Final PCB Test Plan

Test Plan Overview

Test Name	Overview	Aspect/Deliverable Tested
Pre-Soldering Test	Verifying board integrity after shipping.	• Hardware defects
Post-Soldering Test	Verifying there are no defects with the soldering work and to make sure fundamental features such as MCU power-up, MCU debugging, and power regulation is functioning as intended.	• Hardware defects • Power stability • Debug capability
E-Fuse Control	Basic e-fuse test to verify that e-fuses can be turned off and on, allowing current flow through. 1. MCU firmware simply turns the e-fuses on and off periodically. 2. Could also include reading of the current sense data and diagnostic information in this section. a. Cross reference current readings with expected values. b. Cross reference diagnostic information with oscilloscope measurements to ensure that the MCU correctly recognizes the state of an e-fuse.	• E-fuse enable control • Non-PWM'ed diagnostic information sampling • Current sensing
Basic E-Fuse Trip Test	Verifying that all e-fuses without a variable current limit trip when a load draws current above the specified limit. 1. Will also verify the auto restart behavior of relevant e-fuses.	• E-fuse trip capability
Programable E-fuse Trip Test	Verifying that e-fuses with variable current limit and trip time can have those parameters modified through the MCU and trip as expected. 1. Will require firmware that can control the current limit and trip time selection multiplexers. a. Unsure if multiple firmware version should be created, or if we should periodically loop through current limit and trip time settings.	• Variable current limit • Variable trip time

PWM E-Fuse Control Test	Verifying that relevant e-fuses can be controlled by a PWM signal. 1. MCU will need to generate a PWM signal with 50 and 100% duty cycle periodically, and with a frequency of 500Hz. 2. MCU also needs to sample the diagnostic signal from such e-fuses and be able to distinguish between an e-fuse stuck in auto restart versus an e-fuse operating normally.	• PWM generation • Diagnostic of PWM'ed e-fuses.
CAN Communication Test	Verifying that CAN messages can be received and sent. 1. Ideally would like to be able to communicate between two MCU's, but if this is not possible, it may be sufficient to do a self-test with the CAN controller test mode.	• CAN sending and receiving

Comprehensive Test Plan

Pre-Soldering

Step/Test	Test Description	Expected Outcome	Pass/Fail	Comments
1. Visual inspection	Inspect the PCB for chips, scratches, or cracks.	No defects are visually confirmed.		
2. Inter layer short circuit	Inspect and probe the PCB edge layers for a short.	The edges are cut cleanly and no short between layers is apparent.		

Post-Soldering

Step/Test	Test Description	Expected Outcome	Pass/Fail	Comments
1. Component placement	Using a microscope, verify that all components, including IC pins, are soldered securely to the board in their appropriate location.	No components are lifted off the board, all connections are fastened securely.		

2. Solder bridging	Using a microscope, inspect solder job between pins of all ICs' on board.	No solder bridging is visually confirmed.		
3. Power ground short	With an ohmmeter, probe between power and ground planes.	The resistance between rails is in the megaohms range.		
4. Adjacent pin shorts	Using a multimeter on the continuity setting, probe each pair of adjacent IC pins.	No adjacent pins have a confirmed short unless the design specifies they should be connected.		
5. Independent MCU power up	Using a power supply with a 50mA limit, supply 3.3V to the MCU through the 6 pin debug port.	The power supply is not attempting to supply more current than the set limit. The power LED is turned on.		
6. MCU ADC reference voltage	While still powered by the supply, probe the Vref+ and VDDA pins on the MCU with a voltmeter.	Vref+ and VDDA should read 3.3V.		
7. Independent MCU debugging connection	Disconnect the power supply. Connect the ST-Link USB debugger to the MCU through the 6 pin debug port. Plug in the USB debugger to a laptop with the ST-Link driver installed.	The power LED is turned on. The connected laptop opens a file explorer window with a folder and text file. No text file named "Error.txt" is found in the same location.		Deliverable
8. Independent MCU debugging flashing	Using the STM32 IDE, attempt to flash a program onto the MCU. Observe the STM32 IDE console for updates on the download process.	The STM32 IDE reports no issues with flashing the program onto the MCU.		Deliverable
9. Complete board powerup	Disconnect the STM32 debugger from the laptop and PCB.	The power LED is turned on.		

	Using a power supply with current limit set at 50mA, supply 12V to the board through the primary battery connection point.	The power supply does not attempt to supply more current than the set limit.		
10. 3.3V rail stability	Probe the 3.3V rail with a voltmeter at the output of the 3.3V linear regulator.	The voltage remains steady at 3.3V +/- 5%.		Deliverable
11. 5V rail stability	Probe the 5V rail with a voltmeter at the output of the corresponding switching regulator.	The voltage remains steady at 5V +/- 5%.		Deliverable
12. 8V rail stability	Probe the 8V rail with a voltmeter at the output of the corresponding switching regulator.	The voltage remains steady at 8V +/- 5%		Deliverable
13. MCU power stability	Probe one of the power pins on the MCU with a voltmeter.	The voltage remains steady at 3.3 +/- 2%		Deliverable
14. Battery connected powerup	Disconnect the power supply from the PDM. Connect the 12V car battery to the PDM terminals. Repeat steps 10-13	The voltages remain in the specified ranges.		Deliverable

Basic E-Fuse Control and Monitoring

This portion of the testing requires a program for the MCU which simply flips all e-fuses on and off periodically. Setting the current limits and delay times for e-fuse is not of concern in this portion. As a result, the current limits should be set to their maximal values where applicable. The MCU firmware will also periodically sample the diagnostic and current sense outputs, where applicable, to verify that data signals are connected appropriately and can be sampled by the MCU.

Step/Test	Test Description	Expected Outcome	Pass/Fail	Comments
1. Board powerup	Using a power supply with current limit set at 1A, supply 12V to the board through the primary battery connection point.	The power LED is turned on. The power supply does not attempt to supply more current than the set limit.		
2. Firmware flashing	Connect a computer to the MCU through an ST-Link debugger and the 6-pin debug port. Flash the appropriate firmware onto the MCU using the STM32 IDE.	The firmware flashing process completes without issues.		
3. E-fuse load connection	Connect a load to e-fuse 1. The load should be sized to not exceed the current limit of the specified e-fuse. Connect an ammeter between the e-fuse and the load.	The load and ammeter are connected successfully to the appropriate e-fuse.		
4. E-fuse toggling	Probe the output of the e-fuse with an oscilloscope for 10 seconds.	The oscilloscope measurements show that the e-fuse is turning off and on according to the period set by the MCU firmware. The voltage at the output of the e-fuse range across the operating condition of the e-fuse under test. Ie. 0V to 5, 8, or 12V.		
5. Diagnostic information sampling	Observe the ITM communication from the MCU within the STM32 IDE.	The diagnostic information from the e-fuse coincides with the signal shown on the oscilloscope. Eg. The		

		MCU states that the MCU is off when no current flows through the e-fuse. For applicable e-fuses, the current sense data from the MCU coincides with the current measured by the ammeter (+/-10%).		
6. Remaining e-fuse tests	Repeat steps 2-5 for all e-fuses.	All e-fuses pass the specified tests.		

Basic E-Fuse Trip Test

This test will verify that each e-fuse trips when its drawn current exceeds the planned maximum. For relevant e-fuses, the current limit and trip times will be set to their minimal values. The load which will trip each e-fuse will be the power resistor purchased for the project.

Take note of each e-fuse and its designed current limit when conducting the test.

Step/Test	Test Description	Expected Outcome	Pass/Fail	Comments
1. Board powerup	Using the 12V race car battery, connect the terminals to the PDM.	The power LED is turned on.		
2. Firmware flashing	Connect a computer to the MCU through an ST-Link debugger and the 6-pin debug port. Flash the appropriate firmware onto the MCU using the STM32 IDE.	The firmware flashing process completes without issues.		
3. Probe setup	Place an oscilloscope probe on the output of the e-fuse under test.	The oscilloscope displays the ON voltage of the selected e-fuse (5, 8,		

		12V), as the e-fuse is turned on by the firmware.		
4. Load connection	Connect a load to the e-fuse under test which is sized to draw more current than the allowed maximum. Observe the oscilloscope.	The voltage at the output of the e-fuse drops to 0V. Indicating that the e-fuse tripped successfully. Note that certain e-fuses have an auto-restart mechanism which will cause the e-fuse to intermittently turn back on and off. Entering this auto-restart loop is still considered tripping.		
5. Remaining e-fuse tests	Repeat steps 3-4 for the remaining e-fuses.	All e-fuses pass the specified tests.		

Programable E-Fuse Trip Test

This test will require firmware which varies the current limit and trip times for relevant e-fuses. Rather than cycling through each parameter in one program, the firmware will simply set a single set of values and new firmware will be flashed in order to cycle to the next set of values. Each firmware version will set both the current limit and trip time to a new value, such that firmware flashing need only be done 4 times.

For each e-fuse and current limit/trip time combination, take note of the expected current limit and trip time.

Step/Test	Test Description	Expected Outcome	Pass/Fail	Comments
1. Board powerup	Using the 12V race car battery, connect the terminals to the PDM.	The power LED is turned on.		
2. Firmware flashing	Connect a computer to the MCU through an ST-Link debugger and the 6-pin debug port.	The firmware flashing process completes without issues.		

	Flash the appropriate firmware onto the MCU using the STM32 IDE.			
3. Probe setup	Place an oscilloscope probe on the output of the e-fuse under test.	The oscilloscope displays the ON voltage of the selected e-fuse (5, 8, 12V), as the e-fuse is turned on by the firmware.		
4. Load connection	Connect a load to the e-fuse under test which is sized to draw less current than the set maximum. Observe the oscilloscope.	The oscilloscope still displays the ON voltage of the selected e-fuse.		
5. Load increase	Disconnect the load from the e-fuse under test. Increase the load such that it is sized to draw current slightly above the presently set current limit, but below the next highest current limit. Reconnect the load to the e-fuse and observe the oscilloscope. Take notes on the apparent trip time.	The oscilloscope shows that the e-fuse has tripped and begun attempting to auto-restart. The trip time between auto-restarts corresponds to the trip time set by the firmware.		
6. Remaining e-fuse tests	Repeat steps 3-5 for the remaining e-fuses.	All e-fuses pass the specified tests.		
7. Remaining parameter tests	Repeat steps 2-6 in order to test the next combinations of current limit and trip times.	All e-fuses pass the specified tests.		

PWM E-Fuse Control Test

This test requires firmware that feeds a PWM signal to relevant e-fuses with a duty cycle of 50 and 100% periodically. The diagnostic line of these e-fuses will also be sampled in order to verify that the low pass filtering works as expected.

Step/Test	Test Description	Expected Outcome	Pass/Fail	Comments
1. Board powerup	Using the 12V race car battery, connect the terminals to the PDM.	The power LED is turned on.		
2. Firmware flashing	Connect a computer to the MCU through an ST-Link debugger and the 6-pin debug port. Flash the appropriate firmware onto the MCU using the STM32 IDE.	The firmware flashing process completes without issues.		
3. Output probe setup	Place an oscilloscope probe on the output of the e-fuse under test.	The oscilloscope displays the ON voltage of the selected e-fuse (5, 8, 12V), as the e-fuse is turned on by the firmware.		
4. Load connection	Connect a load to the e-fuse under test which is sized to draw less current than the set maximum. Observe the oscilloscope. Observe the diagnostic information being printed in the STM32 IDE console.	The oscilloscope displays a PWM signal with duty cycle and frequency matching that of the input enable signal. The values printed in the console distinguish whether the duty cycle of the input signal is 50 or 100%.		
5. Load increase	Disconnect the load from the e-fuse under test. Increase the load such that it is sized to draw current slightly above the presently set current limit, but	The oscilloscope shows that the e-fuse has tripped and begun attempting to auto-restart. The values printed in the console indicate that the e-fuse in is in the auto restart state.		

	below the next highest current limit. Reconnect the load to the e-fuse. Observe the oscilloscope and the STM32 IDE console.			
6. Remaining e-fuse tests	Repeat steps 3-5 for the remaining e-fuses.	All e-fuses pass the specified tests.		