Index

[Introduction 1](#_Toc12650270)

[1. Unit Test Plan Scope (In Scope – Out of Scope) 2](#_Toc12650271)

[2. Unit Test Cases 2](#_Toc12650272)

[3. Unit Test Results 3](#_Toc12650273)

[4. Integration Test Plan Scope (In Scope – Out of Scope) 4](#_Toc12650274)

[5. Integration Test Cases 4](#_Toc12650275)

[6. Integration Test Results 4](#_Toc12650276)

[5. Addendums & Appendices 5](#_Toc12650277)

## Introduction

In this Document you will find the testcases and their results. The tests in this documents include unit and integration tests.

In the Project Files for the TIVA Board you will find a Folder named Testing, in there you can run all TIVA related tests again.

We decided to move the Tests to this folder and copy and modify the critical parts, in a way that does not change its functionality, but allows us to start parts of the software with predefined variables, as the functions normally don’t take any input variables.

In a Project of this scope we can use this method, as there are not as many critical parts and functions.  
Do not use this approach in bigger project, as it will take a lot of effort to keep the copied test-functions updated when making changes in the actual code.

## Test recreation

To run this tests again, open the TIVA Project. In tcp\_client.c you will find the main function, change this function name to release\_main. Move to the Testing Folder in the TIVA Project. In test\_tcp\_client.c you will find a test\_main function. Change the name to main. In the, now, main function in test\_tcp\_client.c you will find int testscenario. Set this value to the unit test id, from the unit test you want to run, times ten (for Unit Test 2.1 set testscenario to 21).

|  |  |
| --- | --- |
| 1. Unit Test Plan Scope (In Scope – Out of Scope) | |
| **In Scope** | **Out of Scope** |
| * + Decibel Calculation   + Webinterface Function | * Analog-Digital-Converter |

|  |  |  |  |
| --- | --- | --- | --- |
| 2. Unit Test Cases | | | |
| **ID** | **Test Cases** | **Input Value** | **Expected Output** |
| 2.1 | **Decibel Calculation**  Test Procedure:  Run Decibel Calculation with predefined sample array and predefined reference value (decibels\_calibrated)  Expected Results:  Correctly calculated decibel | Array  [1940  ,1994  ,1953  ,1940  ,1964  ,2006  ,2047  ,2020  ,1940  ,1976]  Array  [1835  ,1984  ,1953  ,2010  ,1964  ,2006  ,2047  ,2020  ,1940  ,4976]  Array  [1835  ,1984  ,1953  ,2010  ,1964  ,2006  ,2047  ,2020  ,1940  ,3000] | 69,5781  136,4875  115,4420 |
| 2.2 | **Webinterface**  Test Procedure:  Run webinterface with predefined input value  Expected Results: Only inputs 1,2,3 will generate output | 1  2  3  4  13  0  4/2  6/0  981298312  “Xe§”  “3”  “3/2”  Null  0 | 1  2  3  0  0  0  2  0  0  0  3  0  0  0 |
| 2.3 |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3. Unit Test Results | | |  |  |
| **ID** | **Test Cases** | **Pass/Fail** | **Tested By** | **Date Tested** |
| 3.1 | **Decibel Calculation**  Low\_peak and high\_peak will be set to first value, the adc gets.  Afterwards the adc samples the signal 10 times. Low\_peak and high\_peak will be compared to these sampled values and changed to the highest or lowest values of the sample. If low\_peak or high\_peak are way lower or higher than the sample values, they will never be changed to values from the sample and could result in slightly wrong decibels.  This did never occur in normal conditions and is highly unlikely to happen. | Pass (8 of 9) | Albrecht, Milli | 28.06.2019 |
| 3.2 | **Webinterface** | Pass | Albrecht, Milli | 28.06.2019 |

|  |  |
| --- | --- |
| 4. Integration Test Plan Scope (In Scope – Out of Scope) | |
| **In Scope** | **Out of Scope** |
| * 1. Send Trigger   2. Decibel Calculation   3. Webinterface js Function | Out of Scope *List features/functions that are not tested.* |

|  |  |  |  |
| --- | --- | --- | --- |
| 5. Integration Test Cases | | | |
| **ID** | **Test Cases** | **Input Value** | **Expected Output** |
| 5.1 | **adc\_send\_Fxn Output**  Test Procedure:  Run adc\_send\_Fxn with predefined buffer variable  Expected Results:  Server reads exact same values from input buffer | “1 100” | “1 100” |
| 5.2 | **HTTP POST from Server to Webpage**  Test Procedure:  Run POST calls from webserver to webpage  Expected Results:  The correct value gets send to to webpage and the correct visual clues will be displayed on the webpage |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6. Integration Test Results | | |  |  |
| **ID** | **Test Cases** | **Pass/Fail** | **Tested By** | **Date Tested** |
| 6.1 | **adc\_send\_Fxn Output** | Pass | Albrecht | 28.06.2019 |
| 6.2 | **HTTP POST from Server to Webpage** |  |  |  |

# 5. Addendums & Appendices

*Include any additional documents.*