

IoT Test Cases

Test Scenario #1 - Starting a listening thread

Test case ID	Test case Description	Test Steps	Test Data	Expected Results	Actual Results	Grade
TC11	Starting thread for a new user with correct data	1. Call the specific POST controller method with the wanted user key	user_key = 155	New Hardware User created and saved to persistence. Thread started and connected to specified URL	As expected	Pass
TC12	Starting thread for a new user with incorrect data	1. Call the specific POST controller method with the incorrect user key	user_key = 1Gh6	The user_key fails to be parsed, related terminal message shown and thread not started	As expected	Pass
TC13	Starting thread for an already existing user	1. Call the specific POST controller method with the already saved user key	user_key = 155	Related terminal message shown and thread not started	As expected	Pass

Test Scenario #2 - Returning data to the user

Test case ID	Test case Description	Test Steps	Test Data	Expected Results	Actual Results	Grade
TC21	Requesting data for an existing user	1. Call the specific GET controller method with the wanted user information	user_key = 155 user_token = Vdja6a7dSj	Data returned in a list format and local list cleared to avoid duplicate data transmission	As expected	Pass
TC22	Requesting data for an existing user when no data	1. Call the specific GET controller method with the	user_key = 156	Data returned as an empty list to avoid	As expected	Pass

	was yet received	wanted user information	user_token = Vdja6a7dSj	null pointer exceptions		
TC23	Requesting data for a new user	1. Call the specific GET controller method with the new user information	user_key = 155 user_token = Vdja6a7dSj	Empty list returned as there is no such user thread started yet and related terminal message shown	As expected	Pass

Test Scenario #3 - Forwarding information to the Loriot Network Server

Test case ID	Test case Description	Test Steps	Test Data	Expected Results	Actual Results	Grade
TC31	Send correct data to an existing user	1. Call the specific POST controller method with the correct information	user_key = 155 hweui = 0004A30B002528D3 desired_air_humidity = 50 desired_light_level = 120 desired_air_temperature = 21.5 desired_air_co2 = 600	Json telegram message constructed and send to the Loriot server successfully	As expected	Pass
TC32	Send incorrect data to an existing user	1. Call the specific POST controller method with the incorrect hardware EUI and/or 'desired' information	user_key = 155 hweui = 0004A2528D3 desired_air_humidity = a5 desired_light_level = 120 desired_air_temperature = 21.5 desired_air_co2 = 600	Json telegram fails to send and/or construct successfully and related terminal message is shown	As expected	Pass

TC33	Send data to a new user	1. Call the specific POST controller method with the new user information	user_key = 188 hweui = 0004A30B002528D3 desired_air_humidity = 50 desired_light_level = 120 desired_air_temperature = 21.5 desired_air_co2 = 600	Json telegram not constructed because no related user thread exists yet	As expected	Pass
------	-------------------------	---	---	---	-------------	------

Test Scenario #4 - Stopping a listening thread

Test case ID	Test case Description	Test Steps	Test Data	Expected Results	Actual Results	Grade
TC41	Stopping an existing user's listening thread	1. Call the specific DELETE controller method with the user's key	user_key = 155	User's listening thread is stopped and removed from the local persistence	As expected	Pass
TC42	Stopping a listening for a non-existent or new user	1. Call the specific DELETE controller method with the new user's key	user_key = 1598	No thread is affected and related terminal message is shown	As expected	Pass
TC43	Trying to stop a listening thread with the wrong user key	1. Call the specific DELETE controller method with a wrong key	user_key = hD7!	No thread is affected and related terminal message is shown	As expected	Pass

Test Scenario #5 - Monitoring air temperature and humidity

Test case ID	Test case Description	Test Steps	Test Data	Expected Results	Actual Results	Grade
TC51	Measuring air temperature with sensor and task set up correctly	1.Start the main program for the FreeRTOS application	Related hih8120 driver setup functions called correctly	The measured air temperature will be shown and the value saved in the shared data periodically without affecting other tasks	As expected	Pass
TC52	Measuring air temperature with sensor set up incorrectly	1.Start the main program for the FreeRTOS application	Related hih8120 driver setup functions called incorrectly / not called	Error messages for initialising or waking up the sensor shown in the terminal	As expected	Pass

Test Scenario #6 - Monitoring air carbon dioxide level

Test case ID	Test case Description	Test Steps	Test Data	Expected Results	Actual Results	Grade
TC61	Measuring air co2 level with sensor and task set up correctly	1.Start the main program for the FreeRTOS application	Related mh_z19 driver setup functions called correctly	The measured air co2 level will be shown and the value saved in the shared data periodically without affecting other tasks	As expected	Pass
TC62	Measuring air co2 level with sensor set up incorrectly	1.Start the main program for the FreeRTOS application	Related mh_z19 driver setup functions called incorrectly / not called	Error messages for initialising the sensor shown in the terminal	As expected	Pass

Test Scenario #7 - Monitoring light level

Test case ID	Test case Description	Test Steps	Test Data	Expected Results	Actual Results	Grade
--------------	-----------------------	------------	-----------	------------------	----------------	-------

TC71	Measuring light level with sensor and task set up correctly	1.Start the main program for the FreeRTOS application	Related tsl2591 driver setup functions called correctly	The measured light level will be shown and the value saved in the shared data periodically without affecting other tasks	As expected	Pass
TC72	Measuring light level with sensor set up incorrectly	1.Start the main program for the FreeRTOS application	Related tsl2591 driver setup functions called incorrectly / not called	Error messages for initialising or waking up the sensor shown up	As expected	Pass

Test Scenario #8 - Sending message through LoRaWAN

Test case ID	Test case Description	Test Steps	Test Data	Expected Results	Actual Results	Grade
TC81	Sending a new uplink payload through LoRaWAN when task, payload and transreceiver are setup correctly	1.Start the main program for the FreeRTOS application	<p>Related lora_driver and OTAA join setup steps done correctly</p> <p>LORA_appEUI = 080AE22D17745AEA</p> <p>LORA_appKEY = 12A67C3072B659179BC2216FE32B7DC9</p> <p>_uplink_payload.len = 8</p> <p>_uplink_payload.portNo = 1</p>	The payload will be uploaded correctly and possibility for reading a downlink given. The sent message will be received in the Gateway	As expected	Pass
TC82	Sending a new uplink payload through LoRaWAN when	1.Start the main program for the	Related lora_driver and OTAA join setup steps done incorrectly	Error messages for initialising or failure to join the	As expected	Pass

	task or the transceiver are setup incorrectly	FreeRTOS application	LORA_appEUI = 08!c LORA_appKEY = 12A67C3072B65917 9BC2216FE32B7DC9 _uplink_payload.len = 8 _uplink_payload. portNo = 1	network shown in terminal		
TC83	Sending a new uplink payload through LoRaWAN when payload setup incorrectly	1.Start the main program for the FreeRTOS application	Related lora_driver and OTAA join setup steps done correctly LORA_appEUI = 080AE22D17745AEA LORA_appKEY = 12A67C3072B65917 9BC2216FE32B7DC9 _uplink_payload .len = -1 _uplink_payload. .portNo = 1	Failure to send the uplink and related message shown in terminal	As expected	Pass

Test Scenario #9 - Receiving message through LoRaWAN

Test case ID	Test case Description	Test Steps	Test Data	Expected Results	Actual Results	Grade
TC91	Receiving a new downlink payload through LoRaWAN when task, payload and transceiver are setup correctly	1.Start the main program for the FreeRTOS application	Related lora_driver and OTAA join setup steps done correctly LORA_appEUI = 080AE22D17745AEA LORA_appKEY = 12A67C3072B65917 9BC2216FE32B7DC9	The payload will be read from the message buffer and information saved in the shared data class	As expected	Pass

			downlink_buffer = xMessageBufferCreate (sizeof(lora_driver_payload_t)*2);			
TC92	Receiving a new downlink payload through LoRaWAN when task,or transreceiver are setup incorrectly	1.Start the main program for the FreeRTOS application	Related lora_driver and OTAA join setup steps done incorrectly LORA_appEUI = 080AE2!cG LORA_appKEY = 12A67C3072B65917 9BC2216FE32B7DC9 downlink_buffer = xMessageBufferCreate (sizeof(lora_driver_payload_t)*2)	Error messages for initialising or failure to join the network shown in terminal	As expected	Pass
TC93	Receiving a new downlink payload through LoRaWAN when the message buffer or downlink payload are setup incorrectly	1.Start the main program for the FreeRTOS application	Related lora_driver and OTAA join setup steps done correctly LORA_appEUI = 080AE22D17745AEA LORA_appKEY = 12A67C3072B65917 9BC2216FE32B7DC9 downlink_buffer = NULL	Payload will not be sent and failure message shown in terminal	As expected	Pass

Test Scenario #10 - Managing air humidity and temperature

Test case ID	Test case Description	Test Steps	Test Data	Expected Results	Actual Results	Grade
TC101	Managing air temperature and humidity based on the desired air humidity from downlink	1.Start the main program for the FreeRTOS application	Related motor driver setup done correctly desired_data .desired_hum = 1000	If user's desired humidity is higher than the measured one, motor will move, allowing for the flow of water	As expected	Pass

TC102	Managing air temperature and humidity when desired air humidity is not set up	1.Start the main program for the FreeRTOS application	Related motor driver setup done correctly desired_data .desired_hum = NULL	Error when comparing a NULL to the measured humidity is shown in terminal	As expected	Pass
TC103	Managing air temperature and humidity when no downlink received yet (default desired values)	1.Start the main program for the FreeRTOS application	Related motor driver setup done correctly desired_data .desired_hum = 0	Measured humidity will always be higher and motor will not move	As expected	Pass

Test Scenario #11 - Managing light level

Test case ID	Test case Description	Test Steps	Test Data	Expected Results	Actual Results	Grade
TC111	Managing light level based on the desired light level from downlink	1.Start the main program for the FreeRTOS application	Related motor driver setup done correctly desired_data .desired_light = 150	If desired light is higher than the measured one, motor will move to turn light level down; otherwise it will turn it up	As expected	Pass
TC112	Managing light level when the desired light level is not set up	1.Start the main program for the FreeRTOS application	Related motor driver setup done correctly desired_data .desired_light = NULL	Error when comparing a NULL to the measured light level is shown in terminal	As expected	Pass
TC113	Managing light level when no downlink received yet (default desired values)	1.Start the main program for the FreeRTOS application	Related motor driver setup done correctly desired_data .desired_light = 0	Measured light level will always be higher and motor will turn the light level down	As expected	Pass