Test Specification
Unit tests

TABLE OF CONTENTS

1. INTRODUCTION

This specification is intended to describe how unit tests work for SWT21 labkit. To create this document, a user manual, test plan and software source code were used. These tests are highly specialized and aimed at finding answers to the following questions:

- Is it possible to create unit tests for source code without using a device?
- Does the transmit function match its task, which is described in the manual as a can send?
- Does the parsing function convert the received information correctly?

The way the system is to be tested will depend upon source code capabilities.

2. SCOPE OF TESTING

1.Study of documentation and source code.
2. Selecting functions for testing.
3. Installing and updating software for design test cases.
4. Checking the possibility of testing the sub-system separately from the labkit device.
5. Design test cases.
6. Execute tests
7. Write test result report

3. REFERENCES

#	Document identifier	Document title
D01	Description of device operation	User manual
D02	Test Plan	Test plan “SWT21 labkit”
S01	Source code	swt21_fw/src/can.c
B01	Scope of test cases, meetings summary and teams decisions	Testing backlog

4. FEATURES TO BE TESTED

#	Features	Describing
U01	int parse_message_format	Whether the component is organizing the information correctly for sending to another device.
U03	can_transmit	Whether this component is sending information to another device.

5. TEST CASE SPECIFICATION

This test assumes a one-time run with the ability to create additional autotests. The code is written using the library Platformio which means needs to additionally configure the file Platformio.ini.

The configuration is built in accordance with the technical parameters of the device.

6. TEST PROCEDURES

Step No	Action	Expected results
00	Install pyserial, numpy and matplotlib * pip3 install --user pyserial * pip3 install --user numpy * pip3 install --user matplotlib Install PlatformIO * pip install platformio Install Cygwin Runtime environment https://cygwin.com/install.html Install Visual Studio Code https://code.visualstudio.com/ Install USB-UART driver from https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers	The required libraries, languages, terminals and programs are installed.
01	Open the github repository https://github.com/jlublin/swt21_fw	Link loaded
02	Clone the github repository	The repository was cloned to a test PC

	https://github.com/jlublin/swt21_fw	
03	Open the github repository https://github.com/Mjasha/labkit_unit	Link loaded
04	Clone the github repository https://github.com/Mjasha/labkit_unit	The repository was cloned to a test PC
05	Open folder “swt21_fw” cloned repository from step No 02 on test PC	File opened
06	Replace file Platformio.ini from folder “labkit_unit” (step No 04) in folder “swt21_fw” (Step No 05)	The file is being replaced
07	Replace folder test from folder “labkit_unit” (step No 04) in folder “swt21_fw” (Step No 05)	The folder is being replaced
08	Open Cygwin Terminal	Terminal opened
09	In Terminal, go to the source code folder “swt21_fw”	Folder found and opened
10	Enter command * pio test	The program will build, load, test and give the result (PASSED or NOT PASSED). The result of three tests should be displayed: test_parse_message, test_parse_message_remote test_transmit

6.1. Displaying the result

```
test/test_can.c:44:test_parse_message [RESULT]
test/test_can.c:45:test_parse_message_remote [RESULT]
test/test_can.c:46:test_transmit [RESULT]
```