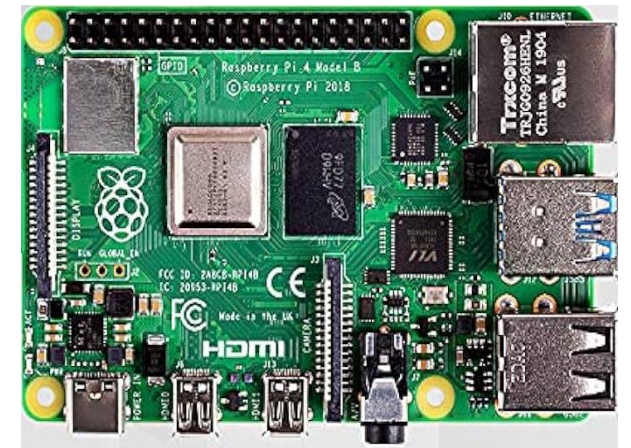
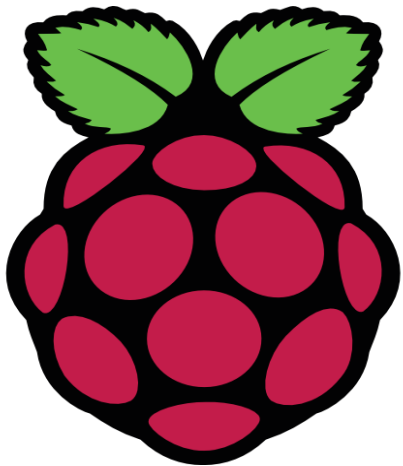


# Avionics Test Engineer Project

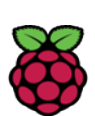
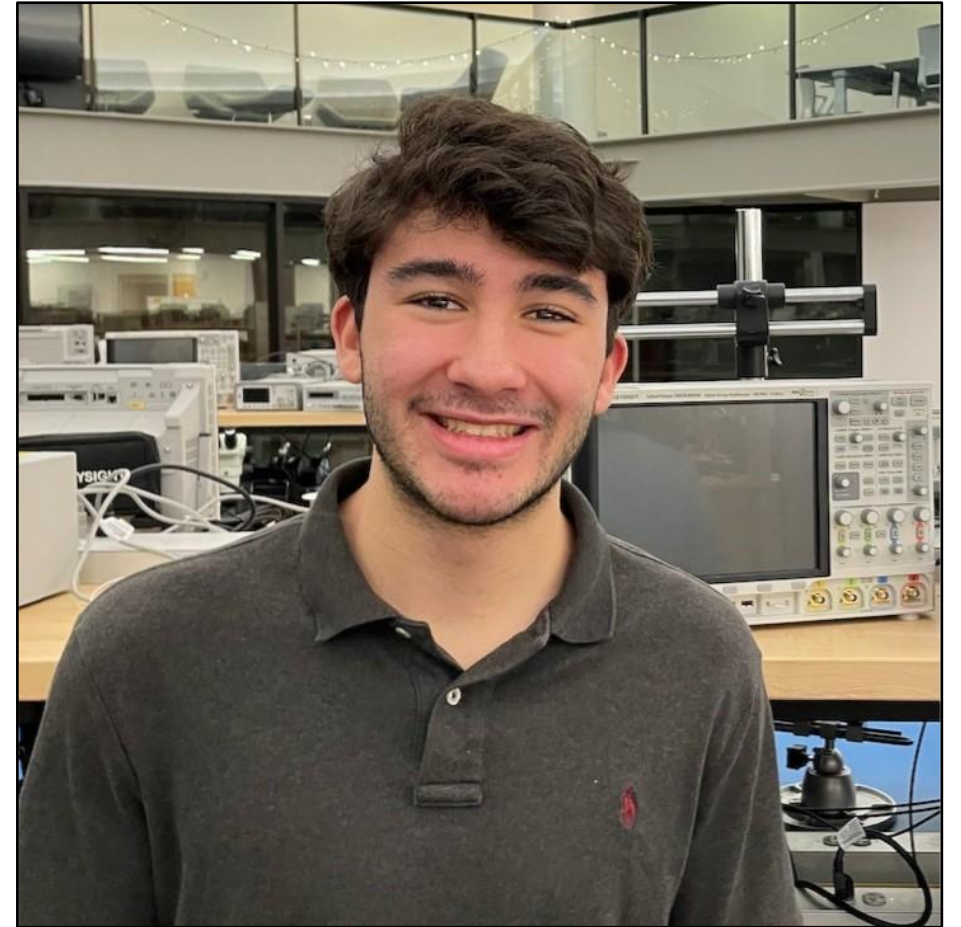
Hardware Test System for a Raspberry Pi 4B

By: Dylan Mira



# Introduction

- Georgia Tech MS ECE
  - *Concentration: Semiconductor Physics and Electromagnetics*
- Started Renewable Energy Startup
  - *Focused in Hydrogen and Generator market*
- Currently Work for Southeast Hydrogen Energy Alliance
  - *Non-profit focused on hydrogen advocacy*
- Previous Work Experience
  - *Includes Test Validation role @ Allegro MicroSystems*



# Table of Contents

## PART 1: Setup & System Design

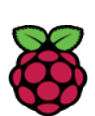
- Bill of Materials **pg. 3 - 7**
- Raspberry Pi Setup
- Hardware Block Diagram
- SW Raspberry Pi Block Diagram
- SW Test Station Block Diagram

## PART 2: Specific Tests & Pass/Fail Criteria

- Processor/Memory Tests **pg. 8 - 12**
- Stereo Audio Output Test
- HDMI Port Test
- GPIO - UART/I2C/SPI Tests
- Gigabit Ethernet Test

## PART 3: Test Execution

- Test Chamber Incorporation **pg. 13**



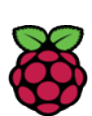
# Bill of Materials

## BOM Hardware List

- 128 GB SD Card
- NI-PXI Chassis
  - Card 1: [Oscilloscope](#) or Cheaper Scope
  - Card 2: [DMM](#)
  - Card 3(optional): [Ethernet Card](#)
  - Card 4(optional): [Logic Analyzer](#)
- [Audio Jack Breakout Board](#)
- [GPIO Breakout Board](#)
- [HDMI Type D Breakout Board](#)
- [Ethernet Breakout Board](#)
- Cables (Need to Meet Temp Spec)

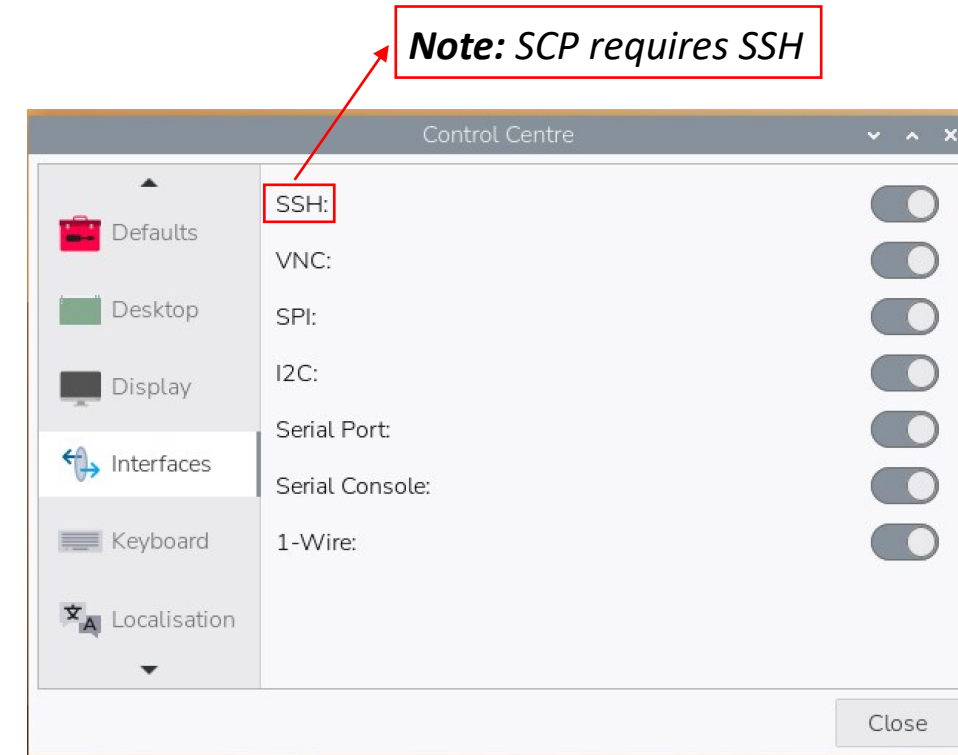
## BOM Software List

- Raspberry Pi Imager
- Raspberry Pi OS
- VNCviewer from Tiger VNC
- SCP Setup Rasp Pi/Test Station Comp
- Stressberry & SysBench
- Memtester
- Iperf3 & Ping
- Libraries include numpy, pygame, scipy



# Raspberry Pi Setup

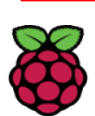
- Download Raspberry Pi Imager
- Select 4b Model, Pi OS (64-bit)
- Customize Board
  - Select Local Network to Run Test
  - Enable SSH & Raspberry Pi Connect
- Activate all “Interfaces” in “Control Centre”
- Select Test Directory Location
- Setup & Activate Python Virtual Environment



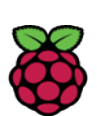
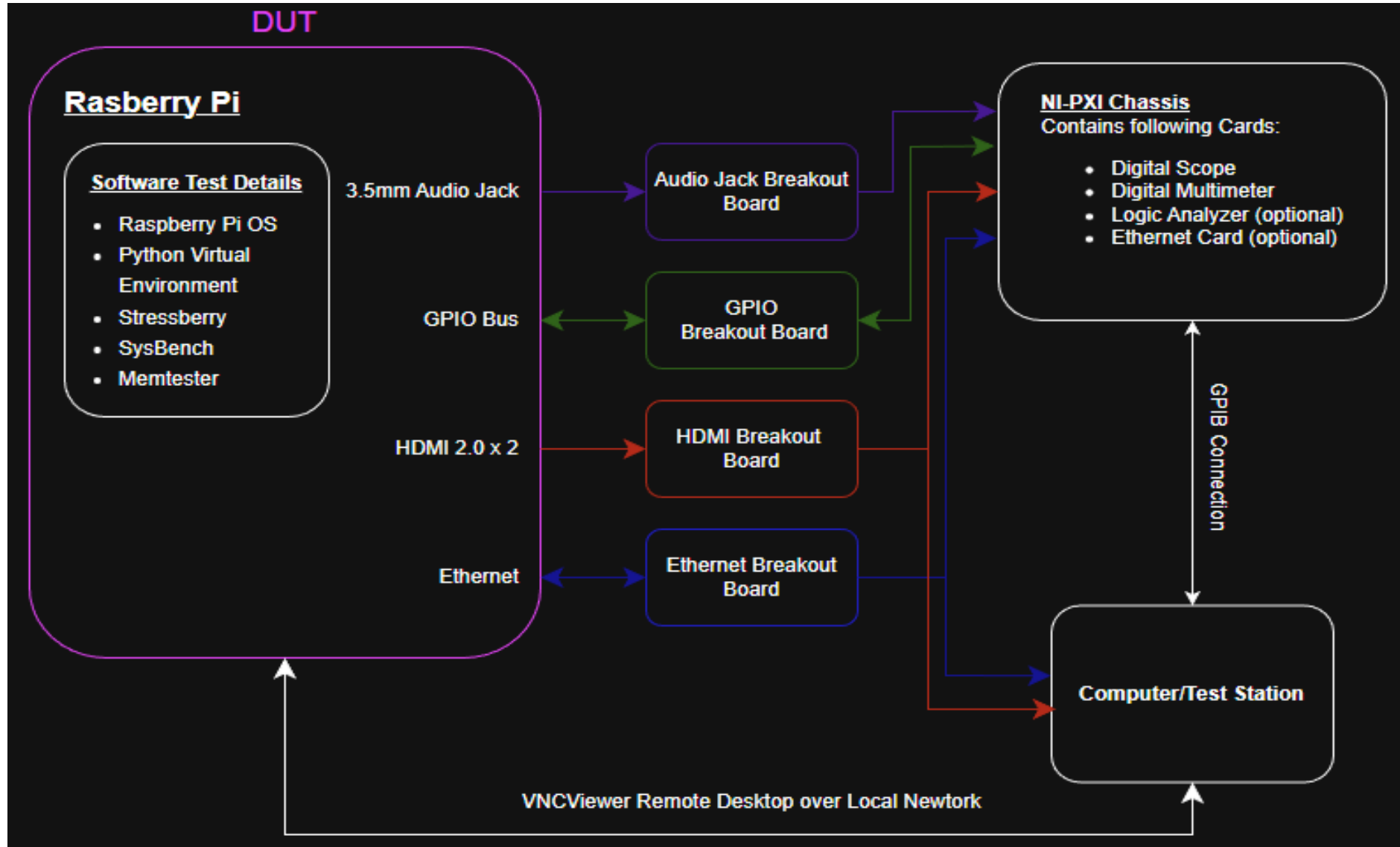
Appears  
when inside  
virtual  
environment

```
dylanmira@dylanmirapi: ~/Reliable-Robotics/Reliable-Robotics
File Edit Tabs Help
dylanmira@dylanmirapi:~ $ cd Reliable-Robotics/Reliable-Robotics/
dylanmira@dylanmirapi:~/Reliable-Robotics/Reliable-Robotics $ source venv/bin/activate
(venv) dylanmira@dylanmirapi:~/Reliable-Robotics/Reliable-Robotics $
```

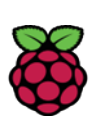
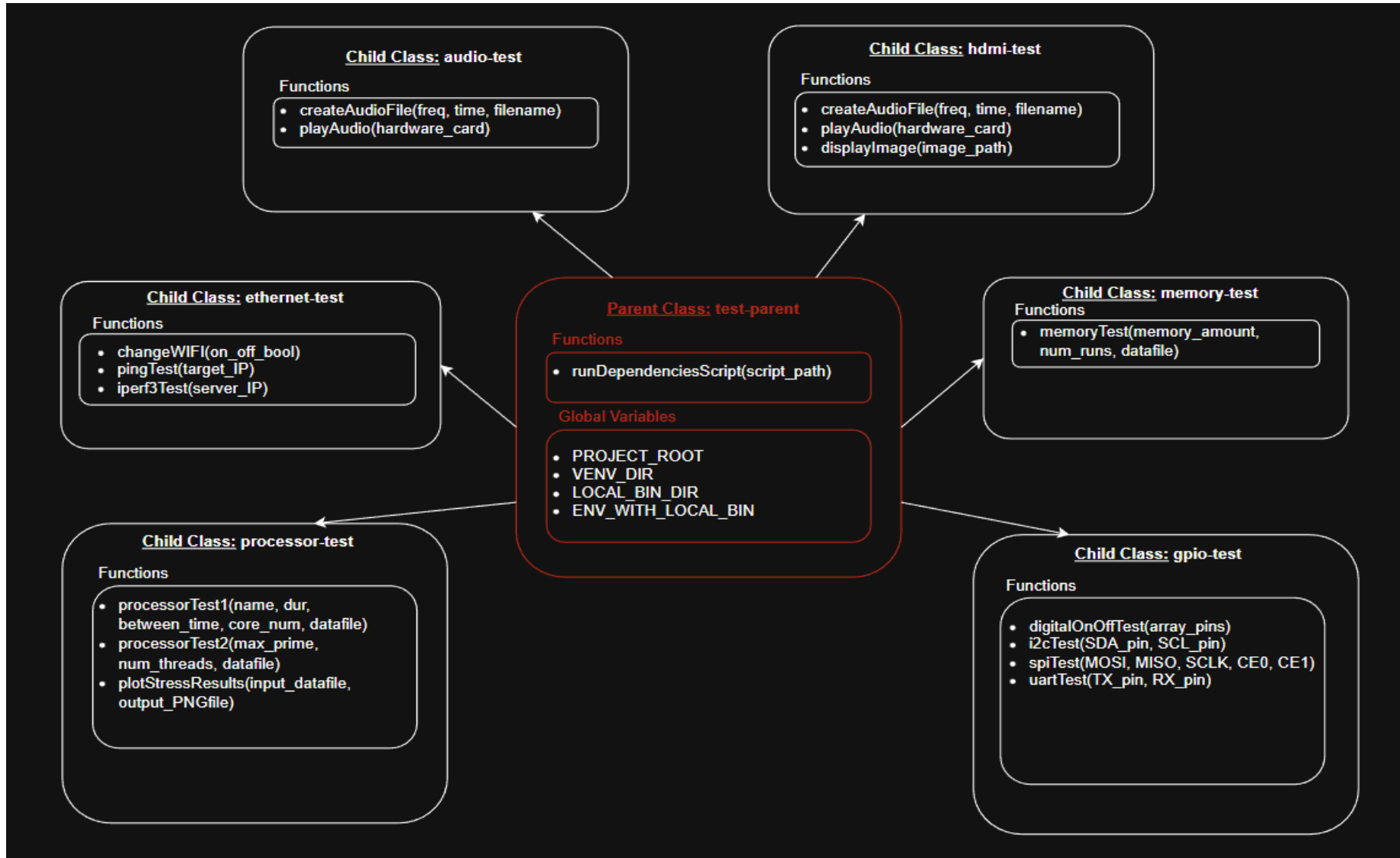
Activation of  
Python Virtual  
Environment



# Hardware Block Diagram



# Software Rasp Pi Block Diagram

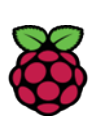


# Software Test Station Computer Block Diagram

## Class: test-execution

### Functions

- sampleScope(address)
- sampleDMM(address)
- sampleLogicA(address)
- manufacturingBaselineTest(pass\_fail\_criteria)
- thermalChamberTest(pass\_fail\_criteria, sample\_rate, temp\_increment, temp\_range)





# Processor Test & Pass/Fail Criteria

- Utilize Stressberry and SysBench
- Assumption: Using Raspberry Pi well below maximum payload
- Failure Criteria
  - CPU Speed  $\leq 7051 * .80$  events/sec
  - Latency  $\geq .57 * 1.20$  ms
  - Core Frequency  $\leq 1798$  or  $\geq 1802$  (when set to 1800)

Chosen Benchmarks

```
--- SYSBENCH CPU TEST RESULTS ---
sysbench 1.0.20 (using system LuaJIT 2.1.1723681758)
```

```
Running the test with following options:
```

```
Number of threads: 4
```

```
Initializing random number generator from current time
```

```
Prime numbers limit: 10000
```

```
Initializing worker threads...
```

```
Threads started!
```

```
CPU speed:
```

```
events per second: 7051.51
```

```
General statistics:
```

```
total time: 10.0005s
```

```
total number of events: 70543
```

```
Latency (ms):
```

```
min: 0.56
```

```
avg: 0.57
```

```
max: 4.68
```

```
95th percentile: 0.57
```

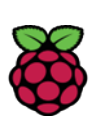
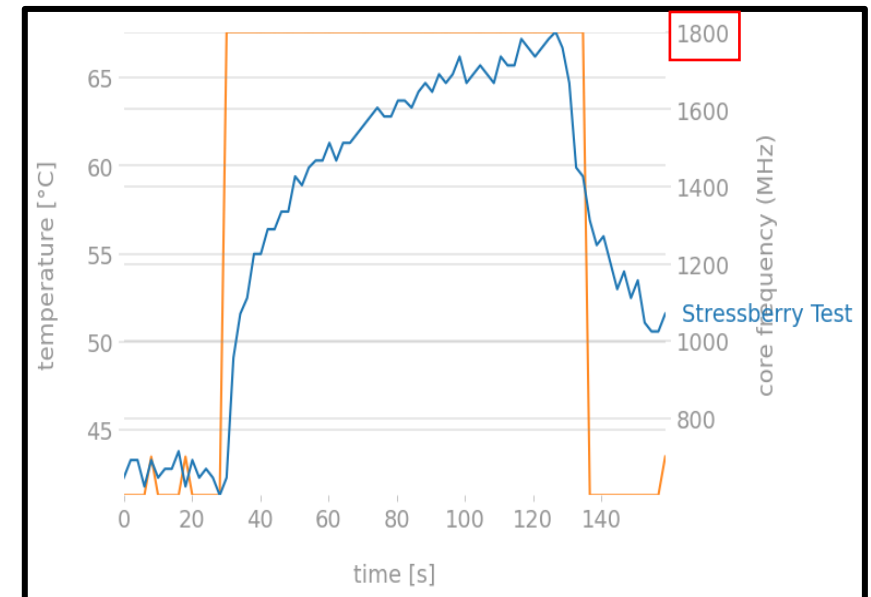
```
sum: 39980.08
```

```
Threads fairness:
```

```
events (avg/stddev): 17635.7500/20.87
```

```
execution time (avg/stddev): 9.9950/0.00
```

CPU Temp  
CPU Frequency



# Memory Test & Pass/Fail Criteria

- Utilize memtester
- Created to catch memory errors due to bad bits
- Failure Criteria
  - Single Data Register Failure
  - Anything but “ok”

```
--- MEMTESTER TEST RESULTS ---
Memory Tested: 2500M, Runs: 2
-----
memtester version 4.7.1 (64-bit)
Copyright (C) 2001-2024 Charles Cazabon.
Licensed under the GNU General Public License version 2 (only).
```

```
pagesize is 4096
pagesizemask is 0xfffffffffff000
want 2500MB (2621440000 bytes)
got 2500MB (2621440000 bytes), trying mlock ...locked.
```

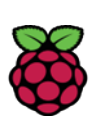
```
Loop 1/2:
```

```
Stuck Address      :      ??????????setting 0????????????testing
```

```
Compare XOR      : ok
Compare SUB      : ok
Compare MUL      : ok
Compare DIV      : ok
Compare OR       : ok
Compare AND      : ok
Sequential Increment: ok
```

*Operations performed on every physical memory address*

*Setting physical mem address to 0*

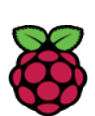
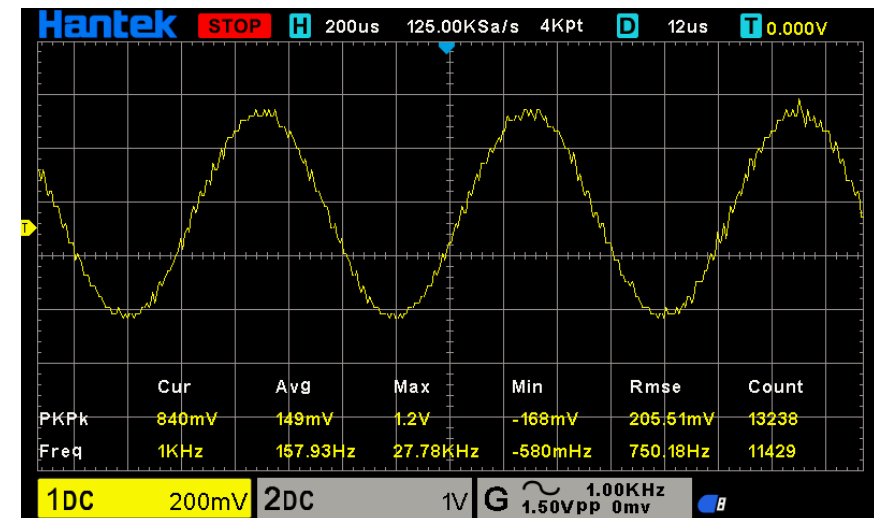
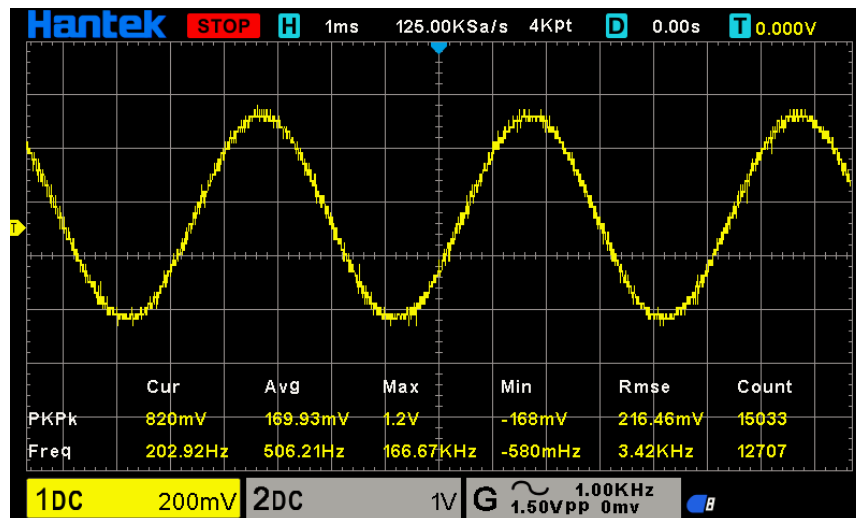
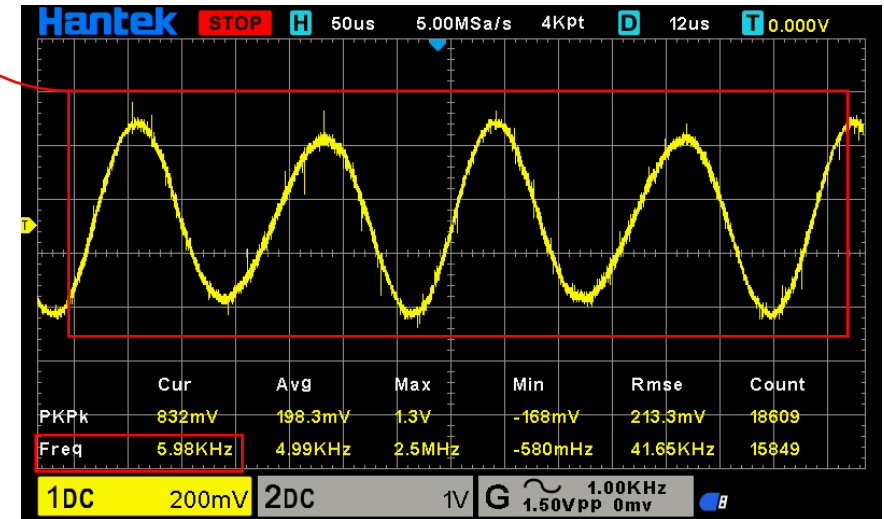


# Audio Test & Pass/Fail Criteria

- Create low, mid, and high frequency audio files
  - Base test off normal speaker design: woofer, mid, and tweeter
- Play audio, read in oscilloscope and compare to WAV file
- Failure Criteria
  - Frequency Error  $\geq |2\%|$
  - Difference from Baseline @ 3x Sample Rate  $\geq |5\%|$

## Higher Freq Distortion

*Note: Assumed distortion is caused by Rasp Pi audio driver. Treated as baseline.*

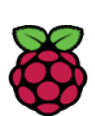


# HDMI Test & Pass/Fail Criteria

- Create low, mid, and high frequency audio files
  - Base test off normal speaker design: woofer, mid, and tweeter
- Use picture with entire color spectrum and intensity
- Failure Criteria
  - Audio Criteria same as on pg. 9
  - Any packet loss
  - Speed of Transmission  $\leq 5\%$



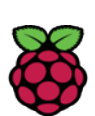
**Note:** Image above has good range of colors and intensity (in form of shade)



# GPIO Test & Pass/Fail Criteria

- Digital Input & Output Test
- I2C Test
  - Utilize i2cdetect to scan the bus & verify functionality of Clock and Data Line
  - Connect to Scope/Logic Analyzer
- SPI Test
  - Connect MOSI to MISO and utilize spidev
  - Connect Scope/Logic Analyzer to CLK, MOSI, and MISO
  - Connect Multimeter to Chip Enable/Select
- UART
  - Connect TX and RX to each other
  - Check to make sure data transmitted matched received
- Power Pins
  - Connect 5V & 3.3V Output Pins to Multimeter

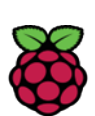
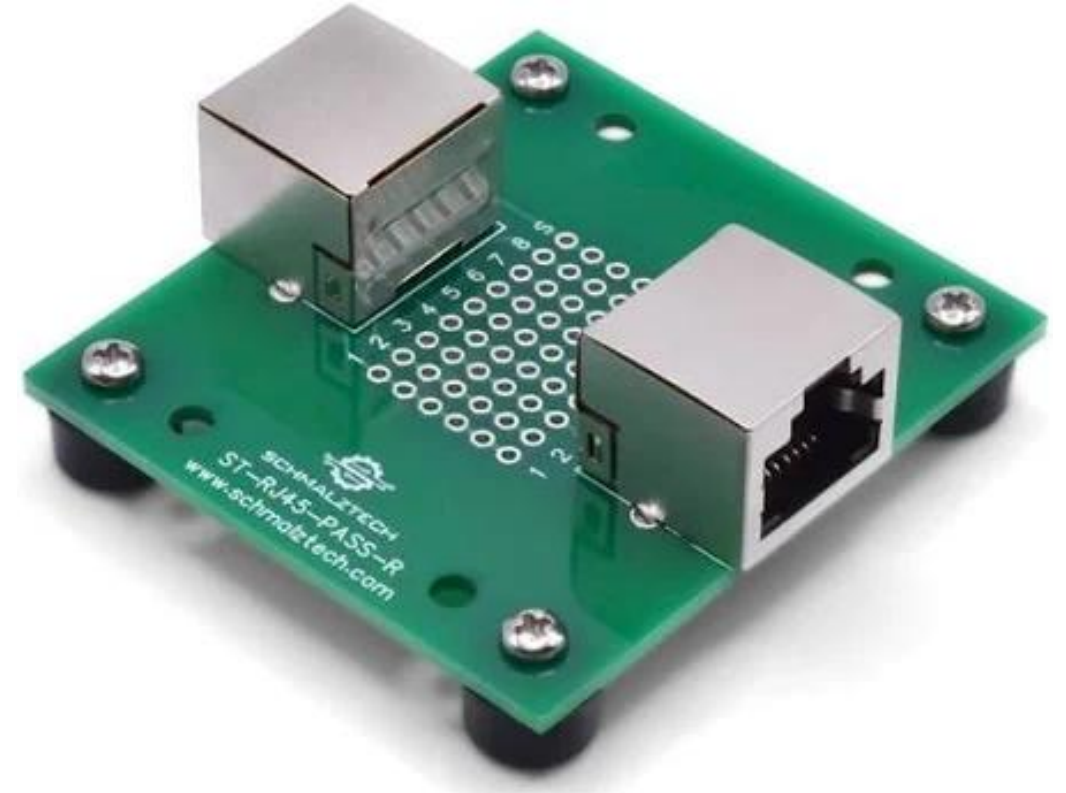
Alternate Function				Alternate Function
	3.3V PWR	1		2 5V PWR
I2C1 SDA	GPIO 2	3		4 5V PWR
I2C1 SCL	GPIO 3	5		6 GND
	GPIO 4	7		8 UART0 TX
	GND	9		10 UART0 RX
	GPIO 17	11		12 GPIO 18
	GPIO 27	13		14 GND
	GPIO 22	15		16 GPIO 23
	3.3V PWR	17		18 GPIO 24
SPI0 MOSI	GPIO 10	19		20 GND
SPI0 MISO	GPIO 9	21		22 GPIO 25
SPI0 SCLK	GPIO 11	23		24 GPIO 8
	GND	25		26 GPIO 7
	Reserved	27		28 Reserved
	GPIO 5	29		30 GND
	GPIO 6	31		32 GPIO 12
	GPIO 13	33		34 GND
SPI1 MISO	GPIO 19	35		36 GPIO 16
	GPIO 26	37		38 GPIO 20
	GND	39		40 GPIO 21



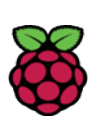
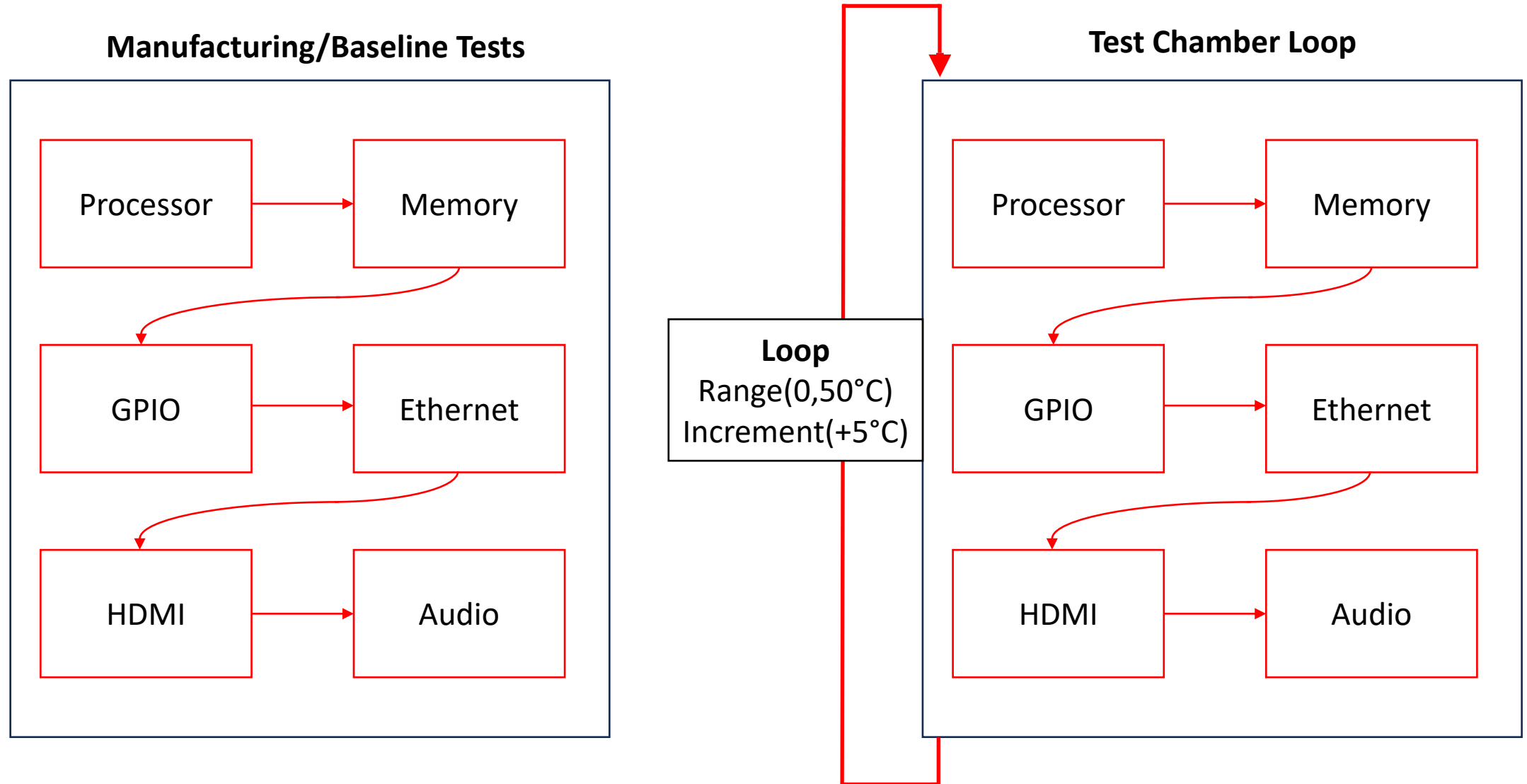


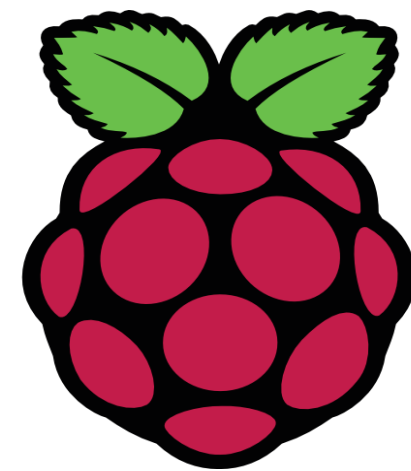
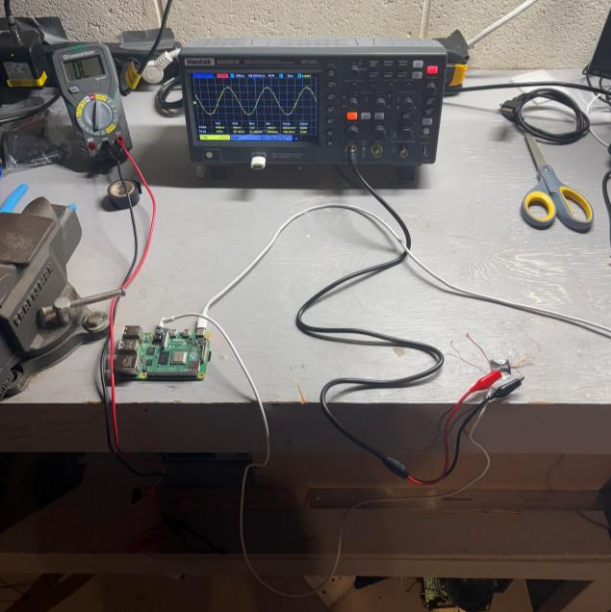
# Ethernet Test & Pass/Fail Criteria

- Utilize iperf3 & ping to test Ethernet
  - Iperf3 test highlight bandwidth & speed
  - ping test highlights packet loss for basic connection
- Utilize logic analyzer/oscilloscope
  - Connect 8 wire test points from breakout board to logic analyzer or oscilloscope to compare
- Pass/Fail Criteria
  - Zero packet data loss
  - 5% within max bandwidth & speed



# Test Order & Loop in Temp Chamber





Thank You!

