
eSim Documentation

Release 2.0.0

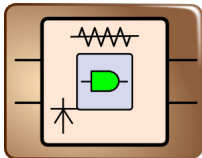
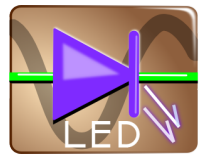
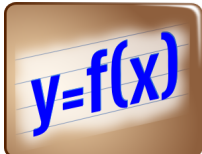
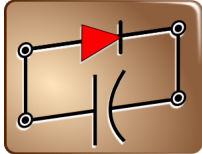
Fossee

Mar 11, 2020

Contents:

1	Configuration Component	3
1.1	Configuration Component	3
2	Browser Component	5
2.1	User Manual Component	5
2.2	Welcome Component	5
3	Front-End Component	7
3.1	Application Component	7
3.2	Dock Area Component	9
3.3	Project Explorer Component	10
3.4	Workspace Component	11
4	kicadtoNgspice Component	13
4.1	Analysis Component	13
4.2	Convert Component	15
4.3	Device Model Component	16
4.4	KicadtoNgspice component	16
4.5	Model Component	18
4.6	Processing Component	18
4.7	Source Component	19
4.8	Subcircuit Tab component	19
4.9	Track Widget Component	20
5	ngspiceSimulation component	21
5.1	Ngspice Widget Component	21
5.2	Python Plotting component	21
6	Project Management Component	23
6.1	Open Kicad Schematic Button	23
6.2	Validation Helper Function	24
6.3	Helper Worker Function	25
6.4	Open Project Button	25
6.5	New Project Function	26
7	Model Editor Component	27
7.1	Model Editor Component	27

8 Subcircuit component	31
8.1 Convert Subcircuit component	31
8.2 New subcircuit component	31
8.3 Open subcircuit component	32
8.4 Subcircuit component	32
8.5 Upload subcircuit component	32
9 Indices and tables	33
Python Module Index	35
Index	37



Configuration Component

1.1 Configuration Component

class configuration.Appconfig.**Appconfig**

All configuration goes here. May change in future for code optimization.

This class also contains function for - Printing error. - Showing warnings. - Displaying information.

2.1 User Manual Component

class browser.UserManual.UserManual

This class opens User-Manual page in new tab of web browser when help button is clicked.

2.2 Welcome Component

class browser.Welcome.Welcome

This class contains content of dock area part of initial esim Window. It creates Welcome page of eSim.

3.1 Application Component

class frontEnd.Application.**Application**(*args)

This class initializes all objects used in this file.

change_mode()

- **This function is used for changing mode of operation for KiCad. - There are three modes of operation :**

- online
- offline
- disable

- If none of the KiCad tools (associated with eSim) are open, then validate this mode by calling the function *validate_mode* and depending on *online_flag*, swap appropriate fp-lib-table files.
- If any of the KiCad tools (associated with eSim) is open, then ask user to close all these tools.
- If *online_flag* is *None*, then disable this feature.

@params

@return None

change_workspace()

This function call changes Workspace

closeEvent(event)

This function closes the ongoing program (process). When exit button is pressed a Message box pops out with exit message and buttons 'Yes', 'No'.

1. If 'Yes' is pressed:

- check that program (process) in *procThread_list* (a list made in *Appconfig.py*):
 - if available it terminates that program.

- if the program (process) is not available, then check it in process_obj (a list made in Appconfig.py) and if found, it closes the program.

2. If ‘No’ is pressed:

- the program just continues as it was doing earlier.

close_project ()

This function closes the saved project. It first checks whether project (file) is present in list.

• If present:

- it first kills that process-id.
- closes that file.
- Shows message “Current project <path_to_file> is closed”

- If not present: pass

help_project ()

This function opens usermanual in dockarea.

- It prints the message “”Function : Help””
- Uses print_info() method of class Appconfig from Configuration/Appconfig.py file.
- Call method usermanual() from ./DockArea.py.

initToolBar ()

This function initializes Tool Bars. It setups the icons, short-cuts and defining functionality for:

- Top-tool-bar (New project, Open project, Close project, Mode switch, Help option)
- Left-tool-bar (Open Schematic, Convert KiCad to NgSpice, Simulation, Model Editor, Subcircuit, NGHDL, Modelica Converter, OM Optimisation)

new_project ()

This function call New Project Info class.

open_OMedit ()

This function calls ngspice to OMedit converter and then launch OMedit.

open_OMoptim ()

This function uses validateTool() method from Validation.py:

- If ‘OMoptim’ is present in executables list then it passes command ‘OMoptim’ to WorkerThread class of Worker.py
- If ‘OMoptim’ is not present, then it shows error message with link to download it on Linux and Windows.

open_modelEditor ()

This function opens model editor option in left-tool-bar. When model editor icon is clicked which is present in left-tool-bar of main page:

- Message shown on screen “Model editor is called”.
- ‘modeleditor()’ function is called using object ‘obj_dockarea’ of class ‘Mainview’.

open_nghdl ()

This function calls NGHDL option in left-tool-bar. It uses validateTool() method from Validation.py:

- If ‘nghdl’ is present in executables list then it passes command ‘nghdl -e’ to WorkerThread class of Worker.py.

- If 'nghdl' is not present, then it shows error message.

open_ngspice()

This Function execute ngspice on current project.

open_project()

This project call Open Project Info class.

open_subcircuit()

This function opens 'subcircuit' option in left-tool-bar. When 'subcircuit' icon is clicked which is present in left-tool-bar of main page:

- Message shown on screen "Subcircuit editor is called".
- 'subcircuiteditor()' function is called using object 'obj_dockarea' of class 'Mainview'.

validate_mode()

This function checks whether proper fp-lib-table* files are available or not. If not, then move appropriate files from library/supportFiles folder and set *self.online_flag* accordingly.

@params

@return None

class frontEnd.Application.MainView(*args)

This class defines whole view and style of main page:

- **Position of tool bars:**
 - Top tool bar.
 - Left tool bar.
- Project explorer Area.
- Dock area.
- Console area.

frontEnd.Application.main(args)

The splash screen opened at the starting of screen is performed by this function.

3.2 Dock Area Component

class frontEnd.DockArea.DockArea

This class contains function for designing UI of all the editors in dock area part:

- Test Editor.
- Model Editor.
- Python Plotting.
- Ngspice Editor.
- Kicad to Ngspice Editor.
- Subcircuit Editor.
- Modelica editor.

closeDock()

This function checks for the project in **dock_dict** and closes it.

createTestEditor ()

This function create widget for Library Editor

kicadToNgspiceEditor (clarg1, clarg2=None)

This function is creating Editor UI for Kicad to Ngspice conversion.

modelEditor ()

This function defines UI for model editor.

modelicaEditor (projDir)

This function sets up the UI for ngspice to modelica conversion.

ngspiceEditor (projDir)

This function creates widget for Ngspice window.

plottingEditor ()

This function create widget for interactive PythonPlotting.

subcircuiteditor ()

This function creates a widget for different subcircuit options.

usermanual ()

This function creates a widget for user manual.

3.3 Project Explorer Component

class frontEnd.ProjectExplorer.ProjectExplorer

This class contains function:

- One work as a constructor(`__init__`).
- For saving data.
- for renaming project.
- for refreshing project.
- for removing project.
- for saving data.

enable_save ()

This function enables save button option.

refreshProject ()

This function refresh the project in explorer area by right clicking on project and selecting refresh option.

removeProject ()

This function removes the project in explorer area by right clicking on project and selecting remove option.

renameProject ()

This function renames the project present in project explorer area. It validates first:

- If project names is not empty.
- Project name does not contain spaces between them.
- Project name is different between what it was earlier.
- Project name should not exist.

After project name is changed, it recreates the project explorer tree.

save_data()

This function saves data before it closes the given file. It first opens file in write-mode, write operation is performed, closes that file and then it closes window.

3.4 Workspace Component

class frontEnd.Workspace.**Workspace** (*parent=None*)

This class creates UI for WorkSpace selection window.

- This window contains text area to select location of your choice or browse location for workspace area.
- By default workspace is set in ~/eSim-Workspace.
- This workspace area contains all the projects made by user.

4.1 Analysis Component

class kicadtoNgspice.Analysis.**Analysis** (*clarg1*)

- This class create Analysis Tab in KicadtoNgspice Window. 4 sections - - Select Analysis Type - AC Analysis - DC Analysis - Transient Analysis
- Set various track widget options here, for tracking purposes across different functions and modules - - AC_entry_var - AC_Parameter - DC_entry_var - DC_Parameter - TRAN_entry_var - TRAN_Parameter - set_Checkbox - AC_type - op_check

createACgroup ()

- Designing of AC group in analysis tab
- 3 radio buttons - Lin | Dec | Oct
- 3 input boxes, with top 2 combos
- If previous values exist then fill default values from previous value xml file

createAnalysisWidget ()

- Create the main analysis widget overview: - Checkbox for analysis type - Place, AC, DC and TRANSIENT analysis tab - *self.acbox*, *self.dcbox*, *self.trbox*...
- Check for *analysis* file, if any in projDir, extract data from it
- Else set the default checkbox to TRAN
- Accordingly set state for track widget options, as TRAN, AC ...

createCheckBox ()

- Create the checkboxes for analysis type, under analysis tab
- checkbox > checkgrid > checkgroupbtn > checkAC | checkDC | checkTRAN
- Trigger enableBox on clicking

createDCgroup ()

- Create DC area under analysis tab
- **Source 1 and 2, each having 4 input boxes as follows**
 - Source
 - Start
 - Increment
 - Stop
- The last 3 have combo box pertaining to their unit as well
- Also in the end a checkbox, for operating system point analysis

createTRANgroup ()

- Creating transient group under analysis and creating it's components
- Contains 3 inout and combo boxes for -
 - Start time
 - Step time
 - Stop time
- Input boxes for values, combo boxes for unit
- Accordingly also event handlers for combo boxes, creates 3 functions

enableBox ()

- Activate analysis areas according to checkBox marked
- Add analysis data to track_obj from TrackWidget

increment_changecombo (text)

Handle increment combo box, ie. units, as mV, V...

increment_changecombo2 (text)

Handle second increment combo box, ie. units, as mV, V...

set_ac_type ()

Sets track object for AC, according to the type of radio box selected.

setflag ()

- Handles the Operating point analysis checkbox

start_changecombo (text)

Handle start combo box, ie. units, as mV, V...

start_changecombo2 (text)

Handle second start combo box, ie. units, as mV, V...

start_combo_change (text)

Handle start combo box, ie. units, as second, ms

start_combovalue (text)

- Handle start_fre_combo box event
- Check where it is Hz, MHz, etc.
- Accordingly set ac_parameter

step_combo_change (*text*)
 Handle step combo box, ie. units, as second, ms...

stop_changecombo (*text*)
 Handle stop combo box, ie. units, as mV, V...

stop_changecombo2 (*text*)
 Handle second stop combo box, ie. units, as mV, V...

stop_combo_change (*text*)
 Handle stop combo box, ie. units, as second, ms...

stop_combovalue (*text*)

- Handle stop_fre_combo box event
- Check where it is Hz, MHz, etc.
- Accordingly set ac_parameter

4.2 Convert Component

class kicadtoNgspice.Convert.**Convert** (*sourcelisttrack*, *source_entry_var*, *schematicInfo*, *clarg1*)

- This class has all the necessary function required to convert kicad netlist to ngspice netlist.

- **Method List**

- addDeviceLibrary
- addModelParameter
- addSourceParameter
- addSubcircuit
- analysisInsertor
- converttosiform
- defaultvalue

addDeviceLibrary (*schematicInfo*, *kicadFile*)

This function add the library details to schematicInfo

addModelParameter (*schematicInfo*)

This function adds the Ngspice Model details to schematicInfo

addSourceParameter ()

- This function extracts the source details to schematicInfo
- **keywords recognised and parsed -**
 - sine
 - pulse
 - pwl
 - ac
 - dc
 - exp

- Return updated schematic

addSubcircuit (*schematicInfo*, *kicadFile*)

This function add the subcircuit to schematicInfo

analysisInsertor (*ac_entry_var*, *dc_entry_var*, *tran_entry_var*, *set_checkbox*, *ac_parameter*,
dc_parameter, *tran_parameter*, *ac_type*, *op_check*)

This function creates an analysis file in current project

converttosiform (*string_obj*)

This function is used for scientific conversion.

defaultvalue (*value*)

This function select default value as 0 if Analysis widget do not hold any value.

4.3 Device Model Component

class `kicadtoNgspice.DeviceModel.DeviceModel` (*schematicInfo*, *clarg1*)

- This class creates Device Library Tab in KicadtoNgspice Window It dynamically creates the widget for device like diode,mosfet, transistor and jfet.
- Same function as the subCircuit file, except for this takes different parameters in the if block
 - q TRANSISTOR
 - d DIODE
 - j JFET
 - m MOSFET
- Other 2 functions same as the ones in subCircuit
 - trackLibrary
 - trackLibraryWithoutButton

trackLibrary ()

This function is use to keep track of all Device Model widget

trackLibraryWithoutButton (*iter_value*, *path_value*)

This function is use to keep track of all Device Model widget

4.4 KicadtoNgspice component

class `kicadtoNgspice.KicadtoNgspice.MainWindow` (*clarg1*, *clarg2=None*)

- This class create KicadtoNgspice window.
- And Call Convert function if convert button is pressed.
- The convert function takes all the value entered by user and create a final netlist “*.cir.out”.
- This final netlist is compatible with Ngspice.
- clarg1 is the path to the .cir file
- clarg2 is either None or “sub” depending on the analysis type

callConvert ()

- This function called when convert button clicked
- Extracting data from the objs created above
- Pushing this data to xml, and writing it finally
- Written to a ..._Previous_Values.xml file in the projDirectory
- Finally, call createNetListFile, with the converted schematic

createMainWindow ()

- This function create main window of KiCad to Ngspice converter
- **Two components**
 - createcreateConvertWidget
 - Convert button => callConvert

createNetlistFile (store_schematicInfo, plotText)

- Creating .cir.out file
- **If analysis file present uses that and extract**
 - Simulator
 - Initial
 - Analysis
- **Finally add the following components to .cir.out file**
 - SimulatorOption
 - InitialCondOption
 - Store_SchematicInfo
 - AnalysisOption
- In the end add control statements and ally, alli, end statements

createSubFile (subPath)

- To create subcircuit file
- Extract data from .cir.out file

createcreateConvertWidget ()

- **Contains the tabs for various convertor elements**
 - Analysis => obj_analysis
 - => Analysis.Analysis(path_to_projFile)
 - Source Details => obj_source
 - => Source.Source(sourcelist, 'sourcelisttrack', 'path_to_projFile')
 - NgSpice Model => obj_model
 - => Model.Model(schematicInfo, 'modelList', 'path_to_projFile')
 - Device Modelling => obj_devicemodel
 - => DeviceModel.DeviceModel(schematicInfo, 'path_to_projFile')
 - Subcircuits => obj_subcircuitTab

```
=> SubcircuitTab.SubcircuitTab(schematicInfo, 'path_to_projFile')
```

- Finally pass each of these objects, to widgets
- `convertWindow > mainLayout > tabWidgets > AnalysisTab, SourceTab ...`

4.5 Model Component

```
class kicadtoNgspice.Model.Model (schematicInfo, modelList, clarg1)
```

- This class creates Model Tab of KicadtoNgspice window. The widgets are created dynamically in the Model Tab.

4.6 Processing Component

```
class kicadtoNgspice.Processing.ProcessNetlist
```

- This class include all the function required for pre-processing of netlist before converting to Ngspice Netlist.

```
convertICintoBasicBlocks (schematicInfo, outputOption, modelList, plotText)
```

- Parses the *schematicInfo* and returns
 - – *SchematicInfo*
 - – *Output Option*
 - – *Model List*
 - – *Unkown Model List*
 - – *Multiple Model List*
 - – *Plot text*
- Parsing info is provided below

```
insertSpecialSourceParam (schematicInfo, sourcelist)
```

- Insert Special source parameter
- As per the parameters passed create source list, start with v or i
- Then check for type whether ac, dc, sine, etc...
- Handle starting with h and f as well

```
preprocessNetlist (kicadNetlist, param)
```

- Preprocess netlist (replace parameters)
- Separate infoline (first line) from the rest of netlist

```
readNetlist (filename)
```

- Read the circuit file and return splitted lines

```
readParamInfo (kicadNetlist)
```

- Read Parameter information and store it into dictionary
- *kicadNetlist* is the .cir file content

separateNetlistInfo (*netlist*)

- Remove the options such as .end, .param, starting with “.” from the netlist file
- This is stored as option info, whereas rest is stored as schematicInfo
- Rest from the * *Sheet Name*: line stored as schematicInfo

4.7 Source Component

class kicadtoNgspice.Source.**Source** (*sourcelist, sourcelisttrack, clarg1*)

This class create Source Tab of KicadtoNgSpice Window.

createSourceWidget (*sourcelist, sourcelisttrack*)

- This function dynamically create source widget in the Source tab of KicadtoNgSpice window
- Depending on the type of source, sourcetab is created
 - – ac
 - – dc
 - – sine
 - – pulse
 - – pwl
 - – exp
- All the entry fields, are kept into the entry_var tracked by self.count
- Finally after each of the sourcelist is mapped to its input component we move to adding these to the track widget
- Also check if any default values present from previous analysis & add them by default
- Each line in sourcelist corresponds to a source
- According to the source type modify the source and add it to the tab

4.8 Subcircuit Tab component

class kicadtoNgspice.SubcircuitTab.**SubcircuitTab** (*schematicInfo, clarg1*)

- This class creates Subcircuit Tab in KicadtoNgspice Window
- It dynamically creates the widget for subcircuits, according to the .cir file
- Creates *lineEdit* and *Add* button, which triggers *fileSelector*
- Also, checks *Previous_value.xml* for previous subcircuit value to autofill, the *lineEdit*
- Add button is bind to *trackSubcircuit*
- Also *trackSubcircuit* without button is triggered if *lineEdit* filled

trackSubcircuit ()

- This function is use to keep track of all Subcircuit widget
- Here the number of ports is tracked using the numPorts and *Add* button objectName property, which is refered using *sender*

- Once a file is selected using the *QFileDialog* validate it
- Pass the path of subcircuit and the number of ports
- According to validation state take further steps
- If validated correctly, add to TrackWidget

trackSubcircuitWithoutButton (*iter_value*, *path_value*)

- Same as trackSubcircuit, but here the count value is passed directly without using any button as in *Add*
- This is triggered only once, initially

4.9 Track Widget Component

class kicadtoNgspice.TrackWidget.**TrackWidget**

- This Class track the dynamically created widget of KicadtoNgSpice Window.
- **Tracks using dictionary and lists ==>**
 - Sources
 - Parameters
 - References
 - Model Details
 - ... etc

ngspiceSimulation component

5.1 Ngspice Widget Component

```
class ngspiceSimulation.NgspiceWidget.NgspiceWidget (command, projPath)
```

5.2 Python Plotting component

```
class ngspiceSimulation.pythonPlotting.MultimeterWidgetClass (node_branch,  
                                                             rmsValue, loc_x,  
                                                             loc_y, voltFlag)
```

```
class ngspiceSimulation.pythonPlotting.plotWindow (fpath, projectName)  
    This class defines python plotting window, its features, buttons, colors, AC and DC analysis, plotting etc.
```

Project Management Component

6.1 Open Kicad Schematic Button

class projManagement.Kicad.Kicad(*dockarea*)

This class called the Kicad Schematic, KicadToNgspice Converter, Layout editor and Footprint Editor Initialises validation, appconfig and dockarea

@params

:dockarea => passed from DockArea in frontEnd folder, consists of all functions for dockarea

@return

check_open_schematic()

This function checks if any of the project's schematic is open or not

@params

@return True => If the project's schematic is not open False => If the project's schematic is open

openKicadToNgspice()

This function creates command to validate and then call KicadToNgSPice converter from DockArea file

@params

@return

openSchematic()

This function creates command to open Kicad schematic after appropriate validation checks

@params

@return

6.2 Validation Helper Function

class projManagement.Validation.Validation

This is Validation class use for validating Project. e.g if .proj is present in project directory or if new project name is already exist in workspace etc

validateCir (*projDir*)

Validate if cir file present in the directory with the appropriate .cir file name, same as the project directory base

@params :projDir => the path to the project directory

@return True False

validateCirOut (*projDir*)

This function checks if “.cir.out” file is present.

validateKicad (*projDir*)

Validate if projDir is set appropriately in the function calling file and if Kicad components are present

@params

:projDir => the path of the project directory, passed from the calling function

@return True False

validateNewproj (*projDir*)

Validate new project created

@params :projDir => Contains path of the new projDir created

@return :”CHECKEXIST” => If same project name folder exists :”CHECKNAME” => If space is there in name :”VALID” => If valid project name given

validateOpenproj (*projDir*)

Takes as input the path of the project and checks if projName.proj file exists or not

@params :projDir => contains the path of the project selected to open

@return True => If the folder contains the projName.proj file False => If the folder doesn’t contain projName.proj file

validateSub (*subDir, givenNum*)

This function checks if “.sub” file is present. Also, if subckt file is present check for ports and check if equal

@params :subDir => the path of the subcircuit directory :giveNum => the number of port calculated and passed for validation

@return True PORT DIREC

validateSubcir (*projDir, fileName*)

This function checks for valid format of .sub file.

Correct format of file is:

- File should start with **.subckt <filename>**
- End with **.ends <filename>**

Function is passed with the file of path it checks the file line by line until it get .subckt as its first word and then check for second word is it <fileName> or not.

Then it checks for second last line if it is “.ends <filename>” it return True if conditions satisfy else return False.

validateTool (*toolName*)

This function check if tool is present in the system.

6.3 Helper Worker Function

class projManagement.Worker.**WorkerThread** (*args*)

Initialise a QThread with the passed arguments WorkerThread uses QThread to support threading operations for other PyQt windows This is a helper functions, used to create threads for various commands

@params

:args => takes a space separated string of comamnds to be execute in different child processes (see subprocess.Popen())

@return None

call_system (*command*)

call_system is used to create childprocess for the passed arguments (self.args) and also pass the process created and its id to config file Apponfig() object contains procThread and proc_dist used to track processes called

@params

:command => (self.args) takes space separated string of comamnds to be executed in different child processes (see subprocess.Popen())

get_proc_threads ()

This function is a getter for the list of project’s workers, and is called to check if project’s schematic is open or not.

@params

@return :self.my_workers

run ()

run is the function that is called, when we start the thread as thisThread.start() Here, it makes system calls for all args passed (self.args)

@params

@return None

6.4 Open Project Button

class projManagement.openProject.**OpenProjectInfo**

This class is called when User click on Open Project Button

body ()

Open a project directory using Qt GUI and validate if .proj file present in it using *Validation* class

@params

@return :dirs => The directories inside the project folder :filelist => The files inside the project folder

6.5 New Project Function

class projManagement.newProject.NewProjectInfo

This class is called when User create new Project.

createProject (*projName*)

This function create Project related directories and files. Before creating also validates using the *Validation* class

Validation codes

- VALID
- CHECKEXIST
- CHECKNAME
- NONE

@params

:projName => name of the project created passed from frontEnd/Application
new_project()

@return :dirs => The directories inside the project folder :filelist => The files inside the project folder

@params

:projName => name of the project created passed from frontEnd/Application
new_project()

@return :dirs => The directories inside the project folder :filelist => The files inside the project folder

7.1 Model Editor Component

class modelEditor.ModelEditor.**ModelEditorclass**

- Initialise the layout for dockarea
- Use QVBoxLayout, QSplitter, QGridLayout to define the layout
- Initialise directory to save new models, savepathtest = 'library/deviceModelLibrary'
- **Initialise buttons and options =====>**
 - Name Function Called
- New opennew
- Edit openedit
- Save savemodelfile
- Upload converttoxml
- Add addparameters
- Remove removeparameter
- Diode diode_click
- BJT bjt_click
- MOS mos_click
- JFET jfet_click
- IGBT igbt_click
- Magnetic Core magnetic_click

addparameters ()

- Called when *Add* button clicked beside QTableWidgetItem
- Open up dialog box to enter parameter and value accordingly
- Validate if parameter already in list of parameters
- Accordingly add parameter and value in modeldict as well as table
- text1 => parameter, text2 => value

bjt_click()

- Set states for other elements
- Initialise types combo box elements
 - – NPN
 - – PNP
- Open the default type in the table
- Add an event listener for type-selection event

converttoxml()

- Called when upload button clicked
- Used to read file from a certain location for .lib extension
- Accordingly parse it and extract modelname and modelref
- Also extract param value pairs
- Take input the name of the library you want to save it as
- Save it in *User Libraries* with the given name,

and input from uploaded file

createXML(model_name)

- Create .xml and .lib file if new model is being created
- Save it in the corresponding component directory, example Diode, IGBT..
- For each component, separate folder is there
- Check the contents of .lib and .xml file to understand their structure

createtable(modelfile)

- Set states for other components
- Initialise QTable widget
- Set options for QTable widget
- Place QTable widget, using *self.grid.addWidget*
- Select the .xml file from the modelfile passed as .lib
- Use ET (xml.etree.ElementTree) to parse the xml file
- Extract data from the XML and store it in *modeldict*
- Show the extracted data in QTableWidgetItem
- Can edit QTable inplace, connect *edit_modeltable* function for editing

diode_click()

- Call function, `openfiletype`, which opens the table view for Diode specs
- Set states for other elements
- Diode has no types, so hide that

`edit_modeltable()`

- Called when editing model inplace in `QTableWidget`
- Set states of other components
- Get data from the `modeltable` of the selected row
- Edit name and value as per needed
- Add the `val` name pair in the `modeldict`

`igbt_click()`

- Set states for other elements
- Initialise types combo box elements
 - – N-IGBT
 - – P-IGBT
- Open the default type in the table
- Add an event listener for type-selection event

`jfet_click()`

- Set states for other elements
- Initialise types combo box elements
 - – N-JFET
 - – P-JFET
- Open the default type in the table
- Add an event listener for type-selection event

`magnetic_click()`

- Set states for other elements
- Initialise types combo box elements
- Open the default type in the table
- Add an event listener for type-selection event
- No types here, only one view

`mos_click()`

- Set states for other elements
- Initialise types combo box elements
 - – NMOS(Level-1 5um)
 - – NMOS(Level-3 0.5um)
 - – ...
- Open the default type in the table

- Add an event listener for type-selection event

openedit ()

- When *Edit* button clicked, this function called
- Set states for other buttons accordingly
- Open the file selector box with path as deviceModelLibrary

and filetype set as .lib, save it in *self.editfile* - Create table for the selected .lib file using *self.createtable(path)* - Handle exception of no file selected

openfiletype (filetype)

- Select path for the filetype passed
- Accordingly call *createtable(path)* to draw tables usingg QTable
- Check for the state of button before rendering

opennew ()

- To create New Model file
- Change state of other buttons accordingly, ex. enable diode, bjt, ...
- Validate filename created, to check if one already exists

removeparameter ()

- Get the index of the current selected item
- Remove the whole row from QTable Widget
- Remove the param,value pair from modeldict

savemodelfile ()

- Called when save functon clicked
- If new file created, call *createXML* file
- Else call *savethefile*

savethefile (editfile)

- This function save the editing in the model table
- Create .lib and .xml file for the editfile path and replace them
- Also print Updated Library with libpath in the command window

setfiletype (text)

- Triggered when each type selected
- Get the type clicked, from text
- Open appropriate table using openfiletype(filetype)

validation (text)

- This function checks if the file (xml type) with the name already exists
- Accordingly show error message

Subcircuit component

8.1 Convert Subcircuit component

class subcircuit.convertSub.convertSub(*dockarea*)

Contains functions that checks project present for conversion and also function to convert Kicad Netlist to Ngspice Netlist.

createSub()

This function create command to call KiCad to Ngspice converter. If the netlist is not generated for selected project it will show error **The subcircuit does not contain any Kicad netlist file for conversion.** And if no project is selected for conversion, it again show error message to select a file or create a file.

8.2 New subcircuit component

class subcircuit.newSub.NewSub

Contains functions to check : - Name of project should not be blank. - Name should not contain space between them. - Name does not match with existing project.

createSubcircuit(*subName*)

- This function create workspace for subcircuit.
- **It also validate file names for Subcircuits:**
 - File name should not contain space.
 - Name can not be empty.
 - File name already exists.

8.3 Open subcircuit component

class `subcircuit.openSub.openSub`

It opens the existing subcircuit projects that are present in Subcircuit directory.

8.4 Subcircuit component

class `subcircuit.Subcircuit.Subcircuit` (*parent=None*)

Creates buttons for New project, Edit existing project and Kicad Netlist to Ngspice Netlist converter and link them with the methods defined for it in other files.

- New Project(NewSub method of newSub).
- Open Project(openSub method of openSub).
- Kicad to Ngspice convertor(convertSub of convertSub).

8.5 Upload subcircuit component

class `subcircuit.uploadSub.UploadSub`

This class contain function for uploading subcircuits in SubcircuitLibrary present in src folder. A folder is created in library/SubcircuitLibrary and desired file is moved to that folder.

upload()

This method opens a dialogue box when Upload subcircuit button is clicked and after entering folder name, it opens directory system to chose file for folder, it only shows file with extension .sub and with the name of project entered earlier as folder name.

It then validates file if it is in proper format or not, for it the file is passed to the function **validateSub** and it returns true if file has valid format or else it shows an error message.

CHAPTER 9

Indices and tables

- `genindex`
- `modindex`
- `search`

b

`browser.UserManual`, 5
`browser.Welcome`, 5

c

`configuration.Appconfig`, 3

f

`frontEnd.Application`, 7
`frontEnd.DockArea`, 9
`frontEnd.ProjectExplorer`, 10
`frontEnd.Workspace`, 11

k

`kicadtoNgspice.Analysis`, 13
`kicadtoNgspice.Convert`, 15
`kicadtoNgspice.DeviceModel`, 16
`kicadtoNgspice.KicadtoNgspice`, 16
`kicadtoNgspice.Model`, 18
`kicadtoNgspice.Processing`, 18
`kicadtoNgspice.Source`, 19
`kicadtoNgspice.SubcircuitTab`, 19
`kicadtoNgspice.TrackWidget`, 20

m

`modelEditor.ModelEditor`, 27

n

`ngspiceSimulation.NgspiceWidget`, 21
`ngspiceSimulation.pythonPlotting`, 21

p

`projManagement.Kicad`, 23
`projManagement.newProject`, 26
`projManagement.openProject`, 25
`projManagement.Validation`, 24
`projManagement.Worker`, 25

s

`subcircuit.convertSub`, 31

`subcircuit.newSub`, 31
`subcircuit.openSub`, 32
`subcircuit.Subcircuit`, 32
`subcircuit.uploadSub`, 32

A

addDeviceLibrary() (kicad-
toNgspice.Convert.Convert method), 15
 addModelParameter() (kicad-
toNgspice.Convert.Convert method), 15
 addparameters() (modelEditor.*ModelEditorclass method*),
 27
 addSourceParameter() (kicad-
toNgspice.Convert.Convert method), 15
 addSubcircuit() (kicadtoNgspice.*Convert.Convert method*), 16
 Analysis (class in kicadtoNgspice.*Analysis*), 13
 analysisInsertor() (kicad-
toNgspice.Convert.Convert method), 16
 AppConfig (class in configuration.*Appconfig*), 3
 Application (class in frontEnd.*Application*), 7

B

bjt_click() (modelEditor.*ModelEditorclass method*),
 28
 body() (projManagement.*openProject.OpenProjectInfo method*), 25
 browser.UserManual (module), 5
 browser.Welcome (module), 5

C

call_system() (projManagement.*Worker.WorkerThread method*), 25
 callConvert() (kicad-
toNgspice.KicadtoNgspice.MainWindow method), 16
 change_mode() (frontEnd.*Application.Application method*), 7
 change_workspace() (frontEnd.*Application.Application method*), 7
 check_open_schematic() (projManagement.*Kicad.Kicad method*), 23

close_project() (frontEnd.*Application.Application method*), 8
 closeDock() (frontEnd.*DockArea.DockArea method*),
 9
 closeEvent() (frontEnd.*Application.Application method*), 7
 configuration.Appconfig (module), 3
 Convert (class in kicadtoNgspice.*Convert*), 15
 convertICintoBasicBlocks() (kicad-
toNgspice.Processing.ProcprocessNetlist method),
 18
 convertSub (class in subcircuit.*convertSub*), 31
 converttosiform() (kicad-
toNgspice.Convert.Convert method), 16
 converttoxml() (modelEditor.*ModelEditorclass method*),
 28
 createACgroup() (kicadtoNgspice.*Analysis.Analysis method*), 13
 createAnalysisWidget() (kicad-
toNgspice.Analysis.Analysis method), 13
 createCheckBox() (kicad-
toNgspice.Analysis.Analysis method), 13
 createcreateConvertWidget() (kicad-
toNgspice.KicadtoNgspice.MainWindow method), 17
 createDCgroup() (kicadtoNgspice.*Analysis.Analysis method*), 13
 createMainWindow() (kicad-
toNgspice.KicadtoNgspice.MainWindow method), 17
 createNetlistFile() (kicad-
toNgspice.KicadtoNgspice.MainWindow method), 17
 createProject() (projManagement.*newProject.NewProjectInfo method*),
 26
 createSourceWidget() (kicad-
toNgspice.Source.Source method), 19
 createSub() (subcircuit.*convertSub.convertSub*)

method), 31
 createSubcircuit() (subcircuit.newSub.NewSub
 method), 31
 createSubFile() (kicad-
 toNgspice.KicadtoNgspice.MainWindow
 method), 17
 createtable() (modelE-
 tor.ModelEditor.ModelEditorclass
 method), 28
 createTestEditor() (fron-
 tEnd.DockArea.DockArea method), 9
 createTRNGroup() (kicad-
 toNgspice.Analysis.Analysis method), 14
 createXML() (modelE-
 tor.ModelEditor.ModelEditorclass
 method), 28

D

defaultvalue() (kicadtoNgspice.Convert.Convert
 method), 16
 DeviceModel (class in kicadtoNgspice.DeviceModel),
 16
 diode_click() (modelE-
 tor.ModelEditor.ModelEditorclass
 method), 28
 DockArea (class in frontEnd.DockArea), 9

E

edit_modeltable() (modelE-
 tor.ModelEditor.ModelEditorclass
 method), 29
 enable_save() (fron-
 tEnd.ProjectExplorer.ProjectExplorer method),
 10
 enableBox() (kicadtoNgspice.Analysis.Analysis
 method), 14

F

frontEnd.Application (module), 7
 frontEnd.DockArea (module), 9
 frontEnd.ProjectExplorer (module), 10
 frontEnd.Workspace (module), 11

G

get_proc_threads() (projManage-
 ment.Worker.WorkerThread method), 25

H

help_project() (frontEnd.Application.Application
 method), 8

I

igbt_click() (modelE-
 tor.ModelEditor.ModelEditorclass
 method),

29

increment_changecombo() (kicad-
 toNgspice.Analysis.Analysis method), 14
 increment_changecombo2() (kicad-
 toNgspice.Analysis.Analysis method), 14
 initToolBar() (frontEnd.Application.Application
 method), 8
 insertSpecialSourceParam() (kicad-
 toNgspice.Processing.ProcprocessNetlist method),
 18

J

jfet_click() (modelE-
 tor.ModelEditor.ModelEditorclass
 method), 29

K

Kicad (class in projManagement.Kicad), 23
 kicadtoNgspice.Analysis (module), 13
 kicadtoNgspice.Convert (module), 15
 kicadtoNgspice.DeviceModel (module), 16
 kicadtoNgspice.KicadtoNgspice (module), 16
 kicadtoNgspice.Model (module), 18
 kicadtoNgspice.Processing (module), 18
 kicadtoNgspice.Source (module), 19
 kicadtoNgspice.SubcircuitTab (module), 19
 kicadtoNgspice.TrackWidget (module), 20
 kicadToNgspiceEditor() (fron-
 tEnd.DockArea.DockArea method), 10

M

magnetic_click() (modelE-
 tor.ModelEditor.ModelEditorclass
 method), 29
 main() (in module frontEnd.Application), 9
 MainView (class in frontEnd.Application), 9
 MainWindow (class in kicad-
 toNgspice.KicadtoNgspice), 16
 Model (class in kicadtoNgspice.Model), 18
 modelEditor() (frontEnd.DockArea.DockArea
 method), 10
 modelEditor.ModelEditor (module), 27
 ModelEditorclass (class in modelE-
 tor.ModelEditor), 27
 modelicaEditor() (frontEnd.DockArea.DockArea
 method), 10
 mos_click() (modelE-
 tor.ModelEditor.ModelEditorclass
 method), 29
 MultimeterWidgetClass (class in ngspiceSimula-
 tion.pythonPlotting), 21

N

new_project() (frontEnd.Application.Application

method), 8
 NewProjectInfo (class in projManagement.newProject), 26
 NewSub (class in subcircuit.newSub), 31
 ngspiceEditor() (frontEnd.DockArea.DockArea method), 10
 ngspiceSimulation.NgspiceWidget (module), 21
 ngspiceSimulation.pythonPlotting (module), 21
 NgspiceWidget (class in ngspiceSimulation.NgspiceWidget), 21

O

open_modelEditor() (frontEnd.Application.Application method), 8
 open_nghdl() (frontEnd.Application.Application method), 8
 open_ngspice() (frontEnd.Application.Application method), 9
 open_OMedit() (frontEnd.Application.Application method), 8
 open_OMoptim() (frontEnd.Application.Application method), 8
 open_project() (frontEnd.Application.Application method), 9
 open_subcircuit() (frontEnd.Application.Application method), 9
 openedit() (modelEditor.ModelEditor.ModelEditorclass method), 30
 openfiletype() (modelEditor.ModelEditor.ModelEditorclass method), 30
 openKicadToNgspice() (projManagement.Kicad.Kicad method), 23
 opennew() (modelEditor.ModelEditor.ModelEditorclass method), 30
 OpenProjectInfo (class in projManagement.openProject), 25
 openSchematic() (projManagement.Kicad.Kicad method), 23
 openSub (class in subcircuit.openSub), 32

P

plottingEditor() (frontEnd.DockArea.DockArea method), 10
 plotWindow (class in ngspiceSimulation.pythonPlotting), 21
 PrcoessNetlist (class in kicadtoNgspice.Processing), 18
 preprocessNetlist() (kicadtoNgspice.Processing.PrcoessNetlist method),

18
 ProjectExplorer (class in frontEnd.ProjectExplorer), 10
 projManagement.Kicad (module), 23
 projManagement.newProject (module), 26
 projManagement.openProject (module), 25
 projManagement.Validation (module), 24
 projManagement.Worker (module), 25

R

readNetlist() (kicadtoNgspice.Processing.PrcoessNetlist method), 18
 readParamInfo() (kicadtoNgspice.Processing.PrcoessNetlist method), 18
 refreshProject() (frontEnd.ProjectExplorer.ProjectExplorer method), 10
 removeparameter() (modelEditor.ModelEditor.ModelEditorclass method), 30
 removeProject() (frontEnd.ProjectExplorer.ProjectExplorer method), 10
 renameProject() (frontEnd.ProjectExplorer.ProjectExplorer method), 10
 run() (projManagement.Worker.WorkerThread method), 25

S

save_data() (frontEnd.ProjectExplorer.ProjectExplorer method), 10
 savemodelfile() (modelEditor.ModelEditor.ModelEditorclass method), 30
 savethefile() (modelEditor.ModelEditor.ModelEditorclass method), 30
 separateNetlistInfo() (kicadtoNgspice.Processing.PrcoessNetlist method), 18
 set_ac_type() (kicadtoNgspice.Analysis.Analysis method), 14
 setfiletype() (modelEditor.ModelEditor.ModelEditorclass method), 30
 setflag() (kicadtoNgspice.Analysis.Analysis method), 14
 Source (class in kicadtoNgspice.Source), 19
 start_changecombo() (kicadtoNgspice.Analysis.Analysis method), 14

`start_changecombo2()` (*kicad-toNgspice.Analysis.Analysis method*), 14
`start_combo_change()` (*kicad-toNgspice.Analysis.Analysis method*), 14
`start_combovalue()` (*kicad-toNgspice.Analysis.Analysis method*), 14
`step_combo_change()` (*kicad-toNgspice.Analysis.Analysis method*), 14
`stop_changecombo()` (*kicad-toNgspice.Analysis.Analysis method*), 15
`stop_changecombo2()` (*kicad-toNgspice.Analysis.Analysis method*), 15
`stop_combo_change()` (*kicad-toNgspice.Analysis.Analysis method*), 15
`stop_combovalue()` (*kicad-toNgspice.Analysis.Analysis method*), 15
`Subcircuit` (class in *subcircuit.Subcircuit*), 32
`subcircuit.convertSub` (module), 31
`subcircuit.newSub` (module), 31
`subcircuit.openSub` (module), 32
`subcircuit.Subcircuit` (module), 32
`subcircuit.uploadSub` (module), 32
`subcircuiteditor()` (*frontEnd.DockArea.DockArea method*), 10
`SubcircuitTab` (class in *kicad-toNgspice.SubcircuitTab*), 19

T

`trackLibrary()` (*kicad-toNgspice.DeviceModel.DeviceModel method*), 16
`trackLibraryWithoutButton()` (*kicad-toNgspice.DeviceModel.DeviceModel method*), 16
`trackSubcircuit()` (*kicad-toNgspice.SubcircuitTab.SubcircuitTab method*), 19
`trackSubcircuitWithoutButton()` (*kicad-toNgspice.SubcircuitTab.SubcircuitTab method*), 20
`TrackWidget` (class in *kicadtoNgspice.TrackWidget*), 20

U

`upload()` (*subcircuit.uploadSub.UploadSub method*), 32
`UploadSub` (class in *subcircuit.uploadSub*), 32
`UserManual` (class in *browser.UserManual*), 5
`usermanual()` (*frontEnd.DockArea.DockArea method*), 10

V

`validate_mode()` (*frontEnd.Application.Application method*), 9

`validateCir()` (*projManagement.Validation.Validation method*), 24
`validateCirOut()` (*projManagement.Validation.Validation method*), 24
`validateKicad()` (*projManagement.Validation.Validation method*), 24
`validateNewproj()` (*projManagement.Validation.Validation method*), 24
`validateOpenproj()` (*projManagement.Validation.Validation method*), 24
`validateSub()` (*projManagement.Validation.Validation method*), 24
`validateSubcir()` (*projManagement.Validation.Validation method*), 24
`validateTool()` (*projManagement.Validation.Validation method*), 25
`Validation` (class in *projManagement.Validation*), 24
`validation()` (*modelEditor.ModelEditor.ModelEditorclass method*), 30

W

`Welcome` (class in *browser.Welcome*), 5
`WorkerThread` (class in *projManagement.Worker*), 25
`Workspace` (class in *frontEnd.Workspace*), 11