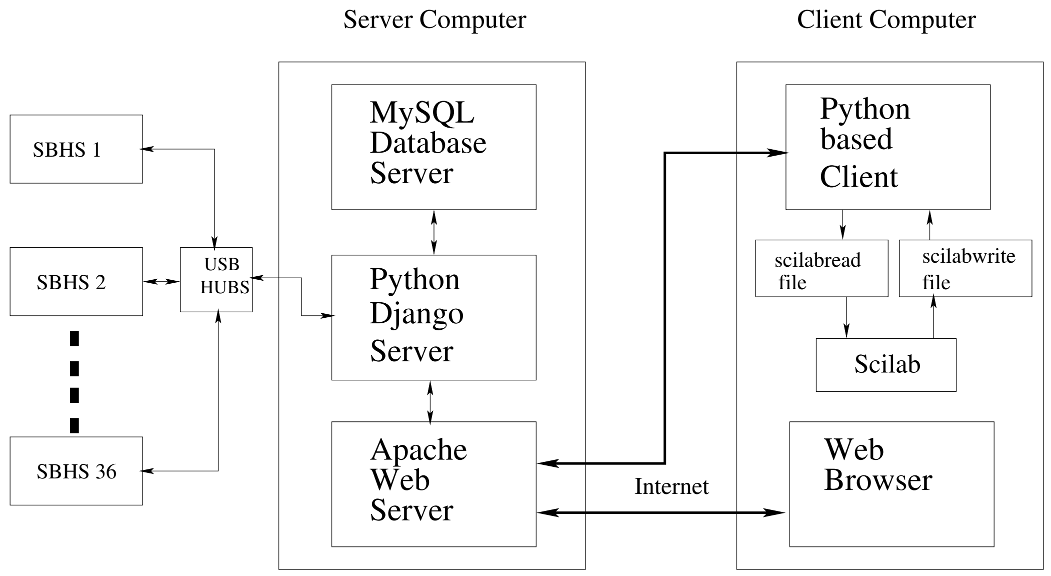


Figure 1.6: Current Architecture of SBHS Virtual Labs

the virtual labs platform. An open loop experiment, step test is used for demonstrating this procedure. The process however remains the same for performing any other experiment explained in this section, unless specified otherwise. Let us first see the required files to be downloaded and installations to be done. Scilab is required to be installed on your computer. Please refer to Section **??** and Section **??** for the procedure to install scilab on Windows and Linux system, respectively.

SBHS scilab code for your OS, under the section `SBHS Virtual Code`, must be downloaded from `http://sbhs.os-hardware.in/downloads`. For example, if you are using a 32-bit linux operating system then you should download the file `SBHS Scilab codes for Linux - (32 bit)`. The code downloaded will be in zip format. After the zip is unpacked, you will see scilab experiment folders such as `Step_test`, `Ramp_Test`, `pid_controller` etc. We will be using the Step_test folder. Do not alter the directory structure. If you want to copy or move an experiment outside the directory then make sure you also copy the `common_files` folder. The `common_files` folder must always be one directory outside the experiment folder. Now given that you have scilab installed and working and the required scilab code downloaded, let us see the step-by-step procedure to do a remote experiment.

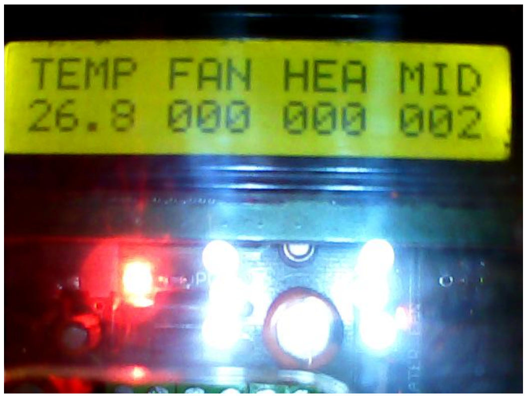


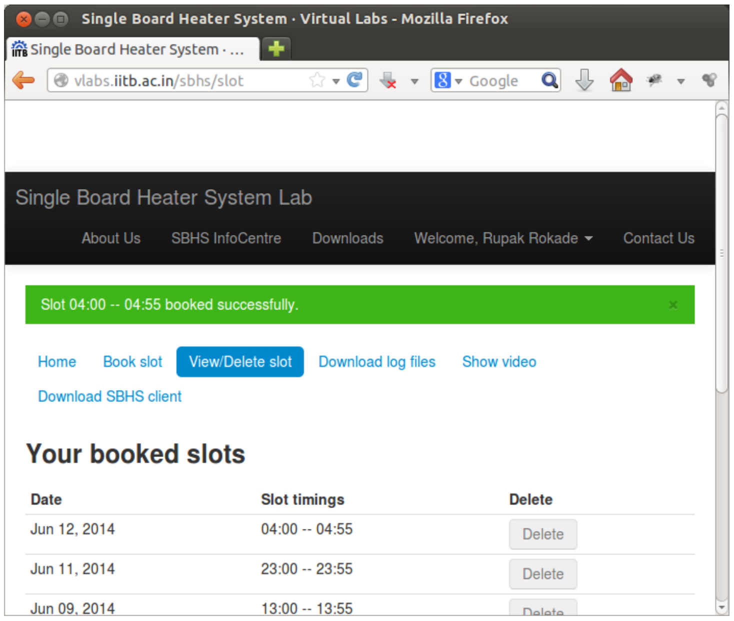
Figure 1.7: Show Video

1.5.1 Registration, Login and Slot Booking

Go to the website `sbhs.os-hardware.in` and click on the `Virtual labs` link available on the left hand side. The home page of Virtual labs is illustrated in Fig 1.5. If you are a first time user, click on the link `Login/Register`. Fill out the registration form and submit it. If the registration form is submitted successfully, you will receive an activation link on your registered Email id. Use this link to complete the registration process. If you skip this step you will not be able to login. Registration is a one time process and need not be repeated more than once. After completing registration login with your username and password. You should now get the options to Book Slot, Delete Slot etc.

View/Delete slot option allows you to delete your booked slots. This option however will work only for slots booked for the future. You cannot delete a past or the current slot. Download log files option gives you the facility to download your experiment log files. Clicking on it will give you a list of all of the experiments you had performed. Show video option can be used to see the live video feed of your SBHS. Web cameras are mounted on every SBHS. You can see the display of your SBHS as shown in Fig. 1.7.

Clicking on the Book slot option will allow you to book an experiment time slot. Slots are of 55 minutes duration. Click on the Book slot option. If the current slot is free, Book now option will appear. Click on it. Else you have to book an advance slot for the next hour or any other future time using the calender that appears on this page. There is a limit to how many slots one can book in a day. We are allowing only two non-consecutive slots, per user, to be booked in a day. However, there is no limit to how many current slots you book and use. Book an


Single Board Heater System · Virtual Labs - Mozilla Firefox

vlabs.iitb.ac.in/sbhs/slot

Single Board Heater System Lab

About Us SBHS InfoCentre Downloads Welcome, Rupak Rokade Contact Us

Slot 04:00 -- 04:55 booked successfully.

Home Book slot View/Delete slot Download log files Show video Download SBHS client

Your booked slots

Date	Slot timings	Delete
Jun 12, 2014	04:00 -- 04:55	Delete
Jun 11, 2014	23:00 -- 23:55	Delete
Jun 09, 2014	13:00 -- 13:55	Delete

Figure 1.8: Slot booking

experiment slot. Once you successfully book a slot a **Slot booked successfully** message highlighted in green color will appear on the top side. This is shown in Fig. 1.8. It will automatically take you to the View/Delete slot page.

1.5.2 Configuring proxy settings and executing python based client

After booking a slot, the web activity is over. You may close the web browser unless you need it open to see live video feed of your SBHS. The next step is to establish the communication link between the server and your computer. A python based application is created which handles the network communication.

Let us first see how to do the proxy settings if you are behind a proxy network. Open the folder `common_files`. Open the file `config`. This files contains various arguments whose values must be eneterd to configure proxy.

Do not change the contents of `config` file if

- You are accessing from inside IIT Bombay OR
- You are accessing from outside IIT Bombay and using an open network

such as at home OR using a mobile internet

Change the contents of `config` file if

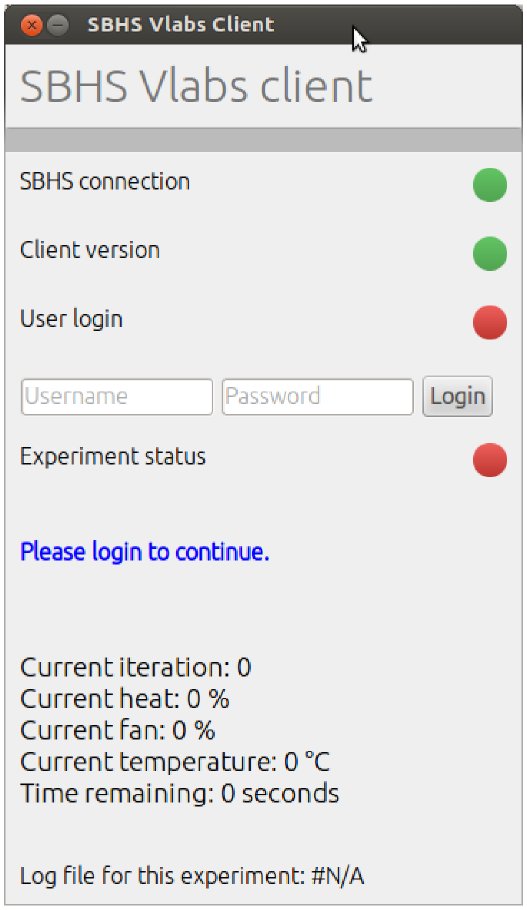
- You are outside IIT Bombay and using a proxy network such as at an institute, office etc.

If you have to put the proxy details, first change the argument `use_proxy` = Yes (Y should be capital in Yes and N should be capital in No). Fill in the other details as per your proxy network. If your proxy network allows un-authenticated login then make the argument `proxy_username` and `proxy_password` blank. This proxy setting has to done only once.

Open the `Step_test` folder. Double click on the file `run`. This will open the client application as shown in Fig. 1.9. Note that for first time execution, it will take a minute to open the client application. It will show various parameters related to the experiment such as SBHS connection, Client version, User login and Experiment status. The green indicators show that the corresponding activity is correct or functional. Here it says that the Client application is been able to connect to the server and the client version being used is the latest. The User login and Experiment status is showing red and will turn green after a registered username and password is entered. If the SBHS is offline or there are some other issues, the corresponding error will be displayed and the respective indicator will turn red. Enter your registered username and password and press login. You should get the message `Ready to execute scilab code`. The application also shows the value of iteration, heat, fan, temperature and time remaining for experimentation. It also shows the name of log file created for the experiment.

1.5.3 Executing scilab code

Inside the `StepTest` folder, if on a windows system, double click on the file `stepc.sce`. This should automatically launch scilab and also open the `stepc.sce` in the scilab editor. It will also automatically change the scilabs working directory. On a linux system, launch scilab manually. Then change the scilab working directory to the folder `StepTest`. This can be done by clicking on `File` menu and then selecting `change current directory`. Next, execute the command `getd ../common_files`. Scilab command `getd` is used to load all functions defined in all .sci files inside a specified folder. Here we have some important function files inside the `../common_files` directory. Executing this command will load all of

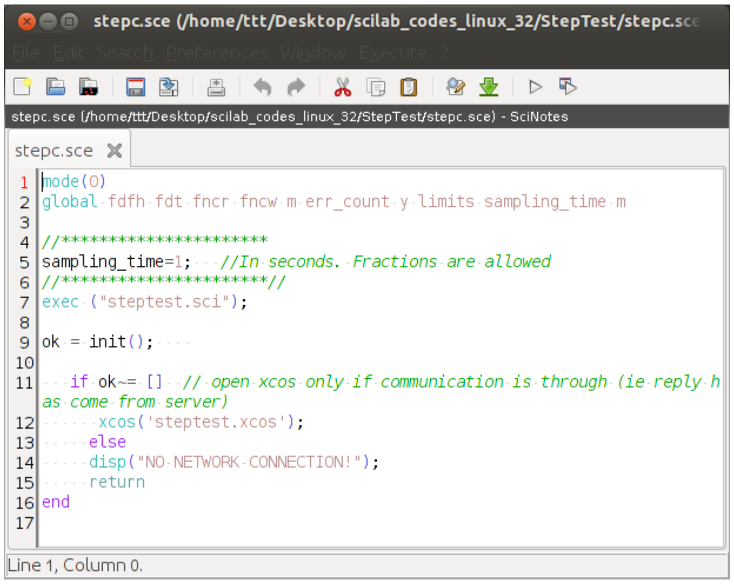


Please login to continue.

Current iteration: 0
Current heat: 0 %
Current fan: 0 %
Current temperature: 0 °C
Time remaining: 0 seconds

Log file for this experiment: #N/A

Figure 1.9: Python Client


stepc.sce (/home/ttt/Desktop/scilab_codes_linux_32/StepTest/stepc.sce
File Edit Search Preferences Window Execute ?
stepc.sce (/home/ttt/Desktop/scilab_codes_linux_32/StepTest/stepc.sce) - SciNotes
stepc.sce
Line 1, Column 0.

```
mode(0)
global fdfh fdt fncr fncw m err_count y limits sampling_time m

//**********************
sampling_time=1;   //In seconds. Fractions are allowed
//************************//
exec ("steptest.sci");

ok = init();

   if ok~= [] // open xcos only if communication is through (ie reply h
as come from server)
      xcos('steptest.xcos');
     else
     disp("NO NETWORK CONNECTION!");
     return
end
```

Figure 1.10: stepc.sce file

the functions that the experiment needs. Open the file `stepc.sce` using the `Open` option inside `File` menu. The file is shown in Fig. 1.10

The experiment sampling time can be set inside the `stepc.sce` file. You may want to change it to a higher value if your network is slow. The default value of 1 second works fine in most cases. On the menu bar, click on `Execute` option and choose option `file with echo`. This will execute the scilab code. If the network is working fine, an xcos diagram will open automatically. If it doesnt open then see the scilab console for error messages. If you get a `No network connection` error message then try executing the scilab code again. The xcos diagram is for the step test experiment as shown in Fig. 1.11. You can set the value of the heat and fan. Keep the default values. On the menu bar of the xcos window, click on `start` button. This will execute the xcos diagram. If there is no error, you will get a graphic window with three plots. It will show the value of Heat in % Fan in % and ..temperature in degree celcius as shown in Fig. 1.12. After sufficient time of experimentation click on the stop button to stop the experiment. Go to the `StepTest` folder. Here you will find a `logs` folder. This folder will have another folder named after your username. It will have the log file for your experiment. Read the log file name as
YearMonthDate_hours_minutes_seconds.txt. This log file contains all the values

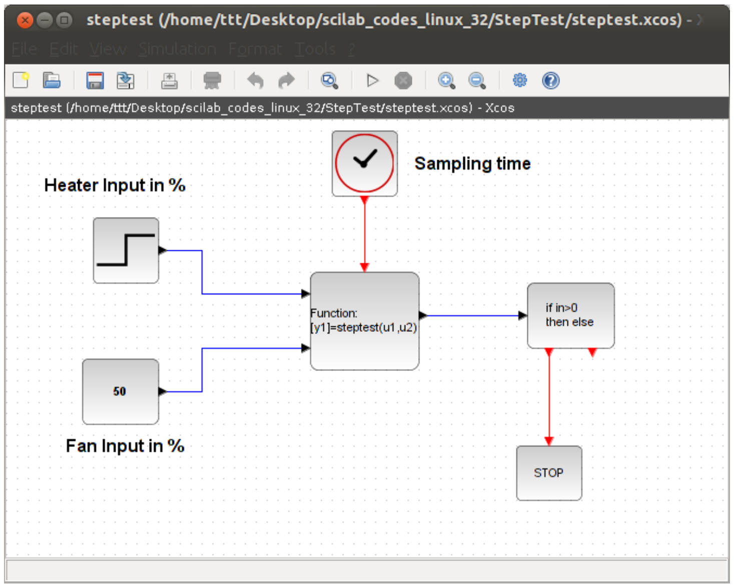

steptest (/home/ttt/Desktop/scilab_codes_linux_32/StepTest/steptest.xcos) -
File Edit View Simulation Format Tools ?
steptest (/home/ttt/Desktop/scilab_codes_linux_32/StepTest/steptest.xcos) - Xcos
Heater Input in %
Sampling time
Function:
[y1]=steptest(u1,u2)
if in>0
then else
50
Fan Input in %
STOP

Figure 1.11: Xcos for step test

of heat fan and temperature. It can be used for further analysis.

1.6 Summary

This section summarizes the process to perform an experiment on SBHS using virtual lab interface. This section assumes that the user has already created an account and booked a slot as explained in section 1.5.1. It also assumes that the proxy settings are already done as explained in section 1.5.2. The user should follow these steps within the booked slot time.

- Step1: Open the StepTest experiment directory
- Step2: Double-click on the file `run`. Expect the SBHS cient application to open.
- Step3: Enter the username and password inside the SBHS cient application and press `login` button. Expect the message `Ready to execute scilab code`
- Step4: Switch to the StepTest experiment directory and double-click on the file `stepc.sce`. This will launch scilab and also open the file `stepc.sce`

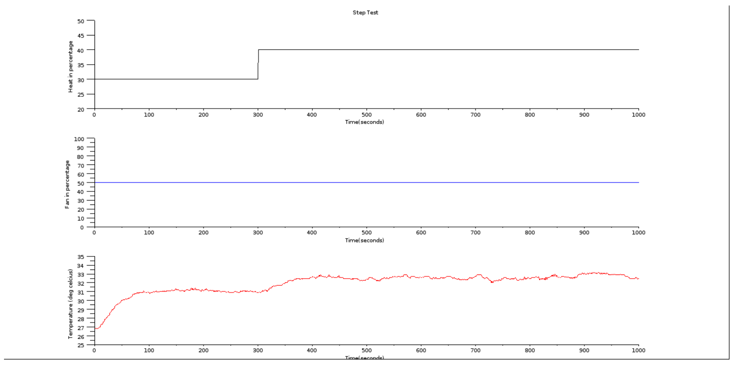


Figure 1.12: Output of Step Test

in the scilab editor. Linux users will have to launch scilab manually. They also have to change the working directory to StepTest and then open the `stepc.sce` file in the scilab editor.

- Step5 Switch to the scilab console and execute the command `getd ../common_files`
- Step6: Execute the file `stepc.sce`. Expect the step test xcos diagram to open automatically. If this doesnt happen, check the scilab console for error message.
- Step7: Execute the step test xcos daiagram. You may change the input parameters, if required, before executing. Expect a plot window to open automatically showing three graphs.
- Step8: Stop the Xcos simulation after the experiment is completed properly.