



Summer Fellowship Report

On

**Fixing eSim Installer for Windows 8 OS, Debugging
NGHDL Error and Solving Path Problem**

Submitted by

Manasi Yadav

Under the guidance of

Prof.Kannan M. Moudgalya
Chemical Engineering Department
IIT Bombay

August 23, 2021

Acknowledgment

It is a pleasure to acknowledge the help and support that has gone to in making this successful project. I express sincere gratitude to Prof. Kannan Moudgalya and FOSSEE Team at IIT Bombay for providing me with this opportunity to work on this project and having faith in my abilities.

I would also like to thank the eSim team for giving me such a great opportunity of learning, being a part of such a wonderful project in such difficult pandemic situations, and also for enabling me for their help in offering me the resources and guiding me throughout the project.

I would also like to thank my Project managers, Usha Viswanathan, Vineeta Ghavri, and Gloria Nandihal for their guidance and support throughout the fellowship.

A special thanks to all my mentors, Sumanto Kar, Rahul Paknikar, and Saurabh Bansode for helping me throughout the fellowship, sharing a lot of valuable and constructive suggestions during the planning and development of this research work.

Finally, I wish to thank my parents and my sister for consistently encouraging me and keeping me positive at all times.

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 3 |
| 1.1 | eSim | 3 |
| 1.2 | NGHDL | 3 |
| 1.3 | NSIS | 3 |
| 1.4 | PyQt5 | 4 |
| 2 | eSim in Windows 8 OS | 5 |
| 3 | Solutions | 6 |
| 3.1 | Solution - 1: Breaking Bluetooth API Module | 6 |
| 3.1.1 | Functions of Bluetooth API | 6 |
| 3.1.2 | Methodology | 6 |
| 3.1.3 | Error in Windows 7 OS | 7 |
| 3.2 | Solution - 2: Downgrading PyQt5 | 7 |
| 3.2.1 | Methodology | 7 |
| 3.2.2 | Addition of dependencies to MinGW64 | 8 |
| 3.3 | Comparing Solution 1 and Solution 2 | 9 |
| 4 | Solving Path Problem | 10 |
| 4.1 | Issue | 10 |
| 4.2 | Solution | 11 |
| 5 | Solving NGHDL Error | 12 |
| 5.1 | Introduction | 12 |
| 5.2 | Solution | 13 |
| 6 | Building eSim Installer for MS Windows | 14 |
| 6.1 | Creating eSim Executable File | 14 |
| 6.2 | Creating NGHDL Executable File | 15 |
| 6.3 | Packaging of NGHDL | 16 |
| 6.4 | Packaging of eSim | 17 |
| 7 | Conclusion | 19 |
| 8 | Future Work | 20 |
| | Bibliography | 21 |

Chapter 1

Introduction

1.1 eSim

eSim is a free/libre and open source EDA tool for circuit design, simulation, analysis and PCB design developed by FOSSEE, IIT Bombay. It is an integrated tool built using free/libre and open source software such as KiCad, Ngspice, NGHDL and GHDL.

1.2 NGHDL

NGHDL is a mixed mode circuit simulator developed by FOSSEE, using NgSpice and GHDL. NGHDL feature facilitates creation of user-defined models for mixed-signal circuit simulation in eSim. Mixed-signal simulation is achieved by interfacing GHDL and Ngspice. In NGHDL, NgSpice is used to simulate the analog components and GHDL is used to simulate the digital components, where the analog and digital components are communicating through socket. As in Ngspice its difficult to write our own code models for digital circuits and many people being familiar with VHDL language is the main reason for introducing NGHDL. NGHDL provides an interface to write VHDL code for a digital model and install it as model in Ngspice. So whenever Ngspice looks for that model, it will actually interface with VHDL code to get the result.

1.3 NSIS

NSIS (Nullsoft Scriptable Install System) is a professional open source system to create installers for Windows. It is released under an open source license and is completely free for any use. NSIS creates installers that are capable of installing, uninstalling, setting system settings, extracting files, etc. The scripting language supports variables, functions and string manipulation, just like a normal programming language - but designed for the creation of installers.

1.4 PyQt5

PyQt5 is a comprehensive set of Python bindings for v5 of the Qt application framework from The Qt Company. Qt is a set of C++ libraries and development tools. PyQt5 comprises PyQt5 itself and a number of add-ons that correspond to Qt's additional libraries. Each is provided as a source distribution (sdist) and binary wheels for Windows, Linux and macOS. PyQt5 supports the Windows, Linux, UNIX, Android, macOS and iOS platforms and requires Python v3.5 or later. (PyQt5 should also build against Python v2.7 and earlier versions of Python v3 using the legacy `configure.py` build script but this is unsupported.)

Chapter 2

eSim in Windows 8 OS

It was reported that eSim 2.1 crashes in Windows 8 OS with a pop-up error which states- "This Windows version (6.3.9600) does not support the required Bluetooth API. Consider updating to a more recent Windows (10.0.10586 or above)".

The snippet of the error is shown in Figure 2.1.



Figure 2.1: Error message window in Windows 8 OS

Chapter 3

Solutions

The error exists due to the incompatibility of Bluetooth API with the Windows version. In order to fix this issue, the following two approaches have been followed.

3.1 Solution - 1: Breaking Bluetooth API Module

3.1.1 Functions of Bluetooth API

The Bluetooth API provides connectivity between Bluetooth enabled devices. The following Bluetooth functionalities are provided:

1. Discovers nearby Bluetooth devices (Device discovery, including Bluetooth LE devices)
2. Gets bonded devices information
3. Controls bonding
4. Connects to a service on a remote device and exchanges data
5. Registers a service (RFCOMM) on a local device, which can be consumed by remote devices to exchange data
6. Advertise for remote devices (including Bluetooth LE devices)
7. Act as a GATT client (Generic Attribute Profile client)

3.1.2 Methodology

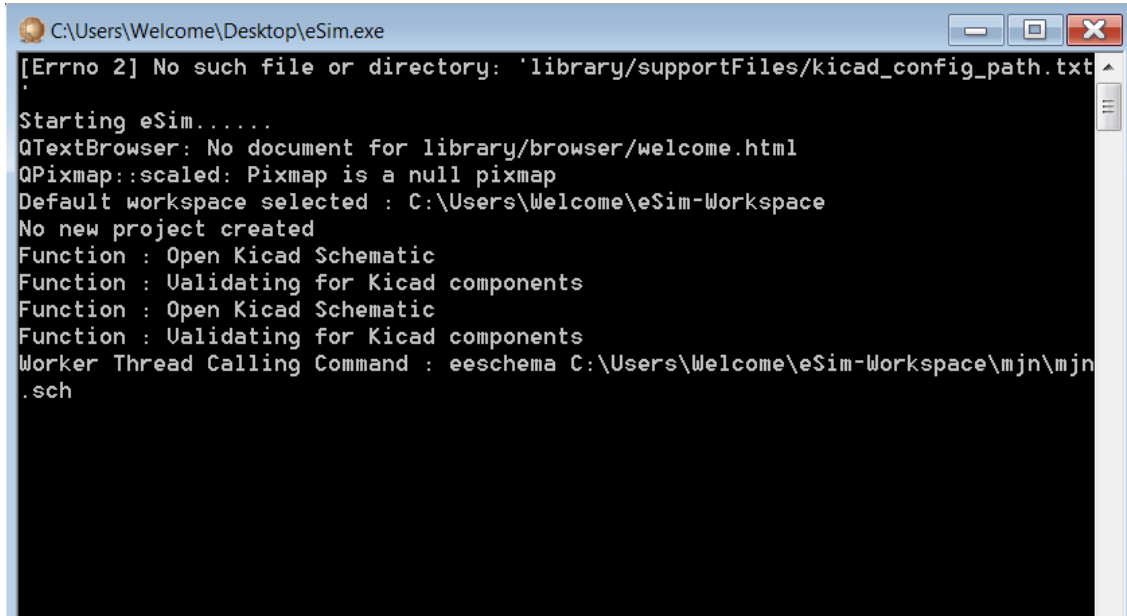
The Bluetooth API implemented into the current version of PyQt5 does not support Windows 8 OS.

Hence, to break the Bluetooth API module, Go to the folder `C:\Python37\Lib\site-packages\PyQt5\Qt5\bin` (or where the Python is installed in) & rename the file `Qt5Bluetooth.dll` to `Qt5Bluetooth.dll1`. After which the installer for eSim is generated in Windows 8 OS using the steps mentioned in Section 6.

3.1.3 Error in Windows 7 OS

After breaking the Bluetooth API Module, eSim installer was generated for Windows 7 OS. While running eSim, a path error was observed on the console window of eSim.

The snippet of the error is shown in Figure 3.1.



```
C:\Users\Welcome\Desktop\eSim.exe
[Errno 2] No such file or directory: 'library/supportFiles/kicad_config_path.txt'
Starting eSim.....
QTextBrowser: No document for library/browser/welcome.html
QPixmap::scaled: Pixmap is a null pixmap
Default workspace selected : C:\Users\Welcome\eSim-Workspace
No new project created
Function : Open Kicad Schematic
Function : Validating for Kicad components
Function : Open Kicad Schematic
Function : Validating for Kicad components
Worker Thread Calling Command : eeschema C:\Users\Welcome\eSim-Workspace\mjn\mjn
.sch
```

Figure 3.1: Error message window in Windows 7 OS

3.2 Solution - 2: Downgrading PyQt5

3.2.1 Methodology

It was observed that this issue has been introduced by higher versions of PyQt5. The existing method of eSim installer generation installs the latest version of PyQt5 i.e. 5.15.4. The Bluetooth API present in this version is not supported by the Windows 8 OS.

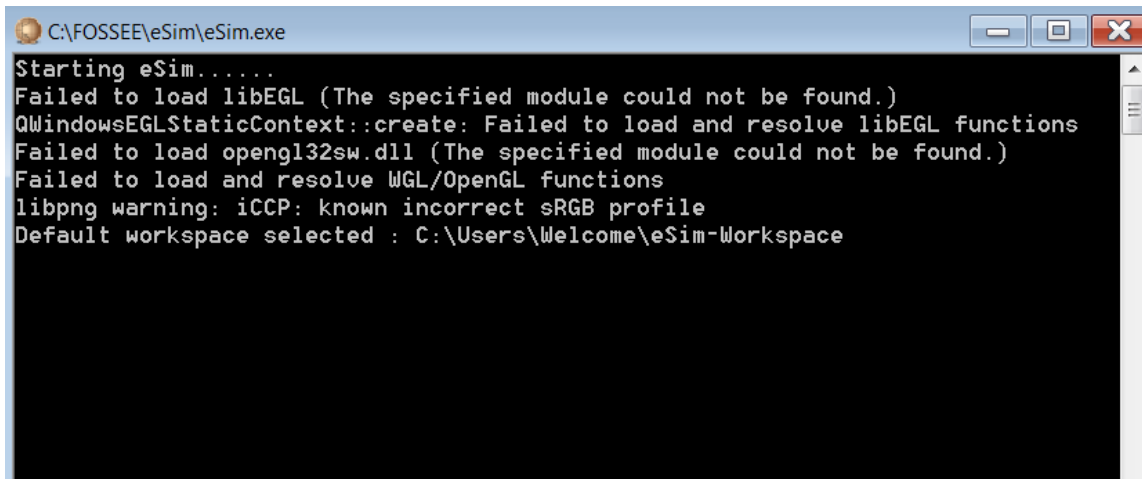
To fix this problem, PyQt5 needs to be downgraded to a version lower than 5.10. The required version of PyQt5 can be installed by specifying its version during the installer generation. After which the installer for eSim is generated in Windows 8 OS using the steps mentioned in Section 6. Thus, PyQt5 is downgraded from version 5.15.4 to 5.9.2 using the following commands.

```
$ pip uninstall pyqt5
$ pip install pyqt5==5.9.2
```


3.2.2 Addition of dependencies to MinGW64

eSim installer was generated for Windows 8 OS after downgrading PyQt5. It was observed that error messages appeared on the console window in Windows 8 OS. The error messages were encountered because of the missing DLL files required for smooth functioning of eSim.

The snippet of the error is shown in Figure 3.2.

A screenshot of a Windows command prompt window titled "C:\FOSSEE\eSim\eSim.exe". The window contains the following text:

```
Starting eSim.....
Failed to load libEGL (The specified module could not be found.)
QWindowsEGLStaticContext::create: Failed to load and resolve libEGL functions
Failed to load opengl32sw.dll (The specified module could not be found.)
Failed to load and resolve WGL/OpenGL functions
libpng warning: iCCP: known incorrect sRGB profile
Default workspace selected : C:\Users\Welcome\eSim-Workspace
```

Figure 3.2: Error message window in eSim

Some Python dependencies required for functioning of eSim were found to be missing, due to which this error occurred. Thus, the following dependencies were added to the C:\FOSSEE\mingw64\bin.

1. **opengl32sw.dll**
The opengl32sw.dll file is created to make the program run more efficiently.
2. **vcruntime140_1.dll**
The vcruntime140.dll is a DLL file where the codes of certain applications are stored.
3. **d3dcompiler_47.dll**
D3DCompiler_47.dll usually contains a set of procedures and driver functions, which may be applied by Windows.
4. **libEGL.dll**
libEGL.dll is essentially a "guide book" that stores information and instructions for executable files.
5. **libGLESv2.dll**
The libglesv2.dll is an executable file on the system's hard drive which contains the machine code.

3.3 Comparing Solution 1 and Solution 2

On comparing both the solutions, solution 2 has been selected and implemented to fix eSim installer for Windows 8 OS. The solution 2 is preferred over solution 1 because of the following reasons:

1. A path error is encountered in Windows 7 OS after breaking Bluetooth API Module, as mentioned in Section 3.1.3.
2. Breaking or removing a module may affect functioning of other modules as well, hence it is considered as unsafe.
3. Downgrading PyQt5 allows Bluetooth API functions to work for eSim and does not compromise with the working of eSim.

Chapter 4

Solving Path Problem

4.1 Issue

In some systems due to presence of an unknown software, the HOME variable of the system gets fixed to C:\SPB_Data\. This is mostly observed in systems with pre-installed ORCAD software. Similar kind of issue is observed in many softwares such as VirtualBox, ORCAD, PSpice, Node-RED, Cygwin, etc.

The path of eSim installation directory also gets fixed to C:\SPB_Data\. Thus, the NSI script **esim-setup-script.nsi** fails to update or modify the path of the HOME variable to C:\Users\<>username>, which results in an error as shown in Figure 4.1.

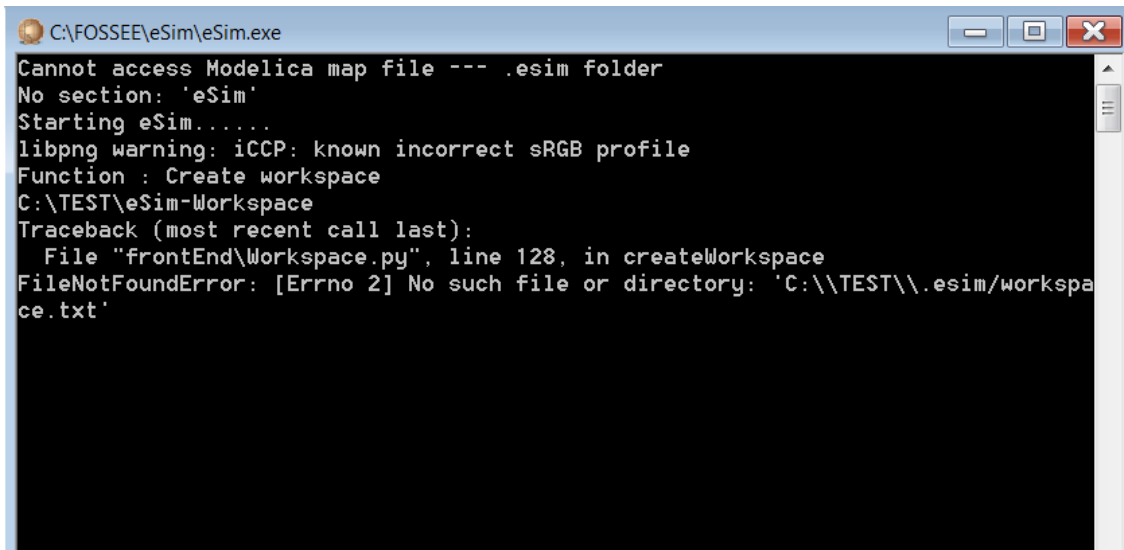


Figure 4.1: Error message window in eSim

4.2 Solution

Till now, the error was solved by moving .esim and .nghdl folders from C:\SPB_Data\ to C:\Users\\.

To avoid this error completely and to make the installation process hassle free, a batch file approach has been followed.

In Windows, the batch file is a file that stores commands in a serial order. Command line interpreter takes the file as an input and executes in the same order. A batch file is simply a text file saved with the .bat file extension. It can be written using Notepad or any other text editor.

By default, a batch file will display its command as it runs. The purpose of `echo off` command is to turn off this display, except for the command itself. The `@` in front makes the command apply to itself as well, thus the command `@echo off` turns off the display for the whole script.

A batch file (extension **.bat**) is packaged along with eSim.exe in FOSSEE\esim folder, in which the HOME variable points to the path of .esim folder. The value of HOME variable is set as C:\Users\\. The batch file closes automatically and prompts eSim.exe to start.

Note: The user has to run this batch file instead of eSim.exe.

The batch file is named as **eSim.bat** for convenience. It is packaged along with eSim.exe (eSim.exe can be renamed as Application.exe so as to avoid confusion), using the steps as mentioned in Section 6.2.

The batch file (eSim.bat) consists of the following commands.

```
@echo off
set HOME=C:\Users\\
start eSim.exe
cd /d %HOME%
```

The following command line is added to the NSI Script `esim-setup-script.nsi` to package the eSim batch file along with eSim executable.

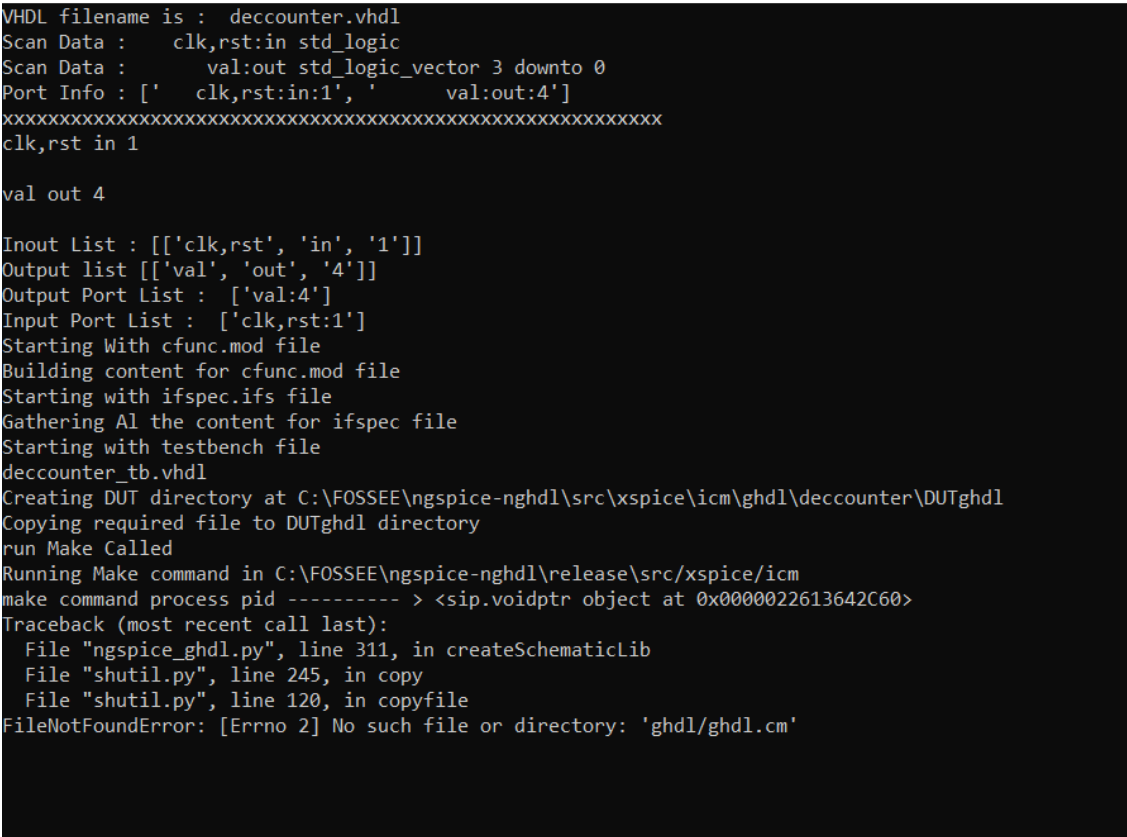
```
FileWrite $0 'eSim = \%(eSim_HOME)s\eSim.bat$\n'
```

Chapter 5

Solving NGHDL Error

5.1 Introduction

Several checks needs to be done before using the NGHDL feature of eSim, as mentioned in Appendix E of the User Manual. One of them is the declaration of each port that has to be done on a new line in a VHDL file. If two or more ports are declared on the same line, an error appears as shown in Figure 5.1.



```
eSim
VHDL filename is : deccounter.vhdl
Scan Data :      clk,rst:in std_logic
Scan Data :      val:out std_logic_vector 3 downto 0
Port Info : ['  clk,rst:in:1', '      val:out:4']
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
clk,rst in 1

val out 4

Inout List : [['clk,rst', 'in', '1']]
Output list [['val', 'out', '4']]
Output Port List : ['val:4']
Input Port List : ['clk,rst:1']
Starting With cfunc.mod file
Building content for cfunc.mod file
Starting with ifspec.ifs file
Gathering Al the content for ifspec file
Starting with testbench file
deccounter_tb.vhdl
Creating DUT directory at C:\FOSSEE\ngspice-nghdl\src\xspice\icm\ghdl\deccounter\DUTghdl
Copying required file to DUTghdl directory
run Make Called
Running Make command in C:\FOSSEE\ngspice-nghdl\release\src\xspice\icm
make command process pid ----- > <sip.voidptr object at 0x0000022613642C60>
Traceback (most recent call last):
  File "ngspice_ghdl.py", line 311, in createSchematicLib
    File "shutil.py", line 245, in copy
    File "shutil.py", line 120, in copyfile
FileNotFoundError: [Errno 2] No such file or directory: 'ghdl/ghdl.cm'
```

Figure 5.1: Error message window in NGHDL

5.2 Solution

Often, the users don't go through the Appendix Section of User Manual and declare ports on the same line. As a result, this error occurs while simulating using NGHDL due to lack of awareness.

This error should be eliminated completely so as to ensure error-free simulation using NGHDL and to increase the efficiency of eSim. In order to do so, the Python file `model_generation.py` present in the folder `FOSSEE\nghdl\src` was modified to accept two ports on the same line. Now users are able to declare ports on the same line without facing any errors.

The snippet for the code is:

```
if item.find(','):  
    temp1 = item.split(",")  
    item = " " + temp1[-1]  
    temp2 = temp1[-1].split(":")  
    for i in range(len(temp1)-1):  
        temp3 = temp1[i] + ":" + temp2[-1]  
    scan_data.append(temp3.rstrip())
```

Chapter 6

Building eSim Installer for MS Windows

6.1 Creating eSim Executable File

1. Add alias to python.exe in Git shell (Need to be done for each new shell created) using the command
`$ alias python='winpty python.exe'`
2. Download `virtualenv` package using the command
`$ pip install virtualenv`
3. Create a virtual environment using the command
`$ virtualenv eSim`
4. Activate the virtual environment using the command
`$ source eSim/Scripts/activate`
5. Install the Python dependencies through pip for eSim within the virtual environment using the following commands
`$ pip install pyinstaller`
`$ pip install matplotlib==3.0.3`
`$ pip install tornado`
`$ pip install setuptools`
`$ pip install PyQt5==5.9.2`
`$ pip install pypiwin32`
6. Test whether only eSim dependencies are available or not using the command
`$ pip freeze`
Note: Following Python packages for eSim along with their dependencies should be available.
 - PyQt5-sip
 - matplotlib
 - PyQt5

- numpy
 - dateutil
 - pyparsing
 - six
 - cyclcr.py
 - pypiwin32
7. Create eSim **spec** file as:

```
$ pyi-makespec --onefile -n eSim <path_to_eSim>/src/frontEnd/Application.py
--icon=<path_to_eSim_logo>.
```
 8. Create eSim onefile executable using **pyinstaller** using the command

```
$ pyinstaller -F --clean eSim.spec
```
 9. Verify whether all eSim src files (*.py) have been included in **Analysis-00.toc** file under the *build* folder generated by PyInstaller.

6.2 Creating NGHDL Executable File

1. Add alias to python.exe in Git shell (Need to be done for each new shell created) using the command

```
$ alias python='winpty python.exe'
```
2. Download **virtualenv** package using the command

```
$ pip install virtualenv
```
3. Create a virtual environment using the command

```
$ python -m virtualenv nghdl
```
4. Activate the virtual environment using the command

```
$ source nghdl/Scripts/activate
```
5. Install the Python dependencies through pip for NGHDL within the virtual environment using the following commands

```
$ pip install pyinstaller
$ pip install setuptools
$ pip install PyQt5==5.9.2
```
6. Test whether only NGHDL dependencies are available or not using the command

```
$ pip freeze
```

Note: Following Python packages for eSim along with their dependencies should be available.

- PyQt5-sip

- pyinstaller
 - PyQt5
7. Create NGHDL **spec** file as:
\$ `pyi-makespec --onefile -n nghdl <path_to_nghdl>/src/ngspice_ghdl.py`
 8. Create NGHDL onefile executable using **pyinstaller** using the command
\$ `pyinstaller -F --clean nghdl.spec`
 9. Verify whether all NGHDL src files (*.py) have been included in **Analysis-00.toc** file under the *build* folder generated by PyInstaller.

6.3 Packaging of NGHDL

1. Take the **master** branch of NGHDL. Rename the folder to **nghdl**.
2. Place the NGHDL executable (nghdl.exe) at the location nghdl/src/. Also, remove following files from the **nghdl** folder:
 - .git folder
 - python files from nghdl/src folder
 - .gitignore
 - GHDLside.md
 - Workflow.md
 - ngspice-nghdl.tar.xz
3. Place this **nghdl** folder containing **src** folder, **Example** folder, **LICENSE** file and **README.md** under folder named **eSim**.
4. Along with **nghdl-src.7z** from step 5, place the following 7z files at the installer folder for **eSim**:
 - ghdl.7z
 - MSYS.7z
 - mingw64.7z
 - ngspice-nghdl.7z

Note: These compressed files can be obtained under Windows/sources/ folder of NGHDL's installers branch. Ensure that the following DLL files are present in **mingw64.7z**.

- opengl32sw.dll
- vcruntime140_1.dll
- d3dcompiler_47.dll
- libEGL.dll

- libGLSLv2.dll

5. Place the script `nghdl-setup-script.nsi` at the eSim's installer folder.

6.4 Packaging of eSim

1. Take the `master` branch of eSim. Rename the folder to `eSim`.
2. Add eSim executable (`eSim.exe`) to the `eSim` folder. Also, remove following files from that folder:
 - `.git` folder
 - `code` folder
 - `src` folder
 - `conf.py`
 - `setup.py`
 - `index.rst`
 - `requirement.txt`
 - `.gitignore`
 - `.travis.yml`
 - `INSTALL`
 - `library/browser/User-Manual/eSim.html`
 - `library/browser/User-Manual/figures` folder
3. Remove entire `library` folder and duplicate files from `template` and `modules` folders from `eSim/library/kicadLibrary` as same will be repeated in KiCad installer. Now, compress it in zip format, place it at `eSim/library` and delete that folder.
4. Add eSim User-Manual `eSim_Manual_<version>.pdf` at location `library/browser/User-Manual`.
5. Replace `LICENSE` file with `LICENSE.rtf` file in `eSim` folder.
6. Compress `eSim` folder in 7z format and add it to the installer folder (Make sure that there is folder named `eSim` inside this compressed file).
7. Add the dependencies for NGHDL to the installer folder.
8. Apply NSIS plugins.
<https://github.com/fossee/nghdl/tree/installers/Windows/NSISplugins/>
9. Add KiCad installer `kicad-4.0.7-i686.exe` to the installer folder.
 All scripts and documentation of steps related to packaging of KiCad are present at the Windows folder under `installers` branch of the eSim repository:
<https://github.com/fossee/KiCad-eSim/blob/executables/README.md>
10. Add following files/folder to the installer folder:
 - License file (rtf format).
 - Logo (ico format).
 - `esim-setup-script` file.

- nghdl-setup-script file.
11. Compile the NSI script `esim-setup-script` file. Now only use the generated installer for distribution.

Chapter 7

Conclusion

Modifications were made in steps to create eSim Executable File and to Package eSim as mentioned in Section 3.2 and Section 4.2. Now user has to run a batch file (eSim.bat) instead of eSim.exe as discussed in Section 4.2. Few dependencies were added to MinGW64 to eliminate Windows OS errors and to ensure smooth running of operations as mentioned in Section 3.2.2. NGHDL now accepts two or more port declaration on the same line without giving any error as discussed in Section 5.2.

In the course of this fellowship the following tasks were achieved:

1. eSim installer was successfully fixed for MS Windows 8 OS and has been tested end-to-end for Windows 7 OS and Windows 10 OS as well.
2. A batch file was incorporated to eliminate path error which was observed in some systems.
3. The NGHDL Python file, `model_generation.py` was modified to accept two ports on the same line.

Chapter 8

Future Work

As of now, downgrading PyQt5 does not cause any problem and eSim is working properly after implementing this solution. However, with more updates in eSim there may be a possibility that the lower version of PyQt5 would not support eSim and create issues.

The path of HOME variable while installing eSim is now fixed to C:\Users\\ in the batch file now. To fix the path of HOME variable of the system permanently, the NSI script `esim-setup-script.nsi` needs to be modified. As of now, a batch file (eSim.bat) has been implemented. In future, the batch file can be eliminated and the path of HOME variable can be set directly using the NSI script `esim-setup-script.nsi`.

Bibliography

- [1] eSim Official website. 2021.
<https://esim.fossee.in/>

- [2] Ngspice Manual
<http://ngspice.sourceforge.net/docs/ngspice-31-manual.pdf>

- [3] PyQt5 Official website. 2021.
<https://pypi.org/project/PyQt5/>

- [4] Github Official Website
<https://github.com/FOSSEE/eSim/blob/installers/Windows/README.md>

- [5] GeekforGeeks Official Website. 2021.
<https://www.geeksforgeeks.org/writing-windows-batch-script/>

- [6] eSim User Manual
https://static.fossee.in/esim/manuals/eSim_Manual_2020_August.pdf

- [7] Nullsoft scriptable Install System Wiki. 2021.
https://nsis.sourceforge.io/Main_Page

- [8] Android Developers Official Website. 2021.
<https://developer.android.com/guide/topics/connectivity/bluetooth>