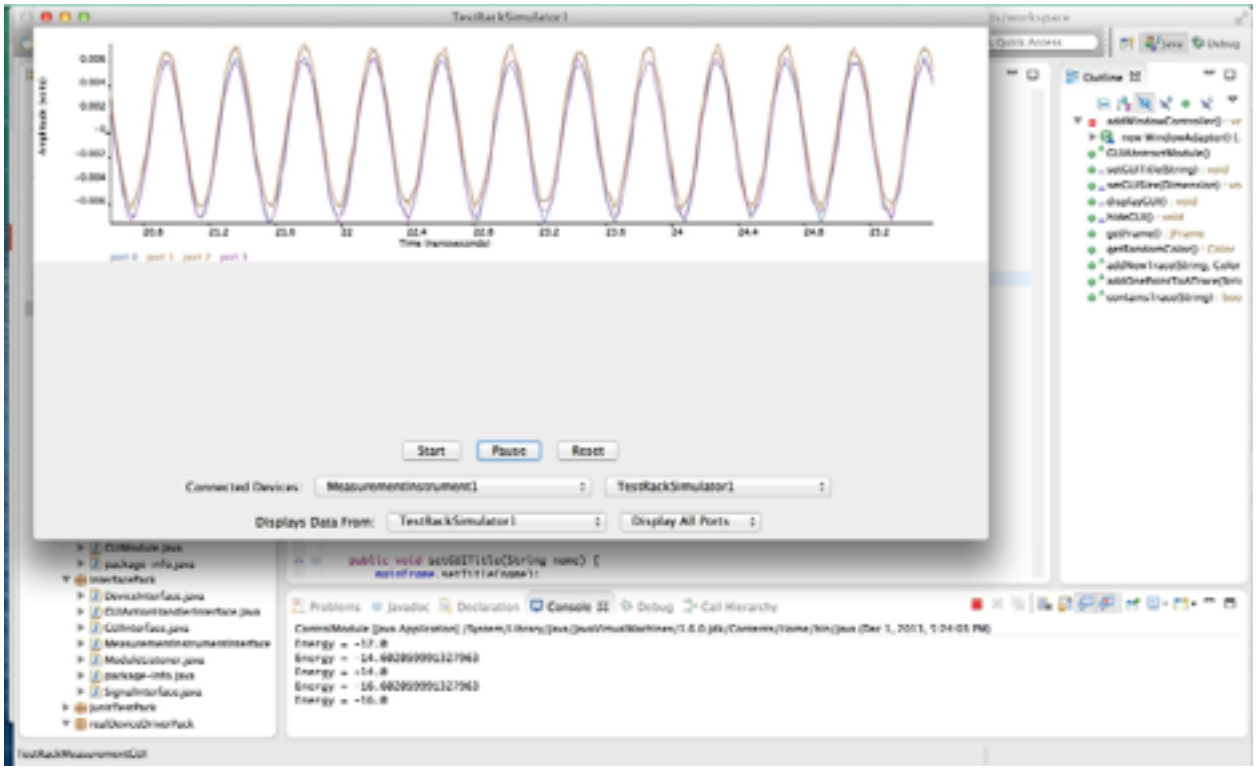
TestRackMeasurementGUI

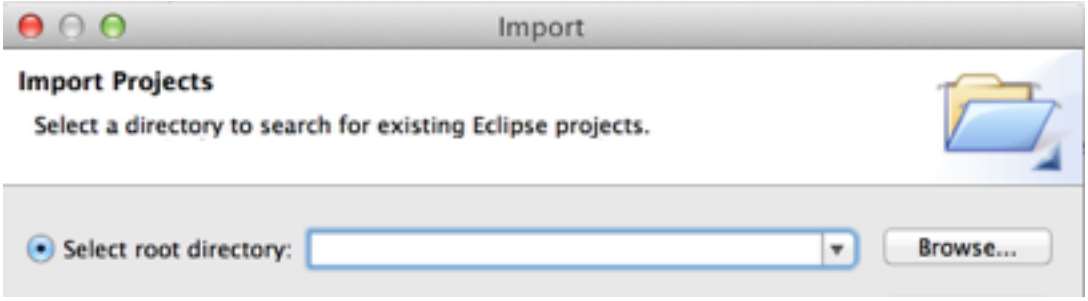


(a screenshot of the GUI)

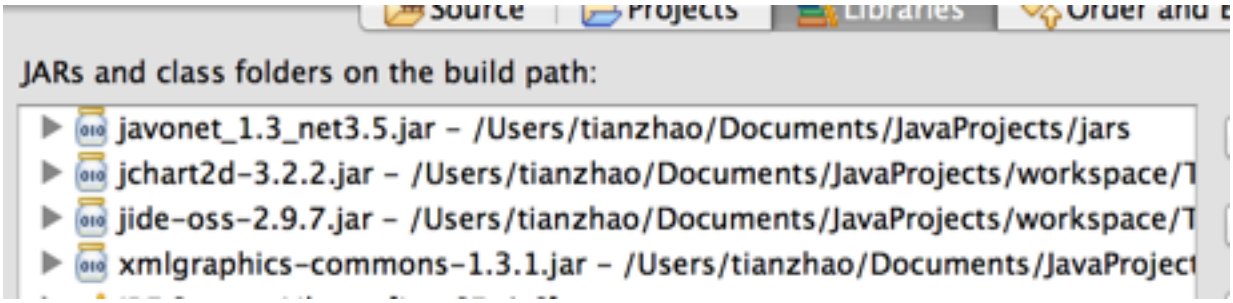
A. Set Up the Project in eclipse:

(Import project into eclipse)

1. In eclipse, go to File > Import. In the wizard, choose General > Existing Project Into Workspace.



2. In the Import wizard:
3. Choose the directory to the project folder (“TestRackMeasurementGUI”) and click on “Finish”

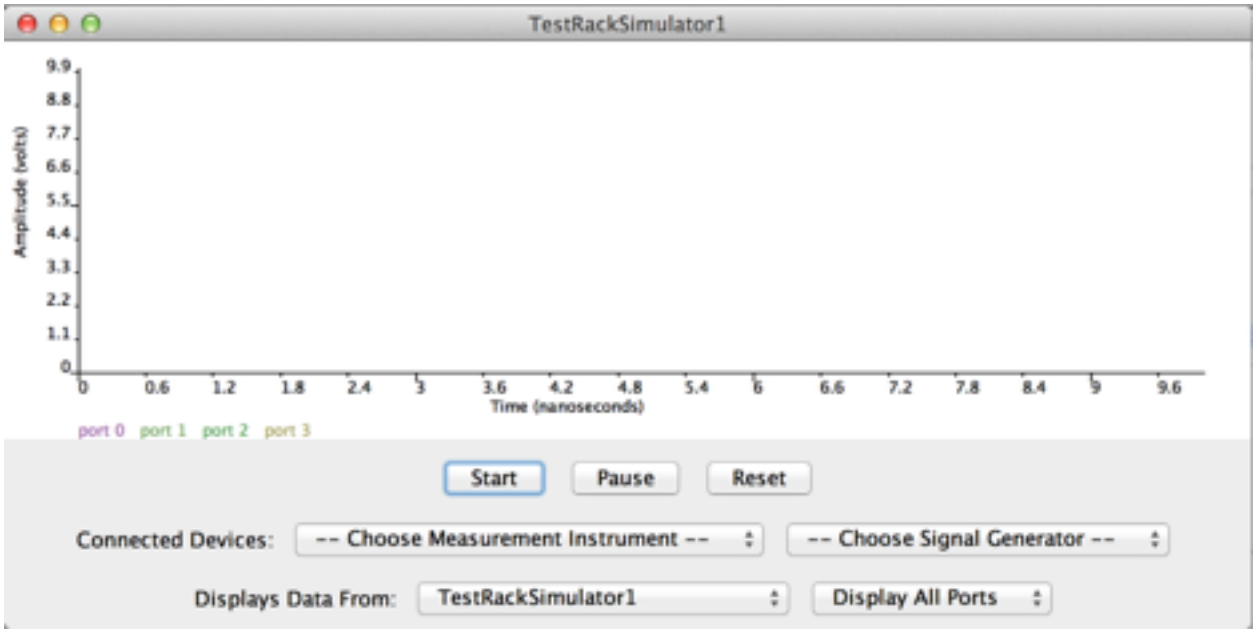


(list of jars)

(Add jar libraries for the GUI and for running the power meter)

4. Right-click on the project in the Workspace, choose Properties > Java Build Path. In the wizard, click on “Add External JARs…” and choose the four jar libraries that I attached to the email. Click on “OK”.
5. Place mcl_pm64.dll under the working directory so that the software can find it and load it into the registry. If the user is using a 32-bit machine, please go to http://www.minicircuits.com/support/software_download.html to download a 32-bit version of the dll. This .dll file is used by the software to talk to the Mini-circuits power meter.
6. Go to RealSignalModule1, RealSignalModule2.java. Change defaultSignalGeneratorName = “xxx.xxx.xxx.xxx" to the IP address of the signal generator that we want to connect. These two files are used by the software to talk to a signal generator using SCPI commands through LAN.

B. GUI Usage



BUTTONS:

- Start Button: Send commands to all the signal generators (can be either real or a simulated one) to start generate data.
- Pause: Pause all the signal generators
- Reset: Reset all the signal generators

COMBO-BOX: “CONNECTED DEVICES”

- User should select from “—Choose Measurement Instrument —“ and “—Choose Signal Generator —“ to choose a pair of measurement instrument and signal generator. The

software will take output data from the chosen signal generator and sends the data to the selected measurement instrument.

COMBO-BOX: "DISPLAY DATA FROM"

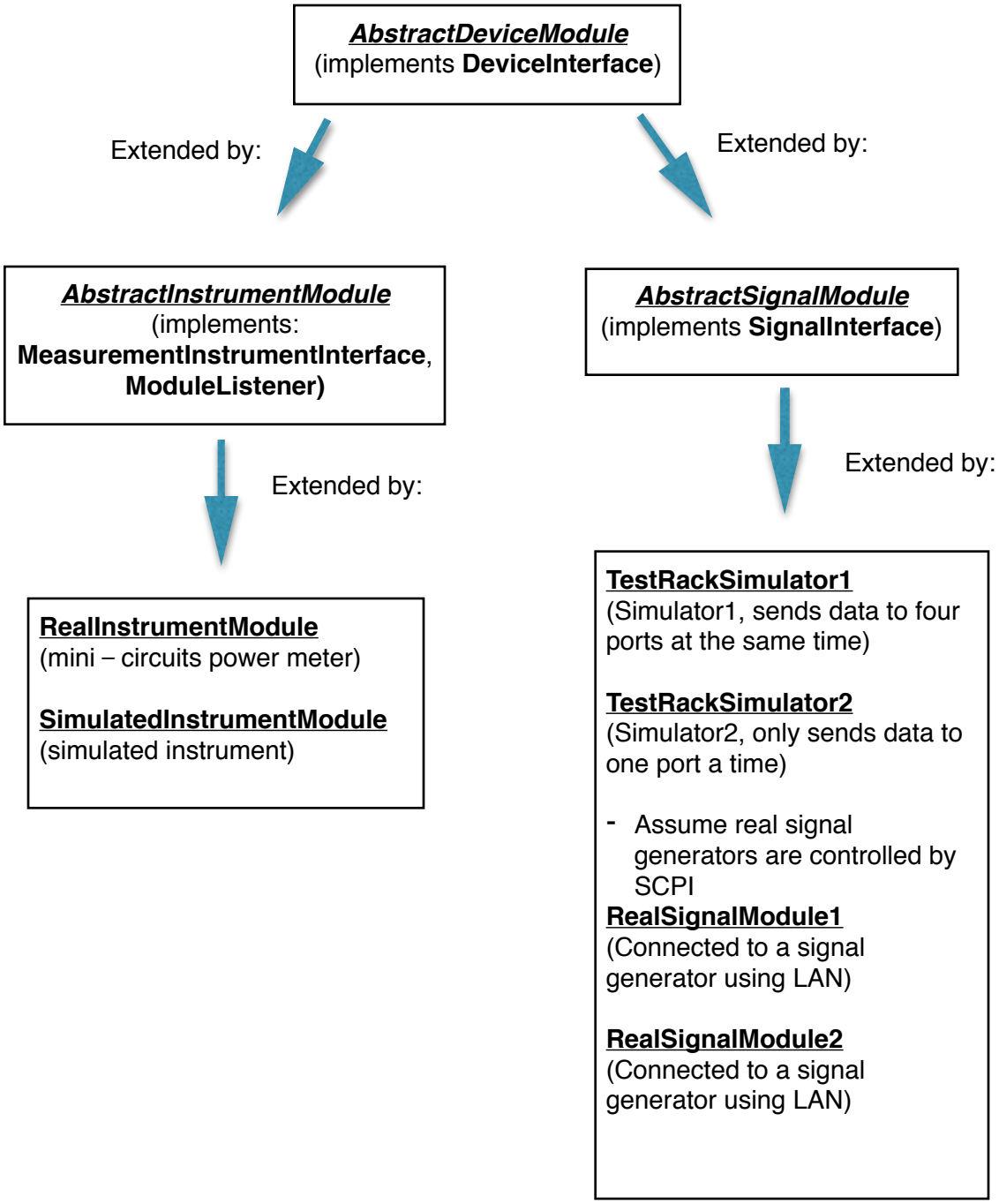
- User should choose a device and choose a display option. In the available display options, if the user chooses "Display All Ports", the GUI will display readings from all ports of the device.

(P.S. In the demo, since I don't have a real power meter and a real signal generator; even though I chose to connect power meter 1 and realSignalModule1 the GUI was not able to display anything. demo video = http://www.youtube.com/watch?v=K9fi0nUReSM)

C. Software Architecture

(The documentations of the whole project can be found at the doc folder. The index.html under doc folder is linked to all the html comment pages)

i. Type Hierarchy of Back-end Modules:



ABSTRACTDEVICEMODULE:
This module encapsulates device information, such as the name of the device, number of ports, etc. The API is defined by DeviceInterface.

ABSTRACTSIGNALMODULE:
This module inherits the AbstractDeviceModule. Apart from that, this module also implements other functions that are needed by a signal generator, such as transmitting data to other devices, etc. The API is defined by SignalInterface.

ABSTRACTINSTRUMENTMODULE:
This module inherits the AbstractDeviceModule. It implements other functions that are needed as a measurement instrument, such as connecting to a signal generator and listen to the data sent by the signal generator. This module also implements the ModuleListener interface in order to communicate with the signal generator modules.

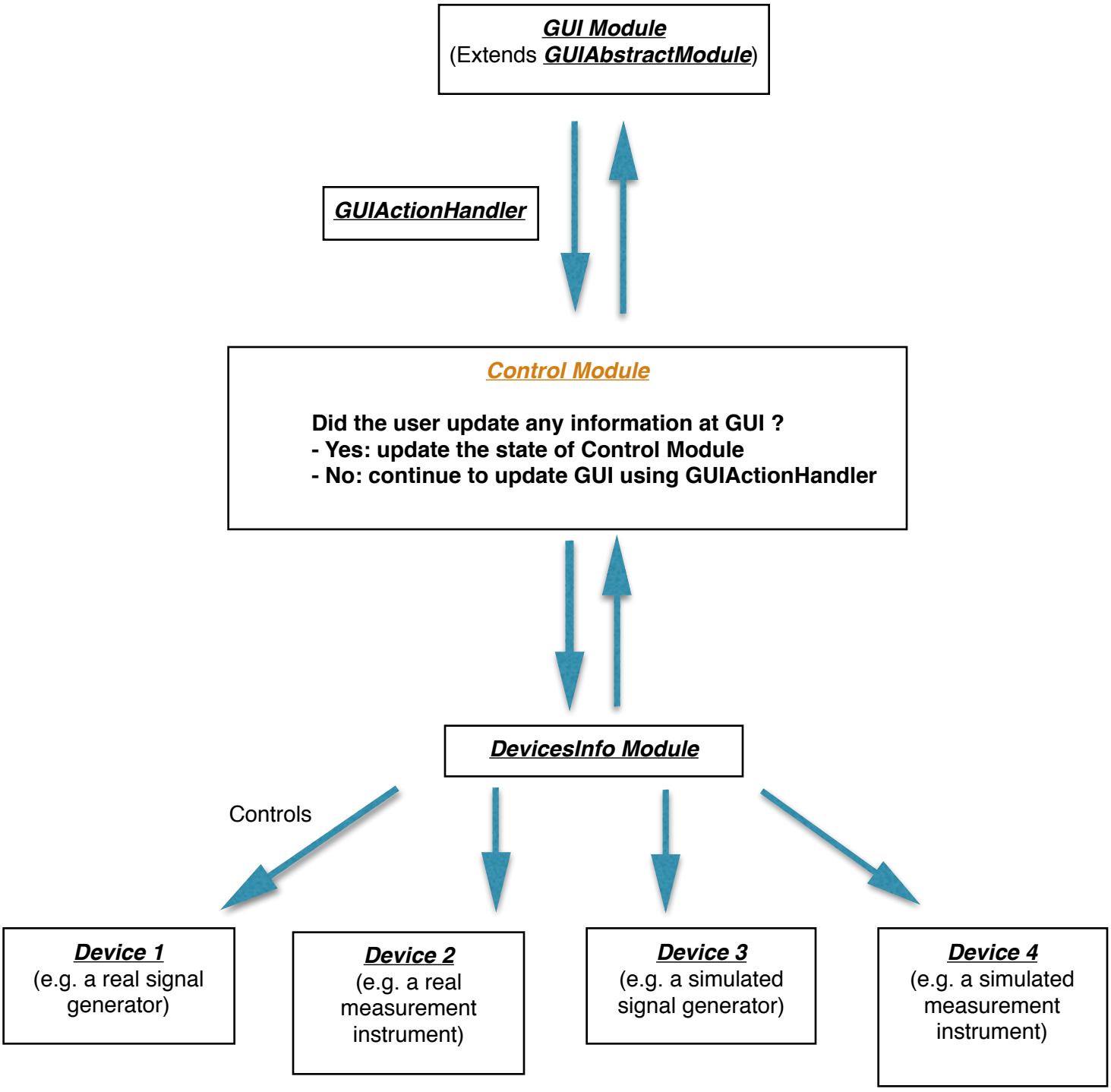
SIMULATED & REAL MEASUREMENT INSTRUMENT MODULE AND SIGNAL GENERATOR MODULE
These modules inherit either an AbstractSignalModule or an AbstractInstrumentModule. These modules need to implement three functions:

1. *configureConnection*: connects this module to a device (can be a simulated or a real device).
2. *getDataFromDevice*: reads data from a device
3. *reset*: resets the device

ii. Communication pattern between signal generator modules and measurement instrument modules:

The communication between a measurement instrument module and a signal generator module is established using an observer-pattern. Essentially, when we ask a measurement module to connect to a signal generator module, the signal generator module will keep a reference of the measurement module in a listener list. When the signal generator is ready to send data, it will iterate over the listener list and notify every measurement module to take data from the signal generator. More information regarding this pattern can be found at: http://en.wikipedia.org/wiki/Observer_pattern .

iii. Front-end Control



GUIABSTRACTMODULE:
This module implements the GUI interface. It gives some basic functionalities to control a GUI. Its API can be found at GUIInterface.

GUI MODULE:
This module extends the GUIAbstractModule. It extends the abstract module's functionalities by attaching Java Swing Listeners to all the GUI components and by notifying GUIActionHandler when the user touches a component on the GUI.

GUIACTIONHANDLER:
This module implements the GUIActionHandler interface. This module is used to transfer data between the GUI and the main control module. Whenever the user selects some component on the GUI (e.g. click on the start button), the GUI module will notify the GUIActionHandler, update it with the information needed by the control module. The GUIActionHandler will then notify the control module, and the control module will decide to change its state based on the information stored in the GUIActionHandler. The API of this module can be found at GUIActionHandlerInterface.

DEVICESINFO:
This module is used by the control module to control all the device modules instantiated in the software. When initiating the program, every device that the developer wants to monitor needs to register itself into a devicesInfo object. Once the program starts to run, the control panel will send command to the devicesInfo object to ask for a certain object, and the devicesInfo object will handle this command and expose the correct device module to the control module.

CONTROLMODULE:
This is the major component of this software. Essentially, the control module runs as a state machine. At every iteration, it checks if the GUI has been used by the user. If the control module finds a change, it will update its internal state and ask devicesInfo object to run the devices based on the current state. After the update, the control module will use the data brought by the devicesInfo object to update the plots on the GUI.

D. Example Device Modules:

SIMULATORS:
TestRackSimulator1: This simulator module sends out a CW tone at 2.4GHz. The power of CW tone decreases from -10 dBm to -30 dBm. The signal is split into four ports.
TestRackSimulator2: This simulator generates the same waveform as TestRackSimulator1. Moreover, this simulator emulates a switch box. Every time after a measurement is done, the simulator will open a random port for reading and close all the other ports.
SimulatedInstrumentModule: This simulator calculates the RMS value of the input data.

REAL DEVICES MODULES:
RealSignalModule1: This module tries to connect to a real signal generator that accepts SCPI commands via LAN. The default settings are: defaultChannel = 101,105,112,113, defaultIP = xxx.xxx.xxx.xxx, default frequency = 2020MHz, default socket port = 5025. If the connection is established correctly, this module will send SCPI commands to get data from the signal generator and split the received data to four ports simultaneously.

RealSignalModule2: This module does the same thing as RealSignalModule1. Moreover, we assume that there is a switch box connected to the signal generator. The output data of the switch box is sent to port 0. The default settings are the same as in RealSignalModule1 except that the default channel is set to 101.
RealInstrumentModule: This module first tries to connect to a Mini-circuits power meter. This module register mcl_pm64.dll in to the registry and send commands to the power meter to ask for data.

E. Appendix

1. Link to the power meter: http://www.minicircuits.com/support/software_download.html
2. Link to SCPI command tutorial provided by Agilent: http://www.home.agilent.com/agilent/editorial.jspx?cc=US&lc=eng&ckey=1688330&id=1688330
3. I use JChart2D library to plot the data to the GUI. A link to JChart2D: http://jchart2d.sourceforge.net/
4. If the console prints error message like following: Error: failed to connect to xxx.xxx.xxx.xxx at port 5025, this error occurs because the connection to a real signal generator failed.
5. If the console prints error message like following: Error occurred while trying to initialize Javonet. Please make sure if you have all dependencies installed. To check list of mandatory prerequisites please visit http://www.javonet.com/download/. Please check if you have .NET 3.5 and JDK1.6 or above installed.
6. I use Javonet to register the .dll file into the registry so that the computer can recognize the mini-circuits power meter. I am currently using a free-trial version and this version only last for 30 days.