This is a list of how each of the Chimera Debugger tests are performed.

**Notes:**

IP address-

Tester Chimera must be 10.1.1.3

DUT Chimera must be 10.1.1.2

**Known issues: More on errors in section of given protocol**

These are known issues that occasionally occur.

Connection – Running test too soon after powerup/reset

"Error: Chimera did not respond. Is it connected?"

Dallas – If CAN communication is established prior to Dallas

“Error: didn’t detect any devices”

I2C – Must establish CAN communication prior to I2C test

“Error: didn’t detect any devices”

Revcomm – Unclear what causes error, seems to occur with Dallas error

"Error: Message was empty!”

Note on connection error - "Error: Chimera did not respond. Is it connected?"

This error occurs when connecting a different chimera and running the test for the first time. This is due to not waiting long enough for the pc to establish a connection. Once this error occurs, simply wait a second, you will see the light on the Ethernet port flash a few times, then run the test again. If the problem proceeds, connect using DevIO before running test.

**Dallas**

Dallas line (D1W) is connected directly to a 1Wire EEprom. This test verifies the ability to write to and read from a 1Wire chip on the debugger board.

Test Steps:

1. Get Dallas ID to establish connection between the DUT chimera and the Dallas chip on the debug board.
2. Write to the scratch pad
3. Read back what was written
4. Check data read is the same as data written

Note on detection error - "Error: didn't detect any devices"

Check Dallas test will return "Error: didn't detect any devices" if CAN communication has been established at any point prior to Dallas test. Not sure what is causing this, but possibly may be something in firmware. To get around this, Dallas test is performed first. This as well as both chimeras being reset at the end of all tests. If problem proceeds, simply reset the both chimeras and wait a bit before running test again.

**SingleWire1 / SingleWire2**

SingleWire1 and SingleWire2 are switched between by a series of relays to connect both to SingleWire1 on the tester chimera. This is to give the ability to test DAC A and DAC B with a limited number of OpenCollectors on the tester for switching the debug boards’ components.

* **Pullup Resistors**

Tested Hardware:

This test primarily tests the functionality of U14, the mux for the RevComm pull-up resistors.

During this test, the DUT chimera switches among the RevComm pullup resistors and reads the given voltage on the ADC 1Wire1 line. This voltage, multiplied by the gain of 5.42 gives the voltage on the Pullup1 line. By switching in a 10k pulldown resistor from the debug board, a voltage divider is formed between Vunreg, the selected pullup resistor on the chimera, the voltage read from Pullup1, and the 10K pulldown resistor. Through this voltage division, the resistance of the pullup resistor can be calculated. During this test, the DUT is isolated from the tester.

Test Steps:

1. Disconnect DUT 1Wire1 from Tester 1Wire1 using tester OC1
2. Connect 10K pulldown resistor to DUT 1Wire1 for desired voltage using tester RE1 and RE2 to select resistor
3. Select desired DUT Pullup resistor to create voltage divider
4. Read voltage on DUT ADC1Wire1
5. Calculate voltage on DUT Pullup1 by multiplying by gain factor of 5.42
6. Pullup Resistance calculated using voltage division formula
7. Check if resistance is within set tolerance
8. Repeat steps 3-6 with all possible pullup resistor selections
9. Reset tester Open collectors and GPIO

Note:

Calculated resistance values from test may be off slightly as the ADC gain and offset that are coded in may not be exact.

* **RevComm (SingleWire1)**

RevComm test checks for ability to send and receive a message over the SingleWire1 line. Here we are using SingleWire1 to send RevComm message over Lin1.

Test Steps:

1. Short DUT 1Wire1 to Tester 1Wire1 using tester open collector OC3 to switch on desired debug board relay
2. Send Transmit data over DUT 1Wire1 (Revcomm over Lin)
3. Receive response from Tester
4. Reset tester open collectors
5. Verify data sent is data received

Note on detection error - "Error: Message was empty!”

It is unclear as to what causes this error. It appears to be something similar to Dallas errors with running the test too soon after power up or reset, and/or something with CAN similar to Dallas. If this error happens, simply run test again. This error appears to disappear after running the test a second time.

* **LIN (SingleWire2)**

SingleWire2 test checks for ability to send and receive message over the SingleWire2 line using LIN.

Test Steps:

1. Short DUT 1Wire2 to Tester 1Wire1 using tester open collectors OC1 and OC2 to switch on desired debug board relays
2. Send Transmit data over DUT 1Wire1 (Revcomm over Lin)
3. Receive response from Tester
4. Reset tester open collectors
5. Verify data sent is data recieved

**UART**

**Dedicated HS CAN**

**Multiplexed CAN**

UART, HS CAN, and Multiplexed CAN tests check ability to send and receive message over the given line. DUT sends a message, then reads a message sent from the tester Chimera. The lines run directly between the two chimera boards.

**AUX IO**

* **Switches**

Switches are not in use on the chimera. The firmware is yet to be written to allow testing functionality.

* **LEDs**

Test turns on all LEDs which we have access to (D5, D15, D14, and D16) then asks user to visually check that the 8 main LEDS (the previous four plus D19, D23, D17, and D18) are all working.

* **Open Collectors**

Each open collector tied to 2 REout pins each, which then are pulled up by a pullup resistor to 3.3v. With the tested Open Collector ON, the corresponding REout pin should read low to pass. Else, with the tested Open Collector OFF, the corresponding REout pin should read high to pass. With that, we are able to test the ability of each OpenCollector to turn on and pull to ground or to be off and let the node high.

Test Steps:

1. Switch OFF all open collectors
2. Switch ON open collector being tested
3. Read corresponding GPIO REout (should be low to pass)
4. Switch OFF open collector being tested
5. Read corresponding GPIO REout (should be high to pass)
6. Repeat steps 2-5 with all open collectors

* **GPIO (REout)**

The GPIO test is testing the individual ability of each REout pin on the DUT chimera to read a logic high and low. The setup is for every two REout pins, there is a pullup resistor to 3.3v as well as an OpenCollector to pull the node to ground.

Test Steps:

1. Switch OFF all open collectors
2. Read all GPIO REout pins (should be high to pass)
3. Switch ON all open collectors
4. Read all GPIO REout pins (should be low to pass)
5. Test pass/fail

**Signal Conversion**

* **ADCs**

The ADC line is connected to the AIN line on the Chimera. AIN1 and AIN2 are tied together and connected to a 1.5 voltage regulating IC. AIN1 and AIN2 are simply reading the 1.5v supplied by the IC.

Test Steps:

1. Read voltage on AIN1 and AIN2 (should be 1.5v)
2. Compare value read with upper and lower limits

* **DAC**

The DAC is tested by changing the upper and lower voltage limits of a signal received from the tester Chimera, while setting the DAC level in between the limits manually. If the signal was read, the DAC is setting properly. This is done for DAC A and B at two separate voltage ranges, which are approximately as follows:

Upper test: 13v to 9v

Lower test: 7.3v to 5.3v

Test Steps:

Upper voltage range

1. Connect chimeras with auto threshold OFF, setting DAC to 11v
2. All tester open collectors OFF, assuring DUT 1Wire1 is connected to tester 1Wire1 with a 3k resistor in between to allow a voltage differential to be established
3. Change tester GPIO to connect 10k pulldown from debugging board mux to 1Wire1, giving the upper voltage range
4. Send data from tester over 1Wire1 connection, and read on DUT
5. If data is received, the DAC was set properly and is working

Lower voltage range

1. ReConnect chimeras with auto threshold OFF, setting DAC to 6.2v
2. All tester open collectors OFF, assuring DUT 1Wire1 is connected to tester 1Wire1 with a 3k resistor in between to allow a voltage differential to be established
3. Change tester GPIO to connect 1k pulldown from debugging board mux to 1Wire1, giving the lower voltage range
4. Send data from tester over 1Wire1 connection, and read on DUT
5. If data is received, the DAC was set properly and is working
6. Repeat all steps for DAC B, turning on OC1 to connect DUT 1Wire2 to the tester 1Wire1 with a 3k resistor between

**Voltage IO**

Due to a limited number of Analog inputs (AIN1 and 2), the three following tests had to be tied together. With this, if all three fail it may only be one of the three failing.

* **Ignition**
* **Battery**
* **Reverse**

These voltages are tested by setting all voltages OFF (open) and then setting the voltage being tested. This voltage is then lowered by a voltage divider and read using the tester AIN1

Test Steps:

1. IGN and BAT switched OFF and REV to OPEN/COMM
2. Voltage read by tester, should read 0v
3. IGN switched ON
4. Voltage read by tester, should read slightly below Vunreg
5. IGN switched OFF
6. Voltage read by tester, should read 0v
7. BAT switched ON
8. Voltage read by tester, should read slightly below Vunreg
9. BAT switched OFF
10. Voltage read by tester, should read 0v
11. REV switched OFF
12. Voltage read by tester, should read 0v
13. REV switched ON
14. Voltage read by tester, should read slightly below Vunreg

Note:

The following is a note on how the given line on the chimera is changed according to the state of V\_REV. Compare with chimera schematic.

V\_REV: OFF

V\_REV\_ON\_HIGH = 0;

V\_REV\_ON\_LOW = 1;

V\_REV: ON

V\_REV\_ON\_HIGH = 1;

V\_REV\_ON\_LOW = 0;

V\_REV: OPEN/COMM

V\_REV\_ON\_HIGH = 0;

V\_REV\_ON\_LOW = 0;

**I2C**

The I2C lines are connected directly to an I2C EEprom. This test verifies the ability to write to and read from this chip on the debugger board.

Test Steps:

1. Get Dallas ID to establish connection between the DUT chimera and the Dallas chip on the debug board.
2. Write to the scratch pad
3. Read back what was written
4. Check data read is the same as data written

Note on detection error - "Error: didn't detect any devices"

Check Dallas test will return "Error: didn't detect any devices" if CAN communication has been established at any point prior to Dallas test. Not sure what is causing this, but possibly may be something in firmware. To get around this, Dallas test is performed first. This as well as both chimeras being reset at the end of all tests. If problem proceeds, simply reset the both chimeras and wait a bit before running test again.

**Other Protocols**

* **SPI**

SPI protocol is not currently used by the chimera. The firmware has not been written to allow testing functionality.