

AUTOMATED TEST BENCH

An Undergraduate Research Scholars Thesis

by

MILLIE KRIEL

Submitted to the Undergraduate Research Scholars program at
Texas A&M University
in partial fulfillment of the requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by Research Advisor:

Dr. Stavros Kalafatis

May 2019

Major: Electrical Engineering

TABLE OF CONTENTS

	Page
ABSTRACT.....	1
DEDICATION.....	2
ACKNOWLEDGMENTS	3
NOMENCLATURE	4
CHAPTER	
I. INTRODUCTION	5
Background.....	5
Overview.....	6
II. OPERATING CONCEPT.....	7
Scope.....	7
Operational Description and Constraints	7
System Description	8
Modes of Operations.....	9
Users	11
User Scenarios	11
User Support	12
III. FUNCTIONAL SYSTEM REQUIREMENTS	13
Performance Requirements.....	13
Electrical Characteristics	16
Failure Propagation.....	16
IV. RESULTS	18
Validation.....	18
Test Criteria	18
Test Cases	18
Test Results.....	19
V. CONCLUSION.....	21

Research Impact and Future Improvements	21
REFERENCES	22
APPENDIX.....	23

ABSTRACT

Automated Test Bench

Millie Kriel

Department of Electrical and Computer Engineering
Texas A&M University

Research Advisor: Dr. Stavros Kalafatis

Department of Electrical and Computer Engineering
Texas A&M University

Printed Circuit Boards are used to electrically connect components that are soldered onto the board in order to perform a function or task. They are manufactured by etching out copper traces from the copper layer that is adhered to a non-conducting substrate. Testing a PCB by manually applying and measuring signals is a very time consuming process. However, designing and building a specific automated functional test for one PCB could be expensive and could become obsolete if another board design needs to be tested. To solve this problem, a software interface will be designed that initializes and completes a test using commonly found, laboratory electrical equipment. The program will allow users to input a PCB's functions and pass/fail criteria and then to run the test to receive a test report, indicating the result of each function's test status.

DEDICATION

To my parents and siblings, for their unconditional love and support.

ACKNOWLEDGEMENTS

Thanks also go to my friends and colleagues and the department faculty and staff for making my time at Texas A&M University a great experience. I also want to extend my gratitude to the Fischer '72 Engineering Design Center, which provided the necessary instrumentation.

Finally, thanks to my mother and father for their encouragement and to my sister, brother, and good friend for their support.

NOMENCLATURE

GUI	Graphical User Interface
Hz	Hertz
ICD	Interface Control Document
kHz	Kilohertz (1,000 Hz)
mA	Milliamp
MHz	Megahertz (1,000,000 Hz)
PCB	Printed Circuit Board
USB	Universal Serial Bus
VISA	Virtual Instrument Software Architecture

CHAPTER I

INTRODUCTION

A Printed Circuit Board (PCB) is designed to perform useful functions by electrically connecting various components and integrated circuits. There are many ways to test whether a PCB will perform correctly. One method for testing the functionality of the board is to test the printed circuit board while it is integrated into the entire system. However, this method becomes cumbersome for test engineers to debug the source of the error. The error could stem from the physical board itself being manufactured or designed incorrectly, the components or sensors having failures, or the program having errors, etc. A simpler method may be to measure continuity across the PCB traces, but this will not provide proof that the printed circuit board design or microcontroller program is performing correctly. A more concrete way of testing the board's functionality would be to control a waveform generator to simulate all inputs and use an oscilloscope to measure outputs. Doing this by hand would consume a lot of time without eliminating the potential human error in recording data. By automating this process with a computer program, data can be collected reliably and quickly. Many automated PCB test benches are designed specifically for one PCB and do not handle the general case. This project involves creating a software interface to commonly found electrical equipment that allows the user to test a generic PCB automatically after inputting information about the PCB.

Background

Within automated testing of PCBs, there are several approaches that have been used. The different approaches fall into three main categories: Automated Optical Inspection, In-Circuit Tests, and Functional Tests.

Automated Optical Inspection tests include taking high resolution scans of the PCB and looking for visual defects. In-Circuit tests involve probing the board, looking for open or short circuit connections as well as measuring electrical properties such as resistance or capacitance. The Automated Optical Inspection and In-Circuit tests validate whether the board was manufactured per its design, but does not prove that the design results in the expected output. Functional tests provide stimulus to the board and verify responses. Functional tests allow the board to uncover problems in analog circuitry, digital circuitry, and connectors which are shortcomings of Automated Optical Inspection and In-Circuit tests.

Overview

The user will define the PCB under test, the test settings, and then will run the test to receive a test report. The user will specify the parameters of the input signals to the board and the expected output signals of the board using a Graphical User Interface (GUI). This GUI will call on scripts which use Virtual Instrument Software Architecture to initialize the test equipment. The four-channel oscilloscope will be set up to measure signals, based on the programmer's guide [1], and to return the results of the measurements. The two-channel function generator, based on the programmer's guide [2], will be set up to sweep a given voltage or create a waveform to simulate. The Arduino Mega will be used to send serial communication strings and to record the received serial communication strings. Using one four-channel oscilloscope, one two-channel signal generator, and one Arduino Mega, the user will be able to test two analog signal inputs, measure four analog outputs, and send and receive serial communication strings to the board at a time.

CHAPTER II

OPERATING CONCEPT

Scope

The scope of this project is to implement a software interface that allows a generic PCB to be tested. Supported functions include sending and receiving serial communication signals, simulating analog sensor outputs, and measuring produced signals features.

The Graphical User Interface (GUI) will allow the user to choose the input test cases to the PCB as well as pass/fail criteria. The GUI will run scripts that interact with the oscilloscope, signal generator, and Arduino Mega in order setup and run the test. The GUI will output whether or not the PCB failed and, upon failure, which output channel did not performed correctly.

Operational Description and Constraints

To use this system, the user must specify on the Graphical User Interface what input signals are needed and what output signals would be produced if the PCB was designed and manufactured correctly. The user will specify the functions mapping to each input channel and output channel. This will be entered by selecting on a drop-down menu of specific functions or measurements for that channel. By selecting the button next to that function, the user will be taken to a separate window to further specify expected levels and pass/fail criteria.

The user will physically wire the oscilloscope connector and waveform generator connector to the printed circuit board's relevant test connectors. The oscilloscope, signal generator, and Arduino Mega will also be connected via USB to the user's laptop that will be running the GUI. Due to the available hardware, this project is constrained to a four-channel

oscilloscope and two-channel waveform generator. This allows for two analog signal inputs and four analog output measurements to be tested at one time. Once the test has run, a window will display the test report. The test report will show the results, the corresponding criteria, and whether or not the system has passed.

System Description

Based on the block diagram, the user will specify the input signals based on the functions of the test PCB. This input signals' settings will then be used to setup the function generator to generate the correct waveforms, the oscilloscope to measure data, and the Arduino Mega to transmit and receive serial communication. The program will then generate a test report specifying whether each function passed or failed. The project can be broken down into two major sub-systems: user interface and equipment interface.

The user interface involves creating a Graphical User Interface that is easy to use and visually appealing. The GUI allows the user to specify details for each input and output channel, pass/fail criteria, and displays the test report after the test has run. The equipment interface involves establishing an USB interface with the equipment with Virtual Instrument Software Architecture and controlling the equipment using SCPI commands. The code will initialize communication from the user's computer running the GUI, through USB, to the oscilloscope or signal generator. The code created will need to handle the generic case when deciding which signals to measure or to create. This means that the GUI will send the settings information to the equipment programs, and the equipment program will then be responsible for calling upon the correct functions and specifying the correct parameters to use. The block diagram of the system can be seen in Figure 1, which details the connections between all of the equipment and the PCB under test.

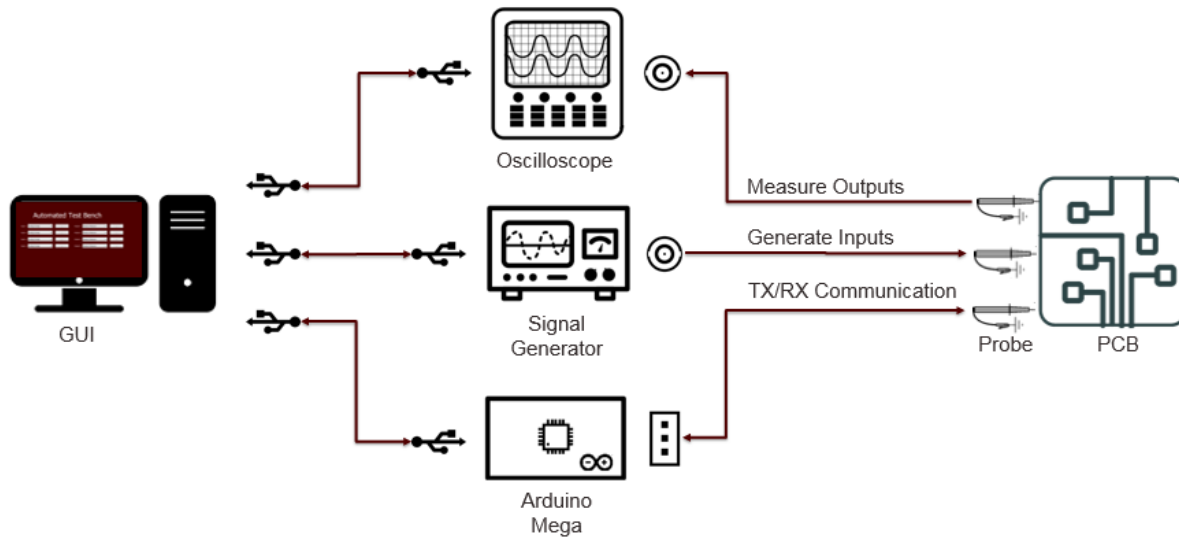


Figure 1. Automated Test Bench Setup

Modes of Operations

The modes of operations include user input, equipment initialization, data collection, signal generation, serial communication transmission and recording, and test report generation. The user input mode involves the initial interaction of the user on the GUI. The user will specify which inputs will be sent to the board and what the expected outputs should be for each measurement. The flow of the GUI program can be seen in Figure 2. Screenshots of each GUI's window can be seen in the Appendix Figures 7 – 14.

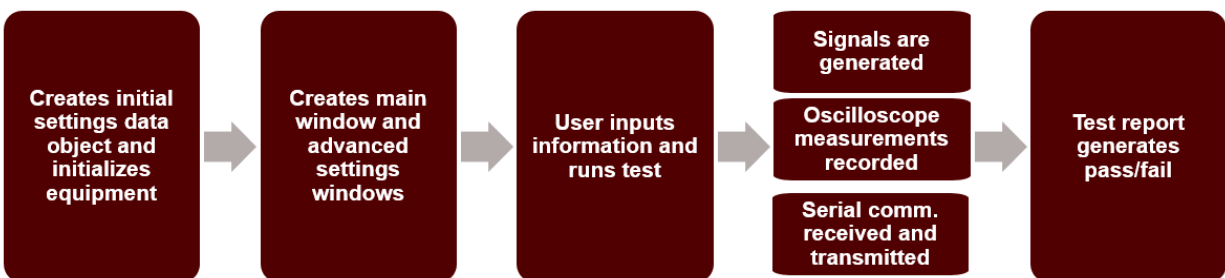


Figure 2. GUI Program Flow Chart

Further information will be needed based on the function. For input channels, the user will specify either what serial communication characters to send or what voltage ranges and waveforms to use for stimulating the board. The flow for the signal generator program can be seen in Figure 3.



Figure 3. Signal Generator Program Flow Chart

For output channels, the user will specify what expected serial communication characters should be received and what features should have been measured from a signal to constitute as a pass criteria. The program flow for the oscilloscope program can be seen in Figure 4 and the serial communication program flow can be seen in Figure 5.



Figure 4. Oscilloscope Program Flow Chart

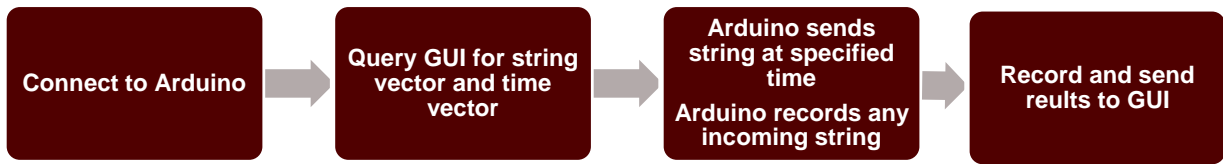


Figure 5. Signal Communication Program Flow Chart

The equipment initialization involves using VISA to open the resource via USB connection and SCPI commands to setup the settings needed to collect data or produce the signal. Data collection and signal generation occur in tandem, while the test is running. Once the test is complete, a test report is generated that displays the data in an easy to read manner.

Users

The intended users for the system are PCB designers who want to efficiently test the functionality of their PCB without having to design a PCB specific test bench set-up. This is ideal for users who would like to pre-test PCB designs before upgrading the PCB to its final version. The software will allow the designer to verify initial designs while also giving the designer the flexibility to test future versions, due to the ease of altering the program's test functionality.

User Scenarios

Functional Test before System Integration

Before integrating a PCB into a larger system, a design engineer will need to validate the performance of his or her PCB. This program may be used as a validation test to ensure the PCB will operate correctly with other systems, in order to minimize the debugging process and reduce risk of damaging other systems. This documentation serves as a continual log for the engineer, in

case of a design failure. The engineer can use the validation testing logs to report progress in design changes. Once all functions have passed, the engineer can use these logs as proof and documentation that a PCB has passed successfully and can be integrated successfully.

Functional Test for Stress Test Experiments

This program may be used as before and after data measurement and documentation for PCBs undergoing experimentation. One example includes stress testing a PCB under mechanical or thermal conditions. The test can be run before undergoing stress, to serve as a baseline for the PCB. Once the PCB has been stressed, the test can be re-run and reports can be compared to show if there was any degradation in the board. The user can define within what error a measurement will be accepted in a passing test. With different error choices, the user can evaluate whether the system passes with a similar or higher rate of error after stress-testing.

User Support

The user may refer to the manual for operating the program as well as best practices for connecting the PCB connectors to the electrical equipment. The manual will include screenshots of the GUI to better instruct users on how to step through setting up a test. The readme file will specify computer requirements to run the program and instructions showing how to install those dependencies. The code will be well-commented, describing each function's purpose and the form of each input and output parameter.

CHAPTER III

FUNCTIONAL SYSTEM REQUIREMENTS

Performance Requirements

1.1 Intuitive User Interface

The Graphical User Interface must be user-friendly and easily understood with the supplement of the user manual. Buttons and drop down menus will be used with clear icons. Text will be meaningful and give clear indication of what information is needed from the user to perform the test. If necessary information is missing, the user will be prompted to give that information before the test is conducted. Error-handling will be implemented to prevent the user from entering numeric data outside the bounds of the equipment. Type checking will be implemented to ensure the correct type of data is entered. The user will be alerted if incorrect information is entered and value will change back to the default value. The specifications of the GUI can be found in Table 1. The intent of the software is to make automated PCB testing easier for users and this should be captured in how the graphical user interface is designed.

Table 1. GUI Specifications

Module	# of Unique Windows
PyQt5	Main: 1 Signal Generator: 4 x 2 channels Oscilloscope: 1 x 4 channels Input Serial Communication: 1 Output Serial Communication: 1 Error: 1

1.2 Transmit and Receive Serial Communication Signal

The Arduino Mega must be able to transmit a serial communication signal based on the user's input. The signal must reproduce each character's signal without skipping characters or adding extra characters. The serial communication speed will be based on the baud rate specified by the user and the logic level is 5V. The buffer size of the Arduino Mega is 64 characters. The Automated Test Bench will support one serial communication port. This requirement validates that digital logic can be implemented into the software program. While implementing every logic protocol is outside of the scope of this project, this validates that the system can be used to analyze generic logic given the time and resources to implement.

Table 2. Serial Communication Specifications

Language	Output	Typical Baud Rates	Testing Model
Python 3.6 and Arduino language (set of C/C++)	String	9600, 14400, 19200, 28800, 38400, 57600, or 115200	Arduino Mega

1.3 Simulate an Analog Sensor's Output

The function generator program must be able to simulate an analog sensor's output by creating generic waveforms. The program will query the function generator to establish the function generator's maximum and minimum bounds for criteria such as voltage and frequency. The program will not run the test if the waveform cannot be created using the signal generator. The Automated Test Bench will support two inputs to the PCB. This requirement is a necessary component of functional testing, to excite the system with signals to verify performance. Many circuits have analog components that are useful to simulate to eliminate the possibility of a broken sensor.

Table 3. Signal Generator Specifications

Language	Error Handling	Waveforms	Parameters
Python 3.6 using SCPI (Standard Commands for Programmable Instruments)	Range Checking <ul style="list-style-type: none"> • Minimum • Maximum Input Type	<ul style="list-style-type: none"> • Sinusoid • Square • Ramp • Pulse 	<ul style="list-style-type: none"> • Frequency • Voltage range • Duty cycle • Rise/fall time • Phase

1.4 Measure Analog Signal Features

The oscilloscope program must measure analog signals and extract meaningful features in order to validate the users PCB. The features must support the decision to pass or fail the board. The program must be able to measure a Pulse Width Modulation signal’s duty cycle. The Automated Test Bench will support four outputs from the PCB. This requirement validates that analog feature extraction for PWM signals can be implemented into the software program. While implementing every possible measurement is outside of the scope of this project, this validates that the system can be used to analyze generic analog features given the time and resources to implement.

Table 4. Oscilloscope Specifications

Language	Error Handling	Measurements	Output
Python 3.6 using SCPI (Standard Commands for Programmable Instruments)	Range Checking <ul style="list-style-type: none"> • Minimum • Maximum Input Type	<ul style="list-style-type: none"> • Duty Cycle • Maximum • Minimum • Rise Time • Frequency 	Results for specified measurements

1.5 Create Meaningful Test Report

The program must output a meaningful test report indicating which functions passed or failed the criteria that the user specified in the GUI. The report must show measurements or data acquired during the test in a readable manner. The purpose of the test bench is to allow users to validate the performance of their PCB through functional testing, requiring a test report showing the results of the test.

Electrical Characteristics

2.1 Inputs

The presence or absence of any combination of the input signals in accordance with the previous specifications applied in any sequence shall not damage the testing equipment. No sequence of command shall crash the Graphical User Interface or cause any malfunction. By design, the program should limit the chance of damage or malfunction by user error.

2.2 Data Output

The Automated Test Bench shall include an interface compatible with the user's computer. This will be in the form of three USB type A interfaces. The Automated Test Bench information may pass directly to the user's system.

Failure Propagation

The Automated Test Bench shall not allow propagation of faults beyond the scope of the program.

3.1 User-Handled Error Messages

The Automated Test Bench system shall have an internal error catching code that will generate error messages for the user to handle, such as a failed initialization of the oscilloscope or signal

generator or missing user input that is critical to executing the test. This is a requirements to ensure there are no faulty tests or commands that the equipment cannot execute.

CHAPTER IV

RESULTS

Validation

Test Criteria

A table was created and used to validate the program's error handling as well as specify each user-editable object's corresponding function which can be seen in the Appendix Table 6. Many of the errors can be grouped into the basic categories of type error or range error. For example, many settings are based on numbers and if the test report was ran with that variable type as a string, an error window would pop up and the test would not be completed. The other common type of error is a range error, where a number is entered but the devices are not capable of being configured in that range.

Test Cases

The test cases for the program include: connecting the signal generator probes directly to the oscilloscope probes and connecting the TX and RX lines of the Arduino Mega together, an analog filter PCB, and the avionics PCB for the Texas A&M University Sounding Rocketry Team. With these three test cases, the functionality of the program was able to be validated. The first test case involving connecting the input generating equipment directly to the output measuring equipment allowed for verifying that the signal generator waveforms were producing according to their specifications and on the correct channel. It also verified that the time scaling and voltage scaling of the oscilloscope was configured correctly. Because the input signals were known, the test report logic was verified using both correct pass criteria and incorrect pass

criteria to test both pass and fail conditions. The system was further tested with known working PCBs that are further specified in Table 5.

Table 5. Test PCBs

Title	Purpose	Necessary Inputs	Produced Outputs
Filter PCB	This PCB includes a low pass, band pass, and high pass analog filter using 1uF capacitors and potentiometers to adjust the cut-off frequency.	Test Case 1: A signal with the appropriate frequency to be passed through Test Case 2: A signal with the appropriate frequency so that it is attenuated	Test Case 1: Measuring amplitude will give similar amplitude to the input Test Case 2: Measuring amplitude will give amplitude that is attenuated compared to the input
Texas A&M University Sounding Rocketry Team Avionics PCB	This PCB measures the output of analog pressure transducers. Once it receives a command string via serial communication, it confirms receiving the command outputs a Pulse Width Modulation signal to turn a servo, which is a step in the process of launching the rocket	An input signal simulating the pressure transducer data, ranging from 0 to 5 V A serial communication command	A serial communication confirmation string A PWM signal of a certain duty cycle

Further information of the Test PCBs can be seen in the Appendix Figures 15 – 17.

Test Results

Equipment Simulating Inputs Connected to Equipment Measuring Outputs

The test went successfully, showing that the Automated Test Bench can successfully stimulate the board and measure the outputs to create a useful test report. One particular test case is shown in Figures 6.

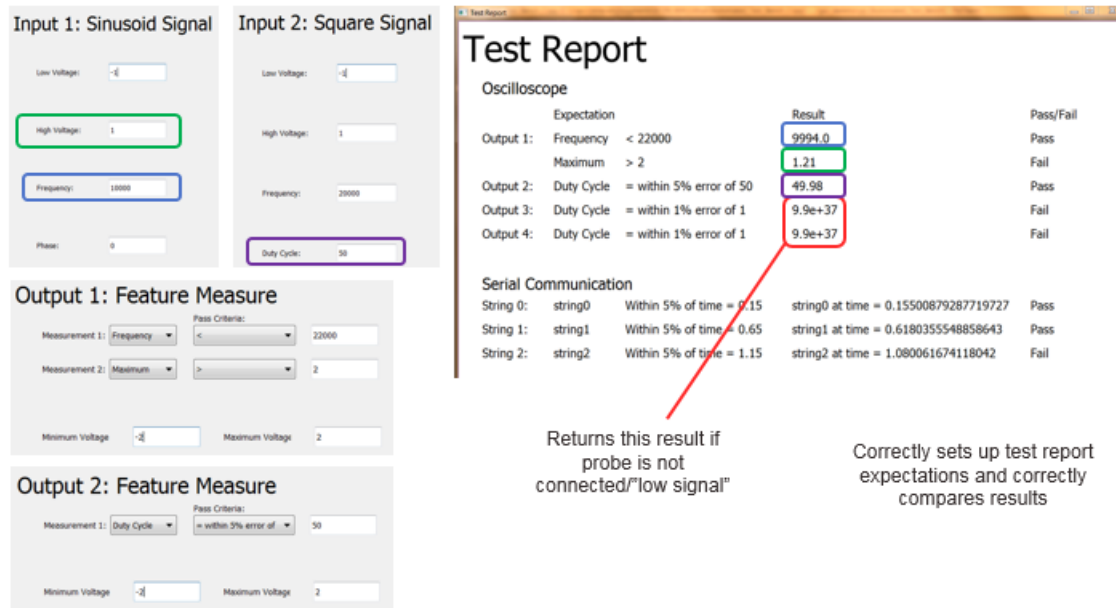


Figure 6. Successful Test Case 1

Filter PCB

Based on the cut-off frequency chosen, the PCB was able to be tested by stimulating the board with a waveform of a certain frequency and having the maximum and minimum voltage amplitude be measured. Several test cases were able to show a “Pass” when the voltage amplitude was correctly specified on the GUI as either being attenuated or passed through, based on the frequency of the input signal.

Texas A&M University Sounding Rocketry Team Avionics

The Automated Test Bench was able to simulate pressure transducers by outputting a signal from 0 to 5V corresponding to a PSI value of 0 to 1000 PSI. The duty cycle of the outputted PWM signal was correctly measured and reported on the GUI generated test report. The serial communication commands to the PCB were sent per specification and confirmation strings from the board were received as intended.

CHAPTER IV

CONCLUSION

Research Impact and Future Improvements

Summary of Proposed Improvements

The system provides an automated printed circuit board functional test which saves users time and increases accuracy compared to operating the equipment by hand. The system also provides a general platform that is flexible enough to allow for different future PCB designs. The system is easy enough to use that if a user changes a functional design on their PCB, they simply have to select a few settings in order to be ready to test the new PCB. The program offers the advantage of reducing human error during data recording and reducing the time to create a test report.

Disadvantages and Limitations

This project is limited by the number of inputs and outputs available to the user. The nature of a functional test means the user will not know exactly what trace or component is malfunctioning so further debugging will need to be done, upon failure.

Alternatives

An alternative solution would be to program a commercial DAQ board with more inputs and outputs. The trade-offs would be that the DAQ board may have lower resolution measurements or could not produce as generic waveforms as the signal generator. The DAQ board may be limited in voltage range. However, if the user did not have access to an oscilloscope or waveform generator, this option may be more cost effective and mobile.

REFERENCES

- [1] “Keysight InfiniiVision 1000 X-Series Oscilloscopes ...” [Online]. Available: https://www.keysight.com/upload/cmc_upload/All/1000_X-Series_prog_guide.pdf. [Accessed: 04-Mar-2019].
- [2] Keysight 81150A and 81160A Pulse Function Arbitrary Noise Generator” [Online]. Available: <http://literature.cdn.keysight.com/litweb/pdf/81160-91020.pdf> [Accessed: 04-Mar-2019].

APPENDIX

Table 6. Error Handling Table

Window	Identifier	Intended Action	Implemented Error Checking	Error Message
Main	Signal Generator Address Line Edit	Program initializes USB connection to signal generator	If program can't establish connection with signal generator	Connection with signal generator could not be established. Verify connection and address.
			If signal generator can't establish connection due to instrument error	Signal Generator Instrument Error: <received instrument error message>
	Input [1/2] Waveform Dropdown	Program creates waveform for channel [1/2] based on dropdown selection		
	Input [1/2] Waveform Button	Program opens window with advanced settings for channel [1/2] based on dropdown selection		
	Serial Communication USB Port Line Edit	Program initializes USB connection to Arduino Mega	If program can't establish connection with Arduino Mega	Connection with Arduino Mega could not be established. Verify connection and address.
			If Arduino Mega can't establish connection due to instrument error	Arduino Mega Instrument Error: <received instrument error message>
	Serial Communication Baud Rate Line Edit	Program initializes Arduino Mega serial port with given baud rate	If entered value is not an integer	Baud Rate must be an integer
			If entered value is above the max, 115200, or below the minimum, 300	Baud Rate must be integer within range 300 to 115200

Table 6. Continued

	Input Serial Communication Number of Packets Line Edit	Advanced serial communication input window will allow for the number of packets set by user	If entered value is not an integer	Number of packets must be an integer
			If entered value is above the max, 10, or below the minimum, 0	Number of packets must be within range 0 to 10
	Input Serial Communication Button	Program opens window with advanced settings for input serial communication based on number of packets		
	Oscilloscope Address Line Edit	Program initializes USB connection to oscilloscope	If program can't establish connection with oscilloscope	Connection with oscilloscope could not be established. Verify connection and address.
			If oscilloscope can't establish connection due to instrument error	Oscilloscope Instrument Error: <received instrument error message>
	Oscilloscope Timescale Line Edit	Program will set up oscilloscope with this timescale	If entered value is not numeric	Timescale must be numeric
			If entered value is not within device range	Oscilloscope timescale must be within range <instrument range>
	Output [1/2/3/4] Number of Measurements Line Edit	Advanced measurement window will allow for the number of measurements set by user	If entered value is not an integer	Number of measurements for Channel<> must be an integer
			If all entered values for channel 1 - 4 exceed 10	Oscilloscope can only record 10 measurements. Only last 10 measurements will be recorded

Table 6. Continued

	Output [1/2/3/4] Measurement Button	Program opens window with advanced settings for channel [1/2/3/4] based on number of measurements		
	Output Serial Communication Number of Packets Line Edit	Advanced serial communication input window will allow for the number of packets set by user	If entered value is not an integer	Number of packets must be an integer
	Output Serial Communication Button	Program opens window with advanced settings for output serial communication based on number of packets	If entered value is above the max, 10, or below the minimum, 0	Number of packets must be within range 0 to 10
Sinusoid Waveform Window	Low Voltage Line Edit	Program will create sinusoid waveform with low voltage peak corresponding to user input	If entered value is not numeric	Low voltage error for Input <>: Must be numeric
			If entered value is above or equal to high voltage value	Low voltage error for Input <>: cannot be equal to or exceed high voltage value
			If entered value is below device voltage minimum	Low voltage error for Input <>: Must be above device minimum of <>
	High Voltage Line Edit	Program will create sinusoid waveform with high voltage peak corresponding to user input	If entered value is not numeric	High voltage value for Input <> must be numeric
			If entered value is above device voltage maximum	High voltage error for Input <>: Must be below device maximum of <>
	Frequency Line Edit	Program will create sinusoid waveform with frequency corresponding to user input	If entered value is not numeric	Frequency error for Input <>: Must be numeric

Table 6. Continued

			If entered value is outside of device's frequency capability for the sinusoid waveform	Frequency error for Input $\langle \rangle$: Must be within device range of $\langle \rangle$
	Phase Line Edit	Program will create sinusoid waveform with phase corresponding to user input	If entered value is not numeric	Phase error for Input $\langle \rangle$: Must be numeric
			If entered value is not within phase range	Phase error for Input $\langle \rangle$: Must be within range 0 to 180 degrees
Square Waveforms Window	Low Voltage, High Voltage, and Frequency Line Edits	Similar functionality and errors as sinusoid low voltage, high voltage, and frequency line edits		
	Duty Cycle Line Edit	Program will create square wave with duty cycle corresponding to user input	If entered value is not numeric	Duty Cycle error for Input $\langle \rangle$: Must be numeric
			If entered value is not within duty cycle range	Duty Cycle error for Input $\langle \rangle$: Must be within range 0 to 100%
			If entered value is not within device range	Duty Cycle error for Input $\langle \rangle$: Minimum duty cycle for device = $100\% * W_{min}/period$ and Maximum duty cycle for device = $100\% * (1 - W_{min}/period)$ where $W_{min} = \langle \rangle$
Ramp Waveform Window	Low Voltage, High Voltage, Frequency, and Phase Line Edits	Similar functionality and errors as sinusoid low voltage, high voltage, frequency, and phase line edits		

Table 6. Continued

	Symmetry Line Edit	Program will create ramp wave with symmetry corresponding to user input	If entered value is not numeric	Symmetry error for Input $\langle \rangle$: Must be numeric
			If entered value is not within symmetry range	Symmetry error for Input $\langle \rangle$: Must be within range 0 to 100%
Pulse Waveform Window	Low Voltage, High Voltage, and Phase Line Edits	Similar functionality and errors as sinusoid low voltage, high voltages, and phase line edits		
	Period Line Edit	Produce pulse waveform with period corresponding to user input	If entered value is not numeric	Period error for Input $\langle \rangle$: Must be numeric
			If entered value is not within device range	Period error for Input $\langle \rangle$: Must be within device range $\langle \rangle$
			If entered value is not greater than the sum of the pulse width and the edge time	Period error for Input $\langle \rangle$: Period must be greater than or equal to Pulse Width + (1.6x Edge Time)
	Pulse Width Line Edit	Produce pulse waveform with pulse width corresponding to user input	If entered value is not numeric	Pulse Width error for Input $\langle \rangle$: Must be numeric
			If entered value is not within device range	Pulse Width error for Input $\langle \rangle$: Must be within device range $\langle \rangle$
			If entered value is not greater than Wmin	Pulse Width error for Input $\langle \rangle$: Must be greater than Wmin = $\langle \rangle$
			If entered value is not less than or equal to Period - Wmin	Pulse Width error for Input $\langle \rangle$: Must be less than or equal to Period - Wmin

Table 6. Continued

	Rise Time Line Edit	Produce pulse waveform with rise time corresponding to user input	If entered value is not numeric	Rise Time error for Input <>: Must be numeric
			If entered value is not within device range	Rise Time error for Input <>: Must be within device range <>
	Fall Time Line Edit	Produce pulse waveform with fall time corresponding to user input	If entered value is not numeric	Fall Time error for Input <>: Must be numeric
			If entered value is not within device range	Fall Time error for Input <>: Must be within device range <>
Input Serial Communication Window	String Input Line Edit	Sends string corresponding to user input via serial communication TX line	If entered value exceeds character limit	Input Serial Communication error for String <>: Must be below character limit of <>
	Time Input Line Edit	Sends string at time corresponding to user input via serial communication TX line	If entered value is not numeric	Input Serial Communication error for Time <>: Must be numeric
			If entered value is equal to or less than previous time	Input Serial Communication error for Time <>: Must be greater than previous specified time
			If entered value exceeds time limit or is negative	Input Serial Communication error for Time <>: Must be within time range 0 to 300
Output Feature Measure Window	Measurement Type Dropdown	Record measurement on output channel <> based on user dropdown selection		

Table 6. Continued

	Pass Criteria Selection Dropdown	Generates test report results for channel <> based on the pass criteria dropdown selection		
	Pass Criteria Number Line Edit	Generates test report results for channel <> based on the pass criteria number	If entered number is not numeric	Feature Measure Pass Criteria error for Output <>: Must be numeric
	Minimum Voltage Line Edit	Configures oscilloscope for channel <> to set voltage range with the minimum being configured by the user	If entered value is not numeric	Minimum voltage error for Output <>: Must be numeric
			If entered value is above or equal to maximum voltage value	Minimum voltage error for Output <>: cannot be equal to or exceed maximum voltage value
			If entered value is below device voltage minimum	Minimum voltage error for Output <>: Must be above device minimum of <>
			If entered value is not numeric	Maximum voltage value for Output <> must be numeric
			If entered value is above device voltage maximum	Maximum voltage error for Output <>: Must be below device maximum of <>
Output Serial Communication Window	String Output Line Edit	Generates test report with the pass criteria based on receiving the corresponding string via serial communication RX line	If entered value exceeds character limit	Output Serial Communication error for String <>: Must be below character limit of <>
	Pass Criteria Selection Dropdown	Generates test report results for based on the pass criteria dropdown selection		

Table 6. Continued

	Time Input Line Edit	Generates test report with the pass criteria time based on receiving the corresponding string via serial communication RX line at time specified by user	If entered value is not numeric and the corresponding dropdown selection is "within 5% of time"	Output Serial Communication error for Time <>: Must be numeric
			If entered value is equal to or less than previous time and the corresponding dropdown selection is "within 5% of time"	Output Serial Communication error for Time <>: Must be greater than previous specified time
			If entered value exceeds time limit or is negative and the corresponding dropdown selection is "within 5% of time"	Output Serial Communication error for Time <>: Must be within time range 0 to 300

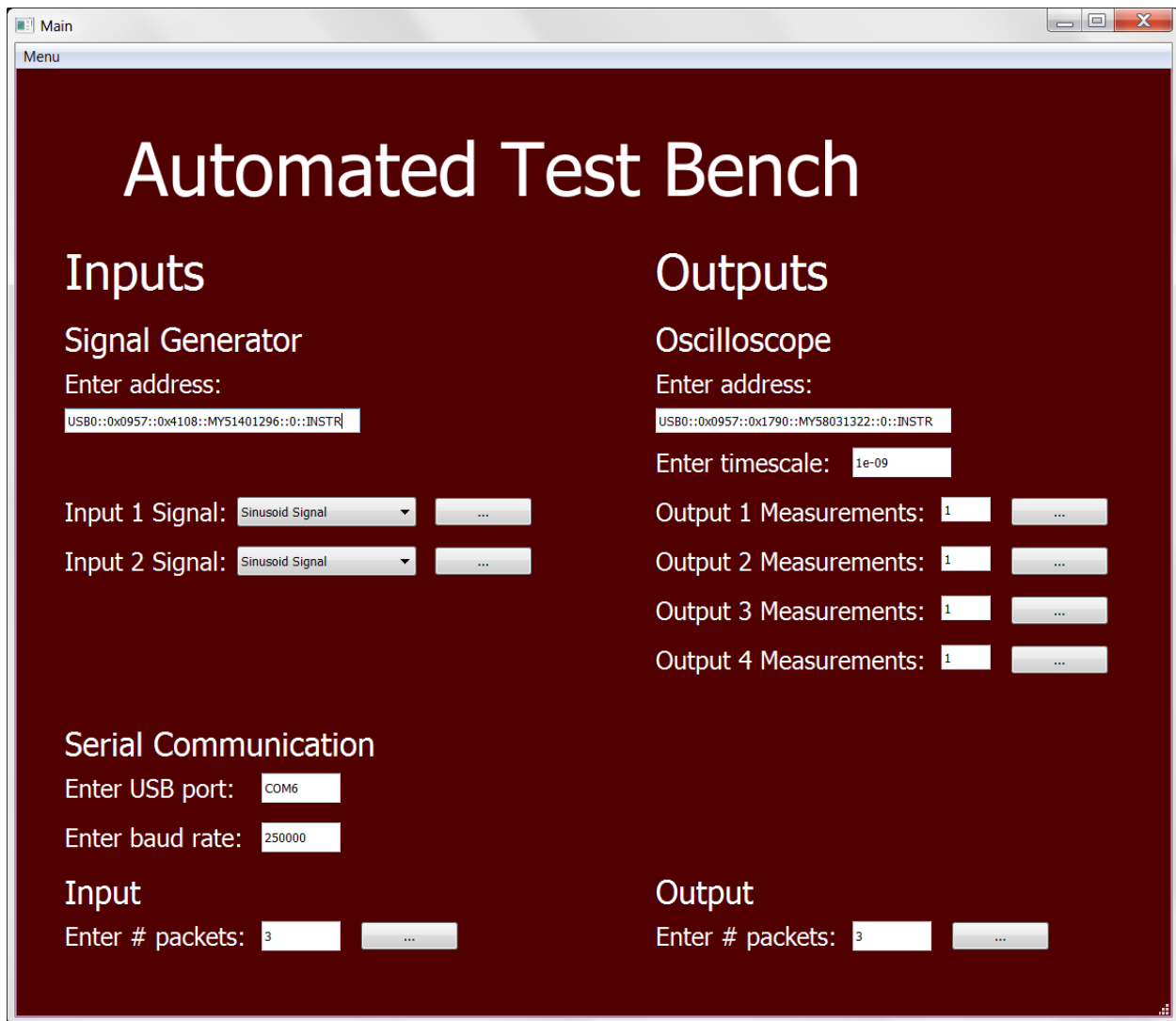


Figure 7. GUI Main Window

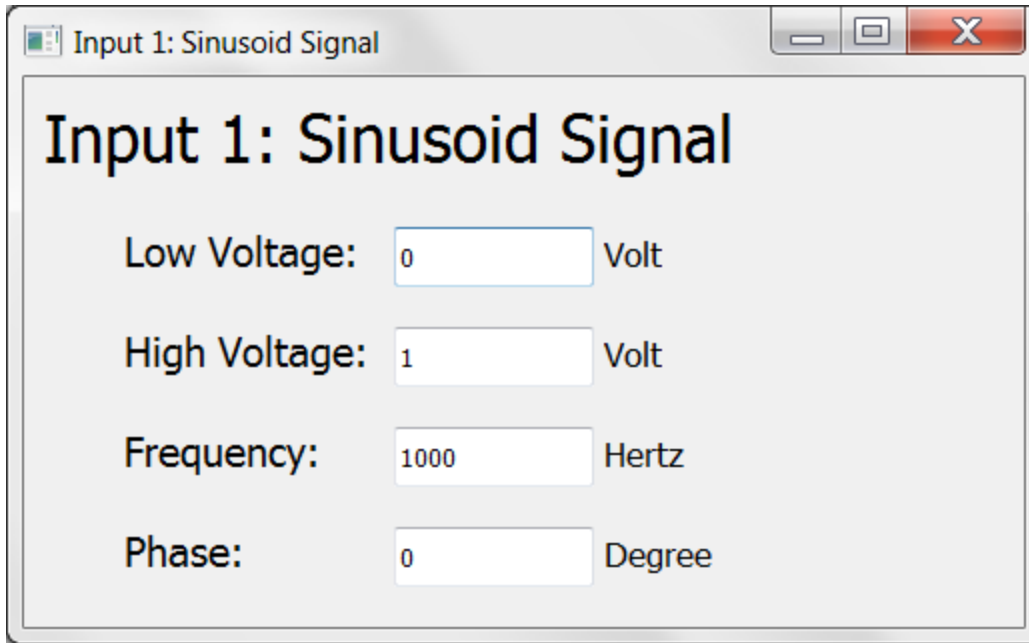


Figure 8. Signal Generator Sinusoid Signal Window

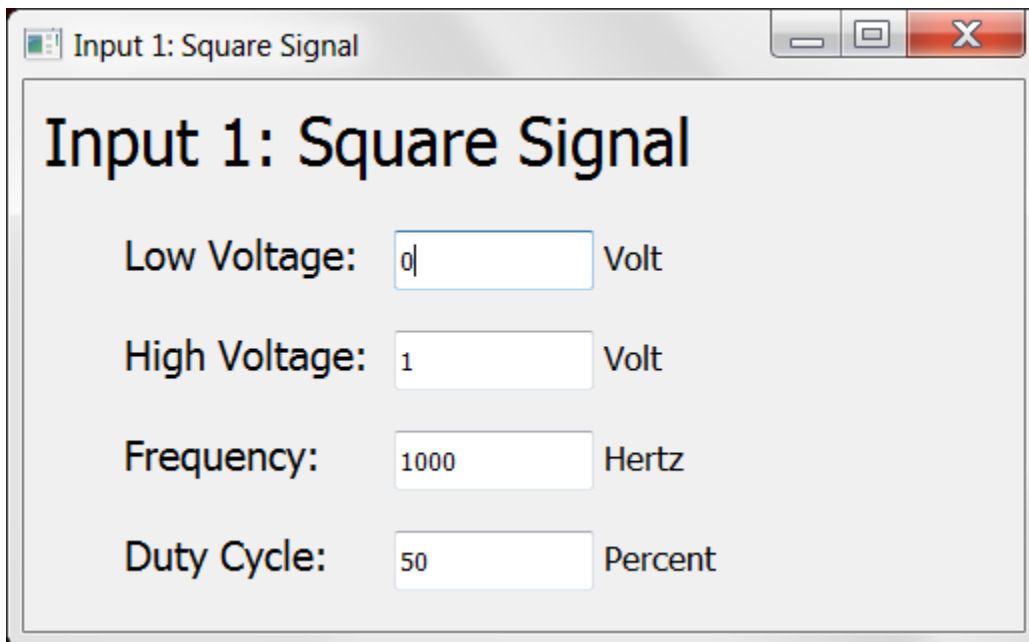


Figure 9. Signal Generator Square Signal Window

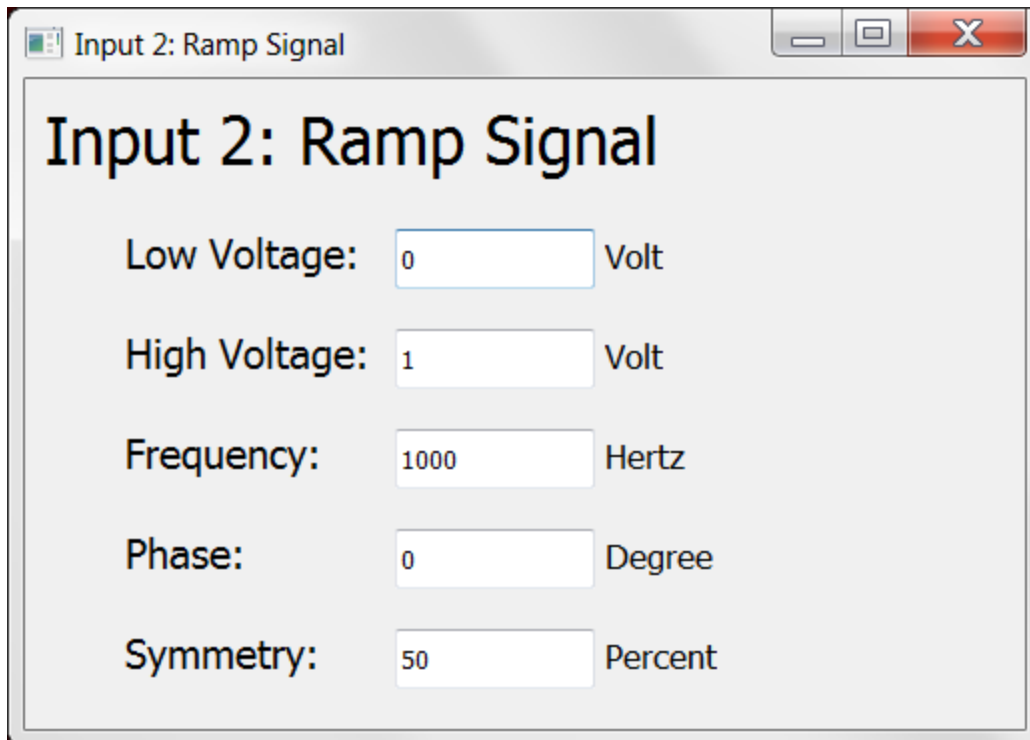


Figure 10. Signal Generator Ramp Signal Window

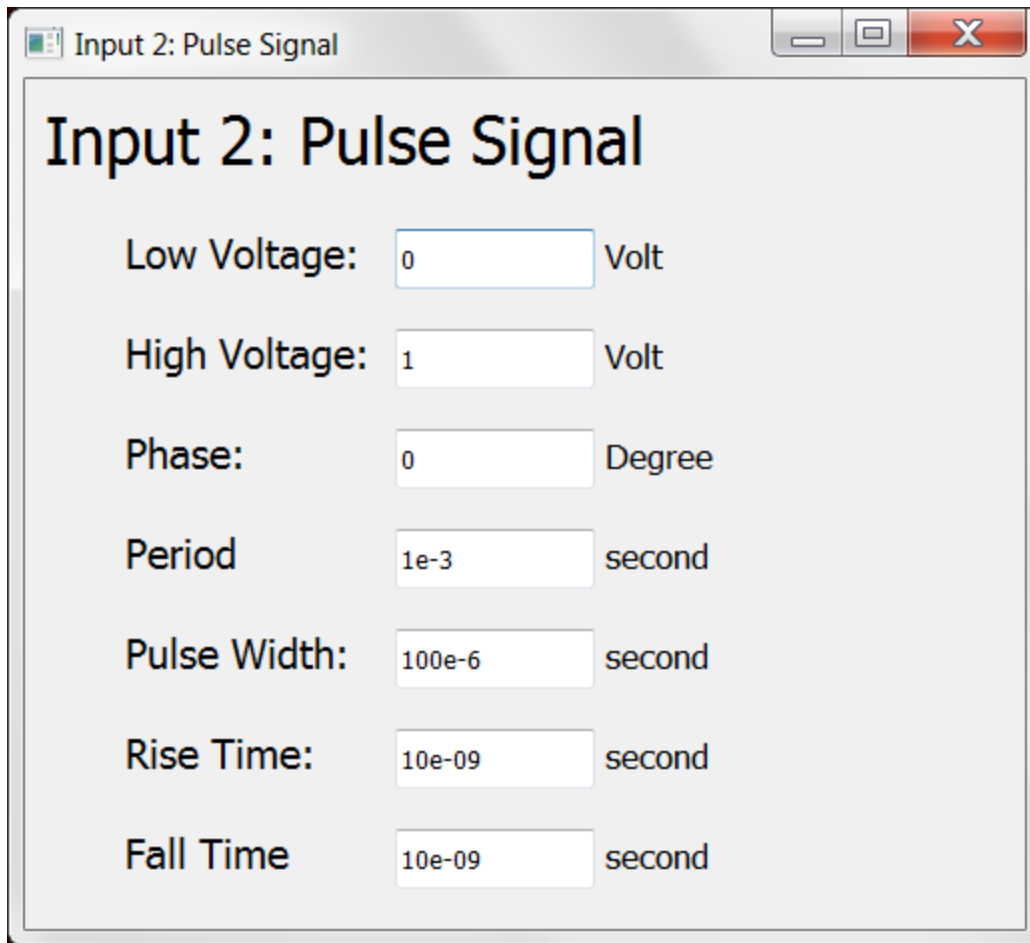


Figure 11. Signal Generator Pulse Signal Window

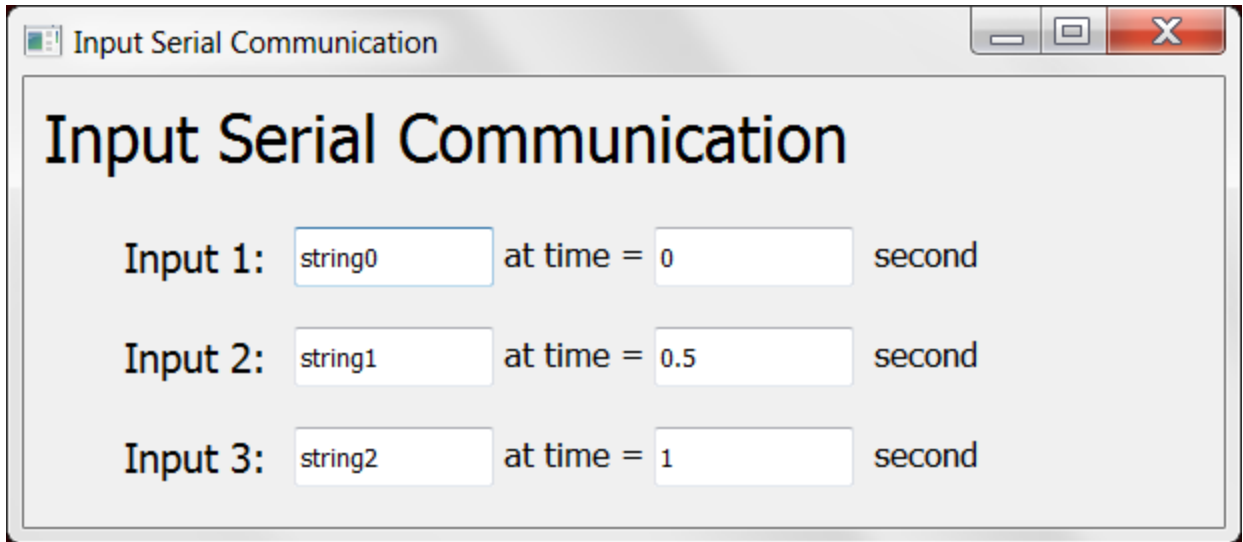


Figure 12. Input Serial Communication Window

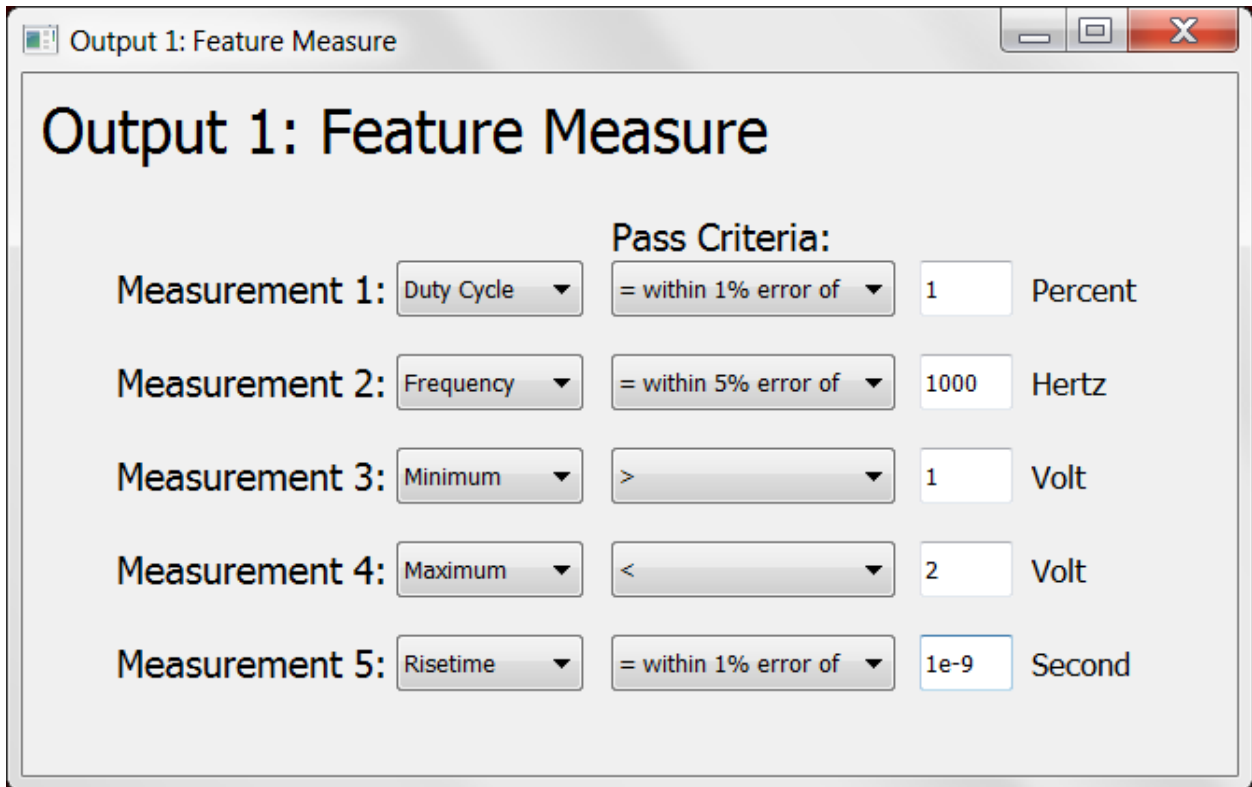


Figure 13. Output Feature Measure Window

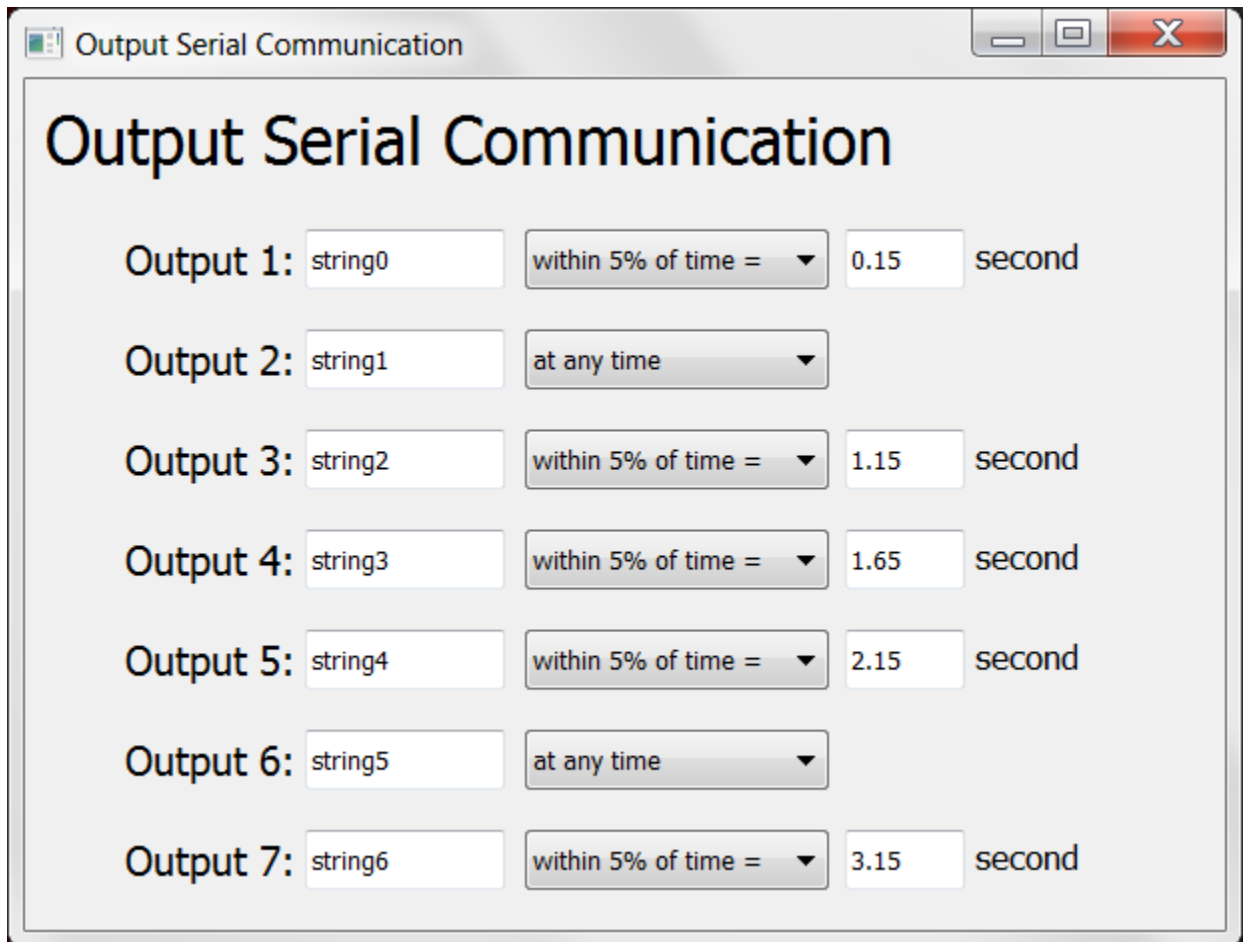
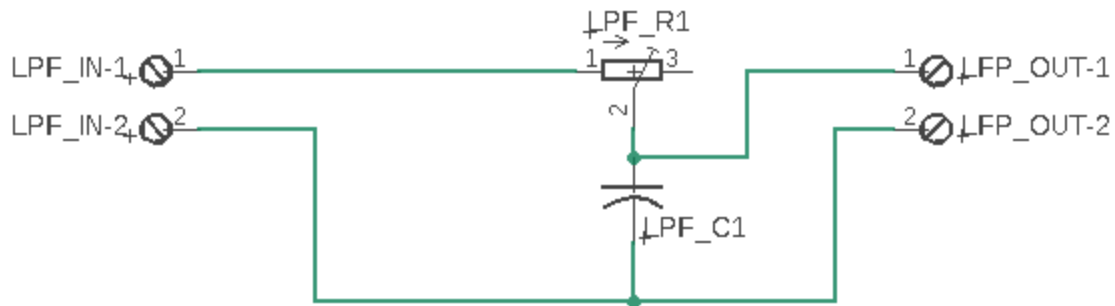


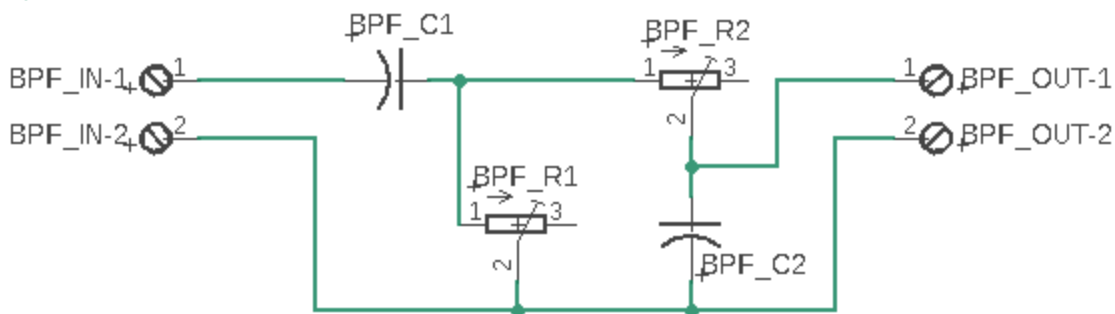
Figure 14. Output Serial Communication Window

Low Pass Filter



+

Band Pass Filter



High Pass Filter

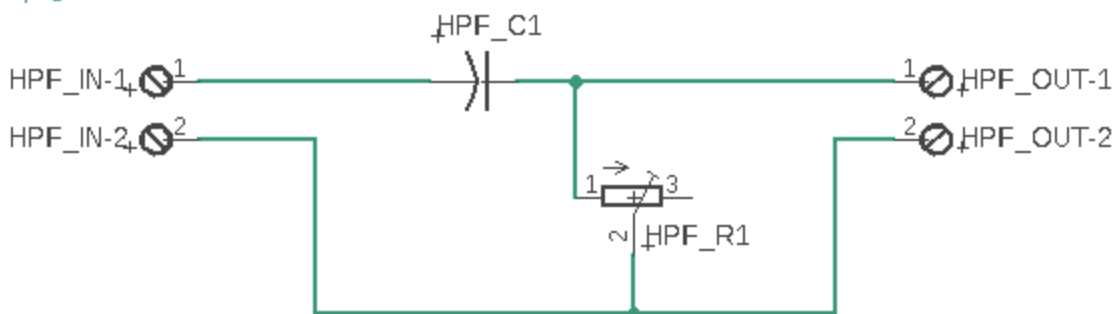


Figure 15. Analog Filter Schematic

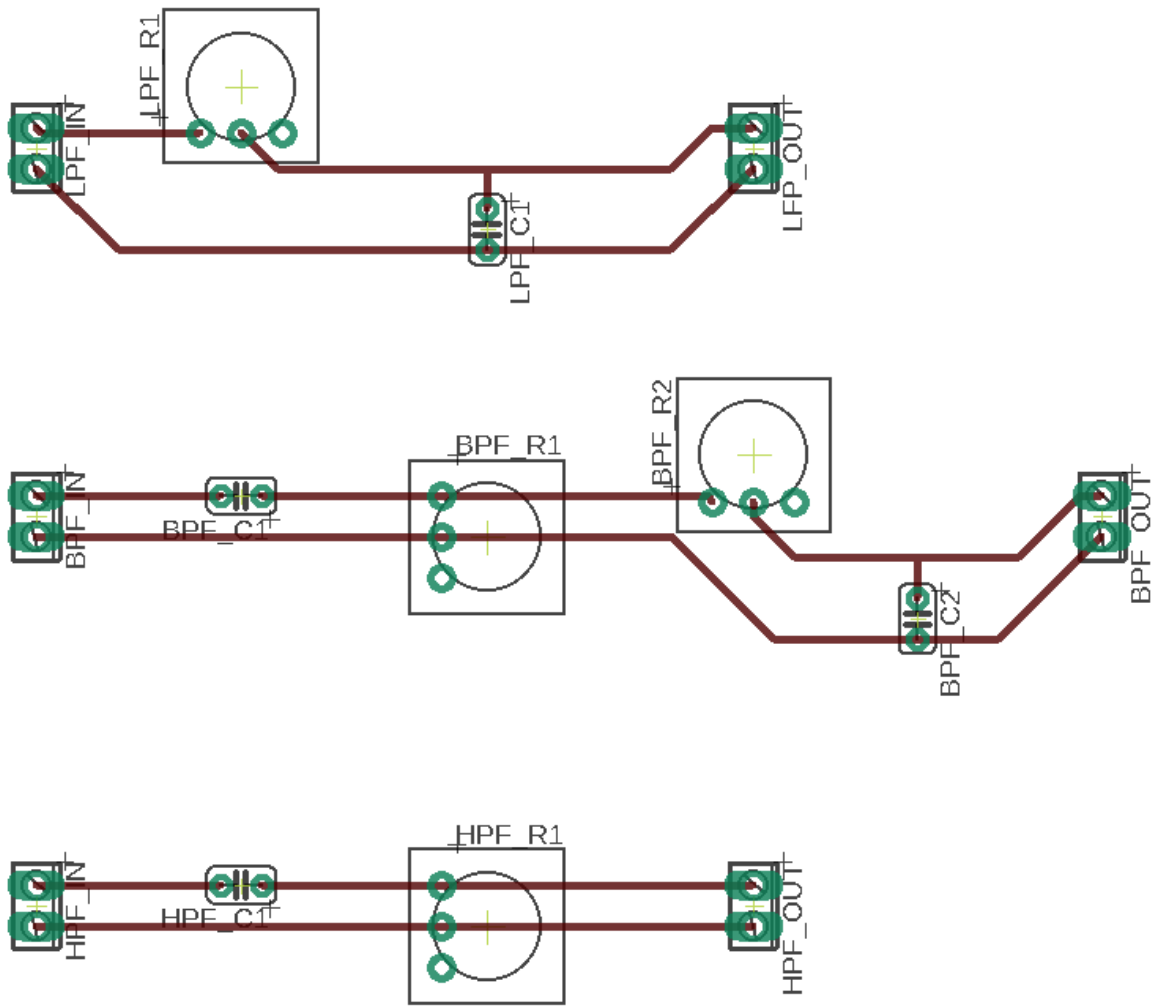


Figure 16. Analog Filter Board View

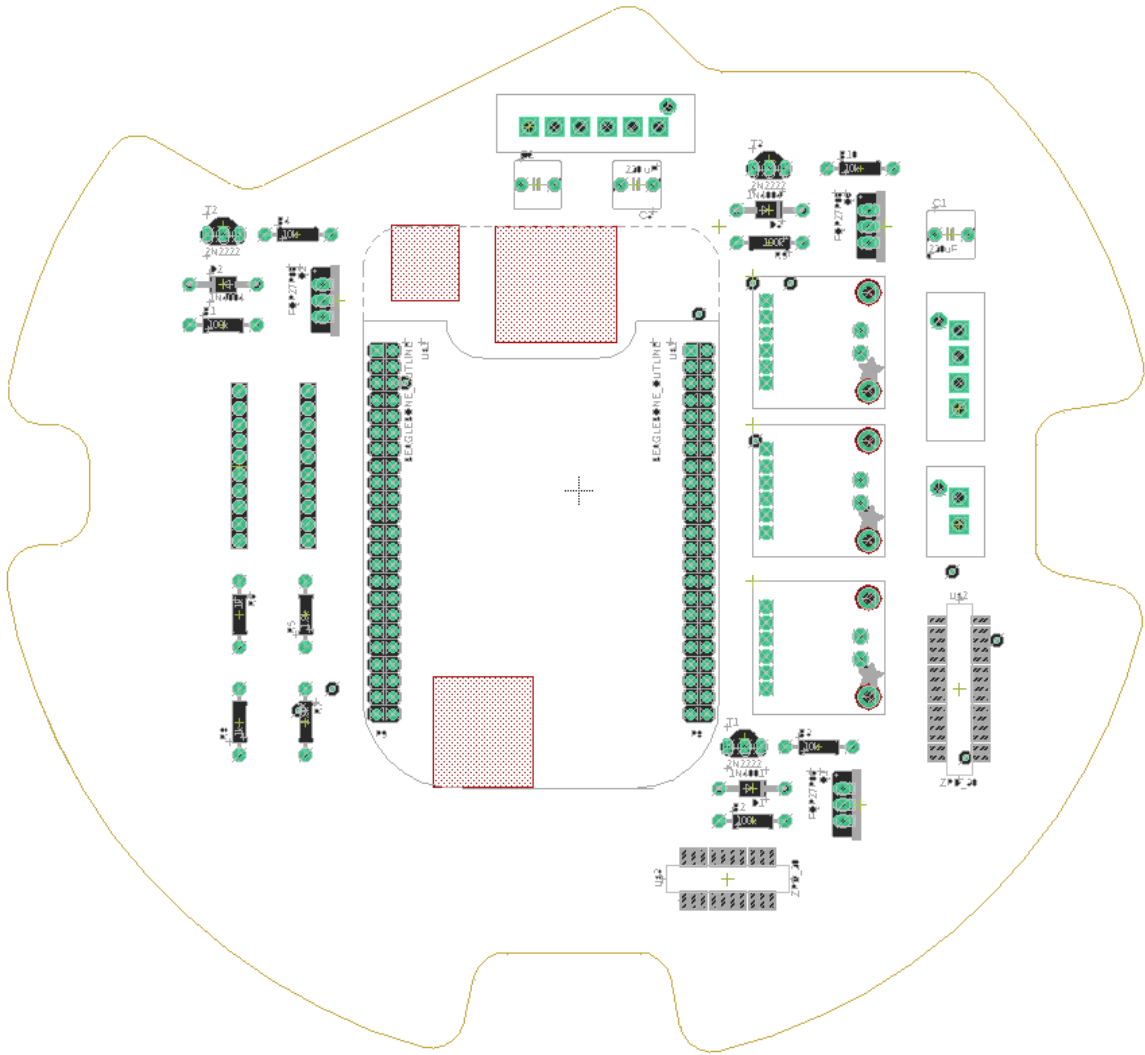


Figure 17. Texas A&M University Sounding Rocketry Team Avionics Board View