TABLE OF CONTENTS

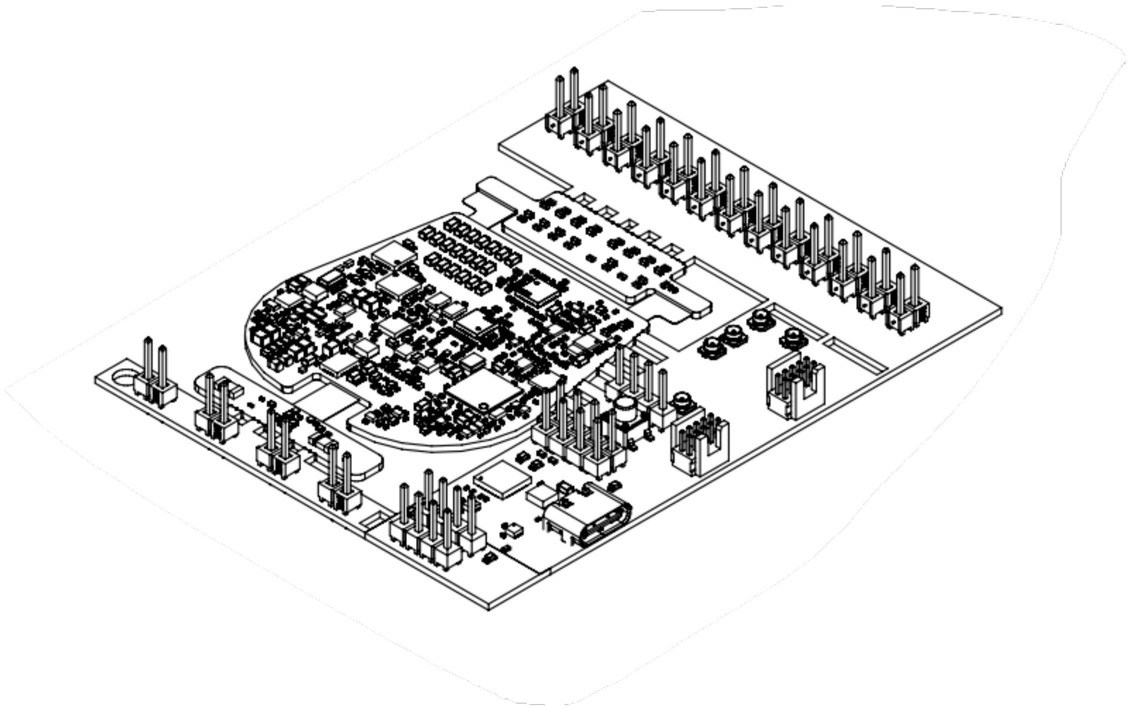
1. Document overview

According to the software design documents, test the software system that contains unit testing and integration testing.

2. Test Article and Equipment Information

2.1. Test Article: Device-Under-Test (DUT)

1	Model: V1
2	Serial No.： 2.1.B
3	Quantity：1
4	FW/SW Ver : FW-MCU-V1

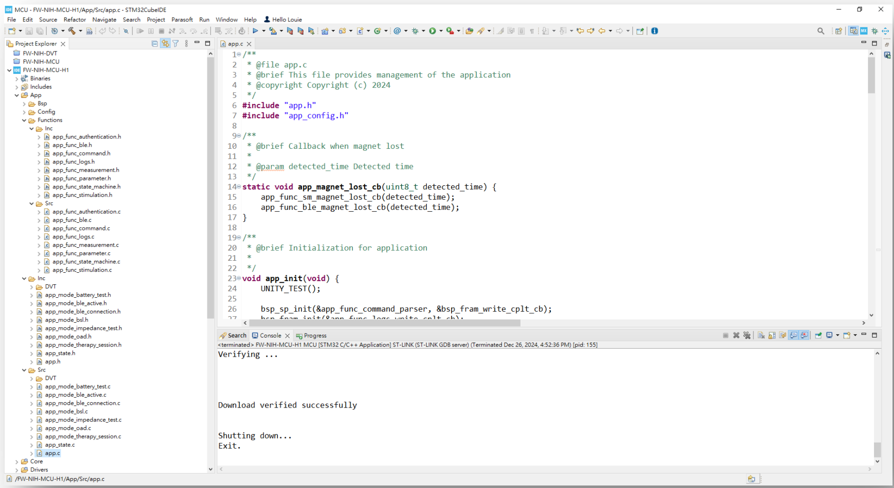


2.2. Test Article: Debugger / Programmer

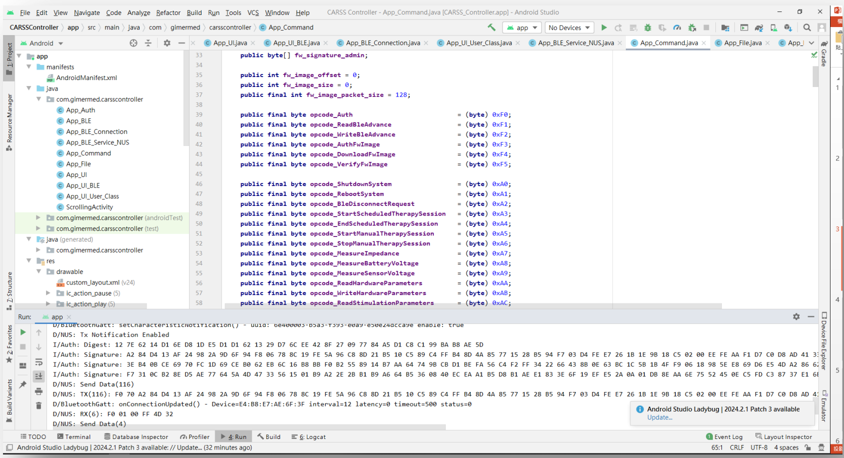
1	Model: STLINK-V3SET
2	Quantity：1
3	FW/SW Ver : V3J15M7B5S1

2.3. Test Article: PC with software

1	OS: Windows 11
2	Quantity：1
3	Software: STM32CubeIDE



2.4. Test Article: Smartphone with APP

1	Model: Google Pixel 8a
2	OS: Android 14
3	Quantity：1
4	APP: CARSS Controller
5	Software: Android Studio
	

3. Test Method

Setup of the test with MCU firmware
● Unit Test ➢ Install Unity Test Framework and add it to the project. ➢ Design unit tests and cases for the following functionality. ➢ Set the initial conditions and expected results for the test. ➢ Compile and run the test firmware. ➢ Unity Test Framework generates test reports. ➢ Confirm the test results report generated. ● Integration Test ➢ Redefine the default state and mode for testing. ➢ Setup the test environment. ➢ Compile and run the firmware in debug mode. ➢ Control the test environment according to test requirements. ➢ Compare actual results to acceptance criteria.

4. Unit Test

VV-UT01 **Authentication Function Test**		
SRS ID	Test Introduction	Acceptance Criteria
SR-BA02 SR-BA03 SR-SA02 SR-SA10 SR-OA01 SR-OA03 SR-BS02	1.Design unit tests and cases for the following functionality. a. Verify ECDSA signature is Admin class. b. Compare the hash message from FRAM and the ECDSA signature. c. Compare the hash message from Flash and the image info. d. When establishing a BLE connection, verify and confirm user class. 2.Set expected results for acceptance. 3.Compile and run tests. 4.Confirm the test results report generated.	a. The actual results are as expected. b. The actual results are as expected. c. The actual results are as expected. d. The actual results are as expected.

Test Results

```
**********[VV-UT-01-a]
<TEST>"Verify ECDSA signature is Admin class."

<COVER>app_func_auth_verify_sign_admin
<NOTE>"Use the private key to sign the data, and then verify whether the signature was signed by Admin."
<CASE>"Enter the signature for each private key and verify the signature."
Testing...3 cases

<RESULT>PASS
**********[VV-UT-01-b]
<TEST>"Compare the hash message from FRAM and the ECDSA signature."

<COVER>app_func_auth_compare_fram_hash
<NOTE>"First write the image to FRAM and then verify the HASH."
<CASE>"Enter the correct or incorrect image size to verify the HASH."
Testing...2 cases

<RESULT>PASS
**********[VV-UT-01-c]
<TEST>"Compare the hash message from Flash and the image info."

<COVER>app_func_auth_compare_flash_hash
<NOTE>"First write the image to Flash and then verify the HASH."
<CASE>"Enter the correct or incorrect image size to verify the HASH."
Testing...2 cases

<RESULT>PASS
**********[VV-UT-01-d]
<TEST>"When establishing a BLE connection, verify and confirm user class."

<COVER>app_func_auth_user_class_get
<NOTE>"Enter the signature and verify the returned User class."
<CASE>"Enter all valid or invalid signatures and verify the returned User class."
Testing...4 cases

<RESULT>PASS

-----------------------
4 Tests 0 Failures 0 Ignored
OK
```

➢ ***Result: Pass***

VV-UT02 BLE Management Function Test		
SRS ID	Test Introduction	Acceptance Criteria
SR-BA01 SR-BA06 SR-BC21 SR-SA11 SR-SA12	1.Design unit tests and cases for the following functionality. a. Enable / Disable BLE chip b. Generates request commands to control the BLE chip. c. Update and get the status and disconnection reason. d. Magnet is detected to reset BLE settings to default. 2.Set expected results for acceptance. 3.Compile and run tests. 4.Confirm the test results report generated.	a. The actual results are as expected. b. The actual results are as expected. c. The actual results are as expected. d. The actual results are as expected.

Test Results

```
**********[VV-UT-02-a]
<TEST>"Enable / Disable BLE chip"

<COVER>app_func_ble_enable
<NOTE>"Check the status and control pins of the BLE chip."
<CASE>"After enabling or disabling, check the status and control pins of the BLE chip."
Testing...2 cases

<RESULT>PASS
**********[VV-UT-02-b]
<TEST>"Generates request commands to control the BLE chip."

<COVER>app_func_ble_adv_start
<COVER>app_func_ble_disconnect
<COVER>app_func_ble_new_state_get
<COVER>app_func_ble_peers_del
<NOTE>"Verify that the generated request command data is as expected."
<CASE>"Enter parameters of the request command and confirm the generated request command data."
Testing...4 cases

<RESULT>PASS
**********[VV-UT-02-c]
<TEST>"Update and get the status and disconnection reason."

<COVER>app_func_ble_curr_state_update
<COVER>app_func_ble_curr_state_get
<COVER>app_func_ble_disc_reason_update
<COVER>app_func_ble_disc_reason_get
<NOTE>"Verify that the updated status and disconnection reason are as expected."
<CASE>"Confirm after updating the status and disconnection reason."
Testing...10 cases

<RESULT>PASS
**********[VV-UT-02-d]
<TEST>"Magnet is detected to reset BLE settings to default."

<COVER>app_func_ble_magnet_lost_cb
<COVER>app_func_ble_default_timer_cb
<COVER>app_func_ble_is_default
<NOTE>"Enter a dummy magnet detection time and confirm that the BLE default state timer is enabled as expected."
<CASE>"Enter the magnet detection time inside and outside the range."
<CASE>"Enter different detection counts during the effective time of magnet detection."
Testing...67 cases

<RESULT>PASS

-----------------------
4 Tests 0 Failures 0 Ignored
OK
```

➢ ***Result: Pass***

VV-UT03 Command Management Function Test		
SRS ID	Test Introduction	Acceptance Criteria
SR-BA01 SR-BA06 SR-BC01 SR-BC02 SR-BC04 SR-BC05 SR-BC06 SR-BC07 SR-BC08 SR-BC09 SR-BC10 SR-BC11 SR-BC12 SR-BC13 SR-BC14 SR-BC15 SR-BC16 SR-BC17 SR-BC18 SR-BC19 SR-BC21 SR-BC22 SR-SA07	1.Design unit tests and cases for the following functionality. a. Parser for response commands. b. Parser for request commands. c. Generate and send a request command for controlling the BLE chip. 2.Set expected results for acceptance. 3.Compile and run tests. 4.Confirm the test results report generated.	a. The actual results are as expected. b. The actual results are as expected. c. The actual results are as expected.

Test Results

```
**********[VV-UT-03-a]
<TEST>"Parser for response commands"

<COVER>app_func_command_resp_parser_set
<COVER>app_func_command_parser
<COVER>app_func_command_crc_confirm
<COVER>app_func_command_resp_parser
<NOTE>"Test whether the CRC value can be confirmed."
<NOTE>"Confirm whether the command to import callback is correct."
<NOTE>"Confirm that the response command returned is as expected."
<CASE>"Enter a known command with correct and incorrect CRCs to check the command parser."
Testing...2 cases

<RESULT>PASS
**********[VV-UT-03-b]
<TEST>"Parser for request commands"

<COVER>app_func_command_req_parser_set
<COVER>app_func_command_parser
<COVER>app_func_command_crc_confirm
<COVER>app_func_command_req_parser
<NOTE>"Test whether the CRC value can be confirmed."
<NOTE>"Confirm whether the command to import callback is correct."
<NOTE>"Confirm that the response command returned is as expected."
<CASE>"Enter a known command with correct and incorrect CRCs to check the command parser."
Testing...2 cases

<RESULT>PASS
**********[VV-UT-03-c]
<TEST>"Generate and send a request command for controlling the BLE chip."

<COVER>app_func_command_req_send
<COVER>app_func_command_resp_send
<NOTE>"Verify that the generated request or response command data is as expected."
<CASE>"Enter the request command format with or without payload and confirm the generated request command data."
Testing...4 cases

<RESULT>PASS

-----------------------
3 Tests 0 Failures 0 Ignored
OK
```

➢ ***Result: Pass***

VV-UT04 Log Management Function Test		
SRS ID	Test Introduction	Acceptance Criteria
SR-SL03 SR-SL07 SR-BC12 SR-BC13 SR-BC18 SR-BC19 SR-TH01 SR-TH05 SR-IM03 SR-IM04 SR-BT04 SR-BT06 SR-EV01 SR-EV02 SR-SA01 SR-FR01 SR-FR02 SR-FR03 SR-IO08	1.Design unit tests and cases for the following functionality. a. Search for the last write address of FRAM during log initialization. b. Write data to the logs. c. Read a log after the input timestamp. d. Search the logs for the event. e. Erase all log data 2.Set expected results for acceptance. 3.Compile and run tests. 4.Confirm the test results report generated.	a. The actual results are as expected. b. The actual results are as expected. c. The actual results are as expected. d. The actual results are as expected. e. The actual results are as expected.

Test Results

```
**********[VV-UT-04-a]
<TEST>"Search for the last write address of FRAM during log initialization."

<COVER>app_func_logs_init
<COVER>app_func_logs_write_cplt_cb
<NOTE>"After the log is initialized or written, confirm that the address is correct."
<CASE>"After initialization, the pointer of the log is consistent with the length of the data written."
Testing...1 case

<RESULT>PASS
**********[VV-UT-04-b]
<TEST>"Write data to the logs."

<COVER>app_func_logs_event_write
<COVER>app_func_logs_batt_volt_write
<COVER>app_func_logs_imped_write
<COVER>app_func_logs_parameter_write
<NOTE>"Confirm that the address and data written to FRAM are correct."
<CASE>"Each type of log is written and checked for pointer addresses and data."
Testing...1 case

<RESULT>PASS
**********[VV-UT-04-c]
<TEST>"Read a log after the input timestamp."

<COVER>app_func_logs_read
<NOTE>"Confirm that the timestamp of the read log is later than the input timestamp."
<CASE>"Use the timestamp of the read log to search for the next log until no more is found."
Testing...4 cases

<RESULT>PASS
**********[VV-UT-04-d]
<TEST>"Search the logs for the event."

<COVER>app_func_logs_event_search
<NOTE>"Detect whether the event exists as expected in the log."
<CASE>"Detect before and after writing specified events in the log."
Testing...2 cases

<RESULT>PASS
**********[VV-UT-04-e]
<TEST>"Erase all log data"

<COVER>app_func_logs_erase
<NOTE>"Check log information and data after erasure."
<CASE>"Check the log information and data before and after erasing all log data."
Testing...1 case

<RESULT>PASS

-----------------------
5 Tests 0 Failures 0 Ignored
OK
```

➢ ***Result: Pass***

VV-UT05 Measurement Function Test		
SRS ID	Test Introduction	Acceptance Criteria
SR-BC08 SR-BC09 SR-BC16 SR-BC22 SR-IM01 SR-IM02 SR-BT01 SR-IO10	1.Design unit tests and cases for the following functionality. a. Enable battery voltage measurement and obtain voltage. b. Enables voltage measurement of impedance and the measurement channel and obtains the voltage. c. Use the voltage measurement of the sensor and the measurement channel to obtain the voltage. d. Enable vrect voltage measurement and obtain voltage. e. Enable thermistor voltage measurement and obtain voltage. f. Turn off all peripheral circuits of measurement. 2.Set expected results for acceptance. 3.Compile and run tests. 4.Confirm the test results report generated.	a. The actual results are as expected. b. The actual results are as expected. c. The actual results are as expected. d. The actual results are as expected. e. The actual results are as expected. f. The actual results are as expected.
Test Results		

```
**********[VV-UT-05-a]
<TEST>"Enable battery voltage measurement and obtain voltage."

<COVER>app_func_meas_batt_mon_enable
<NOTE>"Confirm that the GPIO is controlled as expected."
<CASE>"Set GPIO status High or Low and then confirm the status."
Testing...2 cases
<COVER>app_func_meas_batt_mon_meas
<NOTE>"Confirm that the ADC's parameters are set correctly and that the measured voltage is as expected."
<NOTE>"Voltage factor for battery monitor is 2"
<CASE>"Enter a known dummy data simulation sampling source(0~2.5V) to verify."
Testing...2501 cases

<RESULT>PASS
**********[VV-UT-05-b]
<TEST>"Enables voltage measurement of impedance and the measurement channel and obtains the voltage."

<COVER>app_func_meas_imp_enable
<COVER>app_func_meas_imp_sel_set
<NOTE>"Confirm that the GPIO is controlled as expected."
<CASE>"Set GPIO status High or Low and then confirm the status."
Testing...4 cases
<COVER>app_func_meas_imp_sampPoints_get
<NOTE>"Verify correct ADC sample point calculation."
<CASE>"Case 1: SamplingPoints = 0, expect return 1."
<CASE>"Case 2: SamplingPoints exceeds max limit, expect return ADC_MAX_SAMPLE_POINTS."
<CASE>"Case 3: SamplingPoints within normal range, expect computed value."
<CASE>"Case 4: SamplingPoints exactly ADC_MAX_SAMPLE_POINTS, expect ADC_MAX_SAMPLE_POINTS."
<CASE>"Case 5: Rounding threshold test."
Testing...5 cases
<COVER>app_func_meas_imp_volt_meas
<NOTE>"Confirm that the ADC's parameters are set correctly and that the measured voltage is as expected."
<NOTE>"Voltage factor for impedance monitor is 6"
<CASE>"Enter a known dummy data simulation sampling source(0~2.5V) to verify."
Testing...2501 cases
<COVER>app_func_meas_imp_volt_calc
<NOTE>"Verify correct calculation of differential load voltage in mV."
<CASE>"Compare computed voltage difference between IMP_OUT+ and IMP_OUT- over defined width."
Testing...1 case
<COVER>app_func_meas_imp_calc
<NOTE>"Verify correct calculation of load resistance."
<CASE>"Case 1: vdac_mV = 0, expect 0 \xCE\xA9."
<CASE>"Case 2: vdac_mV = 1400, vload_mV = 700, expect positive finite \xCE\xA9."
<CASE>"Case 3: vdac_mV = 2500, vload_mV = 2500, expect higher resistance."
<CASE>"Case 4: vdac_mV = 1400, vload_mV = 1, expect near 0 \xCE\xA9."
<CASE>"Case 5: vdac_mV = 2500, vload_mV = 1, expect small resistance."
Testing...5 case

<RESULT>PASS
```

VV-UT05 Measurement Function Test

```
**********[VV-UT-05-c]
<TEST>"Use the voltage measurement of the sensor and the measurement channel to obtain the voltage."

<COVER>app_func_meas_vdda_sup_enable
<COVER>app_func_meas_sensor_enable
<NOTE>"Confirm that the GPIO is controlled as expected."
<CASE>"Set GPIO status High or Low and then confirm the status."
Testing...10 cases
<COVER>app_func_meas_sensor_meas
<NOTE>"Confirm that the ADC's parameters are set correctly and that the measured voltage is as expected."
<CASE>"Enter a known dummy data simulation sampling source(0~2.5V) to verify."
Testing...2501 cases

<RESULT>PASS
**********[VV-UT-05-d]
<TEST>"Enable vrect voltage measurement and obtain voltage."

<COVER>app_func_meas_vrect_enable
<NOTE>"Confirm that the GPIO is controlled as expected."
<CASE>"Set GPIO status High or Low and then confirm the status."
Testing...2 cases
<COVER>app_func_meas_vrect_mon_meas
<NOTE>"Confirm that the ADC's parameters are set correctly and that the measured voltage is as expected."
<CASE>"Enter a known dummy data simulation sampling source(0~2.5V) to verify."
Testing...2501 cases

<RESULT>PASS
**********[VV-UT-05-e]
<TEST>"Enable thermistor voltage measurement and obtain voltage."

<COVER>app_func_meas_therm_enable
<NOTE>"Confirm that the GPIO is controlled as expected."
<CASE>"Set GPIO status High or Low and then confirm the status."
Testing...2 cases
<COVER>app_func_meas_therm_meas
<NOTE>"Confirm that the ADC's parameters are set correctly and that the measured voltage is as expected."
<CASE>"Enter a known dummy data simulation sampling source(0~2.5V) to verify."
Testing...2501 cases

<RESULT>PASS
**********[VV-UT-05-f]
<TEST>"Turn off all peripheral circuits of measurement."

<COVER>app_func_meas_off
<NOTE>"Confirm that the GPIO is controlled as expected."
<CASE>"After turning off all peripheral circuits, confirm that all GPIOs are as expected."
Testing...1 case

<RESULT>PASS

-----------------------
6 Tests 0 Failures 0 Ignored
OK
```

➢ ***Result: Pass***

VV-UT06 Parameter Management Function Test

SRS ID	Test Introduction	Acceptance Criteria
SR-SL01 SR-SL02 SR-SL03 SR-SL04 SR-SL05 SR-SL06 SR-SL07 SR-BA01 SR-BA04 SR-BC03 SR-BC06 SR-BC07 SR-BC10 SR-BC11 SR-BC17 SR-BC20 SR-TH03 SR-TH04 SR-TH05 SR-TH06 SR-TH07 SR-TH08 SR-TH09 SR-TH11 SR-TH12 SR-TH13 SR-TH14 SR-IM03 SR-IM04 SR-BT02 SR-BT03 SR-BT05 SR-SA05 SR-SA06	1.Design unit tests and cases for the following functionality. a. Initialization of parameter buffer in flash. b. Get the data format and attributes based on the parameter ID. c. Set / Get data in flash based on parameter ID. d. Check whether the parameter value is within the range. e. Quantizes a parameter value to the nearest multiple of a step and clips it within a specified range. 2.Set expected results for acceptance. 3.Compile and run tests. 4.Confirm the test results report generated.	a. The actual results are as expected. b. The actual results are as expected. c. The actual results are as expected. d. The actual results are as expected. e. The actual results are as expected.

VV-UT05 Measurement Function Test		
SR-SA08		
Test Results		

```
**********[VV-UT-06-a]
<TEST>"Initialization of parameter buffer in flash."

<COVER>app_func_para_init
<NOTE>"Erase EEprom before test."
<NOTE>"Test parameter buffer initialization."
<NOTE>"Check the parameter ID written to Flash."
Testing...63 cases

<RESULT>PASS
**********[VV-UT-06-b]
<TEST>"Get the data format and attributes based on the parameter ID."

<COVER>app_func_para_datatype_get
<COVER>app_func_para_datalen_get
<COVER>app_func_para_defdata_get
<COVER>app_func_para_stepsize_get
<NOTE>"Check that the attributes of the parameter is the same as initialization."
Testing...63 cases

<RESULT>PASS
**********[VV-UT-06-c]
<TEST>"Set / Get data in flash based on parameter ID."

<COVER>app_func_para_data_set
<COVER>app_func_para_data_get
<NOTE>"Compare whether the data written and read are the same."
Testing...63 cases

<RESULT>PASS
**********[VV-UT-06-d]
<TEST>"Check whether the parameter value is within the range."

<COVER>app_func_para_val_in_range
<NOTE>"Test values that are inside and outside the range and critical."
Testing...63 cases

<RESULT>PASS
**********[VV-UT-06-e]
<TEST>"Quantizes a parameter value to the nearest multiple of a step and clips it within a specified range."

<COVER>app_func_para_val_quant_clip
<NOTE>"Test for non-quantified values and out-of-range values."
Testing...63 cases

<RESULT>PASS

-----------------------
5 Tests 0 Failures 0 Ignored
OK
```

➢ ***Result: Pass***

VV-UT07 State Machine Management Function Test		
SRS ID	Test Introduction	Acceptance Criteria
SR-SL01 SR-SL02 SR-SL03 SR-SL04 SR-SL05 SR-SL06 SR-SL07 SR-BA04 SR-BA05 SR-BC01 SR-BC02 SR-BC03 SR-BC06 SR-BC07 SR-BC20 SR-BC21 SR-TH02 SR-TH04 SR-TH14 SR-IM05 SR-BT06 SR-BT07 SR-OA03	1.Design unit tests and cases for the following functionality. a. Initialization of the state machine management. b. Set / Get the current application state. c. Control wakeup timers in sleep state. d. Use magnets to wakeup or go to sleep. 2.Set expected results for acceptance. 3.Compile and run tests. 4.Confirm the test results report generated.	a. The actual results are as expected. b. The actual results are as expected. c. The actual results are as expected. d. The actual results are as expected.
Test Results		

VV-UT05 Measurement Function Test

```
**********[VV-UT-07-a]
<TEST>"Initialization of the state machine management."

<COVER>app_func_sm_impedance_timer_enable
<COVER>app_func_sm_battery_timer_enable
<COVER>app_func_sm_init
<NOTE>"Confirm initialization of application state and timers."
<CASE>"After initialization, confirm that the timer is set as expected."
Testing...1 case

<RESULT>PASS
**********[VV-UT-07-b]
<TEST>"Set / Get the current application state."

<COVER>app_func_sm_current_state_set
<COVER>app_func_sm_current_state_get
<NOTE>"Confirm that the input and output application states are the same."
<CASE>"Enter all application states for testing."
Testing...11 cases

<RESULT>PASS
**********[VV-UT-07-c]
<TEST>"Control wakeup timers in sleep state."

<COVER>app_func_sm_wakeup_timer_cb
<COVER>app_func_sm_confirmation_timer_cb
<COVER>app_func_sm_schd_therapy_enable
<NOTE>"Confirm application state after wake-up timer's callback."
<CASE>"Control the number of callbacks for the wake-up timer and confirm application state."
<CASE>"Confirm that the RTC alarm settings used for wake-up are as expected."
<CASE>"Use the RTC callback to confirm that it wakes up as expected."
Testing...9313 cases

<RESULT>PASS
**********[VV-UT-07-d]
<TEST>"Use magnets to wakeup or go to sleep."

<COVER>app_func_sm_magnet_lost_cb
<NOTE>"Enter a dummy magnet detection time and confirm that the state as expected."
<CASE>"Enter the magnet detection time inside and outside the range."
<CASE>"After triggering magnet detection in each state, confirm that the state is as expected."
Testing...540 cases

<RESULT>PASS

-----------------------
4 Tests 0 Failures 0 Ignored
OK
```

➢ ***Result: Pass***

VV-UT08 Stimulation Management Function Test		
SRS ID	Test Introduction	Acceptance Criteria
SR-BC04 SR-BC05 SR-TH06 SR-TH07 SR-TH08 SR-TH09 SR-TH10 SR-TH11 SR-TH12 SR-TH13 SR-TH14 SR-IO01 SR-IO02 SR-IO03 SR-IO04 SR-IO05 SR-IO09	1.Design unit tests and cases for the following functionality. a. Control the GPIO and settings of HV. b. Control the power supply of VDDS. c. Control the settings of DAC. d. Control the settings of stimulus output. e. Control stimulation waveforms based on waveform settings and current source settings. f. Turn off all peripheral circuits of stimulation. 2.Set expected results for acceptance. 3.Compile and run tests. 4.Confirm the test results report generated.	a. The actual results are as expected. b. The actual results are as expected. c. The actual results are as expected. d. The actual results are as expected. e. The actual results are as expected. f. The actual results are as expected.

Test Results

```
**********[VV-UT-08-a]
<TEST>"Control the GPIO and settings of HV."

<COVER>app_func_stim_hv_supply_set
<NOTE>"Confirm that the GPIO is controlled as expected."
<CASE>"Set GPIO status High or Low and then confirm the status."
Testing...2 cases
<COVER>app_func_stim_hv_sup_volt_set
<NOTE>"Confirm that the HV voltage setting is as expected."
<CASE>"Enter voltages inside and outside the range for verification."
Testing...12000 cases

<RESULT>PASS
**********[VV-UT-08-b]
<TEST>"Control the power supply of VDDS."

<COVER>app_func_stim_vdds_sup_enable
<NOTE>"Confirm that the GPIO is controlled as expected."
<CASE>"Set GPIO status High or Low and then confirm the status."
Testing...2 cases

<RESULT>PASS
**********[VV-UT-08-c]
<TEST>"Control the settings of DAC."

<COVER>app_func_stim_dac_init
<NOTE>"Confirm that the DAC initialization settings are as expected."
<CASE>"Confirm that the register address and data during DAC initialization are as expected."
Testing...1 case
<COVER>app_func_stim_dac_volt_set
<NOTE>"Confirm that the DAC voltage is as expected."
<CASE>"Enter voltages inside and outside the range for verification."
Testing...3000 cases
<COVER>app_func_stim_dac1_ramp_set
<NOTE>"Confirm that the ramp settings are as expected."
<CASE>"Enter the ramp settings for verification."
Testing...1 cases

<RESULT>PASS
**********[VV-UT-08-d]
<TEST>"Control the settings of stimulus output."

<COVER>app_func_stim_vnsb_enable
<COVER>app_func_stim_stimulus_enable
<COVER>app_func_stim_mux_enable
<COVER>app_func_stim_sel_set
<COVER>app_func_stim_curr_src_set
<NOTE>"Confirm that the GPIO is controlled as expected."
<CASE>"Set GPIO status High or Low and then confirm the status."
Testing...8 cases

<RESULT>PASS
```

```
**********[VV-UT-08-e]
<TEST>"Control stimulation waveforms based on waveform settings and current source settings."

<COVER>app_func_stim_iout_to_dac
<COVER>app_func_stim_circuit_para1_set
<COVER>app_func_stim_circuit_para2_set
<COVER>app_func_stim_sine_para_set
<COVER>app_func_stim_stim1_start
<COVER>app_func_stim_stim2_start
<COVER>app_func_stim_sine_start
<COVER>app_func_stim_stim1_stop
<COVER>app_func_stim_stim2_stop
<COVER>app_func_stim_sine_stop
<COVER>app_func_stim_sync
<NOTE>"Confirm the format conversion and timer setting of parameters."
<CASE>"Enter parameters to confirm format conversion and timer settings."
<CASE>"Enter parameters to confirm format conversion and GPIO settings."
Testing...4 cases
Testing...3 case
Testing...6 cases
Testing...1 case
<COVER>app_func_stim_stim1_cb
<COVER>app_func_stim_stim2_cb
<COVER>app_func_stim_sine_cb
<NOTE>"Confirm parameter changes after the timer's interrupt callback."
<CASE>"Simulate the timer interrupt callback, and then confirm each parameter change until a period is reached."
Testing...20000000 cases
Testing...400000 cases

<RESULT>PASS
**********[VV-UT-08-f]
<TEST>"Turn off all peripheral circuits of stimulation."

<COVER>app_func_stim_off
<NOTE>"Confirm that the GPIO is controlled as expected."
<CASE>"After turning off all peripheral circuits, confirm that all GPIOs are as expected."
Testing...1 case

<RESULT>PASS

-----------------------
6 Tests 0 Failures 0 Ignored
OK
```

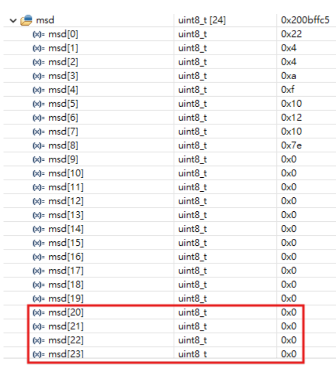
➢ ***Result: Pass***

5. Integration Test

VV-IT01 BLE Active Mode Test		
SRS ID	Test Introduction	Acceptance Criteria
SR-BA01 ~ SR-BA06	1.Redefine the default state and mode for testing. ● Start from "BLE Active Mode". 2.Setup the test environment. ● Run mobile app "CARSS Controller" with Android Studio. 3.Compile and run the firmware in debug mode. 4.Control the test environment according to test requirements. ● Enable DUT. ● Use mobile app to scan the BLE broadcasts sent by DUT. ● Use mobile app to establish a BLE connection with DUT. 5.Compare actual results to acceptance criteria.	a. The BLE ID is the same as HP03 (0x00, 4 bytes). b. BLE broadcast timeout is the same as HP14 (60s). c. Exit BLE Active Mode after BLE broadcast timeout. d. ECDSA authentication successful. e. Complete the authentication process to enter BLE Connection Mode. f. ECDSA authentication failed. g. BLE disconnects.

Test Results

a. The BLE ID obtained during scanning (from STM32CubeIDE & Android Studio):



msd	uint8_t [24]	0x200bffc5
msd[0]	uint8_t	0x22
msd[1]	uint8_t	0x4
msd[2]	uint8_t	0x4
msd[3]	uint8_t	0xa
msd[4]	uint8_t	0xf
msd[5]	uint8_t	0x10
msd[6]	uint8_t	0x12
msd[7]	uint8_t	0x10
msd[8]	uint8_t	0x7e
msd[9]	uint8_t	0x0
msd[10]	uint8_t	0x0
msd[11]	uint8_t	0x0
msd[12]	uint8_t	0x0
msd[13]	uint8_t	0x0
msd[14]	uint8_t	0x0
msd[15]	uint8_t	0x0
msd[16]	uint8_t	0x0
msd[17]	uint8_t	0x0
msd[18]	uint8_t	0x0
msd[19]	uint8_t	0x0
msd[20]	uint8_t	0x0
msd[21]	uint8_t	0x0
msd[22]	uint8_t	0x0
msd[23]	uint8_t	0x0

```
onScanResult, name: CARSS, address: F2:CB:D8:42:A9:2C, rssi: -84, Company ID: 0xF0F0, Manufacturer Data: 22 04 04 0A 0F 10 12 10 7E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

BLE ID: 0x00, 0x00, 0x00, 0x00

b. Obtain BLE broadcast timeout through data trace and timestamp (from STM32CubeIDE):

VV-IT01 BLE Active Mode Test

Access	Value	PC	Cycles	Time
WRITE	0xff		8053058	50.331613 ms
WRITE	0x0		87178162	544.863513 ms
WRITE	0x1		201737645	1.260860 s

WRITE	0x1		9698052883	60.612831 s
WRITE	0xff		9803329620	61.270810 s

Value 0xFF: BLE_STATE_INVALID, Invalid state of BLE (BLE is disabled).

Value 0x00: BLE_STATE_ADV_STOP, BLE advertising stop state.

Value 0x01: BLE_STATE_ADV_START, BLE advertising start state.

BLE Broadcast Timeout = 61.3s – 1.3s = 60s

c. Obtain the status after BLE broadcast timeout through data tracking (from STM32CubeIDE):

Access	Value	PC	Cycles	Time
WRITE	0x201		7828814	48.930088 ms
WRITE	0x200		9802325939	61.264537 s

Value 0x0201: STATE_ACT_MODE_BLE_ACT, Active state with BLE Active mode.

Value 0x0200: STATE_ACT, Active state.

d. Use correct data for ECDSA authentication after connecting (from Android Studio):

```
Digest: C0 25 AA 94 D3 34 A0 D6 6C 5C E3 8E CF A2 AC C7 4B F3 3F 31 63 AB F2 DB AD 99 5D 70 A8 E2 93 AC
Signature: AE 2B 82 9F D1 E2 72 FF F4 E0 EC C1 30 18 B6 36 A0 D2 19 67 FA B0 51 22 32 D8 13 49 E1 84 08 AB 4F EB 4D 10 94 1
Signature: 06 E6 94 91 4C 61 8F 89 02 CE 80 35 D3 2D 8D 0E 81 13 53 BB DD C9 BD 7F 03 47 93 15 23 92 B8 68 2E 9D 1B 86 B3 D
Signature: 96 C7 D6 D8 A7 D2 7E 9B AA E3 92 73 39 7F BA 6A 67 1F 00 47 B7 B0 C8 E6 E1 F1 A6 CA A7 B9 DD A2 BC 64 E7 2B 6F 1
Send Data(110)
TX(110): F0 6A AE 2B 82 9F D1 E2 72 FF F4 E0 EC C1 30 18 B6 36 A0 D2 19 67 FA B0 51 22 32 D8 13 49 E1 84 08 AB 4F EB 4D 10
RX(6): F0 01 00 FF 4D 32
```

After sending the data for ECDSA authentication, the status code returned is STATUS_SUCCESS.

e. Get the mode for completing the certification process through data tracking (from STM32CubeIDE):

WRITE	0x201		10483988314	65.524927 s
WRITE	0x202		10959567494	68.497297 s

Value 0x0201: STATE_ACT_MODE_BLE_ACT, Active state with BLE Active mode.

Value 0x0202: STATE_ACT_MODE_BLE_CONN, Active state with BLE Connection mode.

f. Use incorrect data for ECDSA authentication after connecting (from Android Studio):

```
Digest: EF CD C3 0D 99 37 7F 89 E3 CC E9 AC 7E A2 F0 CB 58 5B 7C D9 07 AB AA C0 8E 69 A8 89 06 2E 57 6D
Signature: EB F3 0A 90 0A 90 42 9D E6 93 9D 14 1D EA 3F 26 8C 14 1D AC 6D B3 89 26 52 C1 97 9E 3A D9 58 9A 91 0B 7B 5C CE 2
Signature: B0 D9 4C FC D4 0C BC 2C 56 99 17 CE B3 43 DB 4F F3 1F E0 F1 2F B7 A7 12 3C 5D CF 17 1A 30 EC 58 C0 B9 BC F8 72 4
Signature: 40 4E DF 3D 50 B9 60 B5 80 39 52 6A DA 45 E8 6A 28 21 F4 3C 7B FE 68 1C 2F 05 8A 99 EB 3E B0 F2 C7 16 5D 68 E6 B
Send Data(110)
TX(110): F0 6A EB F3 0A 90 0A 90 42 9D E6 93 9D 14 1D EA 3F 26 8C 14 1D AC 6D B3 89 26 52 C1 97 9E 3A D9 58 9A 91 0B 7B 5C
RX(6): F0 01 01 00 8C 1F
```

After sending the data for ECDSA authentication, the status code returned is STATUS_INVALID.

VV-IT01 BLE Active Mode Test

g. Obtain the BLE status after the authentication process fails (from Android Studio):
`RX(6): F0 01 01 00 8C 1F`
`onClientConnectionState() - status=19 clientIf=13 device=F2:CB:D8:42:A9:2C`
`status: 19, newState: 0`
`cancelOpen() - device: XX:XX:XX:XX:A9:2C`
`close()`
After the authentication process fails, BLE disconnects.

➢ ***Result: Pass***

VV-IT02 BLE Connection Mode Test		
SRS ID	Test Introduction	Acceptance Criteria
SR-BC01 ~ SR-BC22 SR-SA03 ~ SR-SA10	1.Redefine the default state and mode for testing. ● Start from "BLE Active Mode". 2.Setup the test environment. ● Run mobile app "CARSS Controller" with Android Studio. 3.Compile and run the firmware in debug mode. 4.Control the test environment according to test requirements. ● Enable DUT. ● Use mobile app to establish a BLE connection with DUT. ● Confirm the status is "BLE Connection Mode". ● Send commands using the mobile app. ● For functions triggered by commands, please refer to unit tests. 5.Compare actual results to acceptance criteria.	a. The command opcodes sent and received are the same and the command status codes are both STATUS_SUCCESS. b. Actual time of the BLE disconnect request is the same as HP16 (10s). c. Exit BLE Connection Mode and BLE Active Mode after BLE Disconnect Request time is elapsed. d. Actual time of the BLE Idle Connection is the same as HP15 (300s). e. Exit BLE Connection Mode and BLE Active Mode after BLE Idle Connection time is elapsed. f. Exit BLE Connection Mode, remain in BLE Active Mode, start broadcast
Test Results		
a. Check the request command opcode and the status code of the returned command (from Android Studio and STM32CubeIDE):		

VV-IT02 BLE Connection Mode Test

```
D/NUS: Send Data(4)
D/NUS: TX(4): A0 00 71 00
D/NUS: RX(5): A0 00 00 00 71
```

Expression	Type	Value
(x)= req.Opcode	uint8_t	0xa0
(x)= resp.Status	uint8_t	0x0

0xA0U /*!< The opcode of the command "SHUTDOWN_SYSTEM" */

0x00U /*!< Code for success status */

```
D/NUS: Send Data(4)
D/NUS: TX(4): A1 00 40 33
D/NUS: RX(5): A1 00 00 30 46
```

Expression	Type	Value
(x)= req.Opcode	uint8_t	0xa1
(x)= resp.Status	uint8_t	0x0

0xA1U /*!< The opcode of the command "REBOOT_SYSTEM" */

0x00U /*!< Code for success status */

```
D/NUS: Send Data(4)
D/NUS: TX(4): A2 00 13 66
D/NUS: RX(5): A2 00 00 60 1F
```

Expression	Type	Value
(x)= req.Opcode	uint8_t	0xa2
(x)= resp.Status	uint8_t	0x0

0xA2U /*!< The opcode of the command "BLE_DISCONNECT_REQUEST" */

0x00U /*!< Code for success status */

```
D/NUS: TX(4): A5 00 84 FF
D/NUS: RX(5): A5 00 00 F0 9A
```

Expression	Type	Value
(x)= req.Opcode	uint8_t	0xa5
(x)= resp.Status	uint8_t	0x0

0xA5U /*!< The opcode of the command "START_MANUAL_THERAPY_SESSION" */

0x00U /*!< Code for success status */

VV-IT02 BLE Connection Mode Test

```
D/NUS: Send Data(4)
D/NUS: TX(4): A6 00 D7 AA
D/NUS: RX(5): A6 00 00 A0 C3
```

Expression	Type	Value
(x)= req.Opcode	uint8_t	0xa6
(x)= resp.Status	uint8_t	0x0

0xA6U /*!< The opcode of the command "STOP_MANUAL_THERAPY_SESSION" */

0x00U /*!< Code for success status */

```
D/NUS: Send Data(4)
D/NUS: TX(4): A3 00 22 55
D/NUS: RX(5): A3 00 00 50 28
```

Expression	Type	Value
(x)= req.Opcode	uint8_t	0xa3
(x)= resp.Status	uint8_t	0x0

0xA3U /*!< The opcode of the command "START_SCHEDULED_THERAPY_SESSION" */

0x00U /*!< Code for success status */

```
D/NUS: Send Data(4)
D/NUS: TX(4): A4 00 B5 CC
D/NUS: RX(5): A4 00 00 C0 AD
```

Expression	Type	Value
(x)= req.Opcode	uint8_t	0xa4
(x)= resp.Status	uint8_t	0x0

0xA4U /*!< The opcode of the command "END_SCHEDULED_THERAPY_SESSION" */

0x00U /*!< Code for success status */

```
D/NUS: Send Data(4)
D/NUS: TX(4): A7 00 E6 99
D/NUS: RX(7): A7 02 00 83 07 F8 8F
```

Expression	Type	Value
(x)= req.Opcode	uint8_t	0xa7
(x)= resp.Status	uint8_t	0x0

0xA7U /*!< The opcode of the command "MEASURE_IMPEDANCE" */

0x00U /*!< Code for success status */

VV-IT02 BLE Connection Mode Test

```
D/NUS: Send Data(4)
D/NUS: TX(4): A8 00 D8 89
D/NUS: RX(9): A8 04 00 00 00 00 00 CD A1
```

Expression	Type	Value
(x)= req.Opcode	uint8_t	0xa8
(x)= resp.Status	uint8_t	0x0

0xA8U /*!< The opcode of the command "MEASURE_BATTERY_VOLTAGE" */

0x00U /*!< Code for success status */

```
D/NUS: Send Data(8)
D/NUS: TX(8): AC 04 53 50 30 31 13 FE
D/NUS: RX(11): AC 06 00 53 50 30 31 03 00 8E 15
```

Expression	Type	Value
(x)= req.Opcode	uint8_t	0xac
(x)= resp.Status	uint8_t	0x0

0xACU /*!< The opcode of the command "READ_STIMULATION_PARAMETERS" */

0x00U /*!< Code for success status */

```
D/NUS: Send Data(8)
D/NUS: TX(8): AD 04 53 50 30 31 B3 BB
D/NUS: RX(5): AD 00 00 51 33
```

Expression	Type	Value
(x)= req.Opcode	uint8_t	0xad
(x)= resp.Status	uint8_t	0x0

0xADU /*!< The opcode of the command "WRITE_STIMULATION_PARAMETERS" */

0x00U /*!< Code for success status */

```
D/NUS: Send Data(10)
D/NUS: TX(10): AE 06 17 01 01 00 00 00 C3 86
D/NUS: RX(43): AE 26 00 5B 32 30 32 33 2D 30 31 2D 30 31 54 30 30 3A 3
```

Expression	Type	Value
(x)= req.Opcode	uint8_t	0xae
(x)= resp.Status	uint8_t	0x0

0xAEU /*!< The opcode of the command "READ_IPG_LOG" */

0x00U /*!< Code for success status */

VV-IT02 BLE Connection Mode Test

```
D/NUS: Send Data(4)
D/NUS: TX(4): AF 00 4F 10
D/NUS: RX(5): AF 00 00 31 5D
```

Expression	Type	Value
(x)= req.Opcode	uint8_t	0xaf
(x)= resp.Status	uint8_t	0x0

0xAFU /*!< The opcode of the command "ERASE_IPG_LOG" */

0x00U /*!< Code for success status */

```
D/NUS: Send Data(4)
D/NUS: TX(4): B0 00 02 03
D/NUS: RX(11): B0 06 00 17 01 01 00 25 1D A6 E7
```

Expression	Type	Value
(x)= req.Opcode	uint8_t	0xb0
(x)= resp.Status	uint8_t	0x0

0xB0U /*!< The opcode of the command "READ_TIME_AND_DATE" */

0x00U /*!< Code for success status */

```
D/NUS: Send Data(10)
D/NUS: TX(10): B1 06 18 0C 04 0F 1E 02 86 16
D/NUS: RX(5): B1 00 00 53 05
```

Expression	Type	Value
(x)= req.Opcode	uint8_t	0xb1
(x)= resp.Status	uint8_t	0x0

0xB1U /*!< The opcode of the command "WRITE_TIME_AND_DATE" */

0x00U /*!< Code for success status */

```
D/NUS: Send Data(8)
D/NUS: TX(8): A9 04 01 14 0A 00 00 28
D/NUS: RX(25): A9 14 00 9E 05 27 08 29 08 29 08 29 08 2A 08 2B 08 2B 08
```

Expression	Type	Value
(x)= req.Opcode	uint8_t	0xa9
(x)= resp.Status	uint8_t	0x0

0xA9U /*!< The opcode of the command "MEASURE_SENSOR_VOLTAGE" */

0x00U /*!< Code for success status */

The command opcode sent by the App is the same as the one received by the DUT.

VV-IT02 BLE Connection Mode Test

The returned command status codes are all STATUS_SUCCESS (0x00).

For functions triggered by commands, please refer to unit tests.

b. Get the actual time of the BLE disconnect request (from STM32CubeIDE):

History (disconnect_request_ms_timer)

Access	Value	PC	Cycles	Time
WRITE	-1		1037405008	6.483781 s
WRITE	10000		2543188348	15.894927 s
WRITE	9999		2543299293	15.895621 s
WRITE	9998		2543459218	15.896620 s
WRITE	9997		2543619217	15.897620 s
WRITE	1		4142969221	25.893558 s
WRITE	0		4143129220	25.894558 s
WRITE	-1		4148729746	25.929561 s

Value (-1): The timer is disabled.

BLE disconnect request time = 25.9s – 15.9s = 10s

c. Get mode after BLE disconnect request time has elapsed (from STM32CubeIDE):

History (curr_state)

Access	Value	PC	Cycles	Time
WRITE	0x201		4598946	28.743412 ms
WRITE	0x202		1021238064	6.382738 s
WRITE	0x200		4165369293	26.033558 s

Value 0x0201: STATE_ACT_MODE_BLE_ACT, Active state with BLE Active mode.

Value 0x0202: STATE_ACT_MODE_BLE_CONN, Active state with BLE Connection mode.

Value 0x0200: STATE_ACT, Active state.

d. Get the actual time of the BLE Idle Connection (from STM32CubeIDE):

History (idle_connection_ms_timer)

Access	Value	PC	Cycles	Time
WRITE	300000		464773472	2.904834 s
WRITE	1		48480928304	303.005802 s
WRITE	0		48481088303	303.006802 s
WRITE	-1		48487168833	303.044805 s

Value (-1): The timer is disabled.

BLE Idle Connection time = 303s – 3s = 300s

e. Get mode after BLE Idle Connection time has elapsed (from STM32CubeIDE):

VV-IT02 BLE Connection Mode Test

History (curr_state)

Access	Value	PC	Cycles	Time
WRITE	0x201		4604713	28.779456 ms
WRITE	0x202		891141771	5.569636 s
WRITE	0x200		48929717521	305.810735 s

Value 0x0201: STATE_ACT_MODE_BLE_ACT, Active state with BLE Active mode.

Value 0x0202: STATE_ACT_MODE_BLE_CONN, Active state with BLE Connection mode.

Value 0x0200: STATE_ACT, Active state.

f. Get BLE status and operating mode after BLE connection lost (from STM32CubeIDE):

History (ble_curr_state)

Access	Value	PC	Cycles	Time
WRITE	0xff		4755397	29.721231 ms
WRITE	0x0		88261226	551.632663 ms
WRITE	0x1		120900604	755.628775 ms
WRITE	0xa		7171137636	44.819610 s
WRITE	0xe		7269057577	45.431610 s
WRITE	0x0		14442133134	90.263332 s
WRITE	0xff		14450434516	90.315216 s
WRITE	0x0		14533812652	90.836329 s
WRITE	0x1		14566451827	91.040324 s

Value 0xFF: BLE_STATE_INVALID, Invalid state of BLE (BLE is disabled).

Value 0x00: BLE_STATE_ADV_STOP, BLE advertising stop state (BLE is disconnected).

Value 0x01: BLE_STATE_ADV_START, BLE advertising start state.

Value 0x0A, 0x0E: BLE_STATE_CONNECT, BLE connection state.

History (curr_state)

Access	Value	PC	Cycles	Time
WRITE	0x201		4611079	28.819244 ms
WRITE	0x202		7320829263	45.755183 s
WRITE	0x201		14450289405	90.314309 s

Value 0x0201: STATE_ACT_MODE_BLE_ACT, Active state with BLE Active mode.

Value 0x0202: STATE_ACT_MODE_BLE_CONN, Active state with BLE Connection mode.

➢ ***Result: Pass***

VV-IT03 Therapy Session Mode Test		
SRS ID	Test Introduction	Acceptance Criteria
SR-TH01 ~ SR-TH14	1.Redefine the default state and mode for testing. ● Start from “Therapy Session Mode”. 2.Setup the test environment. ● Connect a resistor in series with the dual output channels (CH1 & CH2) of the stimulus. ● Connect a resistor in series with the dual output channels (CH3 & CH4) of the stimulus. 3.Compile and run the firmware in debug mode. 4.Control the test environment according to test requirements. ● Enable DUT. 5.Compare actual results to acceptance criteria.	a. These stimulation parameters must be the same as the default values. b. Actual of the Train On Duration is the same as SP07 (10s); Actual of the Train Off Duration is the same as SP08 (90s); Actual of the Ramp Up Duration is the same as SP05 (2s); Actual of the Ramp Down Duration is the same as SP06 (2s) c. Actual of the Pulse Frequency is the same as SP04 (5Hz) d. Actual of the Pulse Width is the same as SP03 (500us) e. Actual of the Therapy Session Stop is the same as STX2 (08:30)
Test Results		
a. Get the parameters for stimulus initialization (from STM32CubeIDE):		

VV-IT03 Therapy Session Mode Test

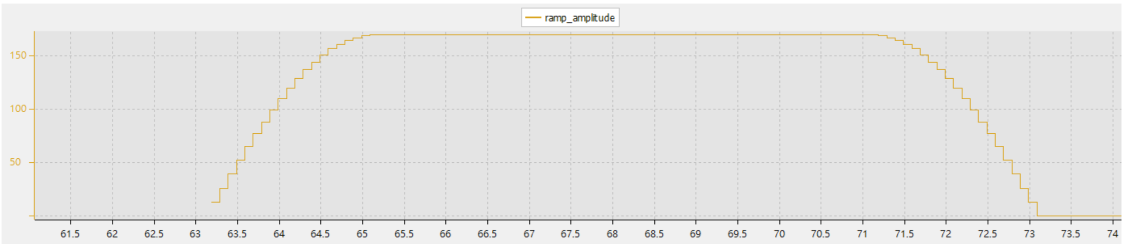
Expression	Type	Value
(x)= pulse_amplitude_mA	_Float64	0.20000000000000001
(x)= pulse_width_us	_Float64	500
(x)= pulse_frequency_hz	_Float64	5
(x)= ramp_up_duration_s	_Float64	0
(x)= ramp_down_duration_s	_Float64	0
(x)= train_on_duration_s	_Float64	10
(x)= train_off_duration_s	_Float64	90
(x)= sns_cathode_electrode_number	_Float64	1
(x)= sns_anode_electrode_number	_Float64	2
(x)= max_safe_amplitude_mA	_Float64	0.5
(x)= sine_amplitude_mA	_Float64	1
(x)= max_safe_sine_amplitude_mA	_Float64	2
(x)= sine_frequency_hz	_Float64	2.5
(x)= vnsb_on_duration_s	_Float64	10
(x)= vnsb_off_duration_s	_Float64	10

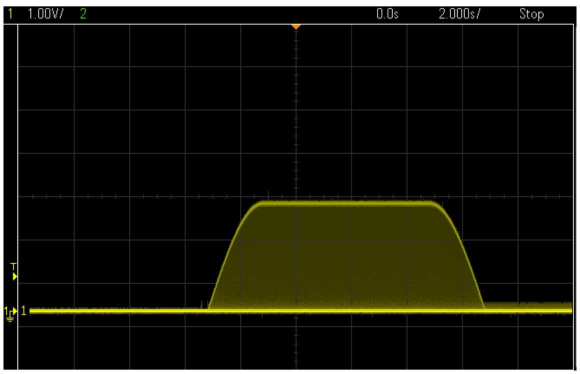
SPID	Parameter Name	Default
SP02	pulse_amplitude_mA	0.2
SP03	pulse_width_us	500
SP04	pulse_frequency_hz	5
SP05	ramp_up_duration_s	2
SP06	ramp_down_duration_s	2
SP07	train_on_duration_s	10
SP08	train_off_duration_s	90
SP09	sns_cathode_electrode_number	1
SP10	sns_anode_electrode_number	2
SP13	max_safe_amplitude_mA	0.5
SP15	sine_amplitude_mA	1
SP16	max_safe_sine_amplitude_mA	2
SP17	sine_frequency_hz	2.5
SP18	vnsb_on_duration_s	10
SP19	vnsb_off_duration_s	10

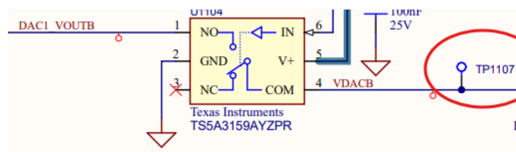
These stimulation parameters are the same as the default values.

b. Get the pulse voltage and time axis of DAC (from STM32CubeIDE):

VV-IT03 Therapy Session Mode Test







※The oscilloscope's yellow signal is the DAC's ramp output, matching the program's voltage variation.

Access	Value	PC	Cycles	Time
WRITE	0xd		10109865367	63.186659 s
WRITE	0x1a		10125929419	63.287059 s
WRITE	0x27		10141929362	63.387059 s
WRITE	0x34		10157929350	63.487058 s
WRITE	0x41		10173929350	63.587058 s
WRITE	0x4d		10189929328	63.687058 s
WRITE	0x58		10205929341	63.787058 s
WRITE	0x63		10221929316	63.887058 s
WRITE	0x6e		10237929314	63.987058 s
WRITE	0x78		10253929303	64.087058 s
WRITE	0x81		10269930358	64.187065 s
WRITE	0x89		10285930399	64.287065 s
WRITE	0x90		10301930302	64.387064 s
WRITE	0x97		10317930296	64.487064 s
WRITE	0x9d		10333930291	64.587064 s
WRITE	0xa1		10349930277	64.687064 s
WRITE	0xa5		10365930062	64.787063 s
WRITE	0xa7		10381930051	64.887063 s
WRITE	0xa9		10397930051	64.987063 s
WRITE	0xaa		10413926915	65.087043 s

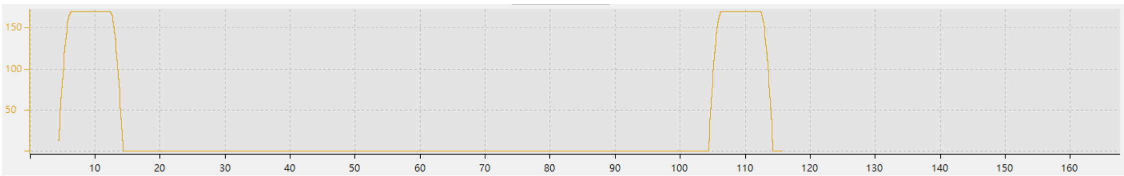
WRITE	0xaa		11373928318	71.087052 s
WRITE	0xa9		11389929487	71.187059 s
WRITE	0xa7		11405929654	71.287060 s
WRITE	0xa5		11421929542	71.387060 s
WRITE	0xa1		11437929740	71.487061 s
WRITE	0x9d		11453929737	71.587061 s
WRITE	0x97		11469929718	71.687061 s
WRITE	0x90		11485929731	71.787061 s
WRITE	0x89		11501929704	71.887061 s
WRITE	0x81		11517929682	71.987061 s
WRITE	0x78		11533929990	72.087062 s
WRITE	0x6e		11549929918	72.187062 s
WRITE	0x63		11565930019	72.287063 s
WRITE	0x58		11581929911	72.387062 s
WRITE	0x4d		11597929878	72.487062 s
WRITE	0x41		11613929884	72.587062 s
WRITE	0x34		11629929867	72.687062 s
WRITE	0x27		11645929861	72.787062 s
WRITE	0x1a		11661929850	72.887062 s
WRITE	0xd		11677929848	72.987062 s

Train On Duration = 73s – 63s = 10s

Ramp Up Duration = 65s – 63s = 2s

Ramp Down Duration = 73s – 71s = 2s

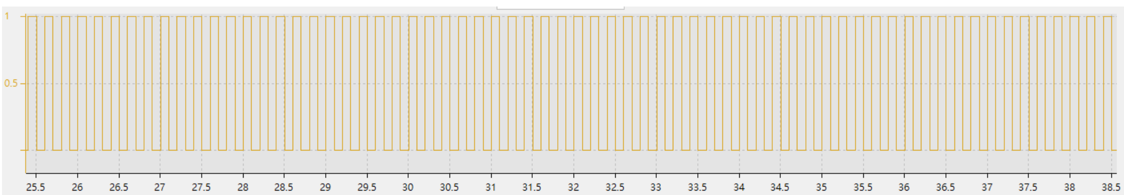
VV-IT03 Therapy Session Mode Test



WRITE	0x0		2282890972	14.268069 s
WRITE	0xd		16698885949	104.368037 s

Train Off Duration = 104s – 14s = 90s

c. Get the direction and time axis of pulse output (from STM32CubeIDE):



WRITE	0x1		5664690999	35.404319 s
WRITE	0x0		5680690991	35.504319 s
WRITE	0x1		5696690983	35.604319 s
WRITE	0x0		5712690975	35.704319 s
WRITE	0x1		5728690967	35.804319 s

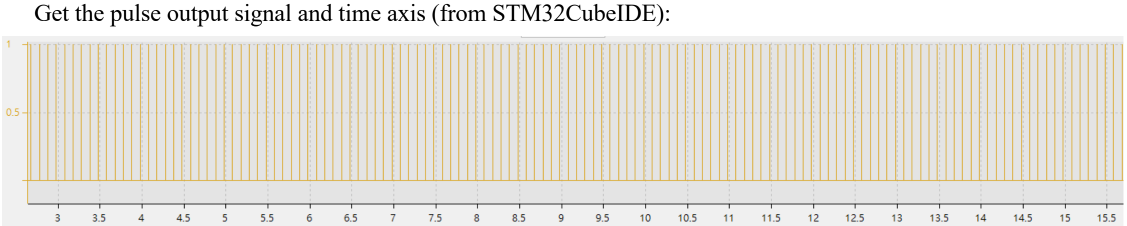
Value 1: The pulse is positive output.

Value 0: The pulse is negative output.

A complete cycle includes both positive and negative pulse outputs.

Pulse Frequency = 1 / (0.1s(positive) + 0.1s(negative)) = 1 / 0.2 = 5Hz

d. Get the pulse output signal and time axis (from STM32CubeIDE):



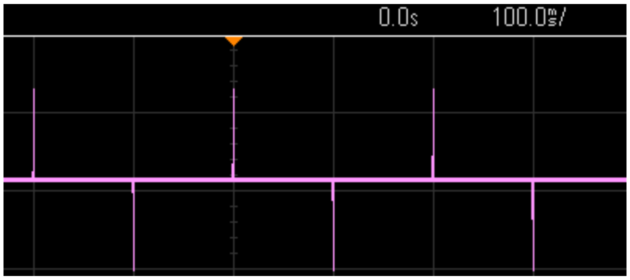
WRITE	1		2028463229	12.677895 s
WRITE	0		2028543265	12.678395 s
WRITE	1		2044463220	12.777895 s
WRITE	0		2044543256	12.778395 s

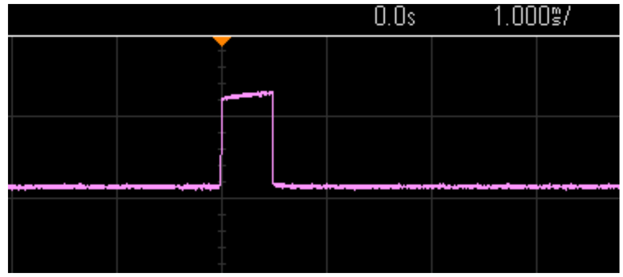
Value 1: Start of pulse output.

Value 0: End of pulse output.

Pulse Width = 500us

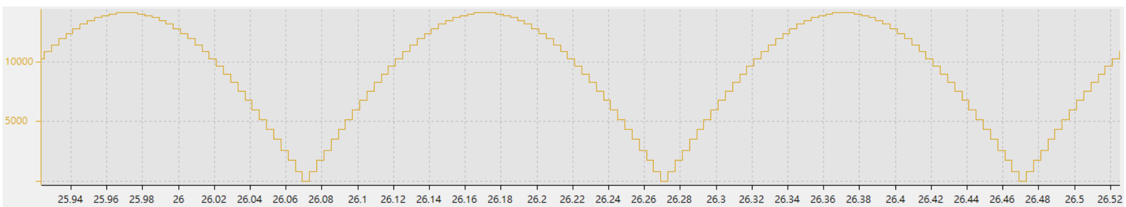
VV-IT03 Therapy Session Mode Test





※The oscilloscope shows the pulse signal with frequency and width matching the program.

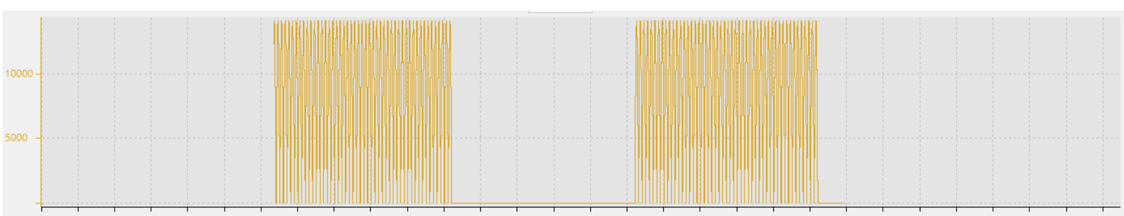
e. Get the sine voltage and time axis of DAC (from STM32CubeIDE):



WRITE	0	5483056460	34.269103 s
WRITE	0	5515056428	34.469103 s

Half period = 0.2s => Period = 0.4s

Sine Frequency = 1 / 0.4s = 2.5Hz



WRITE	0	3587951785	22.424699 s	OFF
WRITE	0	5187949282	32.424683 s	ON
WRITE	0	6787946785	42.424667 s	OFF

VNSb On Duration = 42s – 32s = 10s

VNSb Off Duration = 32s – 22s = 10s

f. Get the timestamp when the RTC alarm for "Therapy Session Stop" was triggered (from STM32CubeIDE):

Expression	Type	Value
(x)= RTC->TR	volatile uint32_t	0x83003

TR: Value 0x83003: 08:30

➢ ***Result: Pass***

VV-IT04 Impedance Test Mode Test		
SRS ID	Test Introduction	Acceptance Criteria
SR-IM01 ~ SR-IM05	1.Redefine the default state and mode for testing. ● Start from "Impedance Test Mode". 2.Setup the test environment. ● Connect a resistor (1000Ω) in series with the dual output channels (CH1 & CH2) of the stimulus. ● Short **CH1** and **CH2** together. 3.Compile and run the firmware in debug mode. 4.Control the test environment according to test requirements. ● Enable DUT. 5.Compare actual results to acceptance criteria.	a. The measured channels are CH1 and CH2. b. The impedance between CH1 (cathode) and CH2 (anode) should be close to 1000Ω. c. The short-circuit event was successfully triggered and recorded in the log. d. The impedance test interval is the same as HP18 (24 hours).

Test Results

a. Get the channels for measurement (from STM32CubeIDE):

(x)= sns_anode_electrode_number	_Float64	2 (Details)
(x)= sns_cathode_electrode_number	_Float64	1 (Details)

The channels selected for measurement are CH1 and CH2.

b. Get the measured value of impedance (from STM32CubeIDE):

(x)= sns_anode_electrode_number	_Float64	2 (Details)
(x)= sns_cathode_electrode_number	_Float64	1 (Details)
(x)= impedance	_Float64	1017.6632566244826 (Details)

The impedance measured between CH1 and CH2 is 1018 ohms, which is within ±5% of the actual impedance.

c. When the measured impedance is below the **Min Safe Impedance**, a **Short Circuit event** will be triggered.

(x)= impedance	_Float64	77.639608585789077 (Details)
(x)= min_safe_impedance_ohm	_Float64	450 (Details)

```
    _Float64 impedance = app_mode_impedance_test_get();
    if (impedance < min_safe_impedance_ohm) {
        app_func_logs_event_write((const char*)EVENT_SHORT_CIRCUIT, NULL);
    }
```

The short-circuit event was successfully triggered and recorded in the log.

d. Get the impedance test interval from the timer (from STM32CubeIDE):

(x)= impedance_test_hour_timer	int32_t	24 (Details)

The impedance test interval is 24 hours.

VV-IT04 Impedance Test Mode Test
➢ ***Result: Pass***

VV-IT05 Battery Test Mode Test		
SRS ID	Test Introduction	Acceptance Criteria
SR-BT01 ~ SR-BT07	1.Redefine the default state and mode for testing. ● Start from “Battery Test Mode”. 2.Setup the test environment. ● Set VBAT Jumpers on DUT. 3.Compile and run the firmware in debug mode. 4.Control the test environment according to test requirements. ● Enable DUT. 5.Compare actual results to acceptance criteria.	a. The voltage of battery A should be the same as the TLV70237DSET output voltage (3.7V), battery B should have no voltage output (0V), and the final sampled voltage should be the same as the voltage of battery A(3.7V). b. Because the battery voltage is greater than HP26 (2.6V), all counters will be reset. c. Since the battery voltage is less than HP27 (2.5V), all counters are incremented by one. d. Enter Shutdown State. e. Battery test timer should be the same as HP19 (24 hours).

Test Results

a. Get the battery voltage after connecting **VBAT1R** to the TLV70237DSET output on J1202 (from STM32CubeIDE):

Expression	Type	Value
(x)= battery_level	_Float64	3.7040000000000002
(x)= batteryA_level	_Float64	3.7040000000000002
(x)= batteryB_level	_Float64	0

battery_level: The final sampled voltage is the higher voltage between batteries A and B, unit: V.

batteryA_level: The voltage of battery A, unit: V.

batteryB_level: The voltage of battery B, unit: V.

b. Get the ER counter and EOS counter (from STM32CubeIDE):

Expression	Type	Value
(x)= battery_er_counter	uint32_t	0
(x)= battery_eos_counter	uint32_t	0

battery_er_counter: ER counter.

battery_eos_counter: EOS counter.

VV-IT05 Battery Test Mode Test
c. Get the ER counter and EOS counter after removing VBAT jumper (from STM32CubeIDE):

Expression	Type	Value
(x)= battery_er_counter	uint32_t	1
(x)= battery_eos_counter	uint32_t	1

battery_er_counter: ER counter.

battery_eos_counter: EOS counter.

d. The operating status is obtained after the EOS counter accumulates to 3 (from STM32CubeIDE):

Expression	Type	Value
(x)= battery_eos_counter	uint32_t	0x3
(x)= curr_state	uint16_t	0x0

battery_eos_counter: EOS counter.

curr_state: The operating status, 0x0000 is “STATE_SHUTDOWN”.

e. Get the battery test timer after the battery test procedure is completed (from STM32CubeIDE):

Expression	Type	Value
(x)= battery_test_hour_timer	int32_t	24

battery_test_hour_timer: The timer of “Battery Test Interval”.

➢ ***Result: Pass***

VV-IT06 OAD & BSL Mode Test		
SRS ID	Test Introduction	Acceptance Criteria
SR-OA01 ~ SR-OA04 SR-BS01 ~ SR-BS03 SR-FR04	1.Redefine the default state and mode for testing. ● Start from “BLE Active Mode”. 2.Setup the test environment. ● Run mobile app “CARSS Controller” with Android Studio. 3.Compile and run the firmware in debug mode. 4.Control the test environment according to test requirements. ● Enable DUT. ● Use mobile app to establish a BLE connection with DUT. ● Confirm the status is “BLE Connection Mode”. ● Enable OAD mode and BSL mode. ● Update the firmware using a special image file. 5.Compare actual results to acceptance criteria.	a. Pass image authentication. b. The image download is complete. c. Pass the verification of the image. d. The version number of the special firmware is "[TEST]".

Test Results

a. Get the result of loading the specified image file for authentication using the mobile app (from Android Studio):

```
D/FILE: Read data length = 163356
D/FILE: fw_image_name: FW-NIH-MCU-H1-TEST.bin

D/NUS: Send Data(100)
D/NUS: TX(100): F3 60 80 32 32 2E D0 67 71 17 9F E2 55 F5 31 2F D0 DC 6B 87 B3 DD B9 37 50 37 79 83 D6 BD 0
D/NUS: RX(5): F3 00 00 9E 76
D/Image: Authentication: Pass
```

The result of the authentication is pass.

b. Download the image file to IPG until it is completed (from Android Studio):

```
D/NUS: Send Data(136)
D/NUS: TX(136): F4 84 00 7E 02 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 0
D/NUS: RX(5): F4 00 00 0E F3
D/Image: Download completed, Time: 327.5 sec, Speed: 498.7969465648855 byte/sec
```

The download completed successfully.

c. Get the verification result of the downloaded image file (from Android Studio):

```
D/NUS: Send Data(8)
D/NUS: TX(8): F5 04 1C 7E 02 00 55 A4
D/NUS: RX(5): F5 00 00 3E C4
D/Image: Verify: Pass
```

The result of the verification is pass.

d. Get the version number after reconnection (from Android Studio):

VV-IT06 OAD & BSL Mode Test
D/NUS: Send Data(8) D/NUS: TX(8): AA 04 48 50 30 35 CE 36 D/NUS: RX(15): AA 0A 00 48 50 30 35 5B 54 45 53 54 5D FC CD The version number of the special firmware is "[TEST]"(ASCII).
➢ ***Result: Pass***

VV-IT07 Active State Test		
SRS ID	Test Introduction	Acceptance Criteria
SR-SL01 SR-SL07 SR-IO07	1.Redefine the default state and mode for testing. ● Start from "Active State". 2.Setup the test environment. ● Set VBAT Jumpers on DUT. 3.Compile and run the firmware in debug mode. 4.Control the test environment according to test requirements. ● Enable DUT. ● Use magnet test DUT. 5.Compare actual results to acceptance criteria.	a. Idle Duration must be the same as HP29 (60s). b. The magnet's detection timer must be the same as the actual detection time.

Test Results

a. Get the Idle Duration of "Active State" (from STM32CubeIDE):

History (curr_state)

Access	Value	PC	Cycles	Time
WRITE	0x200		4723720	29.523250 ms
WRITE	0x100		9624576004	60.153600 s

Value 0x0200: STATE_ACT, Active state.

Value 0x0100: STATE_SLEEP, Sleep state.

When the Idle Duration has elapsed, the active state changes to sleep state.

Idle Duration = 60s

b. Get the detection time of the magnet (from STM32CubeIDE):

History (magnet_detected_sec_timer)

Access	Value	PC	Cycles	Time
WRITE	0		610057642	3.812860 s
WRITE	1		773991849	4.837449 s
WRITE	2		934636929	5.841481 s
WRITE	-1		1035572264	6.472327 s

Value 0: The magnet is detected.

Value (-1): The magnet is removed.

The magnet detection timer is 2s.

The actual detection time of the magnet is 6.5s – 3.8s = 2.7s. However, since the unit is seconds, the actual detection time is 2s.

➢ ***Result: Pass***

VV-IT08 Sleep State Test

SRS ID	Test Introduction	Acceptance Criteria
SR-SL02 ~ SR-SL06 SR-IO07	1.Redefine the default state and mode for testing. ● Start from “Sleep State”. 2.Setup the test environment. ● Temporarily store the RTC timestamp when the interrupt occurs. ● Reset the RTC time for testing. 3.Compile and run the firmware in debug mode. 4.Control the test environment according to test requirements. ● Enable DUT. 5.Compare actual results to acceptance criteria.	a. The RTC interrupt Period must be the same as HP30 (60s). b. Impedance Test Interval must be the same as HP18 (24h), and Battery Test Interval must be the same as HP19 (24h). c. Verify that the time units of the timers are 1 hour. d. The magnet detection timer must be the same as the RTC timestamp interval for magnet detection. e. Therapy Session Start time must be the same as STX1 (08:00).

Test Results

a. Get the timestamp when RTC interrupt Period is elapsed during “Sleep State” (from STM32CubeIDE):

Expression	Type	Value
tr	uint32_t [5]	0x20005d6c
(x)= tr[0]	uint32_t	0x65650
(x)= tr[1]	uint32_t	0x65750
(x)= tr[2]	uint32_t	0x65850
(x)= tr[3]	uint32_t	0x65950
(x)= tr[4]	uint32_t	0x70050

Value 0x65650: 06:56:50

Value 0x65750: 06:57:50

Value 0x65850: 06:58:50

Value 0x65950: 06:59:50

Value 0x70050: 07:00:50

The RTC interrupt Period = 1 minute.

b. Get the timers when entering sleep state (from STM32CubeIDE):

Expression	Type	Value
(x)= impedance_test_hour_timer	int32_t	24
(x)= battery_test_hour_timer	int32_t	24

VV-IT08 Sleep State Test

impedance_test_hour_timer: Timer for impedance test intervals in hours.

battery_test_hour_timer: Timer for battery test intervals in hours.

c. Get the timestamp when the wake timers count down during the "Sleep State" (from STM32CubeIDE):

Expression	Type	Value
(x)= impedance_test_hour_timer	int32_t	23
(x)= battery_test_hour_timer	int32_t	23

The timers first counted down.

Expression	Type	Value
TR	uint32_t [10]	0x20006af4
(x)= TR[0]	uint32_t	0x32519
(x)= TR[1]	uint32_t	0x42519

TR[0]: When entering sleep state, the timestamp at which the timers start counting. Value 0x32519: 03:25:19

TR[1]: The timestamp when the timers first counted down. Value 0x42519: 04:25:19

The time interval between timers counting down = 1 hour.

d. Get the detection time of the magnet (from STM32CubeIDE):

Expression	Type	Value
(x)= magnet_detected_sec_time	uint8_t	2 (Decimal)
TR	uint32_t [10]	0x20006af4
(x)= TR[0]	uint32_t	0x55233
(x)= TR[1]	uint32_t	0x55235

magnet_detected_sec_time: The time of the magnet detection timer, 2s

TR[0]: When the magnet is detected. Value 0x55233: 05:52:33

TR[1]: When the magnet is removed. Value 0x55235: 05:52:35

The RTC timestamp interval for magnet detection is 2s.

e. Get the timestamp when the RTC alarm for "Therapy Session Start" was triggered during "Sleep State". (from STM32CubeIDE):

Expression	Type	Value
(x)= TR[0]	uint32_t	0x80000

TR[0]: Value 0x80000: 08:00

➢ ***Result: Pass***

VV-IT09 Shutdown State Test		
SRS ID	Test Introduction	Acceptance Criteria
SR-IO06	1.Redefine the default state and mode for testing. ● Start from “Shutdown State”. 2.Setup the test environment. ● Set VBAT Jumpers on DUT. 3.Compile and run the firmware in debug mode. 4.Control the test environment according to test requirements. ● Enable DUT. ● Set a breakpoint at the GPIO digital output pin IPG_SHDN to turn off IPG. 5.Compare actual results to acceptance criteria.	a. This breakpoint is to set the GPIO digital output pin IPG_SHDN to turn off the IPG.

Test Results

a. The debugger's current instruction pointer stops at a breakpoint (from STM32CubeIDE):

```
/**
 * @brief Handler for shutdown state
 *
 */
void app_state_shutdown_handler(void) {
    bsp_fram_deinit();
    bsp_sp_deinit();
    app_state_power_off();

    HAL_GPIO_WritePin(IPG_SHDN_GPIO_Port, IPG_SHDN_Pin, GPIO_PIN_SET);
}
```

The debugger's current instruction pointer stops at the specified breakpoint.

➢ ***Result: Pass***

REVISION HISTORY		
Rev.	Date	Description
01	01/01/2026	Validation & Verification and Test Report version 1