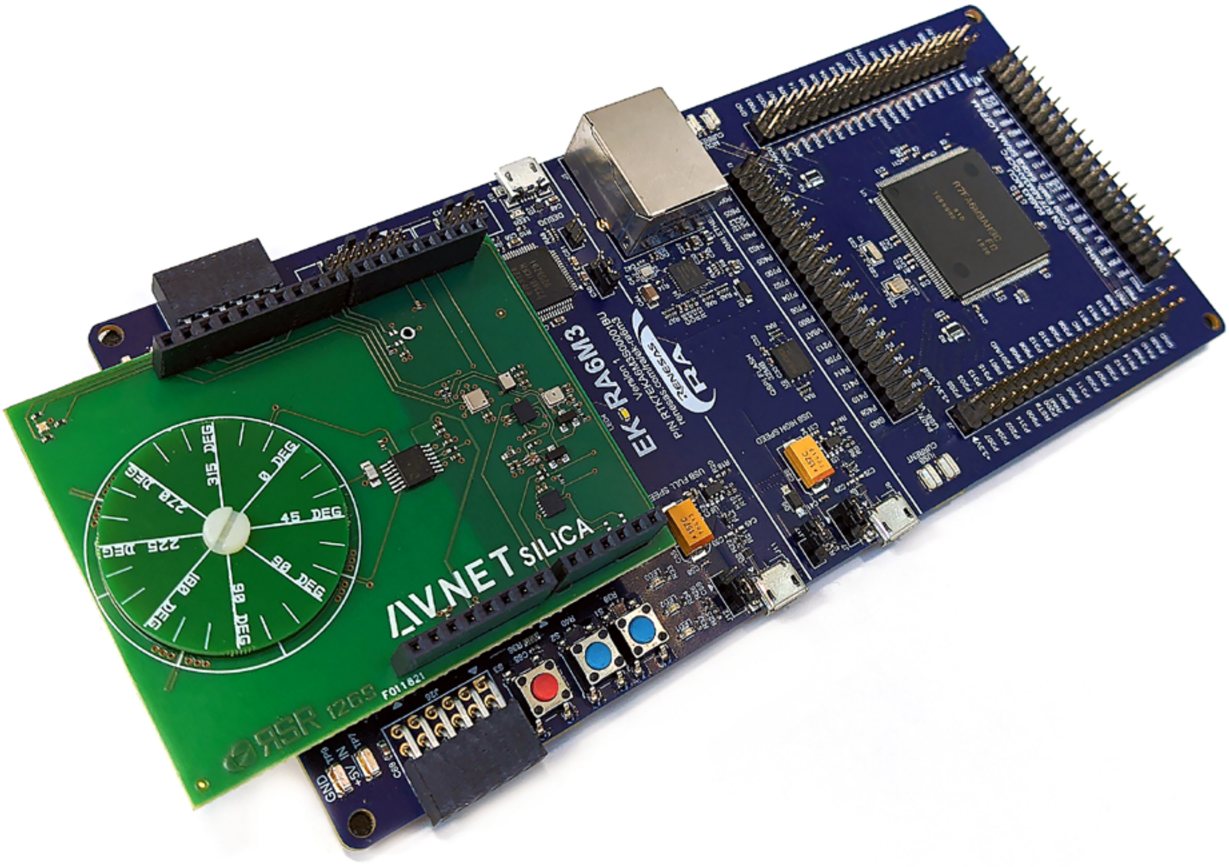
HoriZoneRA IoT Connect 1269



SUPPORTING PARTNERS:

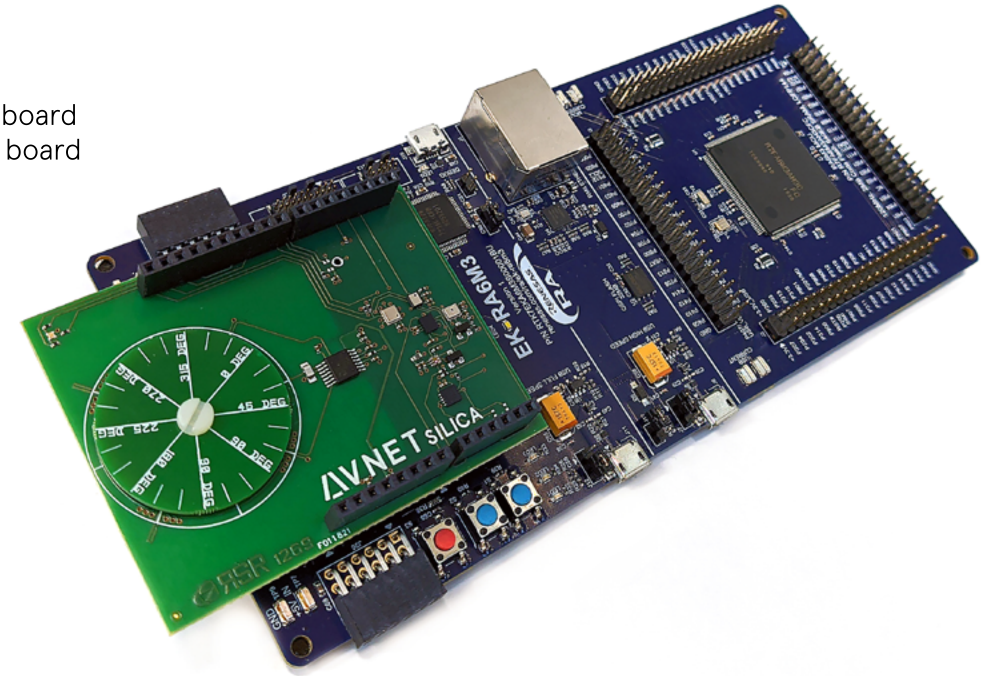
Contents

/IOTCONNECT

Hardware requirements

HARDWARE SETUP

- EK-RA6M3 development board
- Avnet Silica 1269 sensor's board
- Network cable
- USB Micro B cable

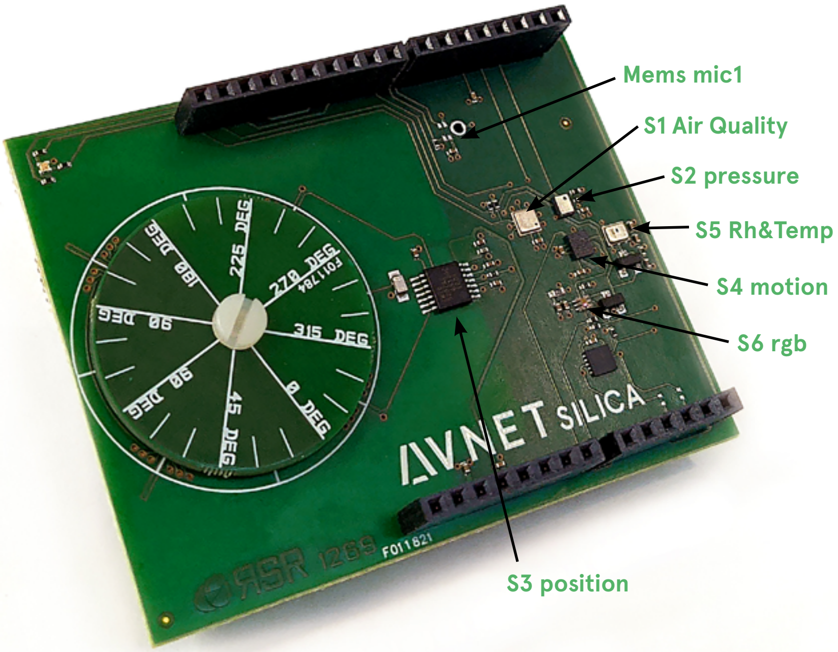


Plug the 1269 board into the EK-RA6M3's Arduino shield.

Plug in the network cable and connect to your own network switch (internet connection).

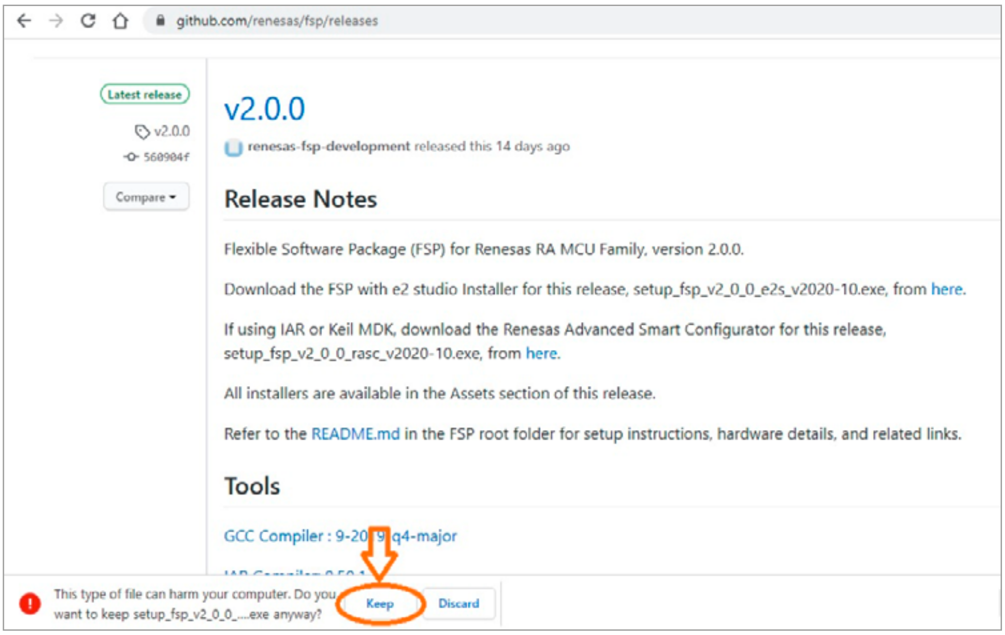
Plug in the USB micro B and connect to the PC's USB port.

SENSOR MAP



Software requirements

Download the FSP with E2-Studio 2020-10 installer setup_fsp_v2_0_0_e2s_v2020-10.exe from here.
Click on "Keep" button to start download



v2.0.0

renesas-fsp-development released this 14 days ago

Release Notes

Flexible Software Package (FSP) for Renesas RA MCU Family, version 2.0.0.

Download the FSP with e2 studio Installer for this release, setup_fsp_v2_0_0_e2s_v2020-10.exe, from here.

If using IAR or Keil MDK, download the Renesas Advanced Smart Configurator for this release, setup_fsp_v2_0_0_rasc_v2020-10.exe, from here.

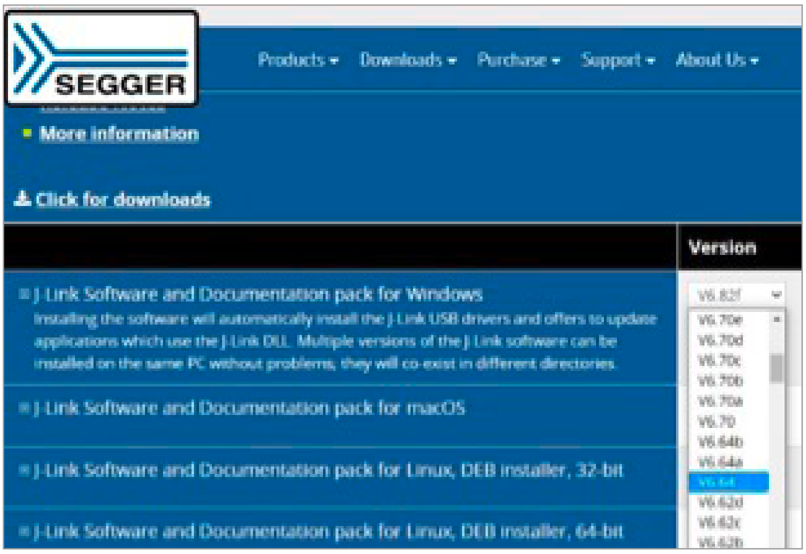
All installers are available in the Assets section of this release.

Refer to the README.md in the FSP root folder for setup instructions, hardware details, and related links.

Tools

GCC Compiler : 9-2019-q4-major

This type of file can harm your computer. Do you want to keep setup_fsp_v2_0_0_....exe anyway?



J-Link Software and Documentation pack for Windows
Installing the software will automatically install the J-Link USB drivers and offers to update applications which use the J-Link DLL. Multiple versions of the J-Link software can be installed on the same PC without problems, they will co-exist in different directories.

J-Link Software and Documentation pack for macOS

J-Link Software and Documentation pack for Linux, DEB installer, 32-bit

J-Link Software and Documentation pack for Linux, DEB installer, 64-bit

Segger J-Link Software pack 6.64 from here.

HoriZoneRA features

FIRMWARE DETAILS BRIEF

Request the design files here:
https://www.avnet.com/wps/portal/silica/products/product-highlights/2020/horizone-renesas-ra/

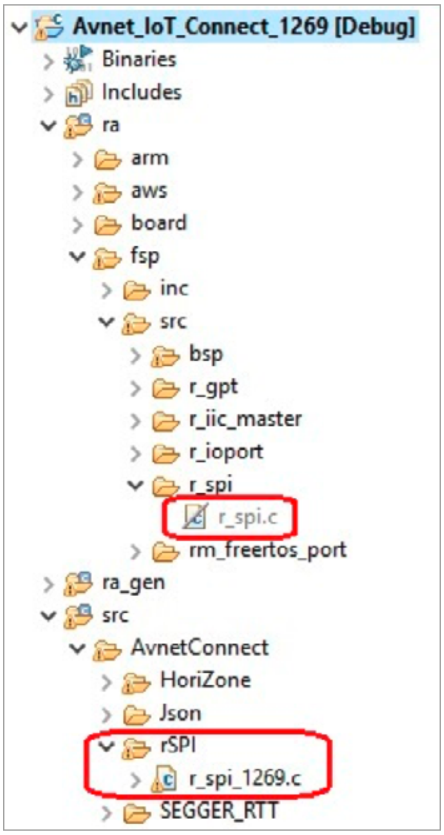
The project comes with all the functionality you need to configure and read data from 1269B board's sensors:

- Accelerometer and gyroscope
- Position sensor (with rotary encoder)
- Light sensor
- Humidity and temperature
- Air pressure
- Noise microphone

The S1 air quality sensor requires a customer NDA agreement; no data is available for this sensor.

The noise microphone uses a particular set of SPI slaves:

- Continuous generated clock (768Khz) using PWM out
- Continuous SPI RXI Irq to sample microphone data
- r_spi.c library file replaced by custom r_spi_1269.c file



```
v Avnet_IoT_Connect_1269 [Debug]
  > Binaries
  > Includes
  v ra
    > arm
    > aws
    > board
    v fsp
      > inc
      v src
        > bsp
        > r_gpt
        > r_iic_master
        > r_ioport
        v r_spi
            r_spi.c
        > rm_freertos_port
  > ra_gen
  v src
    v AvnetConnect
      > HoriZone
      > Json
      v rSPI
        > r_spi_1269.c
      > SEGGER_RTT
```

HoriZoneRA features

FIRMWARE STRUCTURE

The image on the right shows the project structure.

HoriZone folder (circled in green)
This folder contains all the files and functions you need for the IoTConnect portal connection.

It includes 3 files:

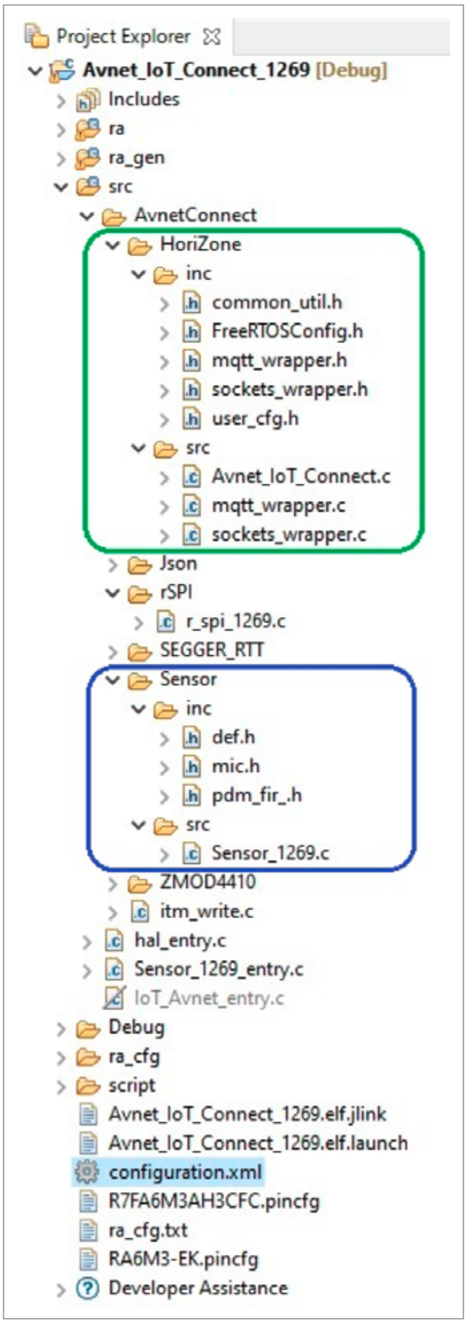
- Avnet_Iot_Connect.c: all application layer functions for IP, MQTT and data formatting in order to send data to backend
- mqtt_wrapper.c, sockets_wrapper.c: middle layer functions to interface with IoT application and functional drivers.

Sensor folder (circled in blue)
This folder contains all the files and functions you need for sensor configuration and data reading:

- Sensor_1269.c: has all drivers and functions to configure driver interface and sensor devices and to read the sensor's data.

The ZMODAir.c is an empty file because the use of this device requires an NDA agreement directly with the end user. Please refer to your Avnet vendor for more information about this.

The IoT_Avnet_entry.c is an autogenerated thread file that is also modified to call the main connection function an Avnet_IoT_Connect.c file. This file is excluded from the building process and will be automatically reincluded once the communication layer has been added.



```
Project Explorer
Avnet_IoT_Connect_1269 [Debug]
  Includes
  ra
  ra_gen
  src
    AvnetConnect
      HoriZone
        inc
          common_util.h
          FreeRTOSConfig.h
          mqtt_wrapper.h
          sockets_wrapper.h
          user_cfg.h
        src
          Avnet_IoT_Connect.c
          mqtt_wrapper.c
          sockets_wrapper.c
      Json
      rSPI
        r_spi_1269.c
      SEGGER_RTT
      Sensor
        inc
          def.h
          mic.h
          pdm_fir_.h
        src
          Sensor_1269.c
      ZMOD4410
      itm_write.c
    hal_entry.c
    Sensor_1269_entry.c
    IoT_Avnet_entry.c
  Debug
  ra_cfg
  script
  Avnet_IoT_Connect_1269.elf.jlink
  Avnet_IoT_Connect_1269.elf.launch
  configuration.xml
  R7FA6M3AH3CFC.pincfg
  ra_cfg.txt
  RA6M3-EK.pincfg
  Developer Assistance
```

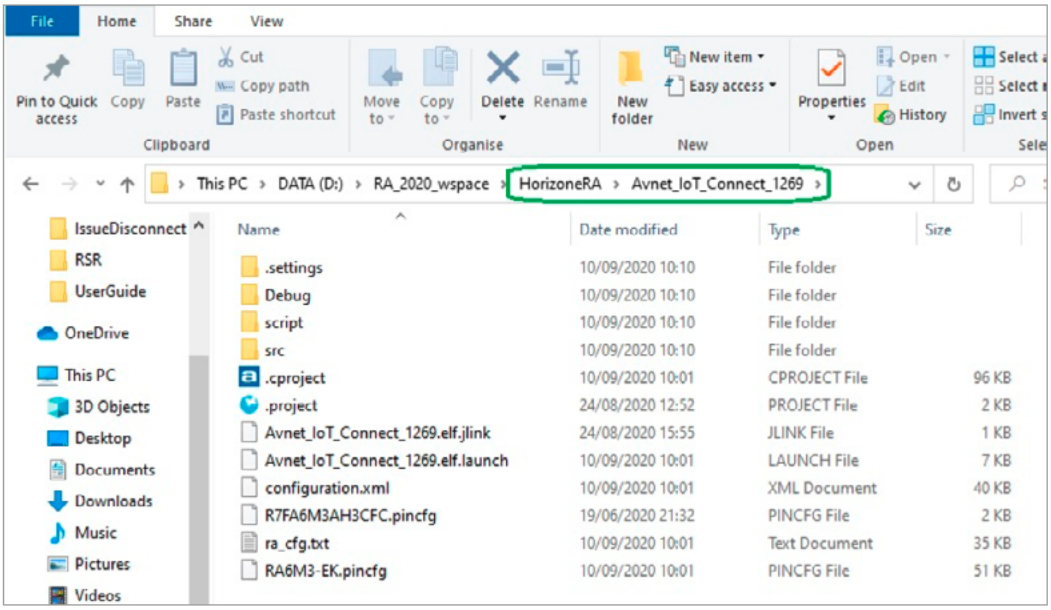
The next steps in this guide will provide you with the instructions you need to be able to add and configure the IoT Connect communication layer. It also gives you a brief guide on how create your own device (to see sent data) on the Avnet IoTConnect Portal.

Please refer to the page Avnet IoT Connect Partner Program (https://partner.iotconnect.io/) on the Avnet website for further details and to get your access credentials.

HoriZoneRA project

STEP 1 – PREPARE WORKSPACE

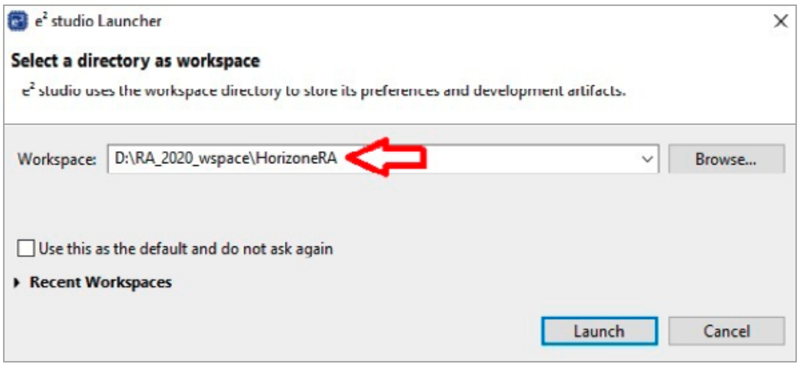
Create a workspace folder named HoriZoneRA. Extract the project here from the HoriZoneRA_1.0.0.zip file and copy it into the workspace folder. Project Avnet_Iot_Connect_1269 is now ready for the next steps.



This PC › DATA (D:) › RA_2020_wspace › HorizoneRA › Avnet_IoT_Connect_1269 ›

Name	Date modified	Type	Size
.settings	10/09/2020 10:10	File folder	
Debug	10/09/2020 10:10	File folder	
script	10/09/2020 10:10	File folder	
src	10/09/2020 10:10	File folder	
.cproject	10/09/2020 10:01	CPROJECT File	96 KB
.project	24/08/2020 12:52	PROJECT File	2 KB
Avnet_IoT_Connect_1269.elf.jlink	24/08/2020 15:55	JLINK File	1 KB
Avnet_IoT_Connect_1269.elf.launch	10/09/2020 10:01	LAUNCH File	7 KB
configuration.xml	10/09/2020 10:01	XML Document	40 KB
R7FA6M3AH3CFC.pincfg	19/06/2020 21:32	PINCFG File	2 KB
ra_cfg.txt	10/09/2020 10:01	Text Document	35 KB
RA6M3-EK.pincfg	10/09/2020 10:01	PINCFG File	51 KB

Open E2 Studio 2020, then select the workspace you just created and click the "Launch" button.



e² studio Launcher

Select a directory as workspace

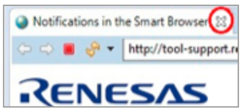
e² studio uses the workspace directory to store its preferences and development artifacts.

Workspace: D:\RA_2020_wspace\HorizoneRA

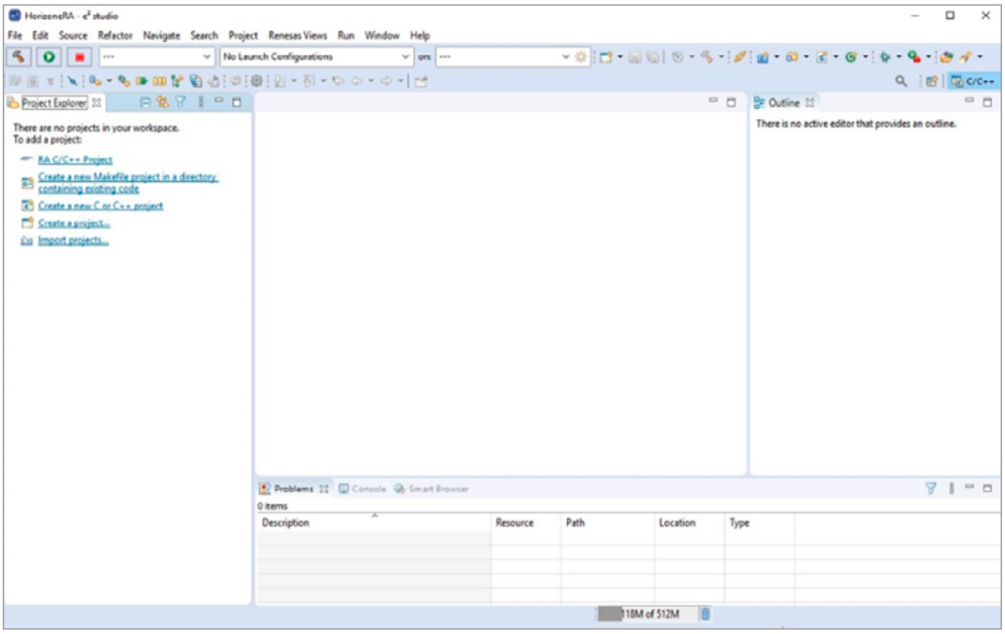
□ Use this as the default and do not ask again

› **Recent Workspaces**

Close all welcome windows and other popups, then close the Notification window in the Smart Browser tab (click the X on upper right).



The E2 Studio workspace is now ready.



There are no projects in your workspace.
To add a project:

- RA C/C++ Project
- Create a new Makefile project in a directory containing existing code
- Create a new C or C++ project
- Create a project...
- Import projects...

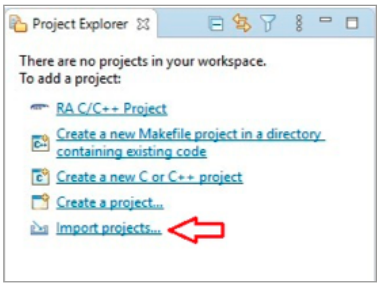
There is no active editor that provides an outline.

0 items

Description	Resource	Path	Location	Type

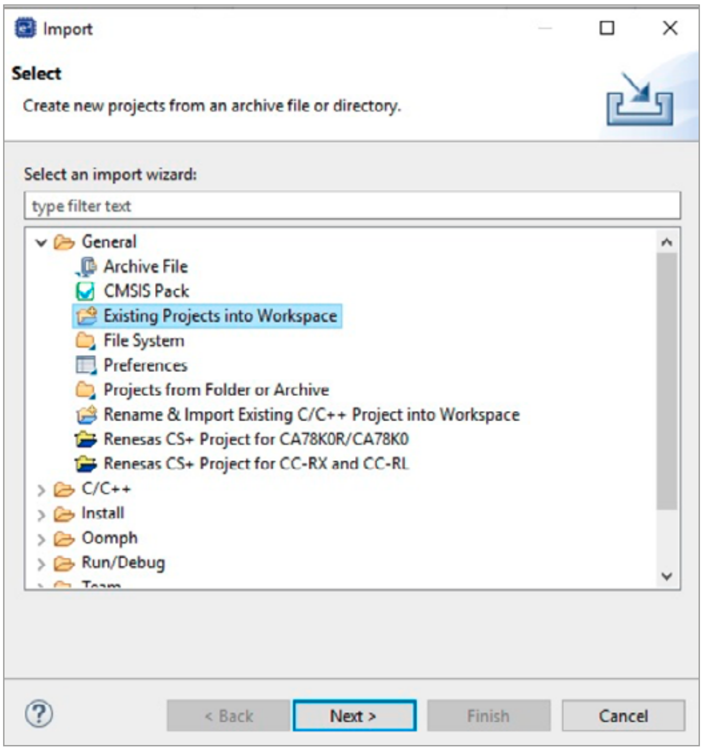
STEP 2 – IMPORT PROJECT

Click on "Import projects" in the Project Explorer tab Select General > Existing Projects into Workspace, then click "Next".



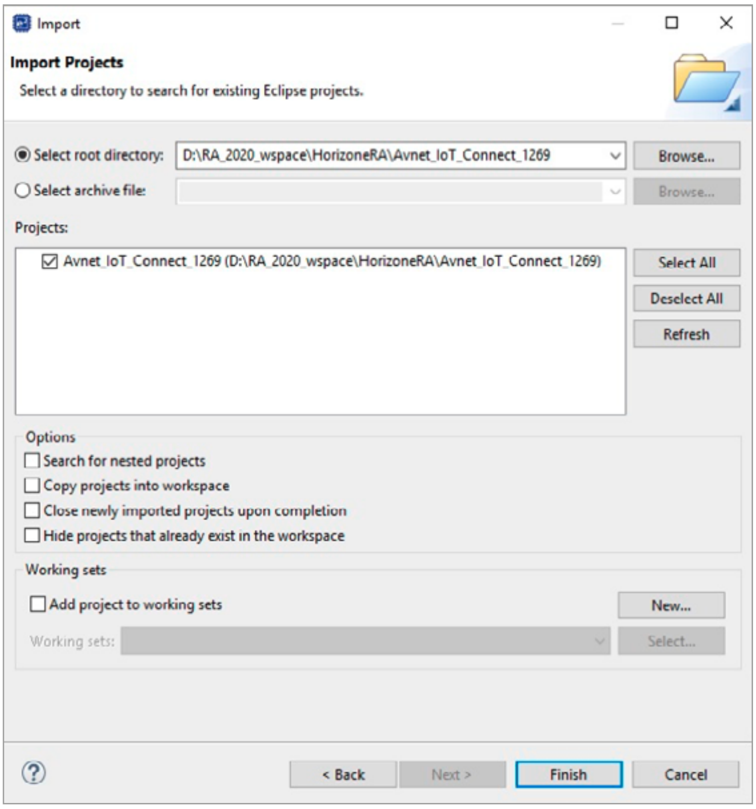
There are no projects in your workspace.
To add a project:

- RA C/C++ Project
- Create a new Makefile project in a directory containing existing code
- Create a new C or C++ project
- Create a project...
- Import projects...


Import
Select
Select an import wizard:
type filter text
General
Archive File
CMSIS Pack
Existing Projects into Workspace
File System
Preferences
Projects from Folder or Archive
Rename & Import Existing C/C++ Project into Workspace
Renesas CS+ Project for CA78K0R/CA78K0
Renesas CS+ Project for CC-RX and CC-RL
C/C++
Install
Oomph
Run/Debug
Team
< Back
Next >
Finish
Cancel

Create new projects from an archive file or directory.

Go to the project folder and select Avnet_IoT_Connect_1269, then click “Finish”.



● Select root directory: D:\RA_2020_wspace\HorizoneRA\Avnet_IoT_Connect_1269

○ Select archive file:

Projects:

☑ Avnet_IoT_Connect_1269 (D:\RA_2020_wspace\HorizoneRA\Avnet_IoT_Connect_1269)

Options

□ Search for nested projects

□ Copy projects into workspace

□ Close newly imported projects upon completion

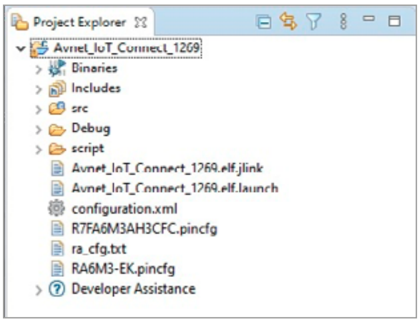
□ Hide projects that already exist in the workspace

Working sets

□ Add project to working sets

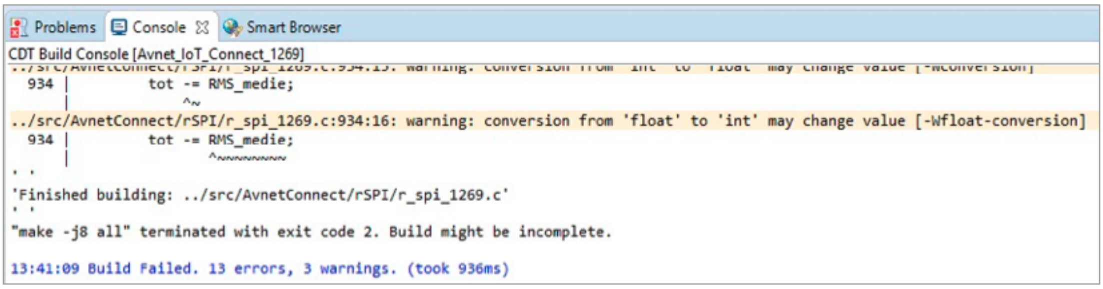
Working sets:

The project is now imported and you should see the project structure in the Project Explorer window.



```
Avnet_IoT_Connect_1269
  Binaries
  Includes
  src
  Debug
  script
  Avnet_IoT_Connect_1269.elf.jlink
  Avnet_IoT_Connect_1269.elf.launch
  configuration.xml
  R7FA6M3AH3CFC.pincfg
  ra_cfg.txt
  RA6M3-EK.pincfg
  Developer Assistance
```

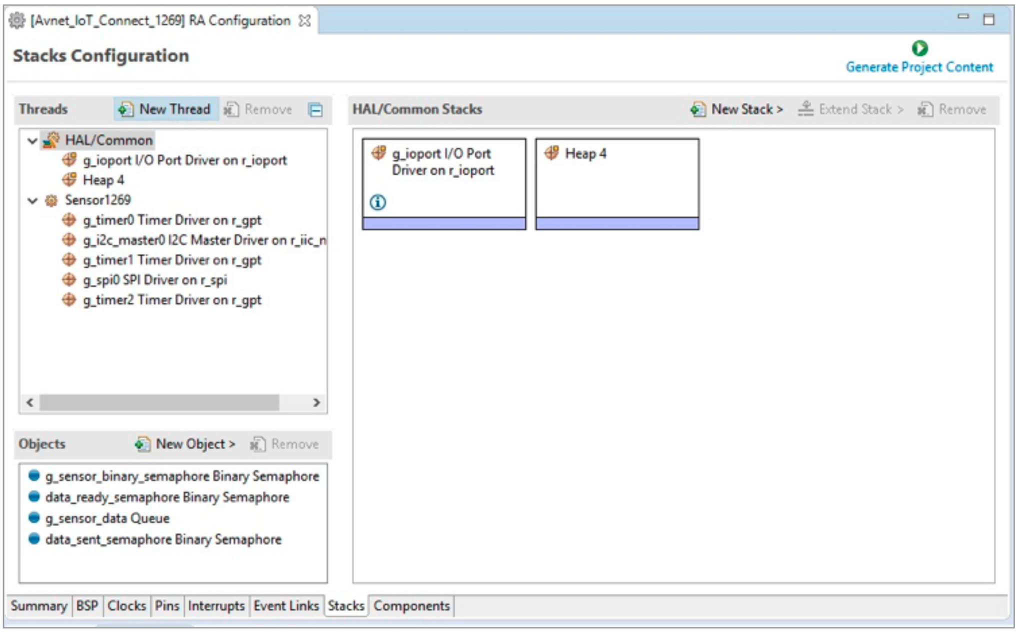
IMPORTANT NOTE: The project needs to be updated and completed by adding the communication task. If you try to compile it as it was imported, some errors will result.



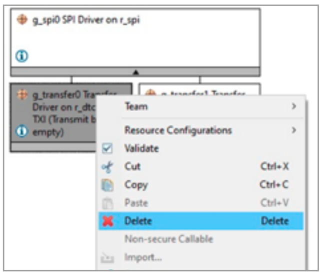
```
CDT Build Console [Avnet_IoT_Connect_1269]
../src/AvnetConnect/rSPI/r_spi_1269.c:934:15: warning: conversion from 'int' to 'float' may change value [-Wconversion]
  934 |           tot -= RMS_medie;
      |               ^~
../src/AvnetConnect/rSPI/r_spi_1269.c:934:16: warning: conversion from 'float' to 'int' may change value [-Wfloat-conversion]
  934 |           tot -= RMS_medie;
      |                  ^~~~~~~~~
' '
'Finished building: ../src/AvnetConnect/rSPI/r_spi_1269.c'
' '
"make -j8 all" terminated with exit code 2. Build might be incomplete.

13:41:09 Build Failed. 13 errors, 3 warnings. (took 936ms)
```

STEP 3 – CHECK SPI CONFIGURATION

Click on "g_spi0 SPI driver on r_spi" inside "Threads" box.
Verify that both DTC drivers are disabled.

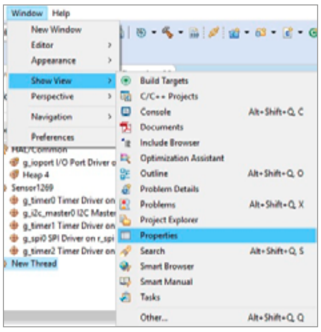

[Avnet_IoT_Connect_1269] RA Configuration
Stacks Configuration
Generate Project Content
Threads
New Thread
Remove
HAL/Common
g_ioport I/O Port Driver on r_ioport
Heap 4
Sensor1269
g_timer0 Timer Driver on r_gpt
g_i2c_master0 I2C Master Driver on r_iic_n
g_timer1 Timer Driver on r_gpt
g_spi0 SPI Driver on r_spi
g_timer2 Timer Driver on r_gpt
HAL/Common Stacks
New Stack >
Extend Stack >
Remove
g_ioport I/O Port Driver on r_ioport
Heap 4
Objects
New Object >
Remove
g_sensor_binary_semaphore Binary Semaphore
data_ready_semaphore Binary Semaphore
g_sensor_data Queue
data_sent_semaphore Binary Semaphore
Summary
BSP
Clocks
Pins
Interrupts
Event Links
Stacks
Components

If the DTC Driver are enabled, select the box, right click and set "Delete"



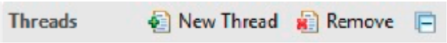
STEP 4 – ADD NEW TASK TO PROJECT

Enable the Properties tab by selecting and clicking on Windows > Show View > Properties in the menu.

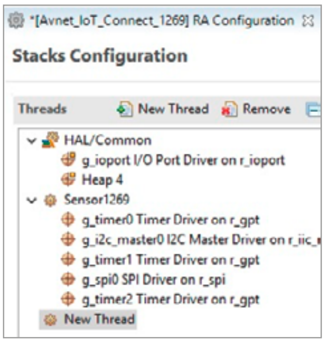


IMPORTANT NOTE: you must use the same name and label as indicated in this guide.

In the Threads tab within the Configuration Stack windows, click "New Thread" button to create a new Rtos thread.

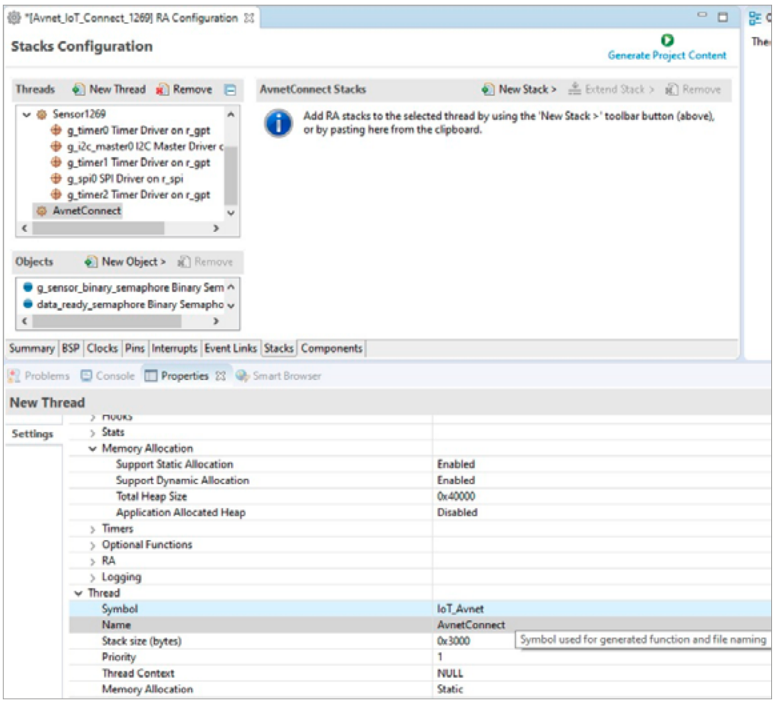


Then you'll need to configure the new task, create stack resources and add them to the project.


*[Avnet_IoT_Connect_1269] RA Configuration
Stacks Configuration
Threads
New Thread
Remove
HAL/Common
g_ioport I/O Port Driver on r_ioport
Heap 4
Sensor1269
g_timer0 Timer Driver on r_gpt
g_i2c_master0 I2C Master Driver on r_iic_r
g_timer1 Timer Driver on r_gpt
g_spi0 SPI Driver on r_spi
g_timer2 Timer Driver on r_gpt
New Thread

STEP 5 – CONFIGURE THE IOT TASK

In the Threads tab, select the "New Thread" you just created, then go to the "Properties" tab and configure as shown in the image below:



Sensor1269
- g_timer0 Timer Driver on r_gpt
- g_i2c_master0 I2C Master Driver c
- g_timer1 Timer Driver on r_gpt
- g_spi0 SPI Driver on r_spi
- g_timer2 Timer Driver on r_gpt

AvnetConnect

Add RA stacks to the selected thread by using the 'New Stack >' toolbar button (above), or by pasting here from the clipboard.

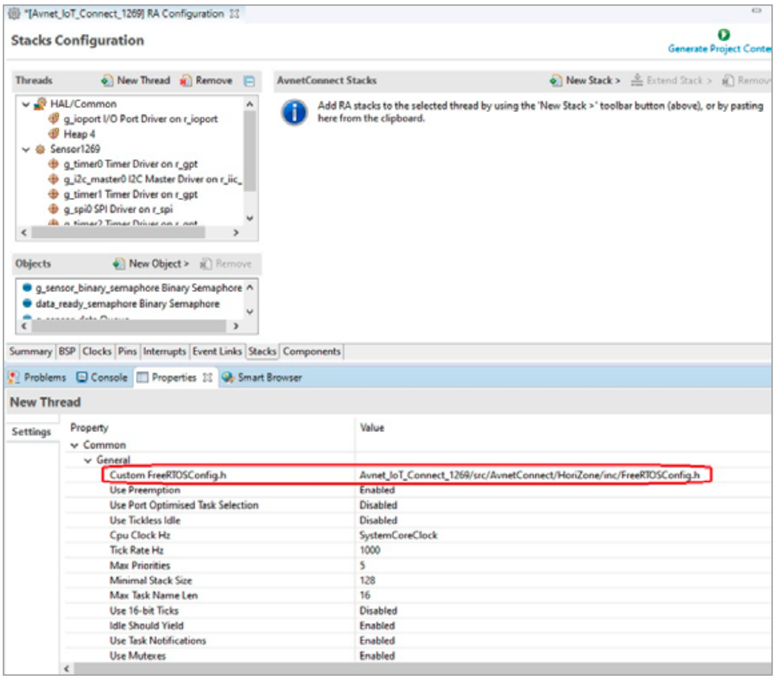
- g_sensor_binary_semaphore Binary Sem
- data_ready_semaphore Binary Semapho

Hooks	
Stats	
Memory Allocation	
Support Static Allocation	Enabled
Support Dynamic Allocation	Enabled
Total Heap Size	0x40000
Application Allocated Heap	Disabled
Timers	
Optional Functions	
RA	
Logging	
Thread	
Symbol	IoT_Avnet
Name	AvnetConnect
Stack size (bytes)	0x3000
Priority	1
Thread Context	NULL
Memory Allocation	Static

Symbol used for generated function and file naming

Now you'll need to check the custom FreeRTOSConfig.h file. In the Properties tab within the AvnetConnect task, go to "Common > General > Custom FreeRTOSConfig.h" and check that it includes the string below: Avnet_IoT_Connect_1269/src/AvnetConnect/HoriZone/inc/FreeRTOSConfig.h

Copy it in it if the Value field is empty.



HAL/Common
- g_ioport I/O Port Driver on r_ioport
- Heap 4

Sensor1269
- g_timer0 Timer Driver on r_gpt
- g_i2c_master0 I2C Master Driver on r_iic_
- g_timer1 Timer Driver on r_gpt
- g_spi0 SPI Driver on r_spi

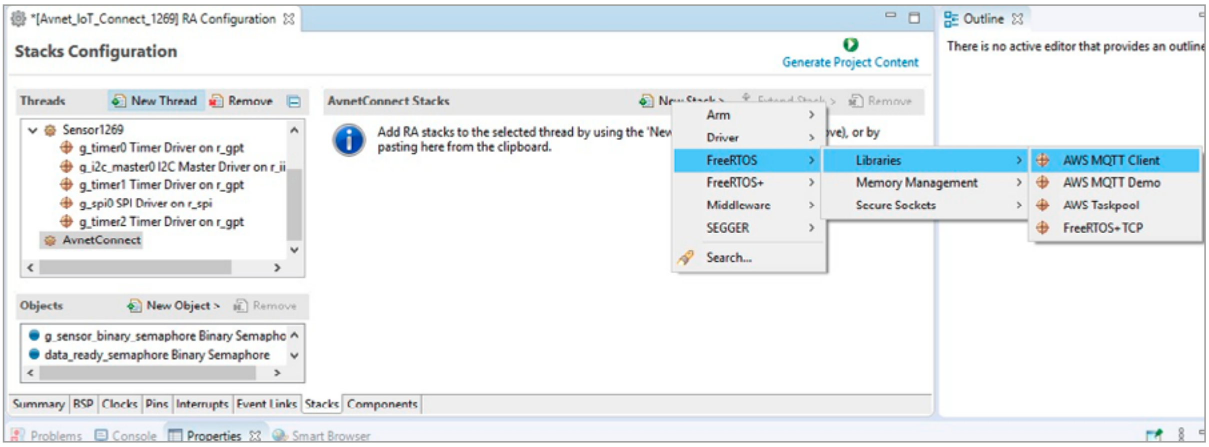
Add RA stacks to the selected thread by using the 'New Stack >' toolbar button (above), or by pasting here from the clipboard.

- g_sensor_binary_semaphore Binary Semaphore
- data_ready_semaphore Binary Semaphore

Property	Value
Common	
General	
Custom FreeRTOSConfig.h	Avnet_IoT_Connect_1269/src/AvnetConnect/HoriZone/inc/FreeRTOSConfig.h
Use Preemption	Enabled
Use Port Optimised Task Selection	Disabled
Use Tickless Idle	Disabled
Cpu Clock Hz	SystemCoreClock
Tick Rate Hz	1000
Max Priorities	5
Minimal Stack Size	128
Max Task Name Len	16
Use 16-bit Ticks	Disabled
Idle Should Yield	Enabled
Use Task Notifications	Enabled
Use Mutexes	Enabled

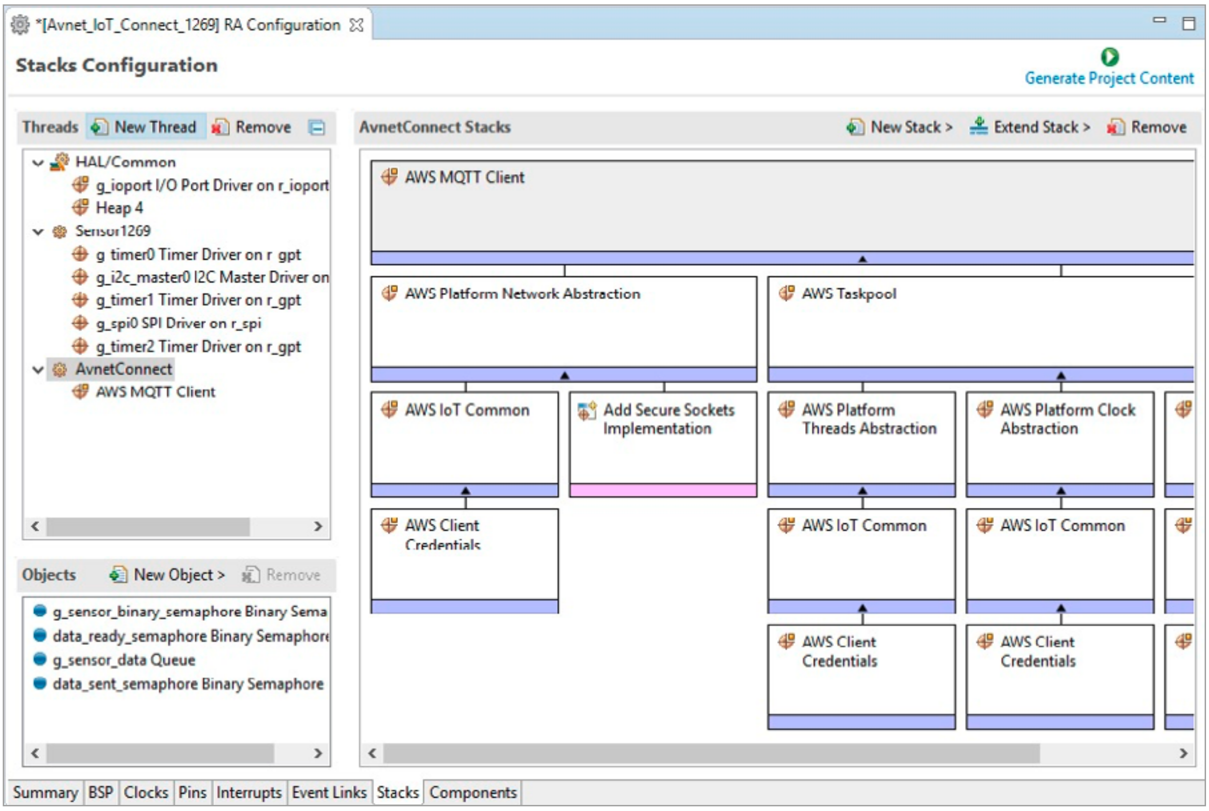
STEP 6 – ADD MQTT CLIENT STACK

In the Threads tab, select the "AvnetConnect" thread, then go to "AvnetConnect Stacks", click on "New Stack" and select FreeRTOS > Libraries > AWS MQTT Client.



*[Avnet_IoT_Connect_1269] RA Configuration
Stacks Configuration
Generate Project Content
Outline
There is no active editor that provides an outline
Threads
New Thread
Remove
Sensor1269
g_timer0 Timer Driver on r_gpt
g_i2c_master0 I2C Master Driver on r_ii
g_timer1 Timer Driver on r_gpt
g_spi0 SPI Driver on r_spi
g_timer2 Timer Driver on r_gpt
AvnetConnect
AvnetConnect Stacks
New Stack >
Extend Stack >
Remove
Add RA stacks to the selected thread by using the 'New ...ve), or by pasting here from the clipboard.
Arm
Driver
FreeRTOS
FreeRTOS+
Middleware
SEGGER
Search...
Libraries
Memory Management
Secure Sockets
AWS MQTT Client
AWS MQTT Demo
AWS Taskpool
FreeRTOS+TCP
Objects
New Object >
Remove
g_sensor_binary_semaphore Binary Semapho
data_ready_semaphore Binary Semaphore
Summary
BSP
Clocks
Pins
Interrupts
Event Links
Stacks
Components
Problems
Console
Properties
Smart Browser

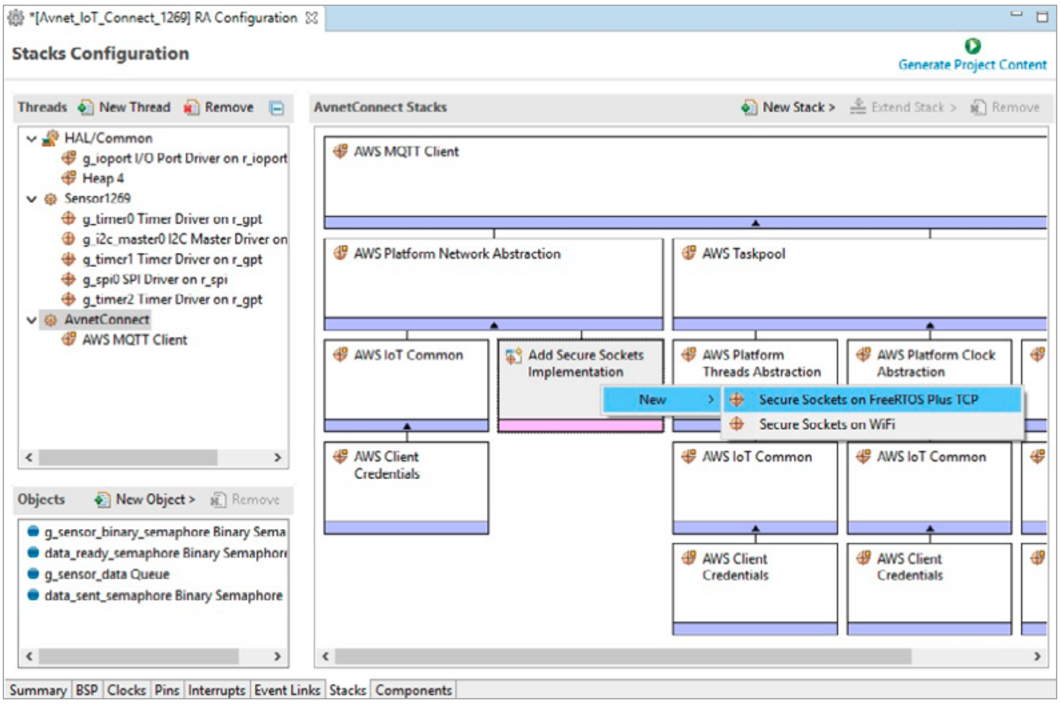
The MQTT stack is now added to your project.



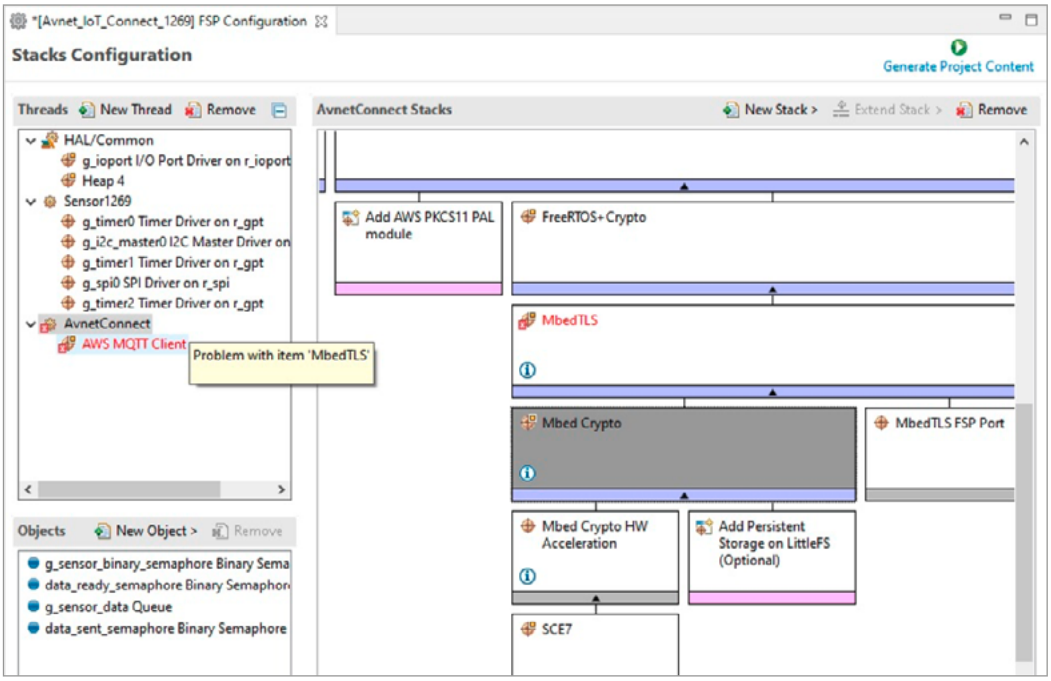
*[Avnet_IoT_Connect_1269] RA Configuration
Stacks Configuration
Generate Project Content
Threads
New Thread
Remove
HAL/Common
g_ioport I/O Port Driver on r_ioport
Heap 4
Sensor1269
g_timer0 Timer Driver on r_gpt
g_i2c_master0 I2C Master Driver on
g_timer1 Timer Driver on r_gpt
g_spi0 SPI Driver on r_spi
g_timer2 Timer Driver on r_gpt
AvnetConnect
AWS MQTT Client
Objects
New Object >
Remove
g_sensor_binary_semaphore Binary Sema
data_ready_semaphore Binary Semaphore
g_sensor_data Queue
data_sent_semaphore Binary Semaphore
AvnetConnect Stacks
New Stack >
Extend Stack >
Remove
AWS MQTT Client
AWS Platform Network Abstraction
AWS Taskpool
AWS IoT Common
Add Secure Sockets Implementation
AWS Platform Threads Abstraction
AWS Platform Clock Abstraction
AWS Client Credentials
AWS IoT Common
AWS IoT Common
AWS Client Credentials
AWS Client Credentials
Summary
BSP
Clocks
Pins
Interrupts
Event Links
Stacks
Components

STEP 7 – CONFIGURE THE MQTT CLIENT STACK

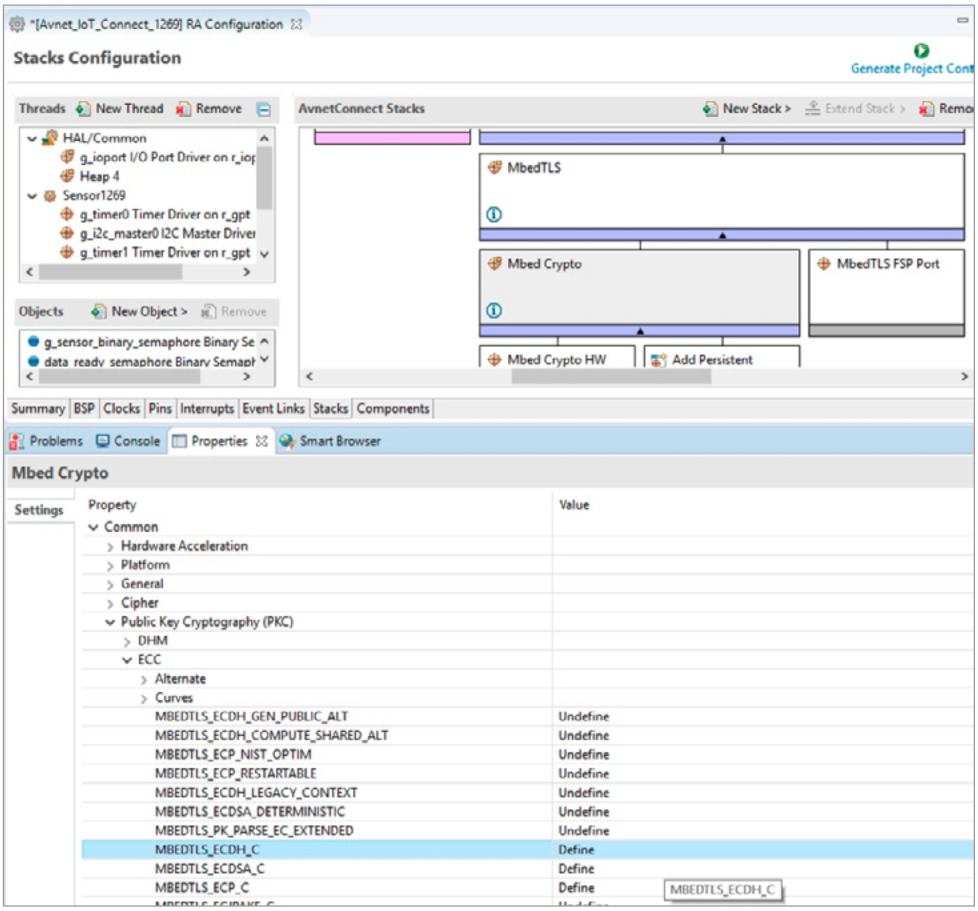
Select and highlight "Add Secure Sockets Implementation", click on the box and select "New > Secure Sockets on FreeRTOS Plus TCP".


*[Avnet_IoT_Connect_1269] RA Configuration
Stacks Configuration
Generate Project Content
Threads
New Thread
Remove
HAL/Common
g_ioport I/O Port Driver on r_ioport
Heap 4
Sensor1269
g_timer0 Timer Driver on r_gpt
g_i2c_master0 I2C Master Driver on
g_timer1 Timer Driver on r_gpt
g_spi0 SPI Driver on r_spi
g_timer2 Timer Driver on r_gpt
AvnetConnect
AWS MQTT Client
Objects
New Object >
Remove
g_sensor_binary_semaphore Binary Sema
data_ready_semaphore Binary Semaphore
g_sensor_data Queue
data_sent_semaphore Binary Semaphore
AvnetConnect Stacks
New Stack >
Extend Stack >
Remove
AWS MQTT Client
AWS Platform Network Abstraction
AWS Taskpool
AWS IoT Common
Add Secure Sockets Implementation
AWS Platform Threads Abstraction
AWS Platform Clock Abstraction
New
Secure Sockets on FreeRTOS Plus TCP
Secure Sockets on WiFi
AWS Client Credentials
AWS IoT Common
AWS IoT Common
AWS Client Credentials
AWS Client Credentials
Summary
BSP
Clocks
Pins
Interrupts
Event Links
Stacks
Components

Once you've created it you may experience some issues with the resources you just created. Here are a few tips to resolve them.

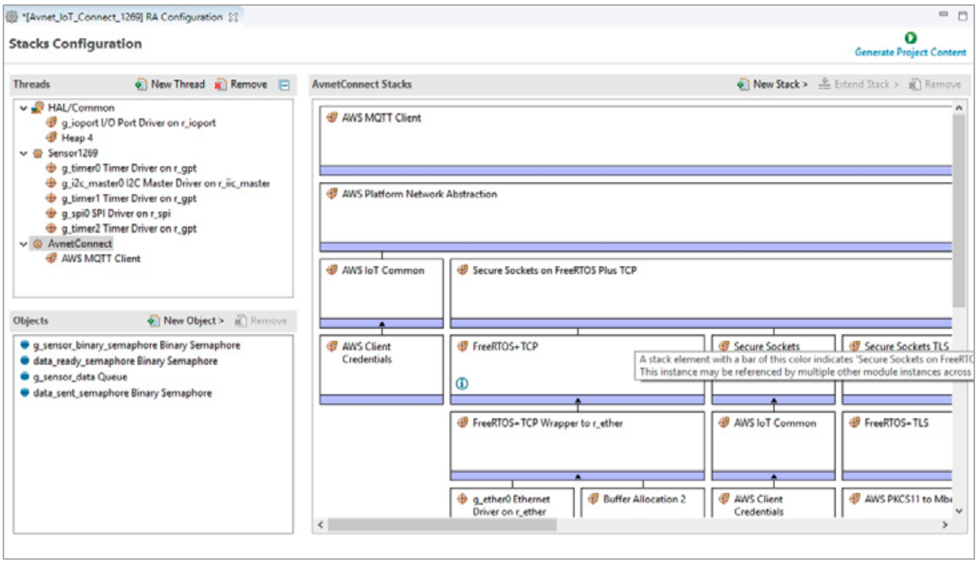

*[Avnet_IoT_Connect_1269] FSP Configuration
Stacks Configuration
Generate Project Content
Threads
New Thread
Remove
HAL/Common
g_ioport I/O Port Driver on r_ioport
Heap 4
Sensor1269
g_timer0 Timer Driver on r_gpt
g_i2c_master0 I2C Master Driver on
g_timer1 Timer Driver on r_gpt
g_spi0 SPI Driver on r_spi
g_timer2 Timer Driver on r_gpt
AvnetConnect
AWS MQTT Client
Problem with item 'MbedTLS'
Objects
New Object >
Remove
g_sensor_binary_semaphore Binary Sema
data_ready_semaphore Binary Semaphore
g_sensor_data Queue
data_sent_semaphore Binary Semaphore
AvnetConnect Stacks
New Stack >
Extend Stack >
Remove
Add AWS PKCS11 PAL module
FreeRTOS+Crypto
MbedTLS
Mbed Crypto
MbedTLS FSP Port
Mbed Crypto HW Acceleration
Add Persistent Storage on LittleFS (Optional)
SCE7

In the AvnetConnect Stacks window, select and highlight the "Mbed Crypto" box. Then go to the "Properties" tab and define the macro as shown in the image below.

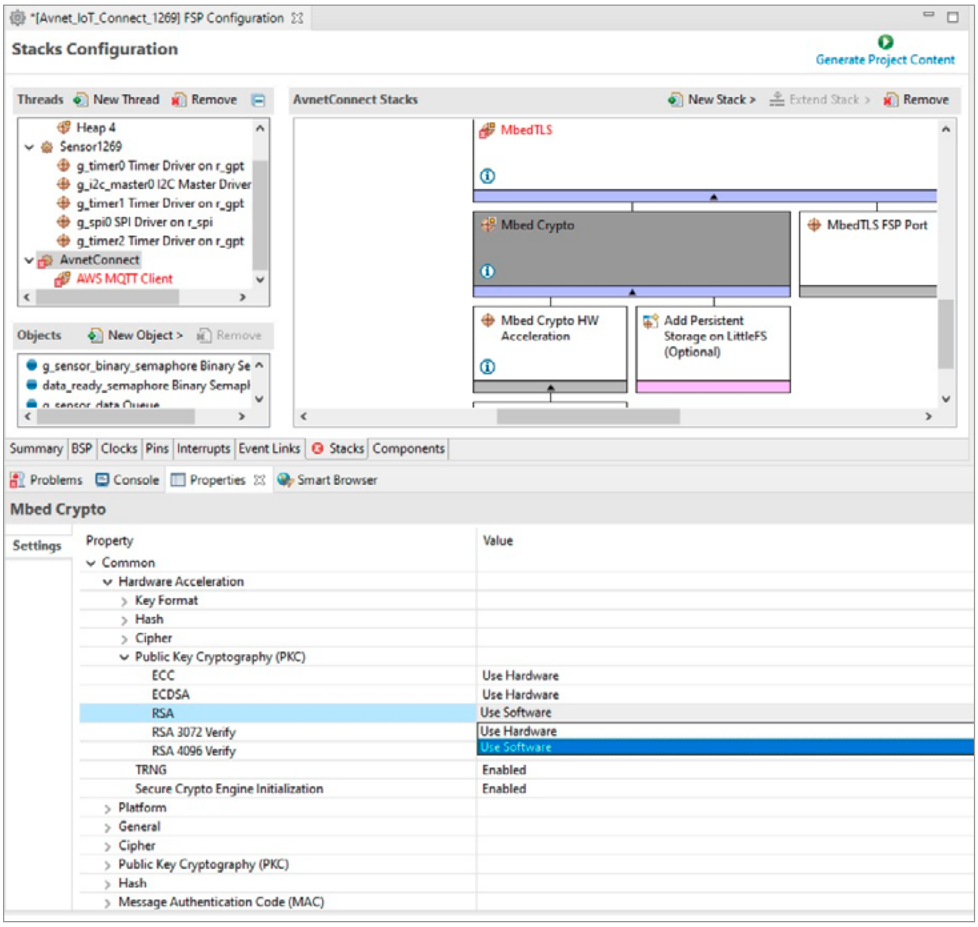


Property	Value
Common	
Hardware Acceleration	
Platform	
General	
Cipher	
Public Key Cryptography (PKC)	
DHM	
ECC	
Alternate	
Curves	
MBEDTLS_ECDH_GEN_PUBLIC_ALT	Undefine
MBEDTLS_ECDH_COMPUTE_SHARED_ALT	Undefine
MBEDTLS_ECP_NIST_OPTIM	Undefine
MBEDTLS_ECP_RESTARTABLE	Undefine
MBEDTLS_ECDH_LEGACY_CONTEXT	Undefine
MBEDTLS_ECDSA_DETERMINISTIC	Undefine
MBEDTLS_PK_PARSE_EC_EXTENDED	Undefine
MBEDTLS_ECDH_C	Define
MBEDTLS_ECDSA_C	Define
MBEDTLS_ECP_C	Define

Error will be solved


*[Avnet_IoT_Connect_1269] RA Configuration
Stacks Configuration
Generate Project Content
Threads
New Thread
Remove
HAL/Common
g_ioport I/O Port Driver on r_ioport
Heap 4
Sensor1269
g_timer0 Timer Driver on r_gpt
g_i2c_master0 I2C Master Driver on r_iic_master
g_timer1 Timer Driver on r_gpt
g_spi0 SPI Driver on r_spi
g_timer2 Timer Driver on r_gpt
AvnetConnect
AWS MQTT Client
Objects
New Object >
Remove
g_sensor_binary_semaphore Binary Semaphore
data_ready_semaphore Binary Semaphore
g_sensor_data Queue
data_sent_semaphore Binary Semaphore
AvnetConnect Stacks
New Stack >
Extend Stack >
Remove
AWS MQTT Client
AWS Platform Network Abstraction
AWS IoT Common
Secure Sockets on FreeRTOS Plus TCP
AWS Client Credentials
FreeRTOS+TCP
Secure Sockets
Secure Sockets TLS
A stack element with a bar of this color indicates 'Secure Sockets on FreeRTO
This instance may be referenced by multiple other module instances across
FreeRTOS+TCP Wrapper to r_ether
AWS IoT Common
FreeRTOS+TLS
g_ether0 Ethernet Driver on r_ether
Buffer Allocation 2
AWS Client Credentials
AWS PKCS11 to Mbe

Set the crypto engine: **this is very important for the TLS certificates managing**

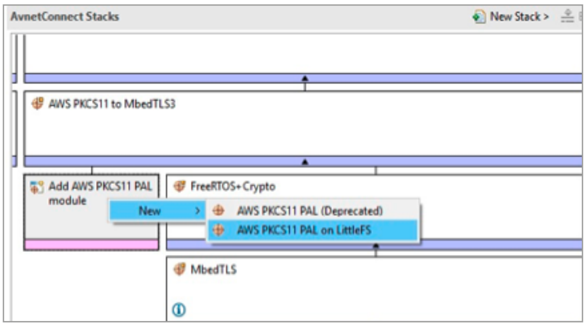


Mbed Crypto

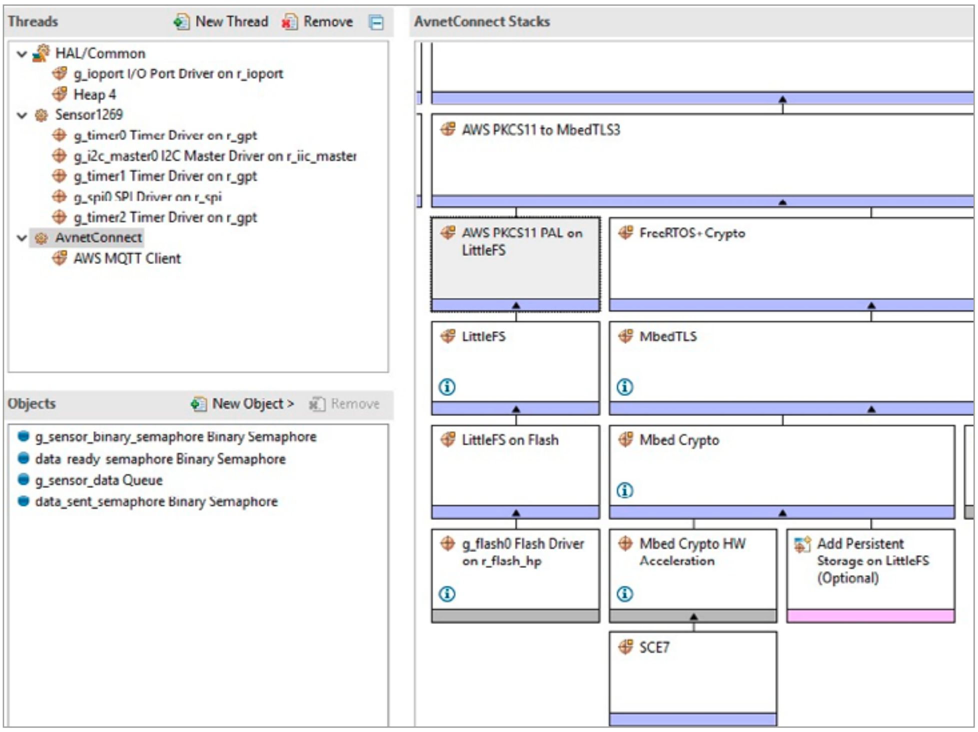
Property	Value
Common	
Hardware Acceleration	
Key Format	
Hash	
Cipher	
Public Key Cryptography (PKC)	
ECC	Use Hardware
ECDSA	Use Hardware
RSA	Use Software
RSA 3072 Verify	Use Hardware
RSA 4096 Verify	Use Software
TRNG	Enabled
Secure Crypto Engine Initialization	Enabled
Platform	
General	
Cipher	
Public Key Cryptography (PKC)	
Hash	
Message Authentication Code (MAC)	

STEP 8 – ADD THE PKCS11 PAL MODULE

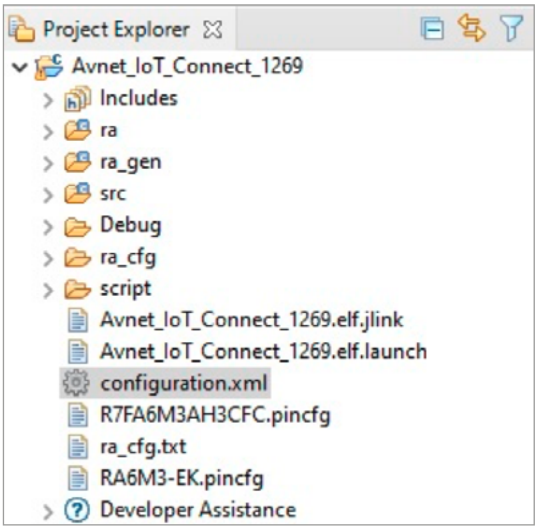
In the AvnetConnect Stacks window, select and highlight "Add AWS PKCS11 PAL module". Click on the box and select "New > AWS PKCS11 PAL on LittleFS".

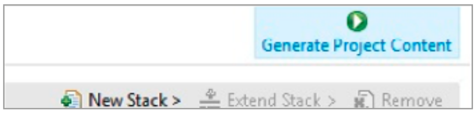


When you have added it as shown here:


Threads
New Thread
Remove
HAL/Common
g_ioport I/O Port Driver on r_ioport
Heap 4
Sensor1269
g_timer0 Timer Driver on r_gpt
g_i2c_master0 I2C Master Driver on r_iic_master
g_timer1 Timer Driver on r_gpt
g_spi0 SPI Driver on r_spi
g_timer2 Timer Driver on r_gpt
AvnetConnect
AWS MQTT Client
Objects
New Object >
Remove
g_sensor_binary_semaphore Binary Semaphore
data_ready_semaphore Binary Semaphore
g_sensor_data Queue
data_sent_semaphore Binary Semaphore
AvnetConnect Stacks
AWS PKCS11 to MbedTLS3
AWS PKCS11 PAL on LittleFS
FreeRTOS+Crypto
LittleFS
MbedTLS
LittleFS on Flash
Mbed Crypto
g_flash0 Flash Driver on r_flash_hp
Mbed Crypto HW Acceleration
Add Persistent Storage on LittleFS (Optional)
SCE7

Click on "Generate Project Content" and wait for the process to complete.



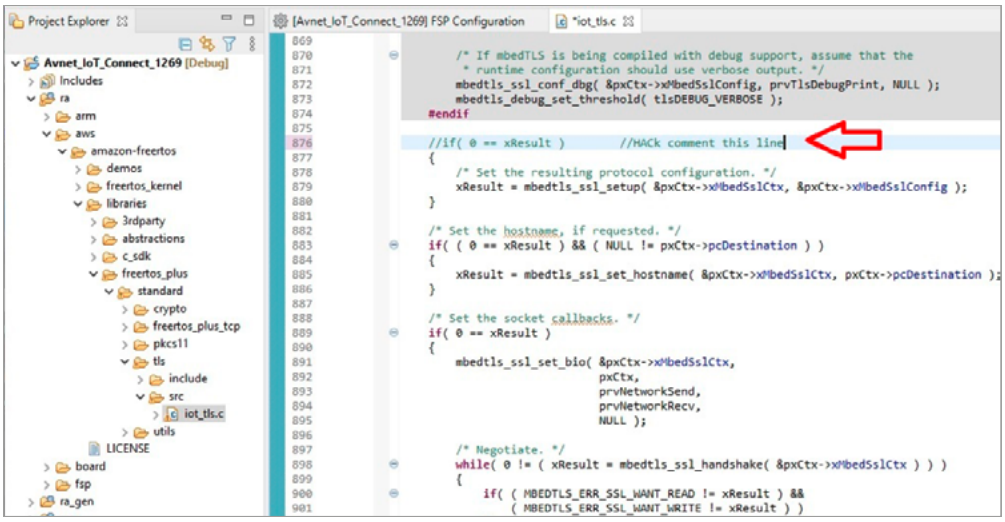


the full project at end of generation process

STEP 9 – TLS "HACK" MODIFICATIONS

In order to make the AWS Mbed TSL fully functional, you'll need to make a simple code modification:

- FILE \HoriZoneRA\Avnet_IoT_Connect_1269\ra\aws\amazon-freertos\libraries\freertos_plus\standard\tls\src\iot_tls.c

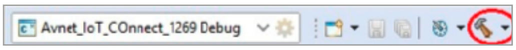


```
        /* If mbedTLS is being compiled with debug support, assume that the
         * runtime configuration should use verbose output. */
        mbedtls_ssl_conf_dbg( &pxCtx->xMbedSslConfig, prvTlsDebugPrint, NULL );
        mbedtls_debug_set_threshold( tlsDEBUG_VERBOSE );
    #endif

    //if( 0 == xResult )        //HACk comment this line
    {
        /* Set the resulting protocol configuration. */
        xResult = mbedtls_ssl_setup( &pxCtx->xMbedSslCtx, &pxCtx->xMbedSslConfig );
    }

    /* Set the hostname, if requested. */
    if( ( 0 == xResult ) && ( NULL != pxCtx->pcDestination ) )
    {
        xResult = mbedtls_ssl_set_hostname( &pxCtx->xMbedSslCtx, pxCtx->pcDestination );
    }

    /* Set the socket callbacks. */
    if( 0 == xResult )
    {
        mbedtls_ssl_set_bio( &pxCtx->xMbedSslCtx,
                             pxCtx,
                             prvNetworkSend,
                             prvNetworkRecv,
                             NULL );

        /* Negotiate. */
        while( 0 != ( xResult = mbedtls_ssl_handshake( &pxCtx->xMbedSslCtx ) ) )
        {
            if( ( MBEDTLS_ERR_SSL_WANT_READ != xResult ) &&
                ( MBEDTLS_ERR_SSL_WANT_WRITE != xResult ) )
```

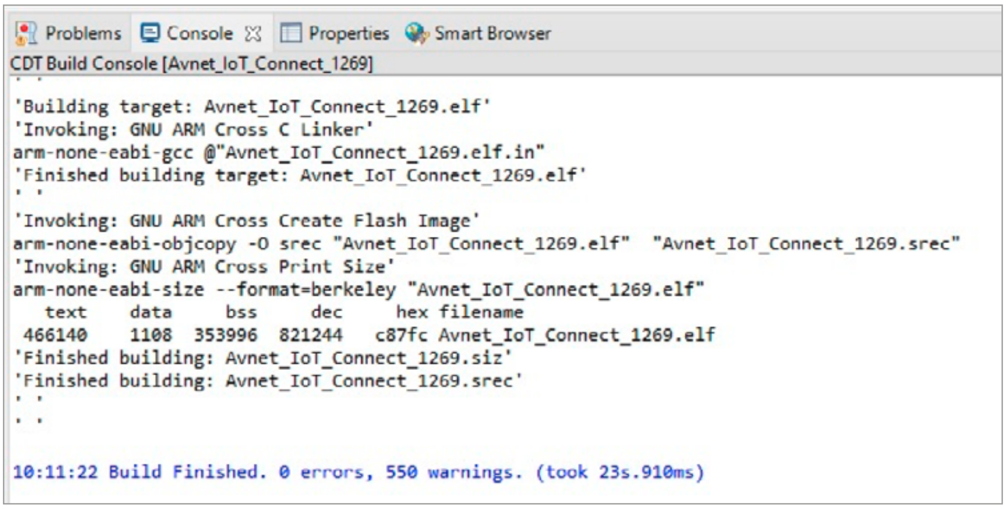
STEP 10 – COMPILE AND LINK

Before you compile the project, you must fill in the fields in the user_cfg.h header file with your own connection credentials. Here's how to do this.

After you have added your credential settings as above, click on the hammer icon



and wait for the compile and link process to end.



CDT Build Console [Avnet_IoT_Connect_1269]

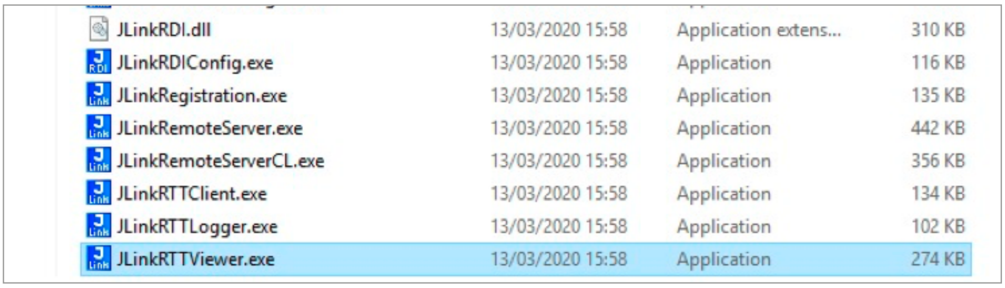
```
'Building target: Avnet_IoT_Connect_1269.elf'
'Invoking: GNU ARM Cross C Linker'
arm-none-eabi-gcc @"Avnet_IoT_Connect_1269.elf.in"
'Finished building target: Avnet_IoT_Connect_1269.elf'
' '
'Invoking: GNU ARM Cross Create Flash Image'
arm-none-eabi-objcopy -O srec "Avnet_IoT_Connect_1269.elf"  "Avnet_IoT_Connect_1269.srec"
'Invoking: GNU ARM Cross Print Size'
arm-none-eabi-size --format=berkeley "Avnet_IoT_Connect_1269.elf"
   text    data     bss     dec     hex filename
 466140    1108  353996  821244   c87fc Avnet_IoT_Connect_1269.elf
'Finished building: Avnet_IoT_Connect_1269.siz'
'Finished building: Avnet_IoT_Connect_1269.srec'
' '
' '

10:11:22 Build Finished. 0 errors, 550 warnings. (took 23s.910ms)
```

Please note that the "text" section space may be different depending on the length of your connection credential string.

HoriZoneRA debug

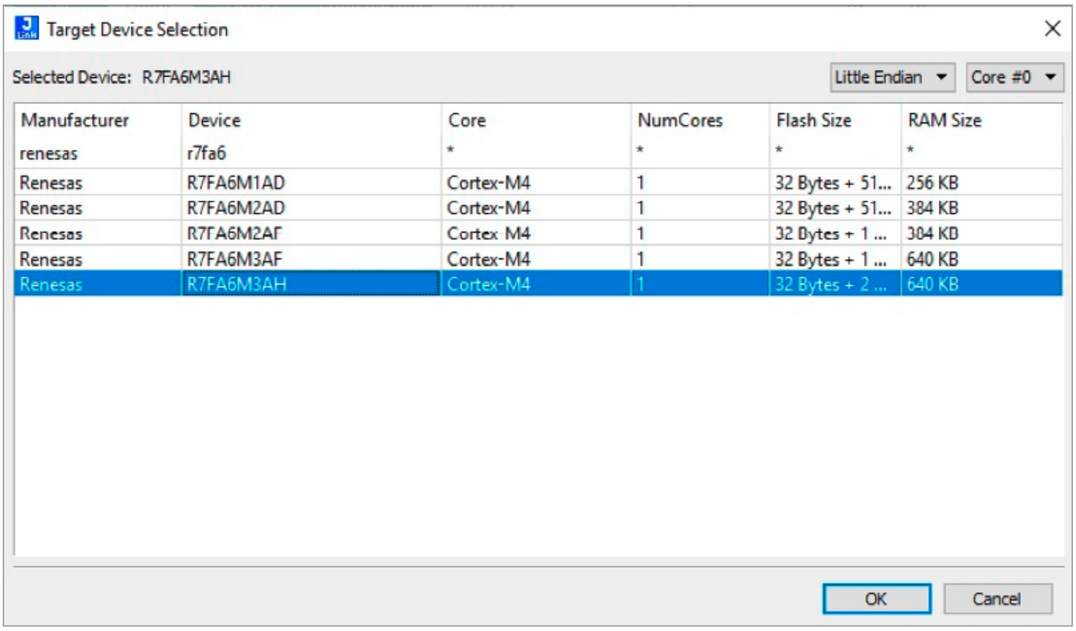
STEP 1 – PREPARE THE SERIAL LOG VIEWER

Go to the Segger J-Link Software pack 6.64 installation path, then open the JLinkRTTViewer.exe file.



JLinkRDI.dll	13/03/2020 15:58	Application extens...	310 KB
JLinkRDIConfig.exe	13/03/2020 15:58	Application	116 KB
JLinkRegistration.exe	13/03/2020 15:58	Application	135 KB
JLinkRemoteServer.exe	13/03/2020 15:58	Application	442 KB
JLinkRemoteServerCL.exe	13/03/2020 15:58	Application	356 KB
JLinkRTTClient.exe	13/03/2020 15:58	Application	134 KB
JLinkRTTLogger.exe	13/03/2020 15:58	Application	102 KB
JLinkRTTViewer.exe	13/03/2020 15:58	Application	274 KB

Configure the correct device.



Target Device Selection

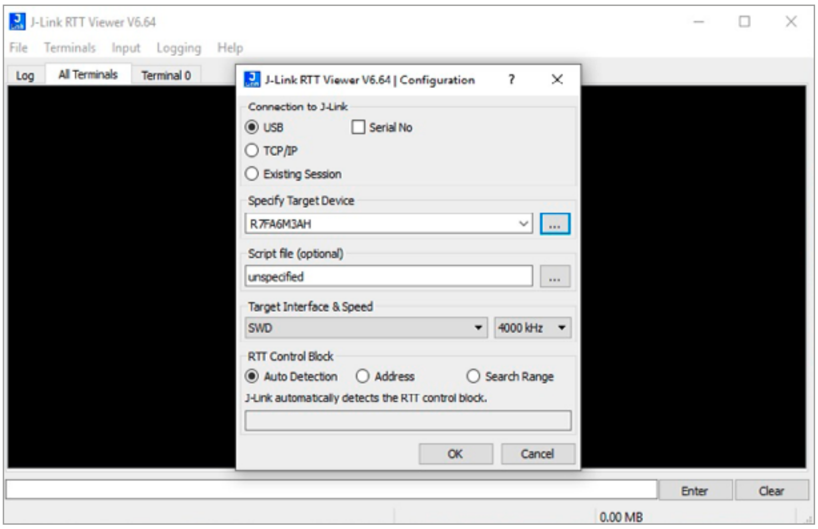
Selected Device: R7FA6M3AH

Little Endian

Core #0

Manufacturer	Device	Core	NumCores	Flash Size	RAM Size
renesas	r7fa6	*	*	*	*
Renesas	R7FA6M1AD	Cortex-M4	1	32 Bytes + 51...	256 KB
Renesas	R7FA6M2AD	Cortex-M4	1	32 Bytes + 51...	384 KB
Renesas	R7FA6M2AF	Cortex-M4	1	32 Bytes + 1 ...	384 KB
Renesas	R7FA6M3AF	Cortex-M4	1	32 Bytes + 1 ...	640 KB
Renesas	R7FA6M3AH	Cortex-M4	1	32 Bytes + 2 ...	640 KB

The terminal will start



J-Link RTT Viewer V6.64 | Configuration

Connection to J-Link

● USB □ Serial No

○ TCP/IP

○ Existing Session

Specify Target Device: R7FA6M3AH

Script file (optional): unspecified

Target Interface & Speed: SWD 4000 kHz

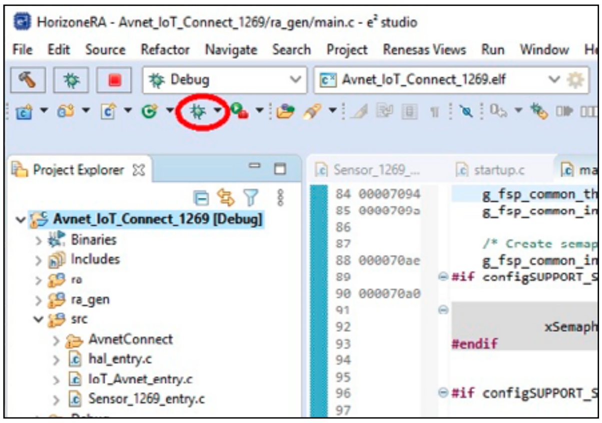
RTT Control Block

● Auto Detection ○ Address ○ Search Range

J-Link automatically detects the RTT control block.

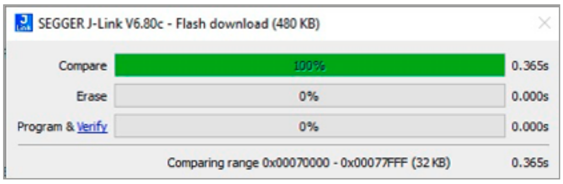
STEP 2 – DOWNLOAD AND DEBUG

In the Project Explorer tab, select and highlight the project, then click on the "bug" icon.

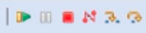


```
00007094      g_fsp_common_th
0000709a      g_fsp_common_in

              /* Create semap
000070ae      g_fsp_common_in
          #if configSUPPORT_S
000070a0

                        xSemaph
          #endif


          #if configSUPPORT_S
```

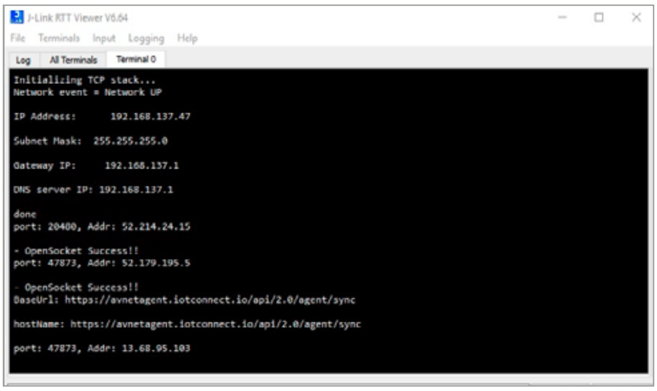
Wait flash programmer…



Comparing range 0x00070000 - 0x00077FFF (32 KB) 0.365s

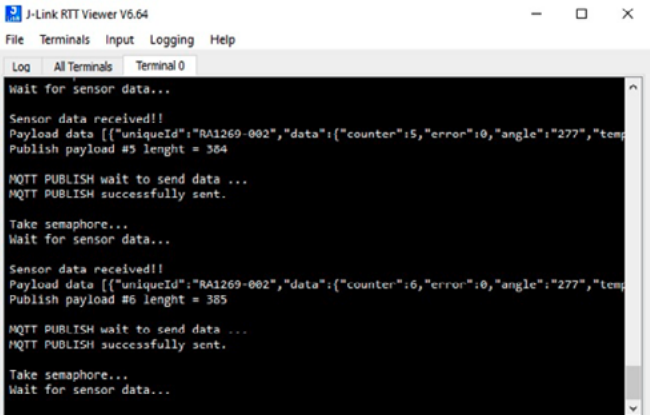
Click the "Play" icon. 

This is how the JLinkRTT Viewer looks at start-up:



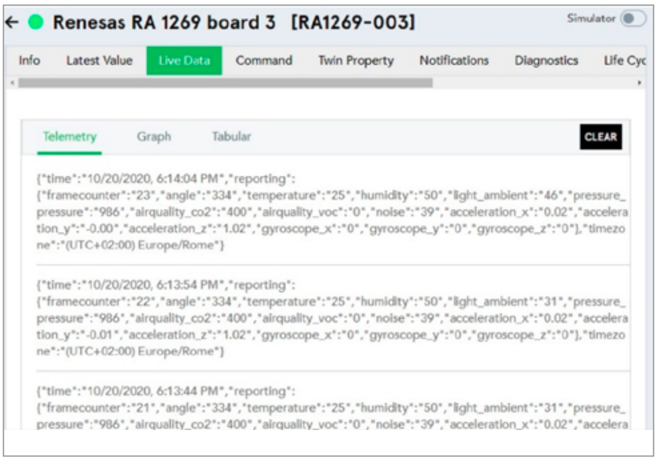
```
Initializing TCP stack...
Network event = Network UP

IP Address:      192.168.137.47

Subnet Mask:  255.255.255.0

Gateway IP:     192.168.137.1

DNS server IP: 192.168.137.1

done
port: 20400, Addr: 52.214.24.15

- OpenSocket Success!!
port: 47873, Addr: 52.179.195.5

- OpenSocket Success!!
BaseUrl: https://avnetagent.iotconnect.io/api/2.0/agent/sync

hostName: https://avnetagent.iotconnect.io/api/2.0/agent/sync

port: 47873, Addr: 13.68.95.103
```

And while it is running:



```
Wait for sensor data...

Sensor data received!!
Payload data [{"uniqueId":"RA1269-002","data":{"counter":5,"error":0,"angle":"277","temp
Publish payload #5 lenght = 384

MQTT PUBLISH wait to send data ...
MQTT PUBLISH successfully sent.

Take semaphore...
Wait for sensor data...

Sensor data received!!
Payload data [{"uniqueId":"RA1269-002","data":{"counter":6,"error":0,"angle":"277","temp
Publish payload #6 lenght = 385

MQTT PUBLISH wait to send data ...
MQTT PUBLISH successfully sent.

Take semaphore...
Wait for sensor data...
```

Go to Backend data, under “Device > (select MyDeviceName) > Live Data



```
{"time":"10/20/2020, 6:14:04 PM","reporting":
{"framecounter":"23","angle":"334","temperature":"25","humidity":"50","light_ambient":"46","pressure_
pressure":"986","airquality_co2":"400","airquality_voc":"0","noise":"39","acceleration_x":"0.02","accelera
tion_y":"-0.00","acceleration_z":"1.02","gyroscope_x":"0","gyroscope_y":"0","gyroscope_z":"0"},"timezo
ne":"(UTC+02:00) Europe/Rome"}

{"time":"10/20/2020, 6:13:54 PM","reporting":
{"framecounter":"22","angle":"334","temperature":"25","humidity":"50","light_ambient":"31","pressure_
pressure":"986","airquality_co2":"400","airquality_voc":"0","noise":"39","acceleration_x":"0.02","accelera
tion_y":"-0.01","acceleration_z":"1.02","gyroscope_x":"0","gyroscope_y":"0","gyroscope_z":"0"},"timezo
ne":"(UTC+02:00) Europe/Rome"}

{"time":"10/20/2020, 6:13:44 PM","reporting":
{"framecounter":"21","angle":"334","temperature":"25","humidity":"50","light_ambient":"31","pressure_
pressure":"986","airquality_co2":"400","airquality_voc":"0","noise":"39","acceleration_x":"0.02","accelera
```

Data units:

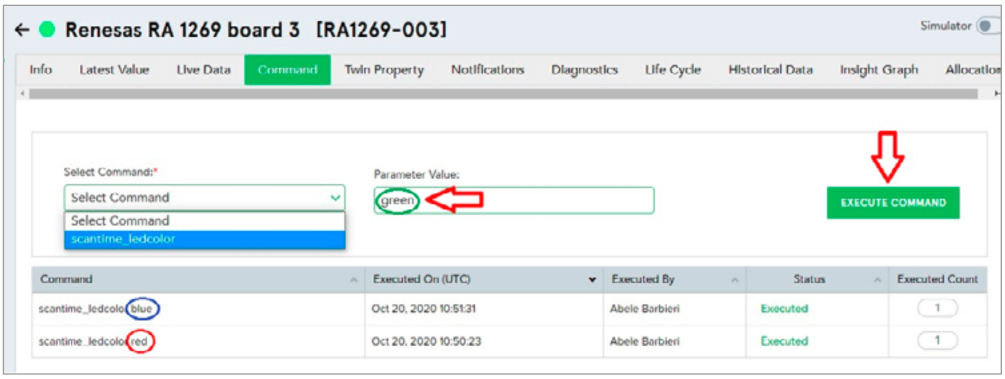
- Angle degree
- Temperature in °C
- Humidity in %
- Light in lux
- Pressure in mbar
- Gas in ppm
- Noise in dB
- Accelerometer in g
- Gyroscope degree/second

STEP 3 – SEND A COMMAND FROM THE BACKEND

On your backend dashboard, go to the “Command” tab of your device.

You can send a LedOn command in order to change the colour of the “send data” indicator led on the 1269B board.

In the “Select Command” dropdown list, choose “LedOn”, then type “red” or “blue” or “green” in the “Parameter Value” field. Then click the EXECUTE COMMAND button.



Select Command:* Select Command

Parameter Value: green

Command	Executed On (UTC)	Executed By	Status	Executed Count
scantime_ledcolo blue	Oct 20, 2020 10:51:31	Abele Barbieri	Executed	1
scantime_ledcolo red	Oct 20, 2020 10:50:23	Abele Barbieri	Executed	1

IoT Connect device configuration

USER DEFINITIONS AND CERTIFICATES

The user_cfg.h file contains all the definitions you'll need to enable device connection.

You must add your own device settings for macros:

```
#define HOSTNAME                         "[MyHostName]"      //used only for DHCP

#define IOTCONNECT_DEVICE_UNIQUE_ID      "[MyDeviceName]"    //device UUID name
#define IOTCONNECT_DEVICE_CP_ID          "[MyCpId]"          //company identifier
#define IOTCONNECT_DEVICE_ENV            "[Environment]"   //company environment
```

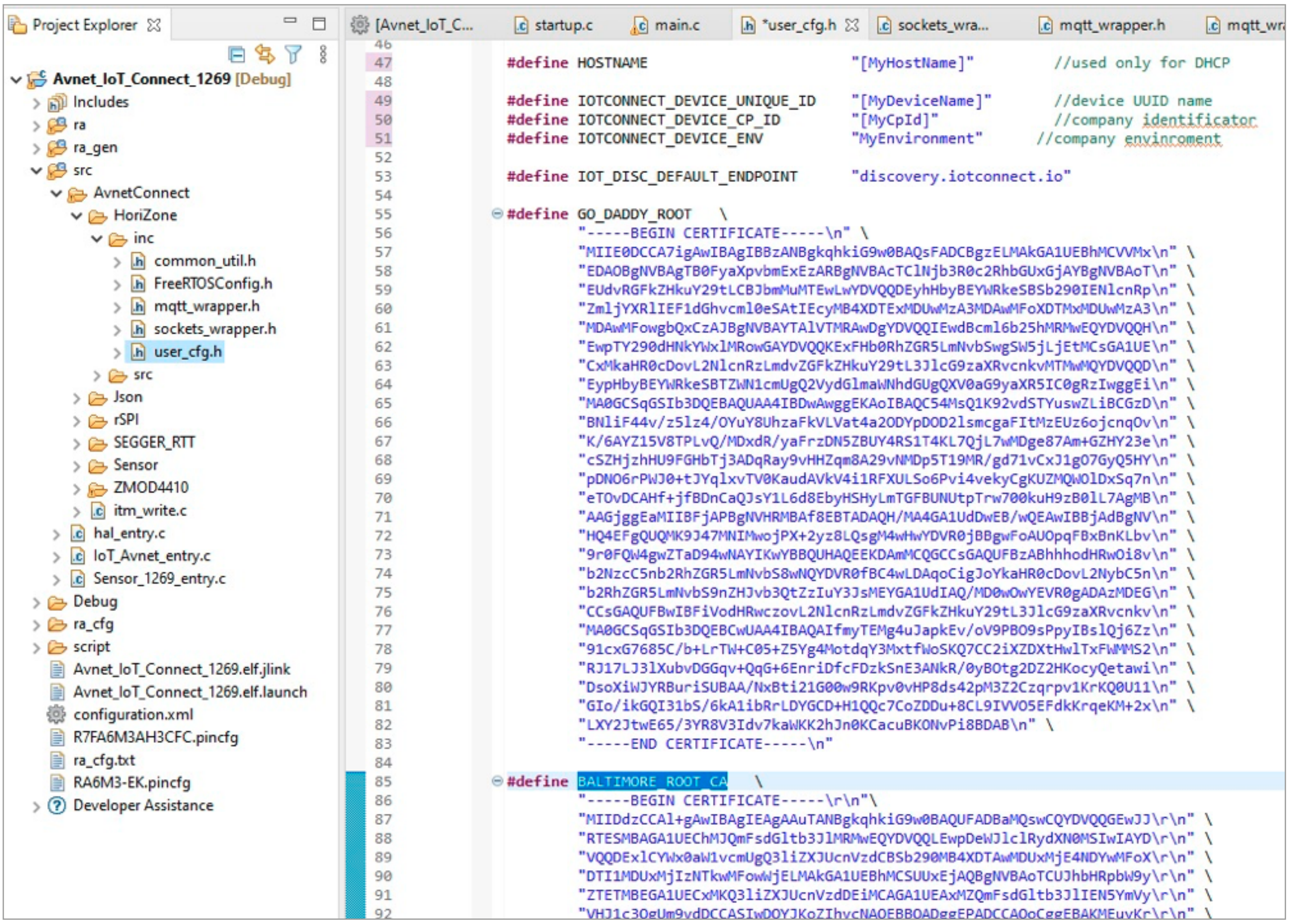
The data values to fill the macro above will be copied from your IoTConnect Portal.

See the next paragraph to learn how to make a new device, get the macro filling data and see the data sent to the device.

In the project, there are 2 certificates (for TSL and MQTT layer connection):

- Root Ca certificate for TLS Secure Socket (GO_DADDY_ROOT)
- Root Ca certificate for MQTT Connection (BALTIMORE_ROOT_CA)

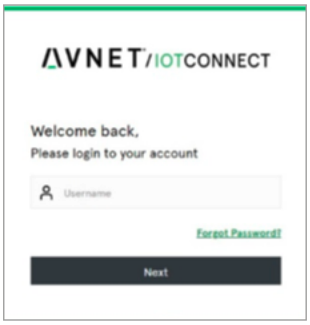
The project includes a working sample but you should replace this with your own certificates.



```
#define HOSTNAME                         "[MyHostName]"       //used only for DHCP

#define IOTCONNECT_DEVICE_UNIQUE_ID      "[MyDeviceName]"     //device UUID name
#define IOTCONNECT_DEVICE_CP_ID          "[MyCpId]"           //company identificator
#define IOTCONNECT_DEVICE_ENV            "MyEnvironment"    //company envinroment

#define IOT_DISC_DEFAULT_ENDPOINT        "discovery.iotconnect.io"

#define GO_DADDY_ROOT   \
        "-----BEGIN CERTIFICATE-----\n" \
        "MIIE0DCCA7igAwIBAgIBBzANBgkqhkiG9w0BAQsFADCBgzELMAkGA1UEBhMCVVMx\n" \
        "EDAOBgNVBAgTB0FyaXpvbmExEzARBgNVBAcTClNjb3R0c2RhbGUxGjAYBgNVBAoT\n" \
        "EUdvRGFkZHkuY29tLCBJbmMuMTEwLwYDVQQDEyhHbyBEYWRkeSBSb290IENlcnRp\n" \
        "ZmljYXRlIEF1dGhvcml0eSAtIEcyMB4XDTExMDUwMzA3MDAwMFoXDTMxMDUwMzA3\n" \
        "MDAwMFowgbQxCzAJBgNVBAYTAlVTMRAwDgYDVQQIEwdBcml6b25hMRMwEQYDVQQH\n" \
        "EwpTY290dHNkYWxlMRowGAYDVQQKExFHb0RhZGR5LmNvbSwgSW5jLjEtMCsGA1UE\n" \
        "CxMkaHR0cDovL2NlcnRzLmdvZGFkZHkuY29tL3JlcG9zaXRvcnkvMTMwMQYDVQQD\n" \
        "EypHbyBEYWRkeSBTZWN1cmUgQ2VydGlmaWNhdGUgQXV0aG9yaXR5IC0gRzIwggEi\n" \
        "MA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC54MsQ1K92vdSTYuswZLiBCGzD\n" \
        "BNliF44v/z5lz4/OYuY8UhzaFkVLVat4a2ODYpDOD2lsmcgaFItMzEUz6ojcnqOv\n" \
        "K/6AYZ15V8TPLvQ/MDxdR/yaFrzDN5ZBUY4RS1T4KL7QjL7wMDge87Am+GZHY23e\n" \
        "cSZHjzhHU9FGHbTj3ADqRay9vHHZqm8A29vNMDp5T19MR/gd71vCxJ1gO7GyQ5HY\n" \
        "pDNO6rPWJ0+tJYqlxvTV0KaudAVkV4i1RFXULSo6Pvi4vekyCgKUZMQWOlDxSq7n\n" \
        "eTOvDCAHf+jfBDnCaQJsY1L6d8EbyHSHyLmTGFBUNUtpTrw700kuH9zB0lL7AgMB\n" \
        "AAGjggEaMIIBFjAPBgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAwIBBjAdBgNV\n" \
        "HQ4EFgQUQMK9J47MNIMwojPX+2yz8LQsgM4wHwYDVR0jBBgwFoAUOpqFBxBnKLbv\n" \
        "9r0FQW4gwZTaD94wNAYIKwYBBQUHAQEEKDAmMCQGCCsGAQUFBzABhhhodHRwOi8v\n" \
        "b2NzcC5nb2RhZGR5LmNvbS8wNQYDVR0fBC4wLDAqoCigJoYkaHR0cDovL2NybC5n\n" \
        "b2RhZGR5LmNvbS9nZHJvb3QtZzIuY3JsMEYGA1UdIAQ/MD0wOwYEVR0gADAzMDEG\n" \
        "CCsGAQUFBwIBFiVodHRwczovL2NlcnRzLmdvZGFkZHkuY29tL3JlcG9zaXRvcnkv\n" \
        "MA0GCSqGSIb3DQEBCwUAA4IBAQAIfmyTEMg4uJapkEv/oV9PBO9sPpyIBslQj6Zz\n" \
        "91cxG7685C/b+LrTW+C05+Z5Yg4MotdqY3MxtfWoSKQ7CC2iXZDXtHwlTxFWMMS2\n" \
        "RJ17LJ3lXubvDGGqv+QqG+6EnriDfcFDzkSnE3ANkR/0yBOtg2DZ2HKocyQetawi\n" \
        "DsoXiWJYRBuriSUBAA/NxBti21G00w9RKpv0vHP8ds42pM3Z2Czqrpv1KrKQ0U11\n" \
        "GIo/ikGQI31bS/6kA1ibRrLDYGCD+H1QQc7CoZDDu+8CL9IVVO5EFdkKrqeKM+2x\n" \
        "LXY2JtwE65/3YR8V3Idv7kaWKK2hJn0KCacuBKONvPi8BDAB\n" \
        "-----END CERTIFICATE-----\n"

#define BALTIMORE_ROOT_CA    \
        "-----BEGIN CERTIFICATE-----\r\n"\
        "MIIDdzCCAl+gAwIBAgIEAgAAuTANBgkqhkiG9w0BAQUFADBaMQswCQYDVQQGEwJJ\r\n" \
        "RTESMBAGA1UEChMJQmFsdGltb3JlMRMwEQYDVQQLEwpDeWJlclRydXN0MSIwIAYD\r\n" \
        "VQQDExlCYWx0aW1vcmUgQ3liZXJUcnVzdCBSb290MB4XDTAwMDUxMjE4NDYwMFoX\r\n" \
        "DTI1MDUxMjIzNTkwMFowWjELMAkGA1UEBhMCSUUxEjAQBgNVBAoTCUJhbHRpbW9y\r\n" \
        "ZTETMBEGA1UECxMKQ3liZXJUcnVzdDEiMCAGA1UEAxMZQmFsdGltb3JlIEN5YmVy\r\n" \
        "VHJ1c3QgUm9vdDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAKMEuvKr\r\n" \
```

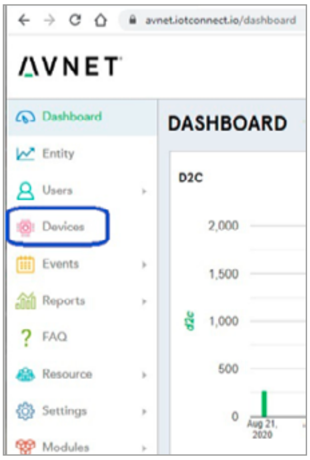
CREATE YOUR OWN DEVICE

Go to https://avnet.iotconnect.io/login, enter your credentials and access the Configuration Dashboard.

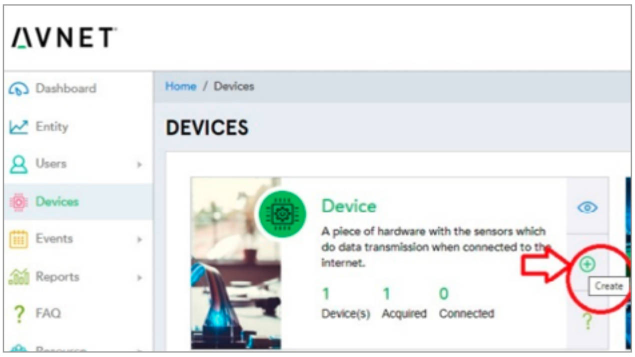


Welcome back,
Please login to your account

Click on "Devices" (circled in blue).

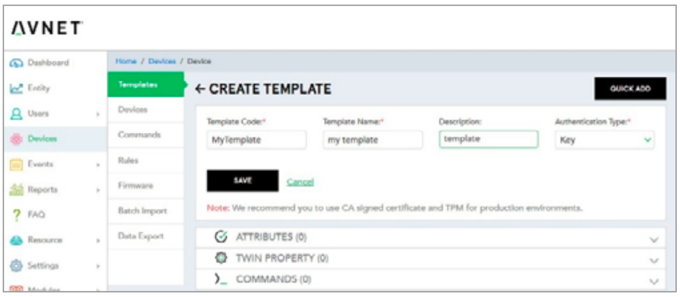


The Device box opens: click on Create (circled in red).



A piece of hardware with the sensors which do data transmission when connected to the internet.

Click on "Templates", then fill in the required fields (as shown in the image below) to create your own template.

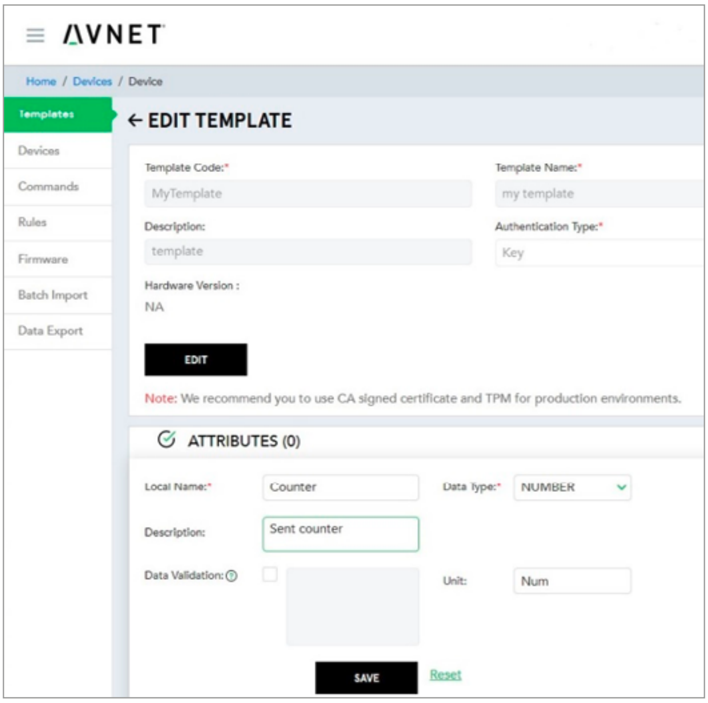


Template Code:* MyTemplate
Template Name:* my template
Description: template
Authentication Type:* Key

Note: We recommend you to use CA signed certificate and TPM for production environments.

Click SAVE.

In the “Attributes” tab, add your first attribute, then click SAVE.



← EDIT TEMPLATE

Template Code:* MyTemplate

Template Name:* my template

Description: template

Authentication Type:* Key

Hardware Version : NA

Note: We recommend you to use CA signed certificate and TPM for production environments.

ATTRIBUTES (0)

Local Name:* Counter

Data Type:* NUMBER

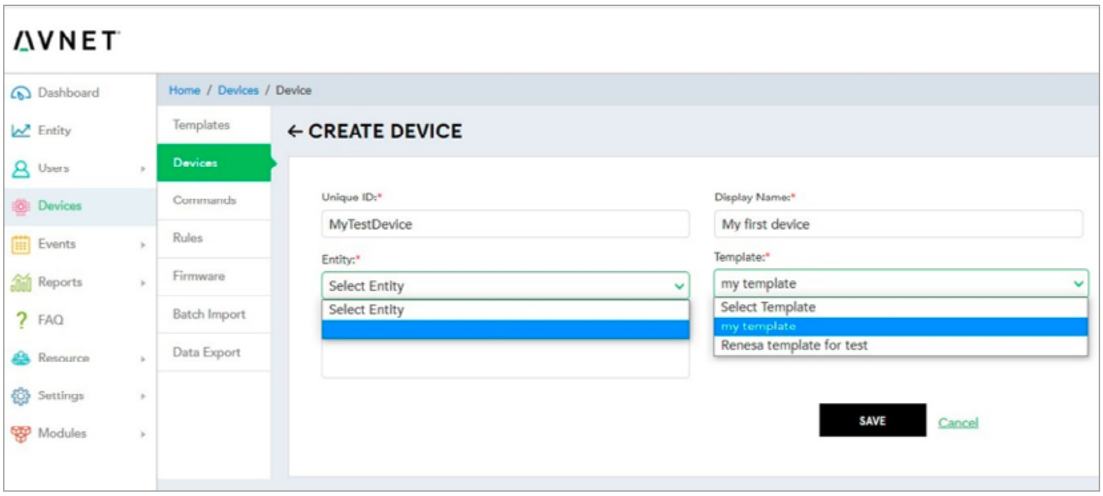
Description: Sent counter

Data Validation: □

Unit: Num

Your first sample template is now ready.

Click “Devices” in the navigation menu on the right, then the “CREATE DEVICE” button on the bottom left. You’ll see the Device creation tab. Fill in the “Unique ID” and “Display Name” fields as you like. Next, you must select the Entity (generally assigned during registration) and the template generates as described above.



← CREATE DEVICE

Unique ID:* MyTestDevice

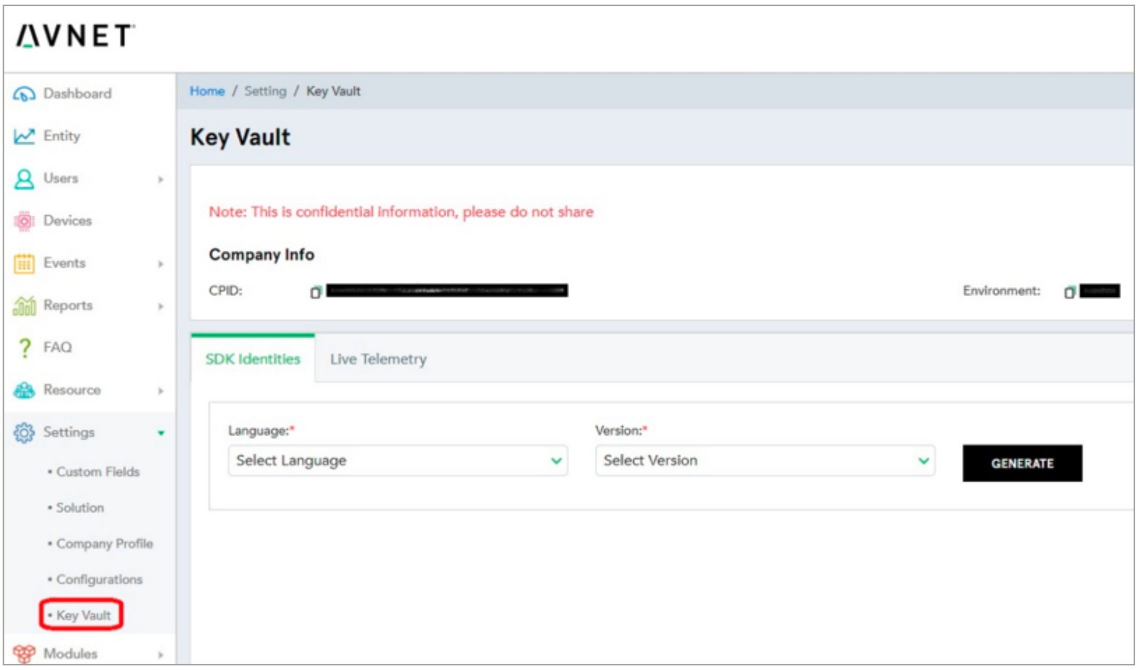
Display Name:* My first device

Entity:* Select Entity

Template:* my template

Now you are ready to send data to the device you just created on the backend.

Click on "Settings" > Key Vault to get the CPID and Environment you need to fill in the application macro.



Key Vault

Note: This is confidential information, please do not share

Company Info

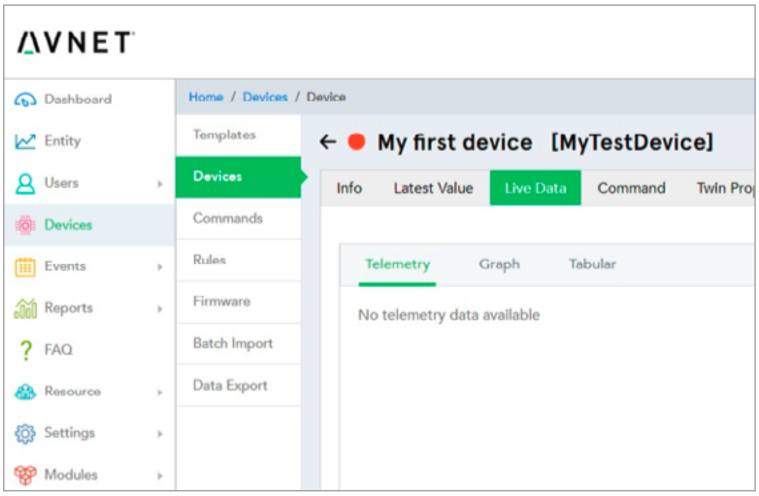
CPID: [redacted]

Environment: [redacted]

Language:* Select Language

Version:* Select Version

Click "Devices" > "Device" > "MyTestDevice" > "Live Data" to see the D2C (device to console) communication data.



My first device [MyTestDevice]

No telemetry data available

Offices

AUSTRIA
Vienna
Phone: +43 186 642 300
Fax: +43 186 642 350
wien@avnet.eu

BELGIUM
Merelbeke
Phone: +32 9 210 24 70
Fax: +32 9 210 24 87
gent@avnet.eu

BULGARIA
Sofia
sofia@avnet.eu

CZECH REPUBLIC (SLOVAKIA)
Prague
Phone: +420 234 091 031
Fax: +420 234 091 030
praha@avnet.eu

DENMARK
Herlev
Phone: +45 432 280 10
Fax: +45 432 280 11
herlev@avnet.eu

ESTONIA
(LATVIA, LITHUANIA)
Pärnu
Phone : +372 56 637737
paernu@avnet.eu

FINLAND
Espoo
Phone: +358 207 499 200
Fax: +358 207 499 280
helsinki@avnet.eu

FRANCE (TUNISIA)
Cesson Sévigné
Phone: +33 299 838 485
Fax: +33 299 838 083
rennes@avnet.eu

Illkirch
Phone: +33 390 402 020
Fax: +33 164 479 099
strasbourg@avnet.eu

Massy Cedex
Phone: +33 164 472 929
Fax: +33 164 470 084
paris@avnet.eu

Toulouse
Phone: +33 05 62 47 47
toulouse@avnet.eu

Vénissieux Cedex
Phone: +33 478 771 360
Fax: +33 478 771 399
lyon@avnet.eu

GERMANY
Berlin
Phone: +49 30 214 882 0
Fax: +49 30 214 882 33
berlin@avnet.eu

Freiburg
Phone: +49 761 881 941 0
Fax: +49 761 881 944 0
freiburg@avnet.eu

Hamburg
Phone: +49 40 608 235 922
Fax: +49 40 608 235 920
hamburg@avnet.eu

Holzwickede
Phone: +49 2301 919 0
Fax: +49 2301 919 222
holzwickede@avnet.eu

Lehrte
Phone: +49 5132 5099 0
hannover@avnet.eu

Leinfelden-Echterdingen
Phone: +49 711 782 600 1
Fax: +49 711 782 602 00
stuttgart@avnet.eu

Leipzig
Phone: +49 34204 7056 00
Fax: +49 34204 7056 11
leipzig@avnet.eu

Nürnberg
Phone: +49 911 24425 80
Fax: +49 911 24425 85
nuernberg@avnet.eu

Poing
Phone: +49 8121 777 02
Fax: +49 8121 777 531
muenchen@avnet.eu

Wiesbaden
Phone: +49 612 258 710
Fax: +49 612 258 713 33
wiesbaden@avnet.eu

HUNGARY
Budapest
Phone: +36 1 43 67215
Fax: +36 1 43 67213
budapest@avnet.eu

ITALY
Cusano Milanino
Phone: +39 02 660 921
Fax: +39 02 660 923 33
milano@avnet.eu

Firenze
Phone: +39 055 428 2301
Fax: +39 055 431 035
firenze@avnet.eu

Modena
Phone: +39 059 348 933
Fax: +39 059 344 993
modena@avnet.eu

Padova
Phone: +39 049 807 368 9
Fax: +39 049 773 464
padova@avnet.eu

Rivoli
Phone: +39 011 204 437
Fax: +39 011 242 869 9
torino@avnet.eu

Roma Tecnocittà
Phone: +39 06 412 319 10
Fax: +39 06 413 116 1
roma@avnet.eu

NETHERLANDS
Breda
Phone: +31 765 722 700
Fax: +31 765 722 707
breda@avnet.eu

NORWAY
Asker
Phone: +47 667 736 00
Fax: +47 667 736 77
asker@avnet.eu

POLAND
Gdansk
Phone: +48 58 307 81 51
Fax: +48 58 307 81 50
gdansk@avnet.eu

Katowice
Phone: +48 32 259 50 10
Fax: +48 32 259 50 11
katowice@avnet.eu

Warszawa
Phone: +48 222 565 760
Fax: +48 222 565 766
warszawa@avnet.eu

PORTUGAL
Vila Nova de Gaia
Phone: +35 1 223 779 502
Fax: +35 1 223 779 503
porto@avnet.eu

ROMANIA (BULGARIA)
Bucharest
Phone: +40 21 528 16 32
Fax: +40 21 529 68 30
bucuresti@avnet.eu

RUSSIA (BELARUS, UKRAINE)
Moscow
Phone: +7 495 737 36 70
Fax: +7 495 737 36 71
moscow@avnet.eu

Saint Petersburg
Phone: +7 812 245 1571
stpetersburg@avnet.eu

SLOVAKIA
Bratislava
Phone: +421 232 242 211
Fax: +421 232 242 210
bratislava@avnet.eu

SLOVENIA
(BOSNIA AND HERZEGOVINA,
CROATIA, MACEDONIA, MONTENEGRO,
SERBIA)
Ljubljana
Phone: +386 156 097 50
Fax: +386 156 098 78
ljubljana@avnet.eu

SPAIN
Barcelona
Phone: +34 933 278 530
Fax: +34 934 250 544
barcelona@avnet.eu

Galdàcano. Vizcaya
Phone: +34 944 572 777
Fax: +34 944 568 855
bilbao@avnet.eu

Las Matas
Phone: +34 913 727 100
Fax: +34 916 369 788
madrid@avnet.eu

SWEDEN
Sundbyberg
Phone: +46 8 587 461 00
Fax: +46 8 587 461 01
stockholm@avnet.eu

SWITZERLAND
Rothrist
Phone: +41 62 919 555 5
Fax: +41 62 919 550 0
rothrist@avnet.eu

TURKEY (GREECE, EGYPT)
Kadikoy Istanbul
Phone: +90 216 528 834 0
Fax: +90 216 528 834 4
istanbul@avnet.eu

UNITED KINGDOM (IRELAND)
Berkshire
Phone: +44 1628 512 900
Fax: +44 1628 512 999
maidenhead@avnet.eu

Bolton
Phone: +44 1204 547 170
Fax: +44 1204 547 171
bolton@avnet.eu

Bucks, Aylesbury
Phone: +44 1296 678 920
Fax: +44 1296 678 939
aylesbury@avnet.eu

Stevenage, Herts, Meadway
Phone: +44 1438 788 310
Fax: +44 1438 788 250
stevenage@avnet.eu

ISRAEL
Tel-Mond
Phone: +972 (0)9 7780280
Fax: +972 (0)3 760 1115
avnet.israel@avnet.com

SOUTH AFRICA
Cape Town
Phone: +27 (0)21 689 4141
Fax: +27 (0)21 686 4709
sales@avnet.co.za

Durban
Phone: +27 (0)31 266 8104
sales@avnet.co.za

Johannesburg
Phone: +27 (0)11 319 8600
Fax: +27 (0)11 319 8650
sales@avnet.co.za

FSC
Mixed Sources
Product group from well-managed forests and other controlled sources
www.fsc.org Cert no. IC-COC-100065
© 1996 Forest Stewardship Council

All trademarks and logos are the property of their respective owners. This document provides a brief overview only, no binding offers are intended. No guarantee as to the accuracy or completeness of any information. All information is subject to change, modifications and amendments without notice.