Amazon FreeRTOS Qualification Developer Guide

Board Qualification Errata

Document Version v1.0.1

Amazon FreeRTOS Qualification: Developer Guide Errata on Board Qualification

Revision History

Date	Version	Change History
July 31, 2018	1.0.0	Initial release
October 08, 2018	1.0.1	Minor typo fix in TI CC3220SF section

Contents

Introduction

The Amazon FreeRTOS Qualification (AFQ) gives confidence to OEM/ODM developers that by using a qualified microcontroller (MCU) from this program for their IoT device, they can run Amazon FreeRTOS on the device without compatibility issues.

What is this document for?

This document gives a summary of the qualification tests that are not passing on the current list of AFQ qualified boards. It is an additional tool for use by OEM/ODMs who are evaluating qualified boards for use in device development. This document will help them be better informed about the port for Amazon FreeRTOS provided by the vendor for each board. We continue to work with AFQ vendors to update each port, with the goal that all tests eventually pass for each board.

Summary of Known Issues

Vendor	Board Name	Category	Test File	Tests that fail	Detailed Description
Texas Instruments	CC3220SF	TCP/IP	aws_test_tcp.c	Intermittent failure of multiple test cases	1.1
STMicroelectronics	STM32L4 Discovery Kit IoT Node	Wi-Fi	aws_test_wifi.c	WiFiSeperateTasksConnectingAndDisconnectingAtOnce	1.2
STMicroelectronics	STM32L4 Discovery Kit IoT Node	TCP/IP	aws_test_tcp.c	SOCKETS_Threadsafe_SameSocketDifferentTasks SECURE_SOCKETS_Threadsafe_SameSocketDifferentTasks	1.3
STMicroelectronics	STM32L4 Discovery Kit IoT Node	TCP/IP	aws_test_tcp.c	SECURE_SOCKETS_SetSockOpt_SERVER_NAME_INDICATION	1.4
STMicroelectronics	STM32L4 Discovery Kit IoT Node	TCP/IP	aws_test_tcp.c	[Intermittent] SECURE_SOCKETS_TwoSecureConnections	1.5
STMicroelectronics	STM32L4 Discovery Kit IoT Node	TLS	aws_test_tls.c	TLS_ConnectEC TLS_ConnectBYOCCredentials	1.6
NXP	LPC54018 IoT Module	Wi-Fi	aws_test_wifi.c	WiFiSeperateTaskConnectingAndDisconnectingAtOnce	1.7
NXP	LPC54018 IoT Module	TCP/IP	aws_test_tcp.c	SOCKETS_Threadsafe_SameSocketDifferentTasks SECURE_SOCKETS_Threadsafe_SameSocketDifferentTasks	1.8
NXP	LPC54018 IoT Module	TCP/IP	aws_test_tcp.c	SECURE_SOCKETS_TwoSecureConnections	1.9
Microchip	PIC32MZEF bundle	Wi-Fi	aws_test_wifi.c	WiFiConnectMultipleAP	1.10
Microchip	PIC32MZEF bundle	Wi-Fi	aws_test_wifi.c	WiFiSeperateTasksConnectingAndDisconnectingAtOnce	1.11
Microchip	PIC32MZEF bundle	TCP/IP	aws_test_tcp.c	[Intermittent] SECURE_SOCKETS_Threadsafe_DifferentSocketsDifferentTasks SECURE_SOCKETS_Threadsafe_SameSocketDifferentTasks SOCKETS_Threadsafe_DifferentSocketsDifferentTasks SOCKETS_Threadsafe_SameSocketDifferentTasks	1.12
Espressif	ESP32-WROVER-KIT and ESP DevKitC	Wi-Fi	aws_test_wifi.c	[Intermittent] WiFiConnectMultipleAP	1.13

Texas Instruments

CC3220SF-LAUNCHXL

1.1 Intermittent failure of Secure Sockets tests

1.1.1. Issue

- Category: Secure Sockets
- Test file: aws_test_tcp.c
- Tests: Multiple

1.1.2. Description

The network processor on the CC3220SF can manage a maximum of 6 simultaneous connections. Due to a known issue in the network coprocessor, a socket is not available for reuse after its first use (until the target is reset). This limits the number of socket connections over the lifetime of the application.

1.1.3. Workaround

If the application use case requires opening 6 or less secure sockets which would not be released back during the lifetime of the application until it quits, this limitation would be of less concern during system design and port evaluation. If the application use case includes (or can accommodate) a target reset before requiring the use of more than 6 sockets, the application would not encounter this issue.

STMicroelectronics

STM32L4 Discovery Kit IoT Node

1.2 Separate tasks connecting and disconnecting at once

1.2.1. Issue

- Category: Wi-Fi management
- Test file: aws_test_wifi.c
- Tests: WiFiSeperateTasksConnectingAndDisconnectingAtOnce

1.2.2. Description

This test is a thread safety test to check if the vendor supplied Wi-Fi port for the board is thread safe. It failed due to lack of thread protection.

1.2.3. Workaround

This limitation can potentially be handled by designing the application to not use the resource in more than one task. This is not an issue with the Amazon FreeRTOS supplied libraries.

1.3 Sockets thread safety using different tasks

1.3.1. Issue

- Category: TCP/IP
- Test file: aws_test_tcp.c
- Tests: SOCKETS_Threadsafe_SameSocketDifferentTasks
  SECURE_SOCKETS_Threadsafe_SameSocketDifferentTasks

1.3.2. Description

These tests are thread safety test to check if the vendor supplied secure sockets port for the board are thread-safe. They failed due to a current limitation of the Wi-Fi module that renders the socket unusable when its internal receive/transmit buffers are full.

1.3.3. Workaround

This limitation could potentially be handled in certain application software by controlling the send/receive speed to ensure that these buffers never become full. This is not an issue with the Amazon FreeRTOS supplied libraries.

1.4 Server Name Indication (SNI) test with invalid hostname

1.4.1. Issue

- Category: TCP/IP
- Test file: aws_test_tcp.c

- Tests: SECURE_SOCKETS_SetSockOpt_SERVER_NAME_INDICATION

1.4.2. Description

Secure Sockets Connect worked for an invalid hostname, whereas it should have failed. When using SNI, the TLS handshake should fail if the client indicates a different hostname than the one presented by the server in the server certificate.

1.4.3. Workaround

Feedback from the vendor is that this feature is not supported by the Wi-Fi module on the board. Applications that do not require SNI are not affected.

1.5 Intermittent failure for two concurrent connections

1.5.1. Issue

- Category: TCP/IP
- Test file: aws_test_tcp.c
- Tests: SECURE_SOCKETS_TwoSecureConnections

1.5.2. Description

The test attempts to make concurrent connections to a cloud endpoint and a secure echo server, it intermittently fails to connect to one of the secure endpoints.

1.5.3. Workaround

None. We are working to diagnose and fix this issue.

1.6 Failure connecting with ECDSA certificates

1.6.1. Issue

- Category: TLS
- Test file: aws_test_tls.c
- Tests: TLS_ConnectEC
  TLS_ConnectBYOCCredentials

1.6.2. Description

Socket connect failed while it was expected to succeed. It is not possible to use ECDSA certificate to connect to AWS IoT.

1.6.3. Workaround

Feedback from the vendor is that the Wi-Fi module's onboard TLS seems to have an issue with using ECDSA certificates to connect to AWS IoT. This is currently under investigation by the Wi-Fi module vendor. The BYOC test also fails as the BYOC test use ECDSA credentials.

In the interim, a possible workaround for this limitation is to not to offload TLS to the Wi-Fi module and use mbedTLS instead. Note that this would come at the cost of reduced performance.

NXP

LPC54018 IoT Module

1.7 Separate tasks connecting and disconnecting at once

1.7.1. Issue

- Category: Wi-Fi management
- Test file: aws_test_wifi.c
- Tests: WiFiSeperateTaskConnectingAndDisconnectingAtOnce

1.7.2. Description

This test is a thread safety test to check if the vendor supplied Wi-Fi port for the board is thread safe. It failed due to lack of thread protection.

1.7.3. Workaround

This limitation can potentially be handled by designing the application to not use the resource in more than one task. This is not an issue with the Amazon FreeRTOS supplied libraries.

1.8 Test taking a long time to execute

1.8.1. Issue

- Category: TCP/IP
- Test file: aws_test_wifi.c
- Tests: SOCKETS_Threadsafe_SameSocketDifferentTasks
  SECURE_SOCKETS_Threadsafe_SameSocketDifferentTasks

1.8.2. Description

This test took over 1 hour to complete. The large amount of time taken to complete the test is due to fetching code instructions out of an external flash over the SPI bus.

1.8.3. Workaround

A workaround for this is to have the application (code + data) fit into the first SRAM bank of the board.

1.9 Connecting to the cloud and secure echo server

1.9.1. Issue

- Category: TCP/IP
- Test file: aws_test_wifi.c
- Tests: SECURE_SOCKETS_TwoSecureConnections

1.9.2. Description

This test checks the connection from the board to the cloud and secure echo server. The test failed as it received bytes on a connection when there should have been none to receive.

1.9.3. Workaround

None. The vendor has acknowledged this issue in the porting code. We are working with the vendor to diagnose and fix this issue.

Microchip

PIC32MZEF bundle

1.10 Failure when switching between Access Points

1.10.1. Issue

- Category: Wi-Fi management
- Test file: aws_test_wifi.c
- Tests: WiFiConnectMultipleAP

1.10.2. Description

This test checks if the device can switch from one wireless Access Point (AP) to another. The test failed because the device did not set the IP address correctly after switching access points.

1.10.3. Workaround

For applications that do not require AP switching as a use case, this limitation is of less concern. Applications that can accommodate a device reset between access point switches may reset the device to avoid this issue.

1.11 Separate tasks connecting and disconnecting at once

1.11.1. Issue

- Category: Wi-Fi management
- Test file: aws_test_wifi.c
- Tests: WiFiSeperateTasksConnectingAndDisconnectingAtOnce

1.11.2. Description

This test is a thread safety test to check if the vendor supplied Wi-Fi port for the board is thread safe. It failed due to lack of thread protection.

1.11.3. Workaround

This limitation can potentially be handled by designing the application to not use the resource in more than one task. This is not an issue with the Amazon FreeRTOS supplied libraries.

1.12 Intermittent errors when multiple tasks access sockets

1.12.1. Issue

- Category: TLS
- Test file: aws_test_tls.c
- Tests: SECURE_SOCKETS_Threadsafe_DifferentSocketsDifferentTasks
  SECURE_SOCKETS_Threadsafe_SameSocketDifferentTasks

SOCKETS_Threadsafe_DifferentSocketsDifferentTasks
SOCKETS_Threadsafe_SameSocketDifferentTasks

1.12.2. Description

This is thread safety test to check the vendor supplied port for Secure Sockets on the board. It failed due to a lack of thread protection. We are working to resolve this issue.

1.12.3. Workaround

This limitation could potentially be handled in the application software by providing synchronized access to the shared connections. This is not an issue with the Amazon FreeRTOS supplied libraries.

Espressif

ESP32-WROVER-KIT and ESP DevKitC

1.13 Intermittent failure when switching between Access Points

1.13.1. Issue

- Category: Wi-Fi management
- Test file: aws_test_wifi.c
- Tests: WiFiConnectMultipleAP

1.13.2. Description

This test checks if the device can switch from one wireless Access Point (AP) to another. The test failed because the device did not set the IP address correctly after switching access points.

1.13.3. Workaround

For applications that do not require AP switching as a use case, this limitation is of less concern. Applications that can accommodate a device reset between access point switches may reset the device to avoid this issue.

