**SolarWindModule.py:** This file contains the Python code for the solar and wind modules. There are multiple parts to this code. The first part is setting up the LCDs and searching for a Bluetooth connection. The bluetooth part will try 6 times to look for a connection to the Smart Grid raspberry pi until it assumes that they are not together and it will continue onto the rest of the code.

The rest of the code consists of the two functions that are used to measure our data. To measure the solar power, SPI is used to read values from the ADC. Taking the average of 10 readings, a scaled up value is then displayed onto the LCD. To measure the power generated by wind, values from a rotary encoder must be measured. This is done by finding the frequency of our PWM signal coming from the encoder. The frequency is measured by dividing the time taken by the number of falling edges in that duration of time. Based on this value, different GPIO pins are turned on and sent to a summation amplifier and a scaled up value is then displayed onto the LCD. Finally, if a bluetooth connection was created, then the power generated from both solar and wind are added together and then sent over bluetooth to the connected party.

**EfficiencyModule.py:** This file contains the Python code for the energy efficiency module. This module utilizes an LCD screen to display the different power consumption levels from the different light bulb options. It then uses two GPIO pins as input pins to check when the two buttons on the module are pressed. Based on the values of the pins, a different power level will be displayed which will represent a realistic value based on the light bulbs that are on at the time.

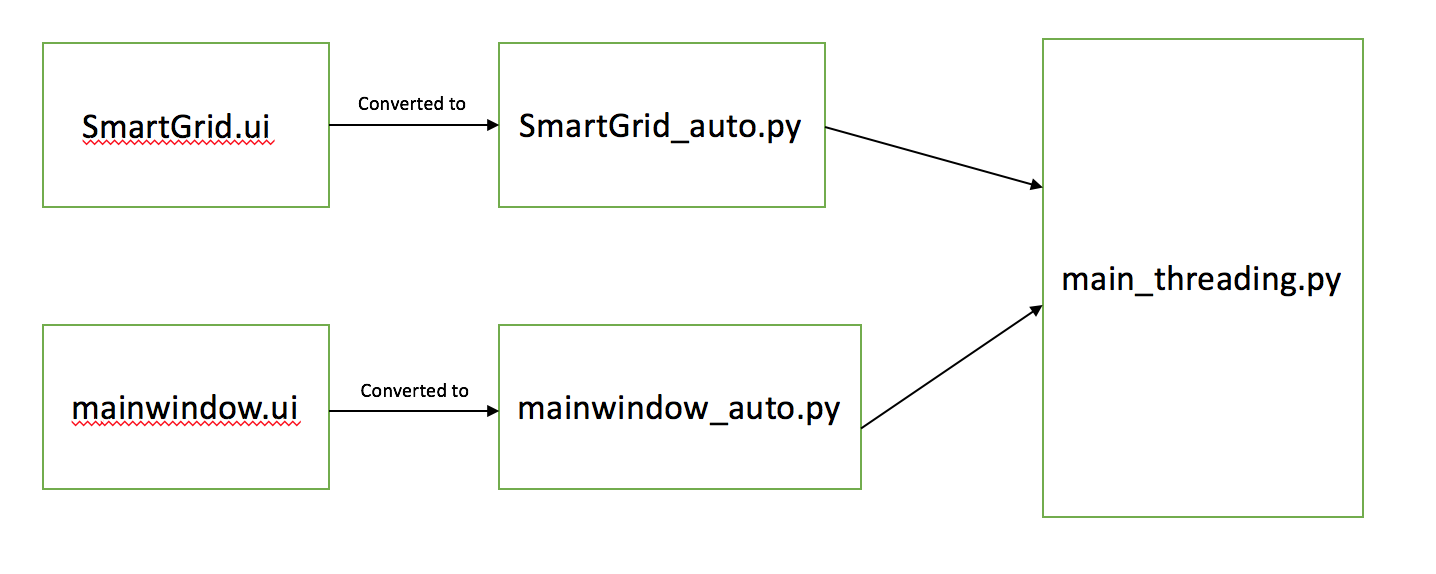
**main\_threading.py:** This file contains the Python code for the Smart Grid module. This module has code that provides functionality for the Smart Grid’s blackout response as well as GUI functionality. Upon startup, the code creates and opens up the MainWindow which will stay open at all times. Upon creating this window, default values for the variables are set and displayed and then different threads are run to set everything up. One thread sets up the blackout response and will be continuously running forever. The thread then waits until a blackout is triggered by disconnecting the transmission line which is detected using an input pin. When this happens, a blackout is triggered and then it switches to the Smart Grid window which will show a timer that runs until the transmission line is reconnected and after the other sectors have turned back on. The second thread is used to detect a bluetooth connection. It opens up to connections from another device and if it connects, it will then be able to receive values and display the values onto our Main Window.

**SmartGrid.ui:** The file containing the components of the Smart Grid user interface, but not in python. This is the file that is opened using QtCreator in order to edit what the Graphical User Interface looks like. This file was and can be (if the GUI is edited) converted into SmartGrid\_auto.py in order to add functionality.

**Mainwindow.ui:** Similar to SmartGrid.ui, this file contains the components of the summary screen user interface, but not in python. Also opened with Qt Creator in order to edit what the GUI looks like. This file was and can be (if the GUI is edited) converted into mainwindow\_auto.py in order to add functionality.

**SmartGrid\_auto.py:** Called in the main.py file. It lists all the components of the Smart Grid user interface in python so that it can be used and called in order to define the functionality.

**Mainwindow\_auto.py:** Similar to SmartGrid\_auto.py, it is also called in the main.py file and lists the components of the summary screen in python so that it can be called to define functionality.



**Figure 1: Flowchart for the main\_threading.py module**

All of the software is placed onto the respective Raspberry Pi for the different module. The code is run using the terminal and Python (3.4.2). By editing files that start at bootup, all of the necessary files will be run once the Raspberry Pi is powered on and booted up. This means that no other connection to USB or ethernet is required. For the bluetooth connection, a specific device address is provided in the bootup file so that it has no danger of connecting to the wrong device. It will automatically connect as long as they are powered on at the same time. If not, then be sure to power on the solar and wind module after the smart grid module. Also, if a connection has been made and the solar and wind module is powered off and then powered back on while the smart grid module stays on, then a connection will not be established and the smart grid module must first be power cycled to reset the connection.