**Three Phase Power Simulator**

**System Overview:**

This simulation setup utilizes a Raspberry Pi 1 and TI’s Launchxl-F28377S to provide the user with customizable three phase power signals. The Raspberry Pi utilizes PyQt4 to provide the easy to use GUI seen in Figure 1. With the addition of a USB keyboard and mouse this system can setup and utilized quickly. For this simulator, the Launchxl functions as a function generator which output the three sinusoids associated with standard three phase power. The Launchxl function output pins can be seen in Table 2 of Appendix A. The Raspberry Pi sends settings to the Launchpad via a 16-bit message (details in Table 3) sent over a standard I2C protocol. The Launchpad decodes this message and effects the outputs accordingly. The proper pin connections for I2C communication between these two devices can be seen in Table 1 of Appendix A. By using the GUI hosted on the Raspberry Pi a user can control the overall frequency of the output as well as the gain and offset of each channel. For more detail please see the System Specifications and GUI Interaction sections.

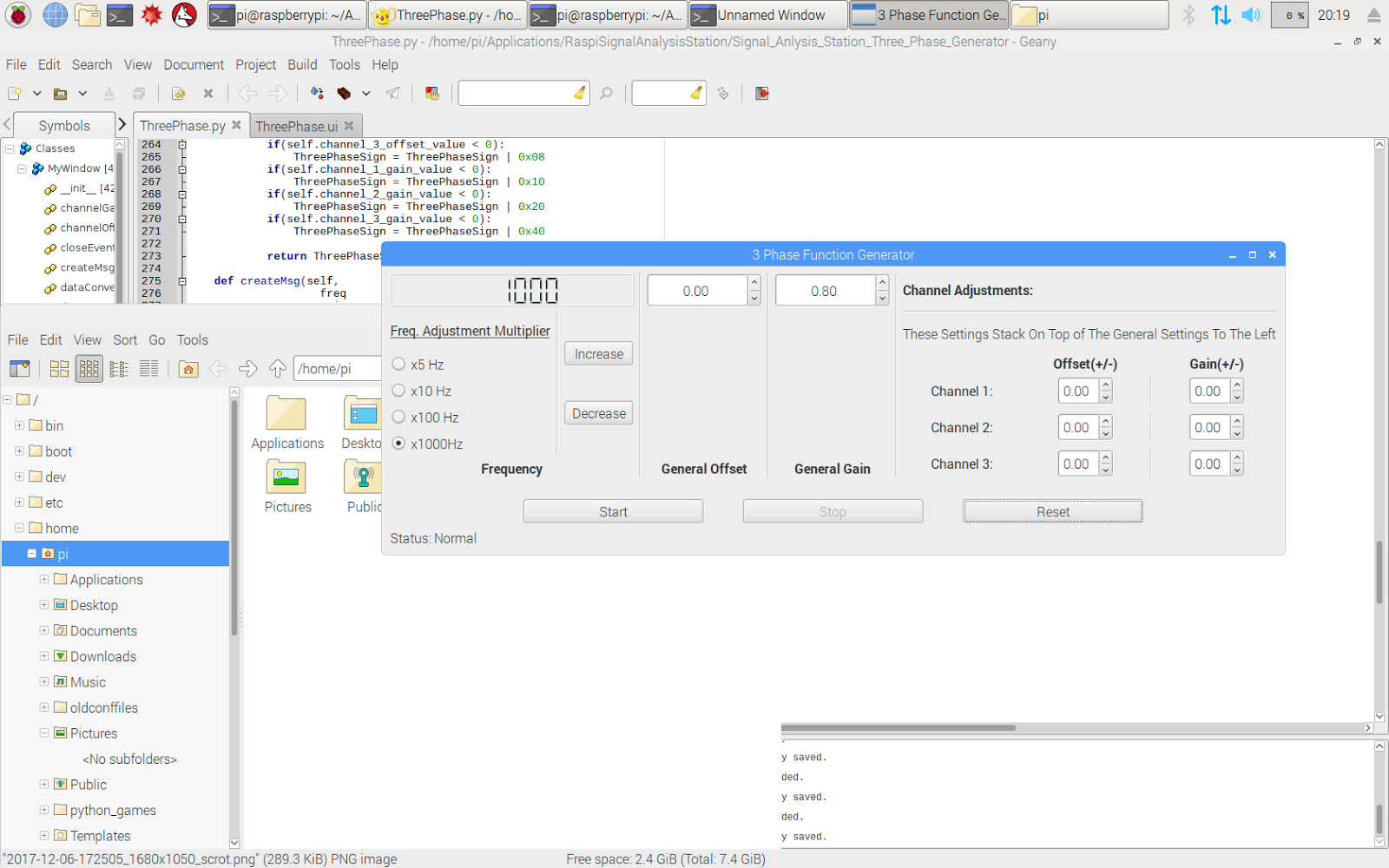


Figure 1: Three Phase GUI

**GUI Interaction:**

* Frequency Control:

Through the GUI a user can increment or decrement the current frequency using the Increase and Decrease buttons located in the Frequency section. In addition, the user can scale the effect these buttons have by selecting a multiplier from the Freq. Adjustment Multipliers. To view these features, navigate to Figure 1.

* General Output Control:

Using the General Offset and General Gain sections the user can affect the output of all three channels simultaneously. To increase or decrease either setting just use the arrows next to the corresponding value.

* Individual Channel Adjustments:

The Channel Adjustments section of the GUI allow each output channel to be modified separately. Both the offset and gain of each channel can be modified with these settings. Do note that these settings stack onto the general settings mentioned above and output rails can be easily reached.

* System Controls:

The system controls include the bottom row of buttons located in Figure 1. These buttons enable the user to stop and start the output on command. The reset button will reset both the GUI and Launchpad to preprogrammed default settings.

**System Specifications:**

* **0 V ≤ VPP ≤ 3.3V**
* **0 Hz ≤ Freq≤ 10kHz**
* **-1 V ≤ DC Offset ≤ 1 V**
* **0 ≤ GAIN≤ 1**

**Appendix A:**

Table 1: Raspberry Pi Pin Connections

|  |  |  |
| --- | --- | --- |
| **Raspberry Pi Pin #** | **Function** | **Launchpad Pin #** |
| 3 | SDA | 10 |
| 5 | SCL | 9 |
| GND | GND | GND |

Table 2: Launchpad Signal Generator Pin Usage

|  |  |
| --- | --- |
| **Launchpad Pin #** | **Function** |
| 24 | Channel 3 Output |
| 27 | Channel 1 Output |
| 29 | Channel 2 Output |
| 28 | VDAC |
| 3V3 | VDAC Connection |

Table 3: I2C Message:

|  |  |
| --- | --- |
| **BIT** | **DATA** |
| 0 | Command |
| 1 | Buffer |
| 2 | Freq MSB |
| 3 | Freq LSB |
| 4 | Buffer |
| 5 | Gain |
| 6 | SignData |
| 7 | Offset |
| 8 | Op\_Code |
| 9 | Channel 1 Offset |
| 10 | Channel 2 Offset |
| 11 | Channel 3 Offset |
| 12 | Channel 1 Gain |
| 13 | Channel 2 Gain |
| 14 | Channel 3 Gain |
| 15 | Buffer |