guPi-Blaster Documentation

Cody Finn, Harrison Beerley, Jacob Smith, Jacob Tarlo

August 4, 2016

Contents

1 What is guPi-Blaster **1**
- 1.1 Pulse Generator . . . 2
- 1.2 Peripheral Reading . . . 2
  - 1.2.1 Voltmeter . . . 2
  - 1.2.2 Ammeter . . . 2
  - 1.2.3 Thermometer . . . 3
  - 1.2.4 Force Meter . . . 3
- 1.3 Data Collection . . . 3

2 How to Use guPi-Blaster **3**
- 2.1 Square Wave Generator State Control (On/Off Button) . . . 4
- 2.2 Pulse Width Incrementation/Decrementation . . . 4
- 2.3 Auto Test Mode . . . 4
- 2.4 Live Data Feed . . . 4
- 2.5 Log File Selection . . . 4
- 2.6 Preference Saving . . . 5

3 How to Build from Source **5**
- 3.1 Dependencies . . . 5
  - 3.1.1 How to install Java . . . 5
  - 3.1.2 How to install Gradle . . . 5
- 3.2 Building the jar file . . . 5

1 What is guPi-Blaster

guPi-Blaster is a multifaceted tool designed to test motor and prop combinations. The software consits of 3 major parts.

- Pulse Generator
- Peripheral Reading
- Data Collection

1.1 Pulse Generator

In order to drive the motors used on our quadcopters we use Electronic Speed Controllers(ESCs). The ESCs control the speed of the motor they are attached to. By varying the pulse width of the electric signal going to the ESC you can control how fast the motor will spin. The ESC uses a pulse with a frequency of 50Hz, and a varying pulse width of 1-2ms. A 1ms pulse width is the slowest the motor will go (It probably wont spin at all) and 2ms is the fastest the motor will spin.

1.2 Peripheral Reading

guPi-Blaster is currently configured to read from a:

- Voltmeter
- Ammeter
- Thermometer
- Force Meter

1.2.1 Voltmeter

The voltmeter is actually a

Digital-to-Analog Converter(DAC). Specifically it is the ADS1015. This is a 12 bit DAC that will convert the analog signal across two of its input pins. The analog signal will be represented by a 12 bit integer.

1.2.2 Ammeter

The ammeter is another Digital-to-Analog Converter(DAC) that uses the same ADS1015 the voltmeter communicates across. In the same fashion as the voltmeter, the 12 bit DAC will convert an analog signal across two of its input pins. The analog signal will be represented by a 12 bit integer.

1.2.3 Thermometer

The thermometer is linked to a Cold-Junction Compensated Thermocouple-to-Digital Converter specifically known as a MAX31855. The MAX31855 reads data from a thermometer and stores it in a register using a digital format with 14-bit resolution.

1.2.4 Force Meter

1.3 Data Collection

The guPi-Blaster application automaticaly logs information and data while in use to a log file. The log file is in the CSV format so that it can easily be imported into Excel or other graphing software. Data that is logged to the log file is shown in parallel in the GUI itself, this is a live feed of the data being read from the instruments. Information stored to the CSV file are calculated averages from the live feed, and additional information from calculations are added based on the raw data, including:

- Lift (in grams), which is calculated using the equation Force = Mass * Acceleration, inputting force from the force meter and acceleration due to gravity, resulting in Force/Acceleration = Mass (in kg), multiplying the resultant by 1000 to get the calculation in grams.
- Power (in watts) is calculated by multiplying voltage and current.

2 How to Use guPi-Blaster

guPi-Blaster implements an intuitive interface that includes the following features:

- Square wave generator state control (on/off button)
- Pulse width incrementation/decrementation
- Auto test mode
- Live data feed
- Log file selection
- Preference saving

2.1 Square Wave Generator State Control (On/Off Button)

The state control button is a big button located at the bottom of the window and defualts to the off state upon program startup. When pressed, the state of the button will change to on and will begin to produce square waves. Every time the state changes from on to off, the pulse width will set itself automatically to 1.0 ms to prevent damage to the motor by rapidly speeding up the next time it is turned on again. If auto test mode is enabled and the state control button is pressed, setting it to off, the auto cycle will be interupted and the pulse width will be set back to 1.0 ms.

2.2 Pulse Width Incrementation/Decrementation

While the state of the program is on and is producing square waves, the pulse width of these square waves can be manually altered by intervals of 0.1 ms by pressing the plus and minus buttons in the top corners of the window, incrementing and decrementing the pulse width by 0.1 ms respectively. The label between the two buttons visualizes the current pulse width being produced; pulse widths below 1.0 ms and above 2.0 ms cannot be achieved.

2.3 Auto Test Mode

By pressing the "Auto" button in the left middle of the window, auto mode is enabled, automatically incrementing the pulse width from 1.0 ms to 2.0 ms at 0.1 ms intervals with a one second buffer between each one to allow for data to be recorded at each pulse width. Again, this can be cancelled at any time by turning the sqaure wave generator off.

2.4 Live Data Feed

In the center of the window, a live feed of the data being recorde from all four sensors can be read.

2.5 Log File Selection

On the righthand side of the window in the middle a button can be found reading "Choose File". Pressing this button will open a file chooser menu that allows you to select a file you wish data to be written to. If no such file exists already, you have the option to create one.

2.6 Preference Saving

When the program is first launched you will be asked to specify a preference file location. With such a file provided, your preference as to where your log file is will be saved even if you close and reopen the program. If you do not specify a preference file, you will simply have to specify the location of a log file each time the program is used.

3 How to Build from Source

3.1 Dependencies

- Java
- Gradle

3.1.1 How to install Java

1. Windows
2. Linux
3. OSX

3.1.2 How to install Gradle

1. Windows
2. Linux
3. OSX

3.2 Building the jar file

In order to build the jar file you simply type into the command line "gradle build jar"