

**ePITome**

Getting Started Guide

|  |  |
| --- | --- |
| **Name** | Getting Started Guide |
| **Purpose** | The purpose of this document is to provide detailed instruction on how to get started with the ePITome PCBA. |
| **Target audience** | General public with firmware, FreeRTOS and electronics experience. |
| **Expected outcome** | By following this guide, you will be able to utilise FreeRTOS on this PCBA. |
| **version** | 1.1 |
| **Author** | Hayden |
| **Reviewer** | Sanket |

**Table of Contents**

[***1***](#_heading=h.30j0zll) ***Document Information 2***

[***2***](#_heading=h.1fob9te) ***Overview 2***

[***3***](#_heading=h.3znysh7) ***Hardware Description 2***

[***4***](#_heading=h.2et92p0) ***Set up your Development Environment 3***

[***5***](#_heading=h.tyjcwt) ***Set up your hardware 4***

[***6***](#_heading=h.3dy6vkm) ***Setup your AWS account and Permissions 4***

[***7***](#_heading=h.1t3h5sf) ***Provision the device with AWS IoT 5***

[***8***](#_heading=h.4d34og8) ***Download FreeRTOS 5***

[***9***](#_heading=h.2s8eyo1) ***Configure FreeRTOS 5***

[***10***](#_heading=h.17dp8vu) ***Build the FreeRTOS demo 5***

[***11***](#_heading=h.3rdcrjn) ***Run the FreeRTOS demo project 5***

[***12***](#_heading=h.26in1rg) ***Debugging 5***

[***13***](#_heading=h.lnxbz9) ***Troubleshooting 5***

# 

# Document Information

## Naming Conventions

## Glossary

## Revision History (Version, Date, Description of change)

# Overview

ePITome is the hardware component to Senstra’s real-time asset monitoring solution. ePITome leverages the power of IoT and smart sensors to communicate real-time telemetry and changes in the asset environment to deliver live data for machine learning, instant alerts or data through Senstra’s Android/iOS app.It provides visibility to assets that are usually unmonitored such as telecommunication pits, cabinets and manholes. It’s flexible, as it’s sensor and cloud platform are agnostic allowing you to monitor what you want, how you want.

Senstra’s real time asset monitoring solution allows businesses to be smarter and drive greater efficiencies through:

• Active Asset and environmental monitoring (with early event warnings).

• AI and machine learning with IoT for proactive maintenance to reduce maintenance costs.

• Accurate asset activity records.

• Logging of field staff attendance.

• Augmented Health & Safety for Field staff.

• Detecting unauthorized access.

• Reduce downtime and improved customer experience through real-time monitoring and field awareness.

# Hardware Description

## DataSheet

Refer to ePITome[*Technical Sheet*](https://www.senstra.com.au/assets/Brochure/Senstra_Brochure.pdf)*.*

## Key Components

## [Refer](https://docs.google.com/document/d/15SMcofkSRO6AwVAy7JCHNpyTu5YQOuJh2s9DZKahNsQ/edit#heading=h.tlpsz11pis5u) to ePITome [Set-up guide](https://github.com/senstraAU/ePITome/blob/29eb1ad742de021e8f0220d6afc76df4f31aa5db/ePITome%20-%20Set-Up%20Guide.docx).

## Hardware requirements to run FreeRTOS demo

### Standard Kit Contents provided:

* *1 x ePITome PCB*
* *1 x 2-Pin JST to bare-ends cable (For power).*
* *1 x getting started guide.*

**Optional:**

* *1 x ePITome Enclosure with battery pack (battery pack comes with 2-pin JST terminated.)*
* *Battery pack only (comes with 2-pin JST terminated.)*
* *Selection of compatible sensors*

### User Provided items

* *Micro-USB cable for debugging*
* *STM32 programmer ST-Link/V2 (Will require JTAG to SWD adapter + SWD-SWD cable)*

### 3rd Party purchasable items

*STM32 programmer -*[*https://au.rs-online.com/web/p/chip-programmers/8801599/?cm\_mmc=AU-PLA-DS3A-\_-google-\_-PLA\_AU\_EN\_Semiconductors\_Whoop-\_-(AU:Whoop!)+Chip+Programmers-\_-8801599&matchtype=&pla-339373444555&gclid=Cj0KCQiA-K2MBhC-ARIsAMtLKRsDV6evDQNgVj\_oKhyW4POUIbA0cDo5qoCvjAqEveSF1xTCxZDMlQkaAvfoEALw\_wcB&gclsrc=aw.ds*](https://au.rs-online.com/web/p/chip-programmers/8801599/?cm_mmc=AU-PLA-DS3A-_-google-_-PLA_AU_EN_Semiconductors_Whoop-_-(AU:Whoop!)+Chip+Programmers-_-8801599&matchtype=&pla-339373444555&gclid=Cj0KCQiA-K2MBhC-ARIsAMtLKRsDV6evDQNgVj_oKhyW4POUIbA0cDo5qoCvjAqEveSF1xTCxZDMlQkaAvfoEALw_wcB&gclsrc=aw.ds)

*JTAG (2x10 2.54mm) to SWD (2x5 1.27mm) Cable Adapter Board-*

[*https://core-electronics.com.au/jtag-2x10-2-54mm-to-swd-2x5-1-27mm-cable-adapter-board.html?utm\_source=google\_shopping&gclid=Cj0KCQiA-K2MBhC-ARIsAMtLKRvS7WcbtfXhnN7RgCCYG9QC3aeuer1zOaaYJfMrd4qjEo5NvnmMHUoaAocTEALw\_wcB*](https://core-electronics.com.au/jtag-2x10-2-54mm-to-swd-2x5-1-27mm-cable-adapter-board.html?utm_source=google_shopping&gclid=Cj0KCQiA-K2MBhC-ARIsAMtLKRvS7WcbtfXhnN7RgCCYG9QC3aeuer1zOaaYJfMrd4qjEo5NvnmMHUoaAocTEALw_wcB)

*10-pin 2x5 Socket-Socket 1.27mm IDC (SWD) Cable - 150mm long-*[*https://core-electronics.com.au/10-pin-2x5-socket-socket-1-27mm-idc-swd-cable-150mm-long.html*](https://core-electronics.com.au/10-pin-2x5-socket-socket-1-27mm-idc-swd-cable-150mm-long.html)

## Additional Hardware References

[*https://www.senstra.com.au/#/epitome*](https://www.senstra.com.au/#/epitome)

# Set up your Development Environment

## Supported IDEs

* Supported IDE for ePITome board is System workbench. The System Workbench toolchain, called SW4STM32, is a free multi-OS software development environment based on Eclipse, which supports the full range of STM32 microcontrollers.
* The SW4STM32 toolchain can be obtained from <https://www.st.com/en/development-tools/sw4stm32.html>
* Once registered to [www.openstm32.org](https://www.openstm32.org/Installing%2BSystem%2BWorkbench%2Bfor%2BSTM32%2Bwith%2Binstaller) site, users will get installation instructions at the Documentation > System Workbench page to proceed with the download of the free toolchain. This site also includes forums, blogs, and training for technical support.

1. *CLI based (e.g. CMake)*

Refer to [Using CMake with FreeRTOS](https://docs.aws.amazon.com/freertos/latest/userguide/getting-started-cmake.html) for prerequisites.

* Download **cmake** from <https://cmake.org/download/> and install it.
* Add cmake into the PATH environment variable.
* Download [GNU Arm Embedded Toolchain](https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-rm/downloads) for your operating system.
* (Optional) Add path to GNU Arm Embedded Toolchain into the PATH environment variable.
* You must have a compatible native build system.

CMake can target many native build systems, including [GNU Make](https://www.gnu.org/software/make/) or [Ninja](https://github.com/ninja-build/ninja/releases). Both Make and Ninja can be installed with package managers on Linux, macOS and Windows. If you are using Make on Windows, you can install a standalone version from [Equation](http://www.equation.com/servlet/equation.cmd?fa=make), or you can install [MinGW](https://sourceforge.net/projects/mingw-w64/files/), which bundles make.

* NOTE: The Make executable in MinGW is called *mingw32-make.exe*, instead of *make.exe*.
* We recommend that you use Ninja, as it is faster than Make and also provides native support to all desktop operating systems.

## Generating build files (CMake command-line tool)

* Use the following command to generate build files:

cmake -DVENDOR=senstra -DBOARD=epitome \

-DCOMPILER=arm-gcc -S . -B <build-directory>

* If toolchain is not into the system PATH variable then specify it with -DAFR\_TOOLCHAIN\_PATH="<toolchain-path>"
* By default Demos will be built, To build Tests add *-DAFR\_ENABLE\_TESTS=1* into the build command.

## Building FreeRTOS from generated build files

* Use the CMake command-line tool to build FreeRTOS with the following command:

cmake --build *<build-directory>*

* The build command will generate .hex and .elf file into the build directory.
* For Demos file name is aws\_demos while aws\_tests for Tests, depending on the selection you made while generating build files.

## Flashing FreeRTOS build files:

* Install STM32 Cube programmer.
* Open terminal and change directory to STM32 Cube programmer’s installed location.
* Use below command to flash the build file into the device

STM32\_Programmer\_CLI -c port=SWD freq=4000 mode=UR -d “<full\_path\_to\_built\_hex\_file>” -v -rst

## Toolchains

* STM32 Cube programmer: <https://www.st.com/content/st_com/en/products/development-tools/software-development-tools/stm32-software-development-tools/stm32-programmers/stm32cubeprog.html>
* ST-Link driver:

<https://www.st.com/content/st_com/en/products/development-tools/software-development-tools/stm32-software-development-tools/stm32-utilities/stsw-link009.html>

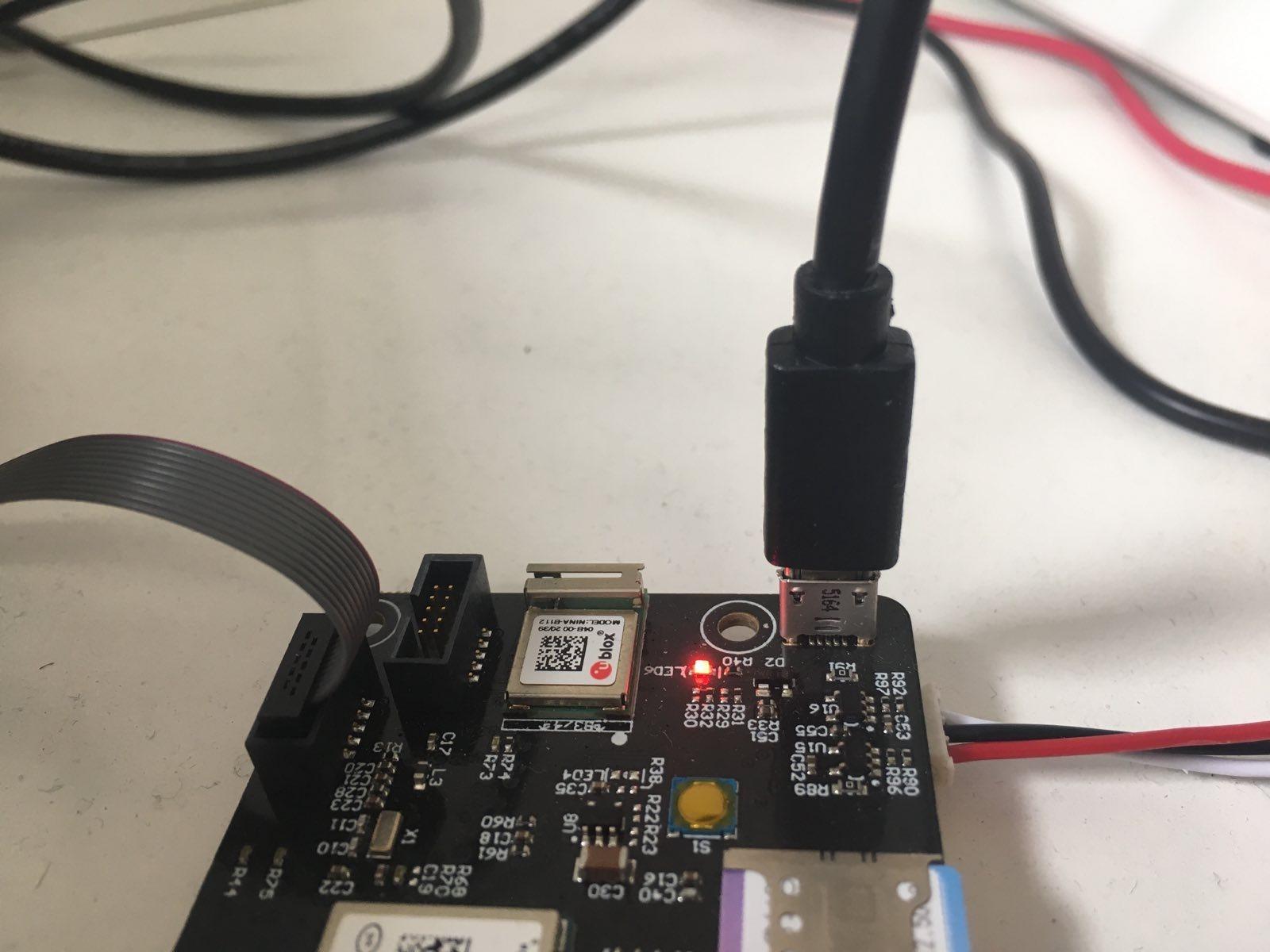
## Establishing a serial connection

Serial terminal emulators that can be used with ePITome board are Putty, Docklight, Hercules etc.

* Download Putty: <https://www.putty.org/>
* Download Docklight : <https://docklight.de/downloads/>
* Download Hercules: <https://www.hw-group.com/software/hercules-setup-utility>

Instruction for debug probe setup:

Connect the USB-A to the system and connect micro usb to the ePITome board as shown below



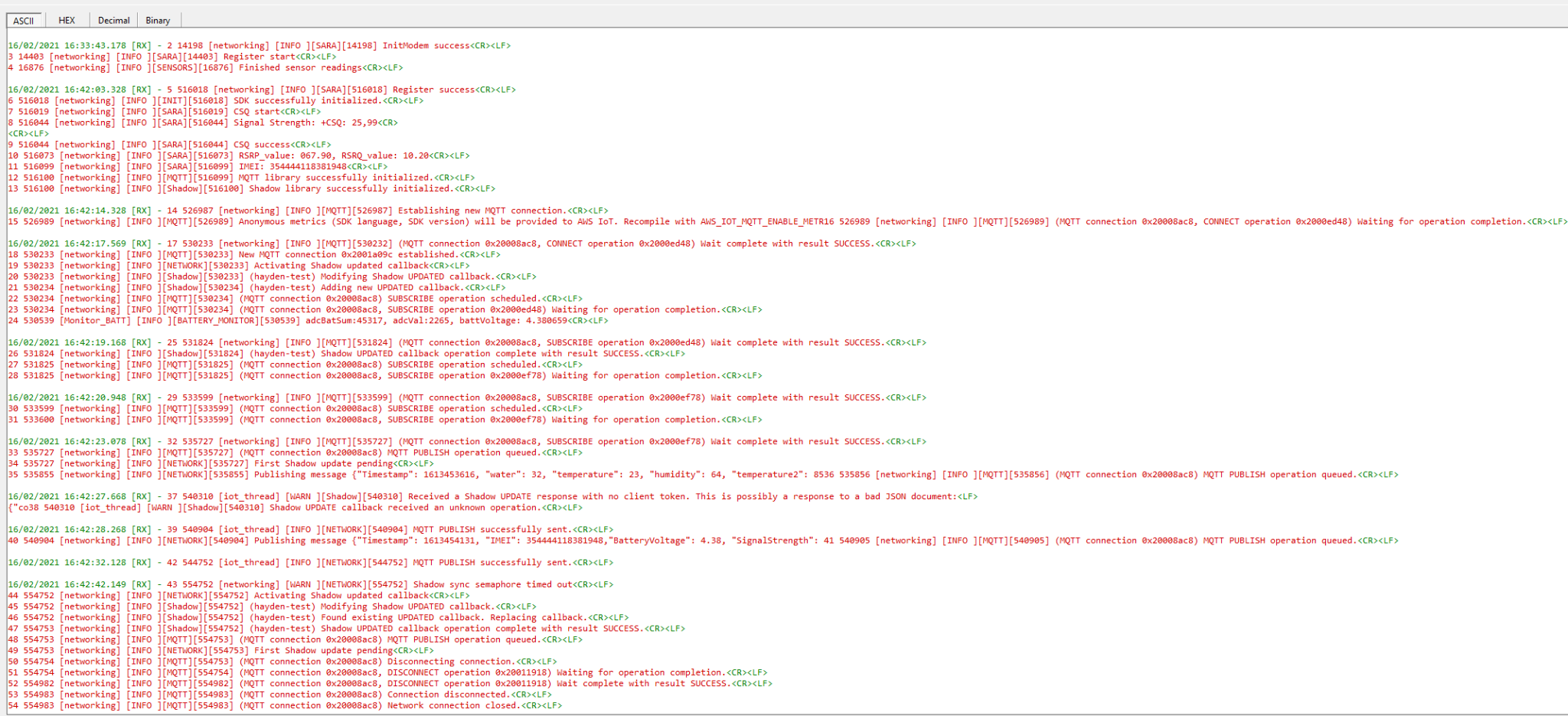
Instructions for terminal emulator:

Step 1: Open terminal emulator of your choice and create a new blank project

Step 2: Open serial project settings and set the following parameters

* Select the COM port shown by the system once the debug cable is connected to ePITome board and the system
* Select baud rate: 115200
* Select data bits: 8
* Select stop bits: 1

Step 3: Start communication



# Set up your hardware

## [Refer](https://docs.google.com/document/d/15SMcofkSRO6AwVAy7JCHNpyTu5YQOuJh2s9DZKahNsQ/edit#heading=h.tlpsz11pis5u) to ePITome [Set-up guide](https://github.com/senstraAU/ePITome/blob/29eb1ad742de021e8f0220d6afc76df4f31aa5db/ePITome%20-%20Set-Up%20Guide.docx).

# Setup your AWS account and Permissions

To create an AWS account, see [Create and Activate an AWS Account](https://aws.amazon.com/premiumsupport/knowledge-center/create-and-activate-aws-account/)

To add an IAM user to your AWS account, see [IAM User Guide](https://docs.aws.amazon.com/IAM/latest/UserGuide/). To grant your IAM user account access to AWS IoT and FreeRTOS, attach the following IAM policies to your IAM user account:

* AmazonFreeRTOSFullAccess
* AWSIoTFullAccess

*NOTE – The examples in this document are intended only for dev environments. All devices in your production fleet must have credentials with privileges that authorize only intended actions on specific resources. The specific permission policies can vary for your use case. Identify the permission policies that best meet your business and security requirements. For more information, refer to* [*Example policies*](https://docs.aws.amazon.com/iot/latest/developerguide/example-iot-policies.html) *and* [*Security Best practices*](https://docs.aws.amazon.com/iot/latest/developerguide/security-best-practices.html)*.*

**To attach the AmazonFreeRTOSFullAccess policy to your IAM user**

1. Browse to the [IAM console](https://console.aws.amazon.com/iam/home), and from the navigation pane, choose **Users**.
2. Enter your user name in the search text box, and then choose it.
3. Choose **Add permissions**.
4. Choose **Attach existing policies directly**.
5. In the search box, enter AmazonFreeRTOSFullAccess, choose it, and then choose **Next: Review**.
6. Choose **Add permissions**.

**To attach the AWSIoTFullAccess policy to your IAM user**

1. Browse to the [IAM console](https://console.aws.amazon.com/iam/home), and from the navigation pane, choose **Users**.
2. Enter your user name in the search text box, and then choose it.
3. Choose **Add permissions**.
4. Choose **Attach existing policies directly**.
5. In the search box, enter AWSIoTFullAccess, choose it, and then choose **Next: Review**.
6. Choose **Add permissions**.

For more information about IAM and user accounts, see [IAM User Guide](https://docs.aws.amazon.com/IAM/latest/UserGuide/).

For more information about policies, see [IAM Permissions and Policies](https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction_access-management.html).

# Provision the device with AWS IoT

Refer to [Register your board manually](https://docs.aws.amazon.com/freertos/latest/userguide/freertos-prereqs.html#get-started-freertos-thing).

Follow steps 1-6 under the heading **To create an AWS IoT policy**. In step 1, note that the AWS region for your account can also be found in the drop-down between the account name and Support drop-downs in the top menu bar.

Follow steps 1-10 under the heading **To create an IoT thing, private key, and certificate for your device**.

# Download FreeRTOS

*You can use* [*https://docs.aws.amazon.com/freertos/latest/userguide/freertos-download.html*](https://docs.aws.amazon.com/freertos/latest/userguide/freertos-download.html) *as a guide*

* To clone using HTTPS:

git clone https://github.com/aws/amazon-freertos.git \ -recurse-submodules

* Using SSH:

git clone git@github.com:aws/amazon-freertos.git --recurse-submodules

* If you have downloaded the repo without using the **--recurse-submodules** argument, you need to run:

git submodule update --init --recursive

# Configure FreeRTOS

Follow the instructions under the heading [Configuring the FreeRTOS demos](https://docs.aws.amazon.com/freertos/latest/userguide/freertos-prereqs.html#get-started-freertos-thing)

# Build the FreeRTOS demo

* Open STM32 SystemWorkbench.
* From the menu bar **File -> Open Projects from file system**
* Choose Directory as <FreeRTOS\_path>/projects/senstra/epitome/ac6/aws\_demos
* Click on **Finish** to open the project.
* Select the project from the Project Explorer view.
* From the menu bar **Project -> Build Project**
* On successful build aws\_demos.elf file will be created inside <FreeRTOS\_path>/projects/senstra/epitome/ac6/aws\_demos/<Debug/Release> directory, depending on the selection of build type Debug or Release.

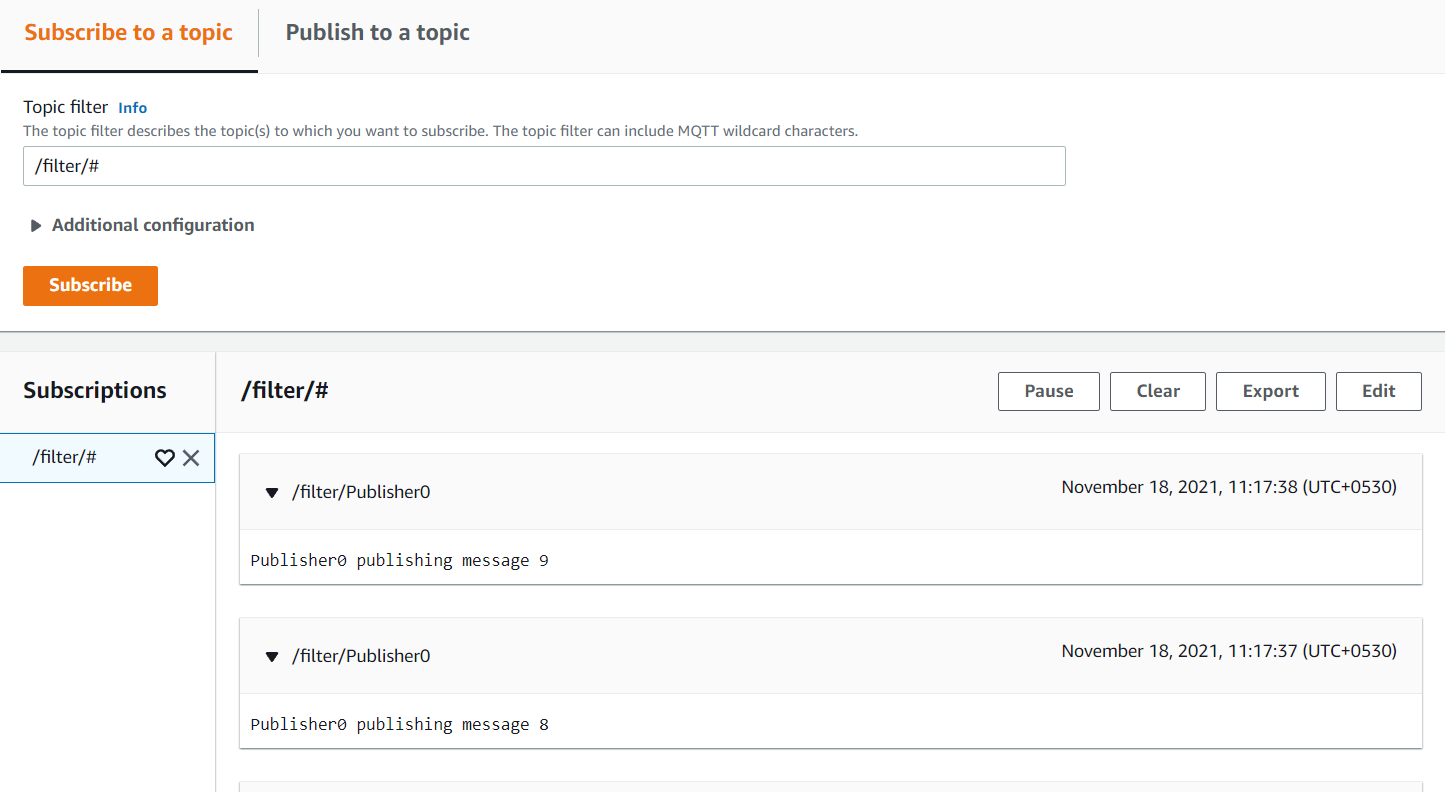
# Run the FreeRTOS demo project

* Open the project as described in the section **10 Build the FreeRTOS demo**
* Follow instruction from [Configuring the FreeRTOS demos](https://docs.aws.amazon.com/freertos/latest/userguide/freertos-prereqs.html#freertos-configure) to set up thing-name and certificates to connect to your AWS account for the demos.
* To build MQTT demo application enable CONFIG\_CORE\_MQTT\_AGENT\_DEMO\_ENABLED in config\_fles/aws\_demo\_config.h
* Define root CA in mqtt\_agent\_demo\_config.h under config\_files as democonfigROOT\_CA\_PEM
* Build the project using **Project->Build Project** from the menu bar.
* Run the project by **Run->Run** from the menu bar.
* Open Serial terminal of your choice to view the logs, as described in the section **4.6 Establishing a serial connection**



## Monitoring MQTT message on the cloud

* Sign in to AWS IoT console
* In the navigation pane choose **Test** to open MQTT client.
* In **Subscribe to a topic** panel enter /filter/# as topic filter.
* Under **Additional configuration** section choose option **Display payloads as strings (more accurate)** under section **MQTT payload display**.
* Click on **Subscribe** to start monitoring messages.



# Debugging

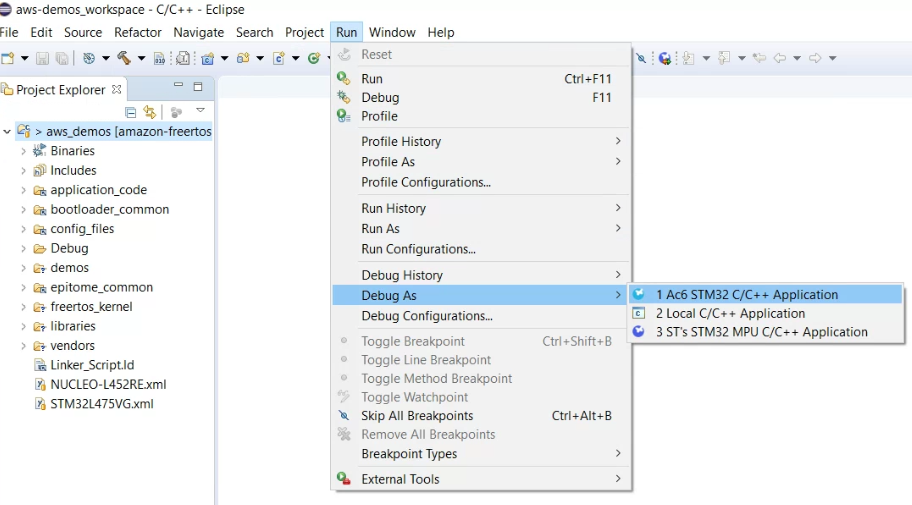
ePITome device does run execution logs and provides state of the device over USB. It provides run time boot up logs with appropriate initialization state of each module, that points out any error in any process or exception with details. For step by step instructions of setting up serial port monitor and to view device logs please refer to section 4.6

## Debugging with JTAG debugger

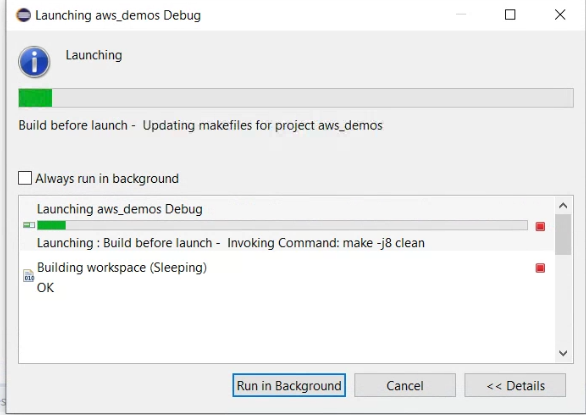
* + 1. Prerequisites
* *STM32 programmer (see section 3.3.3)*
* *JTAG (2x10 2.54mm) to SWD (2x5 1.27mm) Cable Adapter Board (see section 3.3.3)*
* *10-pin 2x5 Socket-Socket 1.27mm IDC (SWD) Cable - 150mm long (see section 3.3.3)*

## Debugging using STM32 SystemWorkbench

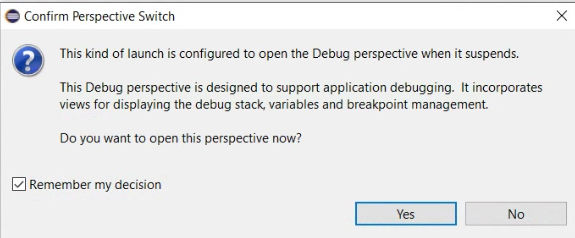
* Connect STM32 programmer to System and ePITome board.
* Open the project as described in the section **10 Build the FreeRTOS demo**
* From the menu bar select **Run->Debug as->Ac6 STM32 C/C++ Application**



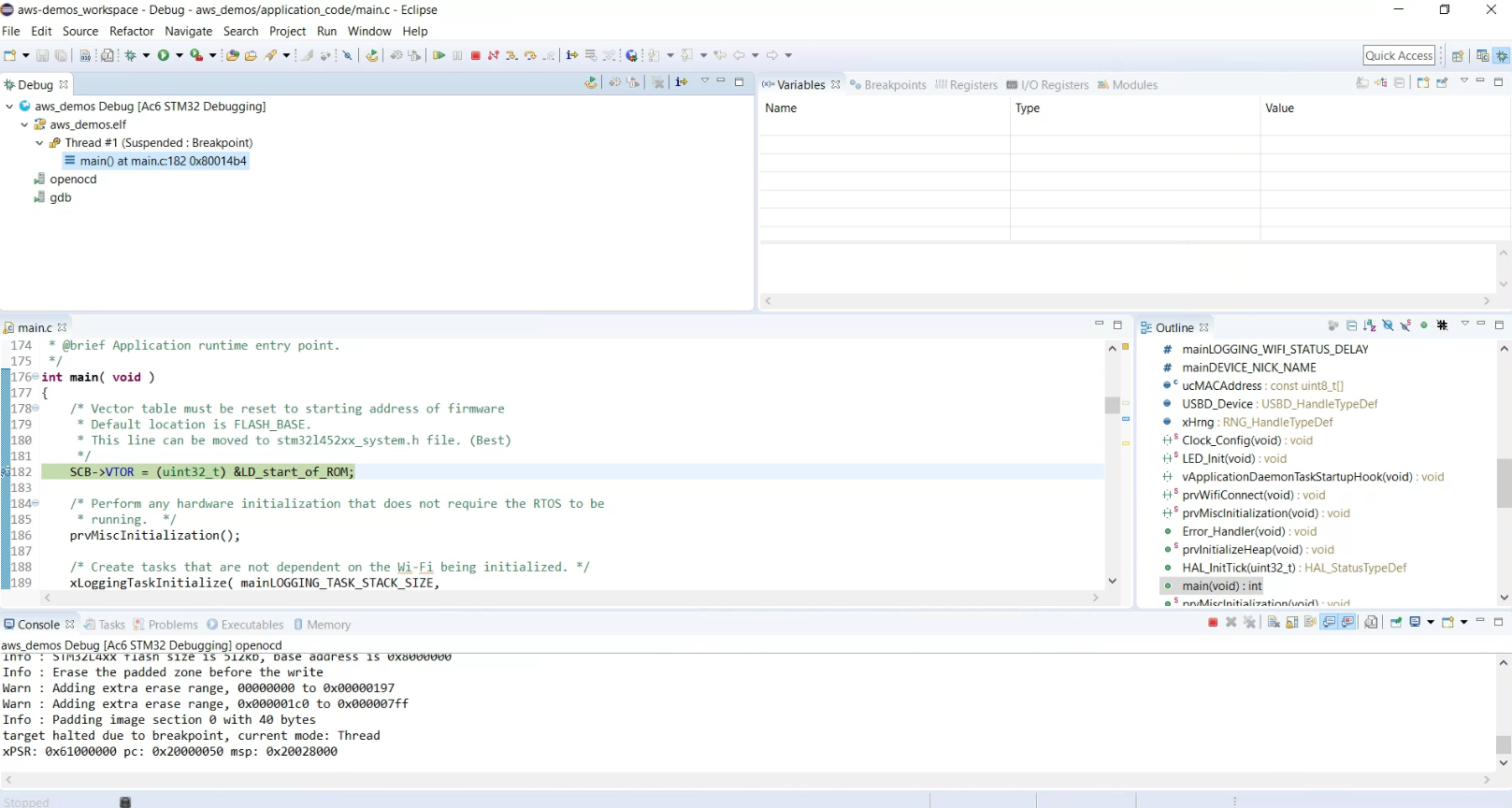
* Wait until the project builds and starts debugging



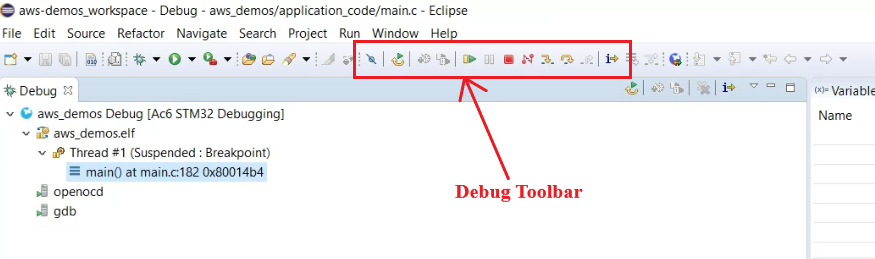
* On prompt to switch Perspective choose **Yes**



* Wait for the debugging to start. Debugging process will start and pause at the first line of the **main** function.



* Use the Debug Toolbar to navigate through code as needed.



# Troubleshooting

General troubleshooting for ePITome board

* If ePITome board does not power up after power supply connection, make sure you have provided the appropriate supply voltage to the board as specified in the [Set-up guide](https://github.com/senstraAU/ePITome/blob/29eb1ad742de021e8f0220d6afc76df4f31aa5db/ePITome%20-%20Set-Up%20Guide.docx).
* If ePITome board is not able to connect to the network, make sure to check the antenna interface connection on the board, and the correct sim card placement.

For general troubleshooting information about Getting Started with FreeRTOS, see [Troubleshooting Getting Started](https://docs.aws.amazon.com/freertos/latest/userguide/gsg-troubleshooting.html).