San Diego State University

College of Engineering, Department of Electrical and Computer Engineering
EE/COMPE 491 Senior Design

System Description, Requirements, and Validations

Team SD Cables

ECE Team 2 – Masimo Cable Tester

ECE Members:
Jonatan Guerra, Kaitlyn Huynh, Phillip Mcleroy, Quincy Nguyen, and Ryan Walb

Sponsor:

Masimo

Jonatan Guerra
COMPE

Kaitlyn Huynh
EE

Phillip Mcleroy
EE

Quincy Nguyen
COMPE

Ryan Walb
EE

Executive Summary:

This document describes SD Cables sponsored project: “Automated multi conductor cable break-short detector,” hereinafter referred to as the “Cable Tester.” The Cable Tester continuously monitors a multi-conductor cable that is being subjected to destructive mechanical testing, measuring resistance and detecting any breaks or shorts. The team is developing a Printed Circuit Board (PCB) that will have serval inputs and outputs (I/O), including SD card port, 20-pin connector, power switch, DC power jack, a debugger, banana jack, and a Human Machine Interface (HMI). For the HMI, the team will be using a Liquid Crystal Display (LCD) to display the Graphic User Interface (GUI), which is shown for use case. We will be using the AVR128DB64 microcontroller (MCU) help to drive our device.

This document serves to describe what functionality our device must provide. Furthermore, it covers a comprehensive summary of the use case illustrating how a user would connect wires to the bend cycle tester, operate the graphic user interface (GUI), and then extract data about the wire’s usability. This is accompanied by a physical description of the Cable Tester, showing the PCB layout, the inputs/outputs of the device, and the GUI display with product dimensions. The next section describes how the system shall be implemented using components, such as a PCB, a microcontroller, a 20-pin connector, and an SD card. This is followed by a table of specifications which describe all performance aspects, such as cable details, measurement accuracy, budget, and PCB components. The final section describes a systems test that verify that the Cable Tester meets the goals specified in this document.

Functionality:

In terms of the functionality, the Cable Tester is a device that must test and analyze the durability and integrity of wires. The device must extract data from the wire connected to characterize it and monitor the changes in the wire that may indicate a short or a break in the conductors while the wire is subjected to a destructive mechanical test, such as a bend cycle test. All data must be captured in an SD card inserted into the device for post-test analysis. Currently, the Cable Tester that Masimo possesses operates with preset assumptions about the wire it will attach. The Cable Tester that we design must be universal to all wires, more specifically, it must characterize the cable regardless of what wire is connected and then proceed with the tests. The universalizability of the Cable Tester will assist Masimo, and potentially other companies, in the development of electrical systems by predetermining faults in the wires before it will be implemented into the system, which prevents future technical issues.

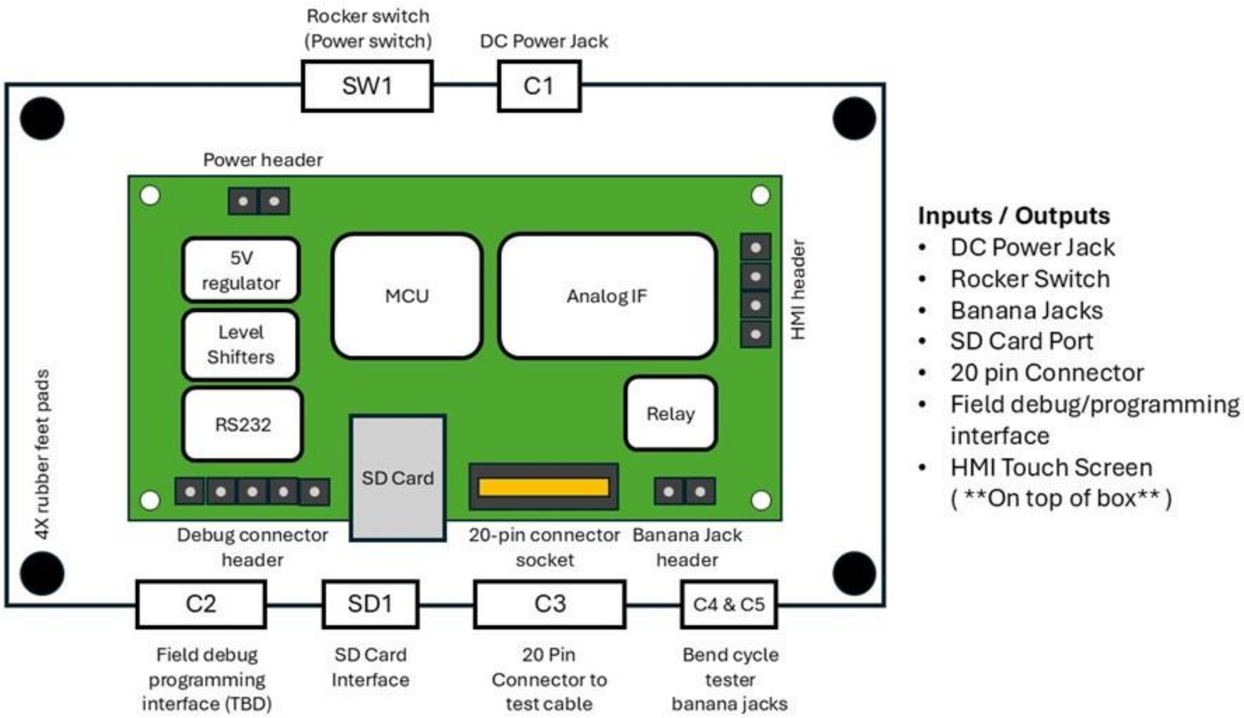

Figure 1 shows the proposed design of the Masimo short-break detector. The input and output connections are labeled. The HMI touch screen (not shown) will be embedded on top of the enclosure.

Inputs/Outputs:

The inputs and outputs of a device are how the user interacts with it. This device is intended to be used by a Masimo test engineer, so our connection ports and human machine interface placements are designed to be intuitive for the user. Our design contains four interaction points on the front, power control in the back, and an LCD touchscreen on top to control the test machine. The four interaction points include three that are used for normal operation: the SD card port (SD1), the 20-pin cable port (C3), and two banana jack ports (C4 & C5), each designed to operate specific functions during the cable test. A fourth interaction point is the field debug and programming interface (C2) for future development and debugging, as needed. Inputs to the graphical user interface (GUI) will be addressed in the next section.

Power to the device will be provided through a DC power jack connection. The delivered power will come from a 6V DC wall mount power supply. The power is then controlled via a rocker switch (SW1) connected between the DC power jack and the voltage regulator. The main connector to the device is the 20-pin connector (C3) provided by Masimo. Connector C3 will then connect to a 20 pin PCB header connected to the printed circuit board (PCB). This connection will run continuously while the device is in use. The MCU will sequentially run a current divider from the MCU’s 5V supply, though each conductor cable and will be tested continuously by the MCU’s analog-to-digital converter (ADC) to monitor voltage changes from the characterized value.

The banana jack connectors (C4 & C5) are linked to the power relay via the banana jack headers on the PCB. If a short or break is detected in the test cable, the relay will close the connection between the banana jack terminals. This will provide an indication to the external bend cycle tester that a fault in the cable has occurred, causing the bend cycle tester to stop. To preserve test data an SD card port is connected to the microcontroller. The device will be powered via the 5V supply. The SD Card reader operates on 3.3V, regulated by the on-board converter during operation. Communication between the SD card and MCU will be facilitated though SPI communication protocol.

To interact with the device, we will use a touch screen LCD. The human machine interface (HMI) device will run on 5V and will contain an on-board real-time clock (RTC). Choosing an HMI with its own contained RTC will minimize the HMI footprint on the PCB. The touchscreen display will provide a graphical user interface for the test engineer to interact with the testing software. For debugging and programming, there will be a connection to the microcontroller unit (MCU) via the debug connector header. This connection will communicate with the MCU via the microchips proprietary UPDI for debugging, as well as the RS-232 for terminal I/O. The exact connection type is yet to be determined. Additionally, a validation device will be developed. The device will be inserted into the 20-pin connector and will contain multiple switches to configure the cable. The configuration of the switches will simulate shorts or breaks within the cable. This device will be used for validation testing of the device. Lastly, four stick-on rubber pads will be attached to the bottom to deter the device from unwanted movement.

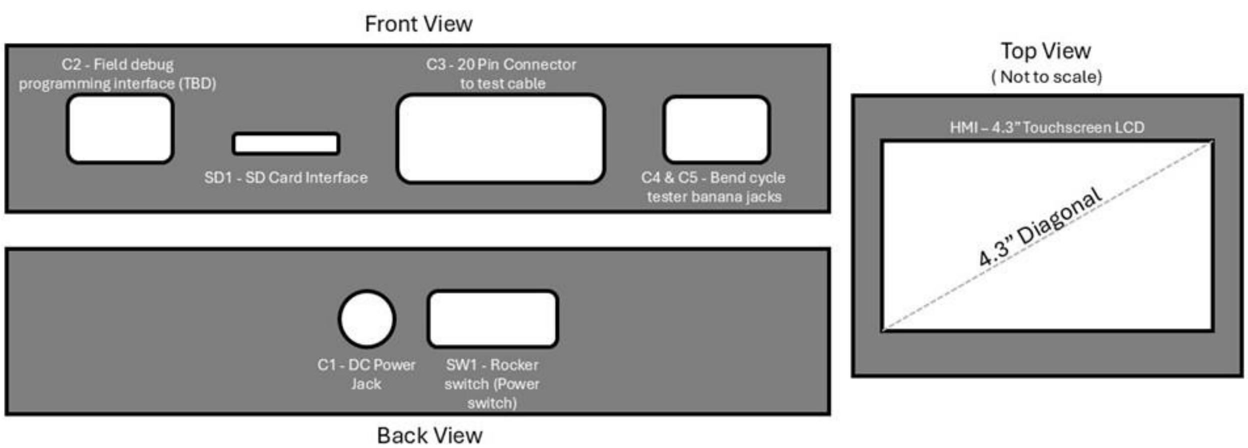

Figure 2 shows the proposed placement of the input and output connections. The front of the device will have 4 connections: debug interface, test cable connector, and two banana jacks. The front of the device will also contain an SD card slot for data preservation. The back of the device will have a DC power jack, and an interactive rocker switch to control power to the device. The top of the device will contain a 4.3" touchscreen LCD display for a graphical user interface to control the cable testing device.

Use Case:

The Cable Tester is being developed to assist Masimo in testing multi-conductor cables for durability and integrity before they are implemented into an electrical system. Specifically, it helps Masimo by allowing its users to quickly determine whether a cable meets its durability target, and if not, what the problem might be. The process is as follows: After switching on the device, the GUI screen at the top left of Figure 3 will appear, which displays "No Cable Inserted" if no cable is inserted. Next, the user connects one end of the cable that's to be tested to the device and the other end of the cable to the bend cycle tester as shown in Figure 4. The GUI will display the screen on the top right of Figure 3, which lists requirements that must be checked before tests can be completed. The user inserts the SD card into the device. Once the first three requirements are met, the user clicks the "characterize" button that will extract data from the cable and store it on the SD card. The screen at the middle left of Figure 3 will appear while the cable is being characterized. Once the characterization is completed, the user starts the bend cycle test, and the GUI will change to the screen on the middle right of Figure 3. If the user pauses the test, the screen shown on the bottom right of Figure 3 will display. The Cable Tester will run until there is a short or break detected in the wire or if the user manually ends the test. If a fault is detected the GUI will change to the screen shown in the bottom right of Figure 3. The user then removes the SD card to read the wire's data.

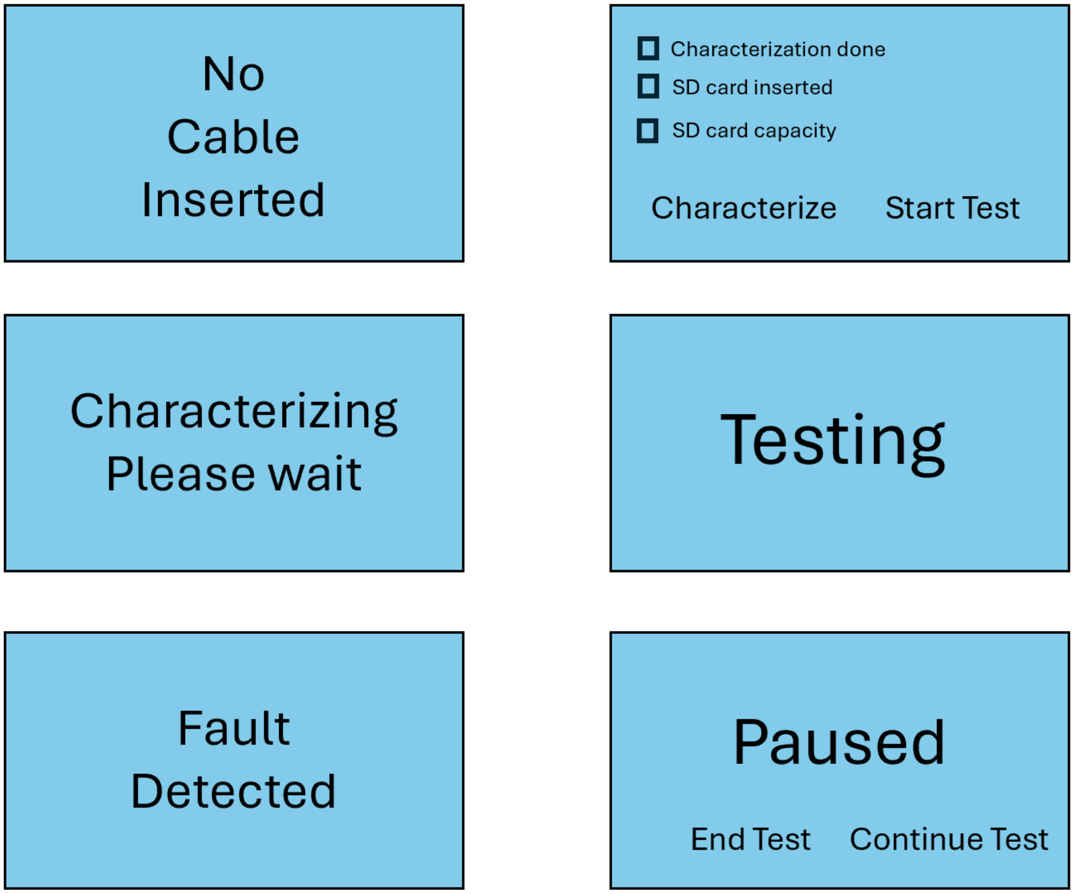

Figure 3: The GUI enables user interaction with the Cable Tester. When the device is powered on, the GUI will display a screen that prompts the user to input a cable. Once a cable is connected, the GUI will list requirements that the user must complete in order to characterize the connected cable and start a bend cycle test. To click the buttons, the requirements must be checked off. The user can also pause the screen from the "testing" screen. The test finishes once there is a fault detected or the user ends the test from the paused screen.

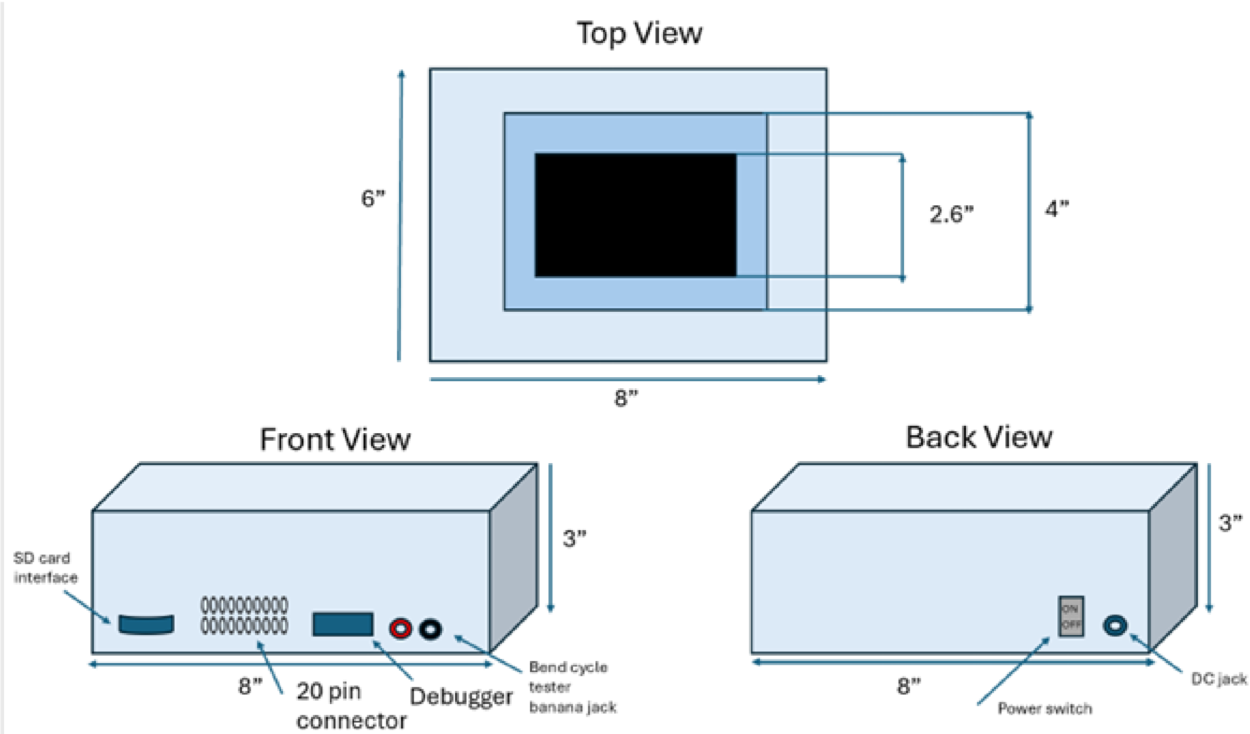

Figure 4: The device comprises a GUI, an SD card input, and a 20-pin connector, as well as banana jacks that connect the system to the bend cycle tester. The user must connect the cable that's to be tested to both the bend cycle tester to undergo stress conditions and the 20-pin connector so that data can be read. The user must also input an SD card into the device so that data can be stored.

Procured Materials:

AVR128DB64 Microcontroller

This project requires a microcontroller (MCU) with 20 ADC channels and 20 available input/output (I/O) pins to drive the cable being tested, as well as other peripherals to operate and implement the entire device. The AVR128DB64 microcontroller was chosen because it has a 22-channel analog to digital converter (ADC), serial peripheral interface (SPI) connectivity, USARTs for the debugging and HMI interfaces, and 64 I/O pins. The ADC is used for gathering voltage values from the wire, collecting and storing data from it. A SPI will attach the MCU to a SD card to allow for external storage. Other I/O pins are connected to a human machine interface (HMI) which provides a method for a user to instruct the device. The AVR128DB64 MCU also has provisions to attach an external debugger.

SD Card Reader and Driver

An SD card is needed to transfer data from the microcontroller to a portable memory storage. The MCU requires a SD driver to read and write data to and from the card. The SD card reader will take the information gathered from the tests. It can then be analyzed on a separate computer

20 Pin Connector

The 20-pin connector is the standard connector Masimo uses for connecting patient cables and sensors to its medical devices. It is through this connector that all testing of the cables will be done. The Masimo 20 pin connector interacts with the MCU through the ADC and 20 I/O pins. The MCU will be able to apply current to the wires in the cable under test and measure changes in resistance of the cable through the connector. Masimo will supply the standard component they use to connect a 20-pin connector to a PCB assembly.

Power Supply, Voltage Regulators, and Electrical Components

The power supply will need to power the MCU and HMI. The Cable Tester accepts 6V DC input. Since the MCU and the HMI operate at 5V, a 5V regulator will provide that power. The SD card reader takes 3.3V, so a separate 3.3V regulator will step down the voltage. The specific regulators to be used will be determined after initial prototyping allows us to determine the current requirement for each regulator. The power is distributed through the printed circuit board (PCB). Electrical components such as resistors, wires, diodes, and relays will connect the MCU, HMI, power supply, and I/O all to the PCB.

Developed Materials:

Microcontroller ADC Code

The MCU ADC is used during cable characterization and testing. The ADC will acquire and store voltage data from the test wire. The MCU sends a 5V signal through an analog interface to each specific wire pair in the cable, and the ADC reads a voltage value result to convert and store in memory. This process will occur sequentially testing every inner pair of wires in the cable. The values are stored with 12 bits of precision to ensure accurate and minimal loss from analog to digital conversion. While the wire is testing, the code will compare the data from the wire being tested to the values obtained for the same wire pair during characterization. If there is a significant difference, the test will stop, the relay will be set, and an error message appears on the HMI.

PCB Design

The PCB holds all the components of the device. The layout will be designed on KiCad then manufactured by JLCPCB. The AVR128DB64 will be Surface Mounted. Any additional electrical components will be soldered on by the team.

Validation Test Cable

The validation system will be used to test the Cable Tester. It can be configured to model different types of wire pairs by creating a short or open between them. The device will test cable characterization and the device testing state. During either process, the team can change the properties of the device to simulate a fault from an external source. The implementation of the test cable is to be determined.

Specifications:

Ref	Title	Specification	Notes
		Total Costs	
D-1	Development Cost	The Development cost of the complete 20 pin wire tester must be less than $5000.	
		Physical Design	
D-2	Dimensions	The dimensions of the	Depending on HMI,

		completed assembly shall fit within an enclosure not greater than 6”x4”x3”.	dimension s may be subject to change.
D-3	Enclosure	A sturdy enclosure must be used to house and protect the PCB and allow for connecting accompanying cables to PCB peripherals.	
D-4	Ergonomics	The connectors and controls on the fixture shall be located and labeled for ease of use (Refer to Figure 1).	
D-5	HMI (Human Machine Interface)	The device shall use an HMI such as a touch screen display. Interface shall be bi-directional.	Used for starting and stopping tests etc.
D-6	Stress Testing Functionality	A bend test must be determined and implemented to test ability of cable to withstand outside force.	
D-7	Rubber Feet	Test fixture shall have rubber feet secured to each of its four corners.	
PCB Design			
D-8	Power Input	A DC power jack shall be used to power the device from wall line voltage (120V AC) and shall be mounted to the enclosure. Input voltages shall be within the range of 6V-24V and will connect to a power header on the PCB.	Converting from high line voltage, there is flexibility in supply voltage.
D-9	Rocker Switch	A Rocker Switch shall be used to power on and off the device.	
D-10	Regulators	Two regulators will be required to provide 3.3V and 5V to their respective components.	5V for micro-controller and 3.3V for SD card Subject to change
D-11	Level Shifters	The device will require level shifters to allow converting logical signals of different voltage levels.	3.3V to 5V Subject to change
D-12	Relay	A relay shall be used to close the circuit when a fault is detected.	
D-13	Analog Interface	A 20-pin connector shall be	

		connected to the analog interface in order to communicate logical signals with the microcontroller.	
D-14	Microcontroller	The microcontroller used shall be the AVR128DB64 - 24MHz -5V microcontroller.	
D-15	Real-Time Clock	The system must have a real-time clock (RTC). The operator must have ability to read and set the RTC and system date.	
External Connectors			
D-16	20 Pin Connector Socket	A 20-pin connector shall be mounted to the device enclosure and connected to its respective peripheral on the PCB.	
D-17	Banana Jacks	Two Banana jack headers shall be mounted to the device enclosure and connected to its respective peripheral on the PCB.	Used for bend cycle testing.
D-18	Debug Connector	The device shall have a field debugging interface to allow program debugging. Debugging header shall be mounted to the device enclosure and connected to its respective peripheral on the PCB.	
D-19	SD Card	An SD card interface shall be mounted to the device enclosure to allow output data to be collected. Socket shall extend beyond the PCB to allow for SD card accessibility.	FAT32 formatted SD card/micro-SD
Software Specifications			
D-20	Serial Communication	RS-232 shall be used as the serial communication standard for the device.	Used for communicating with terminal devices.
D-21	Bypass Mode	Device shall be capable of being put into “bypass mode,”	Used so that output can be toggled off

		where binary output is active closed	for testing purposes.
D-22	Data Recording Rate	It must be possible to write data to the memory card 10 times per second as well as longer periods of time relevant to bend testing.	
D-23	Output Inverted	An interface to reverse relay output must be present on the HMI.	
D-24	Cable Characterization	Microcontroller must be able to accept any signal connected and characterize normal function.	

Validation/System Test:
The system test assumes that the test cable is in adequate condition. The device is initially powered off.

1) Plug the device into an outlet and turn it on.
    a. Ensure that the HMI displays the requirement screen with no cable or SD card detected.
2) Insert cable into the device and verify the device is reading the cable.
3) Ensure the SD card is properly formatted.
4) Insert an SD card and test the cable into the device and verify they are properly identified.
5) Begin characterization and ensure the data is written to the SD card.
6) Review the characterization data written to the SD card and confirm the captured characterization data matches the elements in the cable/sensor design by comparing the captured data to the cable/sensor schematics.
7) Begin test and ensure the SD card is reading the information
8) Simulate a bend cycle test:
    a. Start monitoring the cable/sensor system for changes.
    b. For each cable/sensor element that was discovered in characterization, artificially create an <u>open</u> in the circuit[1], and confirm the tester correctly reports the fault on the HMI, writes the fault data to the SD card, and signals the fault by triggering the relay.
    c. For each cable/sensor element that was discovered in characterization, artificially create a <u>short</u> in the circuit, and confirm the tester correctly reports the fault on the HMI, writes the fault data to the SD card, and signals the fault by triggering the relay.
9) Unplug the cable and repeat steps 5 through 8 for different types of cables/sensors.
10) Test fault conditions for the SD card

---

[1] Faults will be interjected into the cable/sensor being tested through a custom test fixture that is to be developed as the design takes shape and more is known about the cables/sensors to be tested.

a. During test unplug the SD card. Verify the test pauses, the fault is reported on the HMI, and the system waits for user input. Once user input is received, test continues, or aborts as requested.
b. Insert an unformatted SD card. Confirm the system detects and reports the fault on the HMI, and the system waits for user input. Once user input is received, test continues, or aborts as requested.
c. Fill an SD card to near capacity and insert into the tester. Start a test and cause the SD card to be filled. Confirm the system detects and reports the fault on the HMI, and the system waits for user input. Once user input is received, test continues, or aborts as requested.

11) Verify the ability of the tester to invert the banana jack outputs.
    a. During step 8, the device should invert the banana jack output through a relay.
    b. Verify that the external bend cycle tester has stopped bending.