



EBV-IoT – Infineon IoTConnect Secure Cloud connected solution

Training Manual



Version: 1.4
Date: January 2024

Foreword

The EBV-IoT – Infineon IoTConnect Secure Cloud connected solution is a powerful and innovative system that brings together the latest advancements in IoT technology and cloud connectivity to provide a highly secure and reliable platform for businesses looking to embrace the benefits of the Internet of Things. This solution offers a range of features and capabilities that make it ideal for a wide range of applications, from smart homes and buildings to industrial automation and beyond. With its advanced security features, real-time data analytics, and flexible deployment options, the EBV-IoT – Infineon IoTConnect Secure Cloud connected solution is poised to become one of the leading solutions in the fast-growing IoT market.

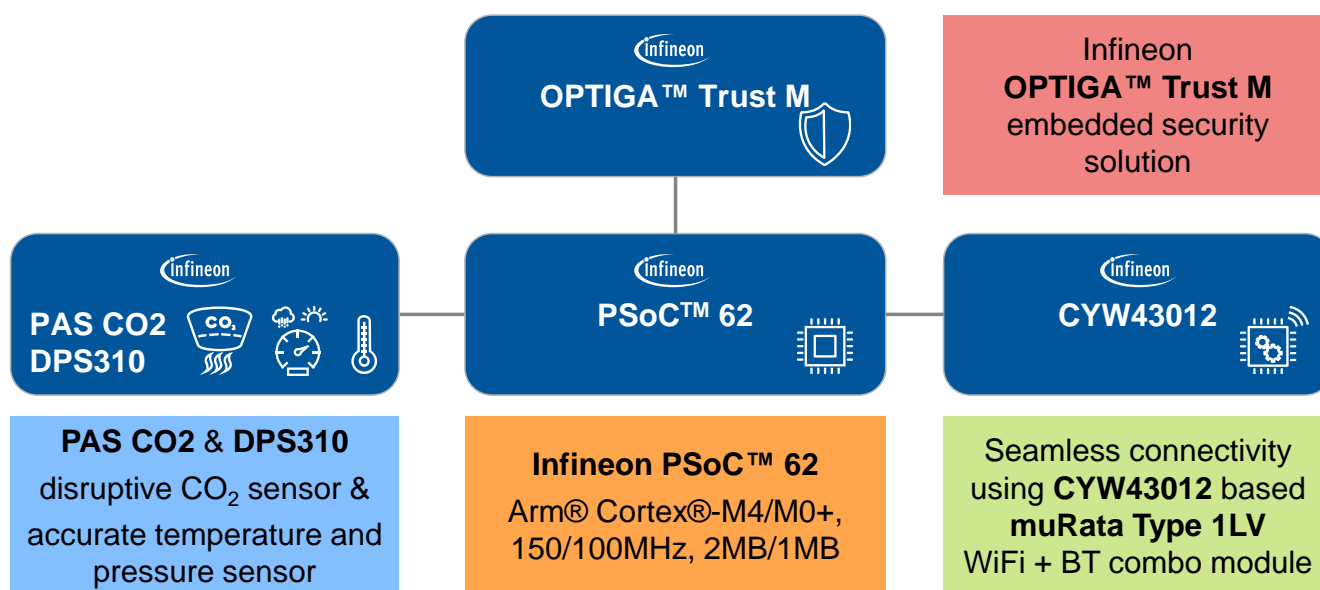
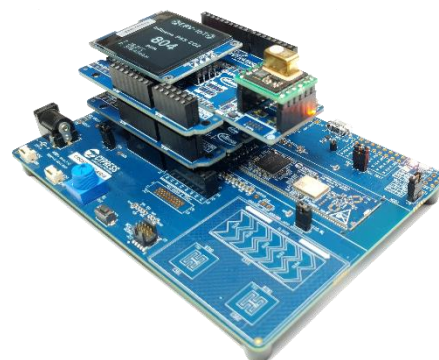


Figure 1: The solution high level overview block diagram

The solution was developed using internal expertise from EBV Elektronik and is built on a foundation of advanced hardware and software components, including Infineon PSoC™ MCU, AIROC™ CYW43012 based muRata Type 1LV WiFi/BT combo module, PAS CO2 and DPS310 pressure sensors using EBV internally developed PAS CO2 and OPTIGA Trust M evaluation shields, as well as the Avnet IoTConnect cloud platform.

The PAS CO2 evaluation shield is designed for seamless evaluation using standard hardware interface and to provide accurate and reliable CO2 monitoring for a wide range of applications, from indoor air quality monitoring to industrial automation. Similarly, the OPTIGA Trust M evaluation shield simplifies access to advanced security features that are critical for ensuring the integrity and confidentiality of data transmitted over the IoT network.

Finally, the Avnet IoTConnect cloud platform provides a highly scalable and flexible cloud infrastructure that enables the secure and efficient management of IoT devices and data. With its advanced analytics and visualization tools, as well as its powerful security features, the Avnet IoTConnect cloud platform is the perfect complement to the above hardware system.

Overall, the EBV-IoT – Infineon IoTConnect Secure Cloud connected solution represents a major leap forward in the world of IoT, providing businesses with a powerful and flexible platform for leveraging the full potential of this exciting technology.



Table of Contents

Foreword	2
Table of Contents	3
1. Introduction	6
1.1. Deliverables	7
1.2. Icon key identifiers.....	7
1.3. Document conventions.....	8
1.4. Basic requirements.....	8
1.5. Notes.....	8
1.6. Links.....	9
2. Prerequisites	10
2.1. Hardware prerequisites	10
2.2. Software prerequisites.....	11
2.2.1 Installing ModusToolbox™	11
2.2.2 Installing ModusToolbox™ Programmer.....	13
2.2.3 Running Python (“modus-shell”).....	14
2.2.4 Installing “optigatrust” Python module	15
2.2.5 Installing other Python modules.....	16
2.3. Create IoTConnect cloud account	17
3. Main system components.....	22
3.1. PSoC 62.....	22
3.1.1 Main features	22
3.2. AIROC™ CYW43012 WiFi/BT combo.....	24
3.3. XENSIV™ PAS CO2 sensor	25
3.4. Xensiv™ DPS310 – digital barometric pressure sensor.....	26
3.5. OPTIGA™ Trust M	27
4. System setup using evaluation hardware	29
4.1. The PSoC Pioneer Kit	29
4.2. EBV IoT – Infineon PAS CO2 Evaluation Shield.....	29
4.3. EBV-IoT – OPTIGA Trust M Evaluation Shield	30
5. Hardware configuration	32
5.1. PAS CO2 hardware setup	32
5.1.1 Out-of-box setup	32
5.1.2 I2C interface configuration (after using any other jumper configuration)	32
5.2. OPTIGA shield hardware setup.....	32
5.2.1 Out-of-box setup	33
5.2.2 I2C interface configuration (after using any other jumper configuration)	33
6. EBV IoTConnect – Solution Accelerator Software	34
7. Task 1: PSoC programming and debugging – Hello world.....	36
7.1. Overview	36



7.2.	ModusToolbox Overview	36
7.3.	Step by step guide.....	36
7.3.1	HW setup	37
7.3.2	Launch Eclipse IDE for ModusToolbox	37
7.3.3	Create a project	38
7.4.	IDE overview.....	41
7.4.1	Project Explorer	42
7.4.2	Quick Panel	43
7.4.3	Build application	43
7.4.4	Program application	44
7.4.5	Accessing the device through serial terminal.....	44
7.4.6	IDE integrated serial communication terminal.....	45
7.4.7	Debug application	45
8.	Task 2: PSoC programming and debugging – Adding sensors.....	47
8.1.	Overview	47
8.2.	Step by step guide.....	47
8.2.1	HW setup	47
8.2.2	Create a project	47
9.	Task 3: OPTIGA™ provisioning for IoTConnect Cloud	54
9.1.	Overview	54
9.2.	Step by step guide.....	55
9.2.1	Retrieve IoTConnect cloud account details	55
9.2.2	Edit OPTIGA programming script	56
9.2.3	Program the secure element	56
9.2.4	Get QR code for seamless device onboarding	58
10.	Task 4: Create a device in IoTConnect Cloud	59
10.1.	Overview.....	59
10.2.	Step by step guide	59
11.	Task 5: Getting hardware ready for IoTConnect cloud.....	64
11.1.	Overview.....	64
11.2.	Step by step guide	64
11.2.1	HW setup	64
11.2.2	Import project	65
11.2.3	Configure WiFi credentials	68
11.2.4	Build.....	68
11.2.5	Program and run the device application	68
12.	Task 6: Flashing device firmware	70
13.	Task 7: Accessing the device through serial terminal	72
13.1.1	Setting WiFi credentials using serial terminal	72
14.	Task 8: Connecting the demo to IoTConnect Cloud	75

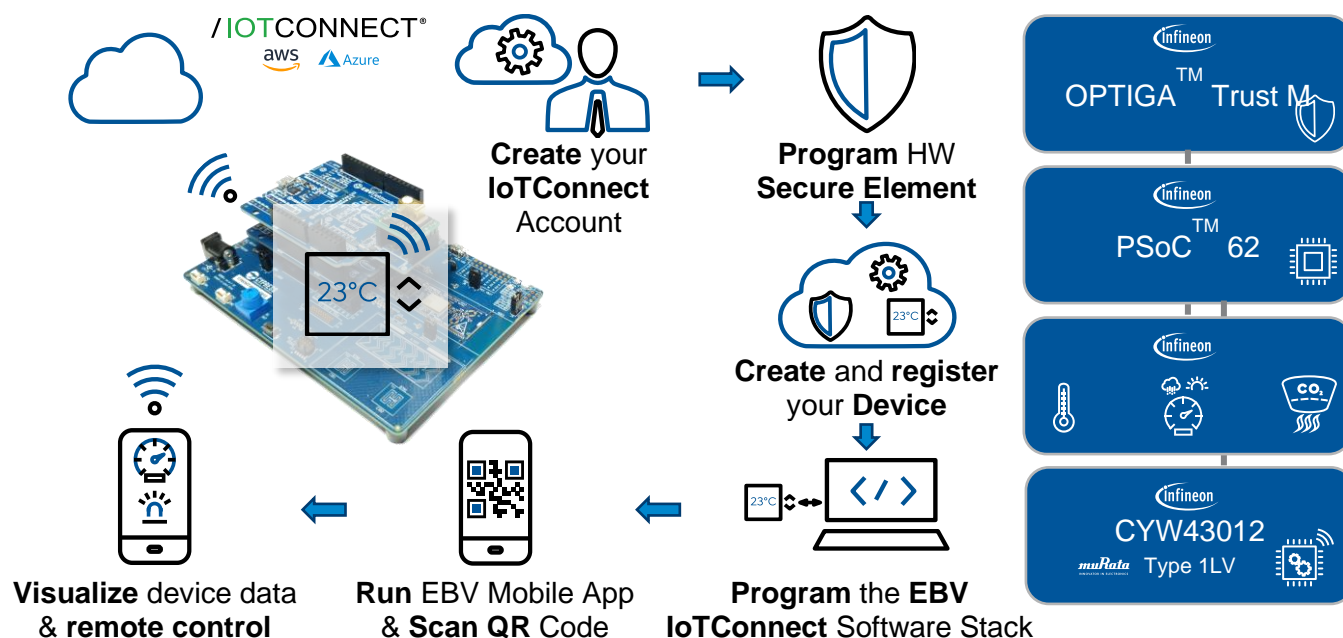


14.1.	Overview.....	75
14.2.	Step by step guide	75
15.	Task 9: Remote access from EBV Mobile App	78
15.1.	Overview.....	78
15.2.	Step by step guide	78
15.2.1	Install the EBV Mobile App on your smartphone.....	78
15.2.2	Open the app and connect to your IoTConnect OEM root account.....	80
15.2.3	End-user signs up to the OEM cloud account.....	80
15.2.4	Add a device to the end-user cloud account.....	81
15.2.5	Device list and menu overview	82
15.2.6	Interacting with the device	83
	Thank you!	85
	Revision history	86

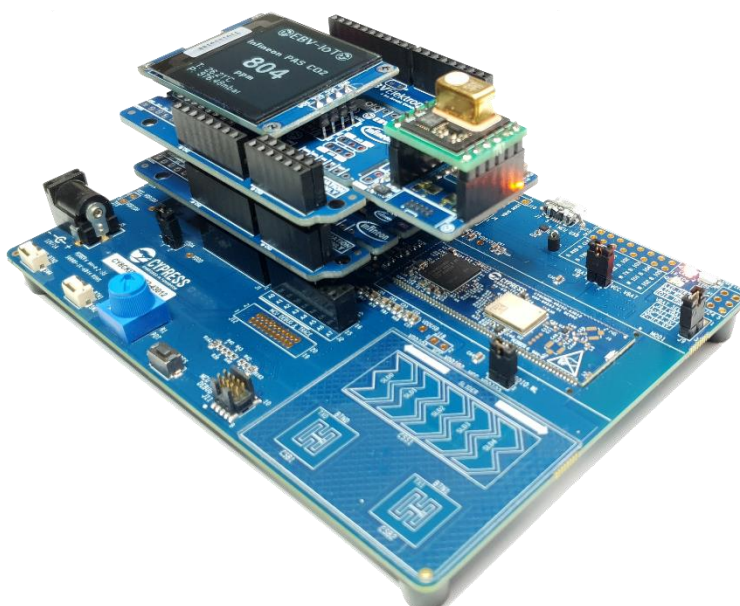


1. Introduction

EBV Elektronik has designed a complete onboarding flow on top of Avnet IoTConnect cloud to simplify IoT device onboarding and cloud platform evaluation. From building a complete cloud environment in an instant, leveraging the challenge of building state of the art secured IoT device (using EBV Elektronik hardware and software), to bringing data to end user via the EBV Mobile App. This cloud ecosystem aims at offering a fast, innovative, and scalable platform which provides real product differentiation to customers at minimal cost.



The goal of this training is to connect the demo setup to the Avnet IoTConnect cloud using one of the best-in-class hardware security thanks to Infineon OPTIGA™ Trust M secure element. As main platform we will be using Infineon PSoC 62 MCU and CYW43012 based Wi-Fi/BT combo connectivity module. A set of Infineon Xensiv™ sensors will complete the device hardware setup to provide a more advanced product.



This training manual will guide you through a hands-on session following every step on the device secure cloud connection path. The training demonstrates how to build, develop and deploy an affordable, state of the art, end-to-end secured solution.



The following topics will be covered:

- Infineon PSoC 62 Programming & JTAG Debugging.
- Infineon OPTIGA Trust M provisioning for IoTConnect Cloud.
- Connecting the demo setup to IoTConnect Cloud.
- Remote Control from the EBV Mobile App.
- Creating custom dashboard



Quick start: There is a “Quick Start” path available marked using fast forward icon. Follow the icon to skip the parts related to device programming and debugging using ModusToolbox™.

1.1. Deliverables

The training material is provided during the Training sessions and comes as:

- EBV-IoTC Training Zip (.zip file)
- EBV-IoTC Training no MTB Zip (.zip file without ModusToolbox)

The different resources needed to complete this training (hands-on documentation, datasheets, application notes, software & tools) are included in the delivery. However, the training manual always refer you to latest documentation provided by the manufacturers on their respective websites.

The training Zip is composed of following main folders:

- **EBV Documentation:** documentation and related information for EBV-IoT products
- **Firmware:** PSoC executable for IoTConnect demo
- **Infineon documentation:** documentation and related information for Infineon products
- **MobileApp:** source code for EBV-IoT Mobile Application
- **MTB Project:** ModusToolbox project used to build the IoTConnect Secured Solution
- **Scripts:** Python scripts for Optiga management
- **Sources:** source files for ModusToolbox projects to be used with the training
- **Template:** IoTConnect device template to upload to the cloud.
- **Tools:** installation files for the tools to be used during this training.

In the ‘root’ folder there are this **training manual** and **training overview**.

1.2. Icon key identifiers

Icons are used to notify about valuable information or point out important steps to follow. These icons are:



Note: Provides valuable information about a specific topic



Tip: Indicates useful tips and techniques



To Do: Highlights objectives to be completed and possible extensions



Task accomplished: Highlights the expected results of the specific task



Important: Indicates important information



Quick start: Showing the path for Quick Start version of the manual



1.3. Document conventions

Table 1: Document Conventions for the manual

Convention	Usage
<i>Italic-Bold</i>	In document or externally referenced link
Bold	Clickable menu item
Bold Squared	Clickable web page or menu item to be clicked during the tasks
Consolas	Commands to be executed or messages displayed in terminal

1.4. Basic requirements

Following will be required during the training:

1. A computer with Windows 10 or above, or Linux
2. Android or Apple device to install mobile app (it also enables to seamlessly follow the training using online manual)
3. Hardware requirements:
 - a. The PSoC™ 62S2 Wi-Fi BT Pioneer Kit (CY8CKIT-062S2-43012)
 - b. EBV-IoT – Infineon PAS CO2 Evaluation Shield
 - c. EBV-IoT – Infineon OPTIGA Trust M Evaluation Shield
 - d. 128x128 pixels OLED display (optional)
4. Software requirements:
 - a. ModusToolBox 3.1 or above
 - b. Python (modus-shell may be used) with installed libraries
5. Avnet IoTConnect cloud platform account



Note: There is a **2 Prerequisites** section with detailed requirements available.

1.5. Notes

Please follow the steps carefully!



Note: Some of the steps marked A, B, C... have same outcome, just different way to get there.

There is online version of the training manual available you can access during the hands-on using your mobile phone or tablet. It improves pace as you do not have to switch between applications. Use following QR code to access the online manual.



Note: Only hands-on relevant sections are available online.

Not all the steps are required to follow in full. Based of your preferences, you can skip some steps and follow the tips to overtake some parts and allows you to focus more on a topic of your choice.



Tip: Tips are providing you information on steps you can eventually skip and focus more on topics of your choice.



1.6. Links

Following please find additional links you may find useful or is being used as extra reading material:

1. ***ModusToolBox landing page***
2. ***MyInfineon landing page***
3. ***PSoC™ landing page***
4. ***Xensiv™ landing page***
5. ***OPTIGA™ Trust M landing page***



2. Prerequisites

Following please find hardware and software prerequisites.

2.1. Hardware prerequisites



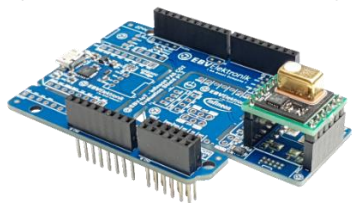
Important: To run the complete workshop hands on part a computer with admin/root rights should be used. Windows OS is preferred, while for Linux should be able to run as well.

Following evaluation hardware is going to be used during the training:

- The PSoC™ 62S2 Wi-Fi BT Pioneer Kit (CY8CKIT-062S2-43012)
- EBV-IoT – Infineon PAS CO2 Evaluation Shield
- EBV-IoT – Infineon OPTIGA Trust M Evaluation Shield
- 128x128 pixels OLED display (optional)



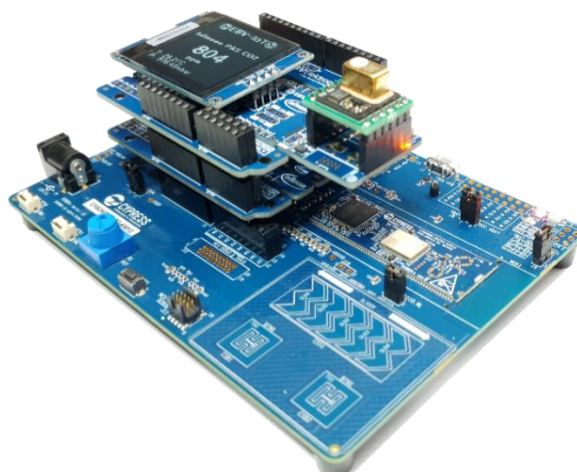
The PSoC™ 62S2 Wi-Fi BT Pioneer Kit
(CY8CKIT-062S2-43012)



EBV-IoT – Infineon PAS CO2 Evaluation Shield



EBV-IoT – Infineon OPTIGA Trust M Evaluation Shield



2.2. Software prerequisites

2.2.1 Installing ModusToolbox™



Quick start: For the “Quick Start” version of the Infineon IoTConnect Cloud connected solution ModusToolbox™ is not required and you can skip to Sub-section 2.2.2 *Installing ModusToolbox™ Programmer*. However, if you would like to get familiar with Infineon ModusToolbox™, we still recommend installing it for your reference.

Please refer to the Infineon’s **ModusToolbox™ Software** pages for application download and “Installation guide” ([link](#)).



Note: It is strongly recommended to install the tool into the root of any of your disk drives e.g. “D:/Infineon”. Please do keep the name as short as possible and do not use spaces or any special characters.

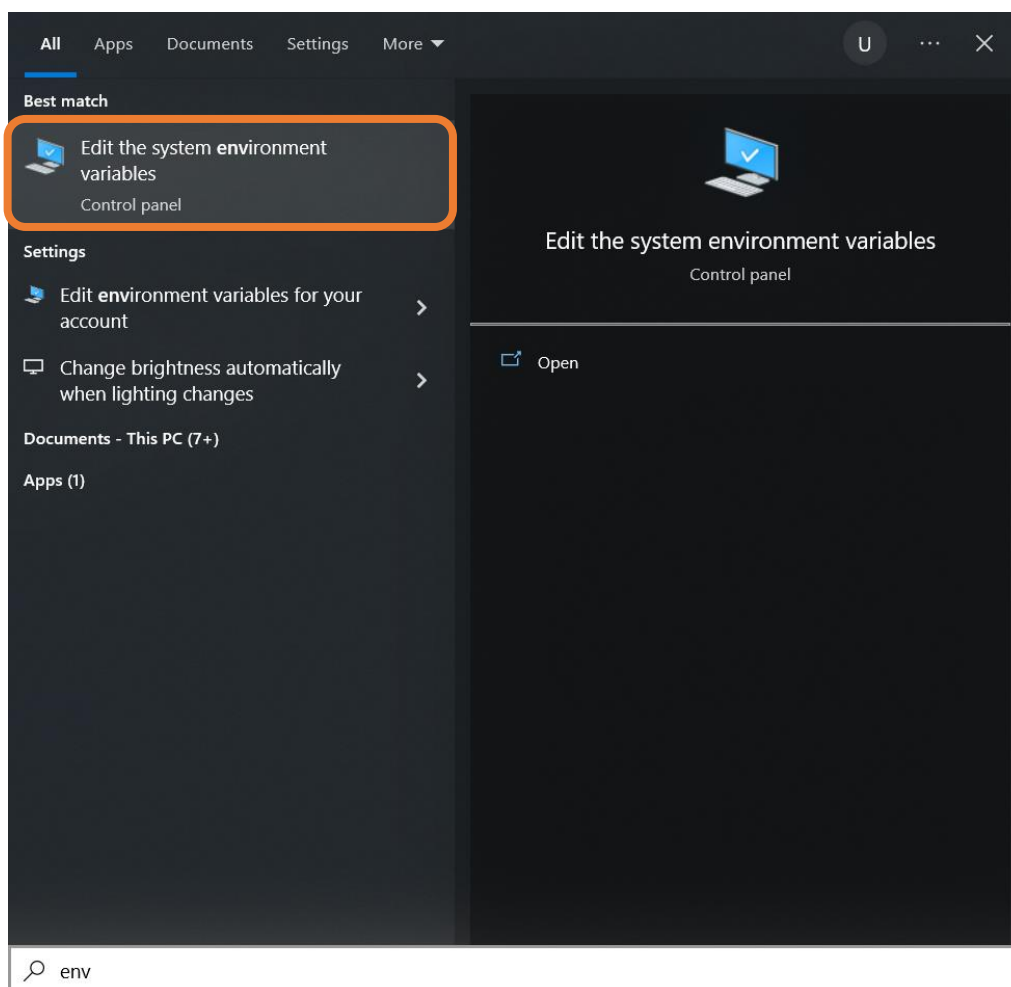
Default location of installation directory for the Eclipse IDE is:

`<install_path>ModusToolbox\ide_<version>\eclipse\`

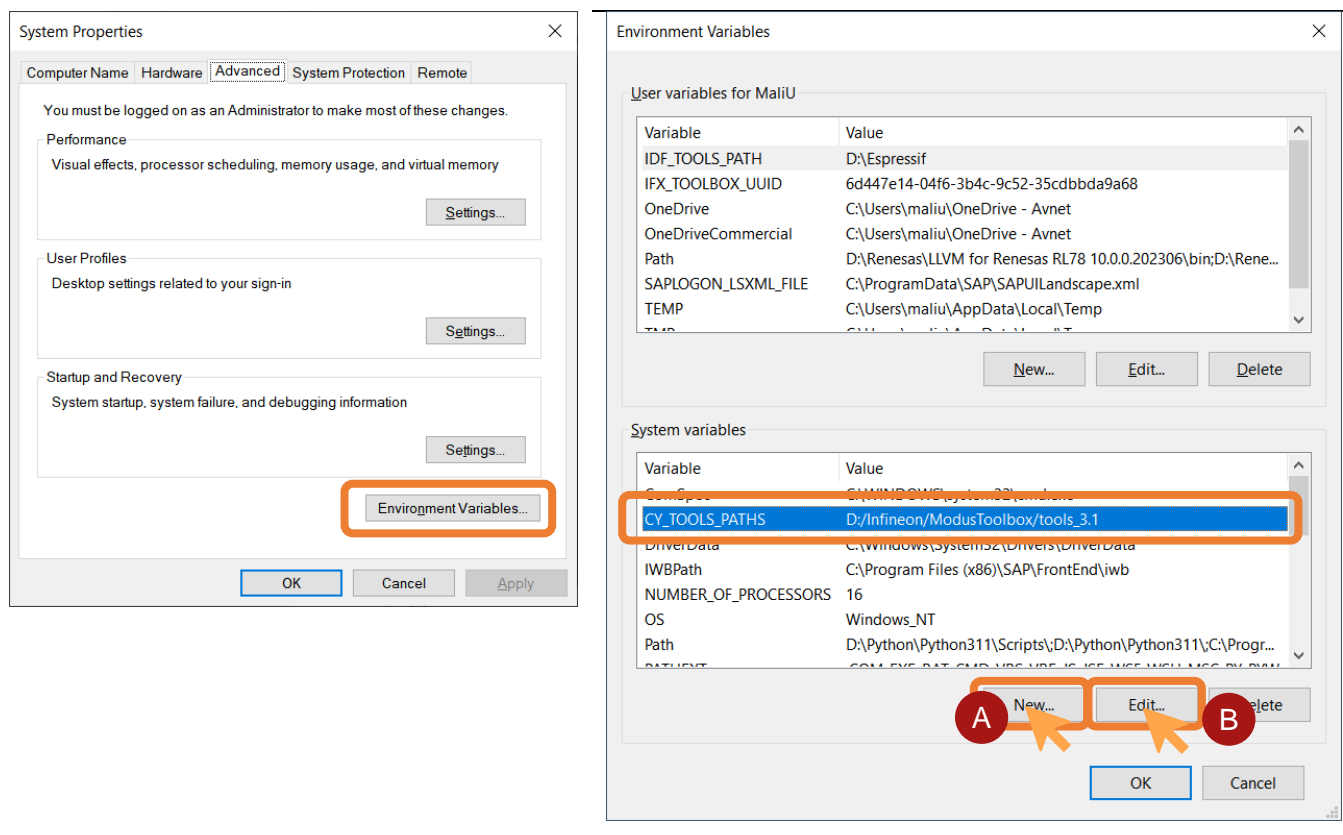


Important: If the software is installed in any other than default location you have to set an environment variable. Please follow the **ModusToolbox installation guide** or follow next steps.

To check for ModusToolBox environmental variable setting, under Windows **Start menu** or **Windows search** type in “env”... . The **Edit the system variables** should soon be listed on top as shown on Figure below.

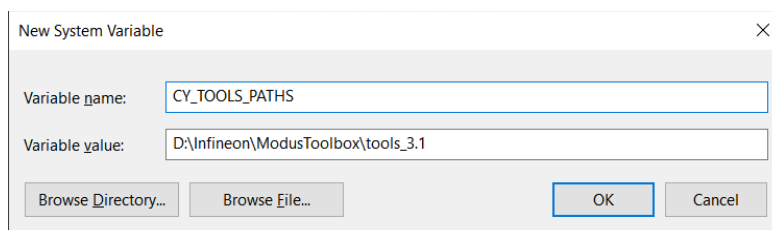


Now click on **Environmental Variables** and look for “System Variable” CY_TOOLS_PATHS.

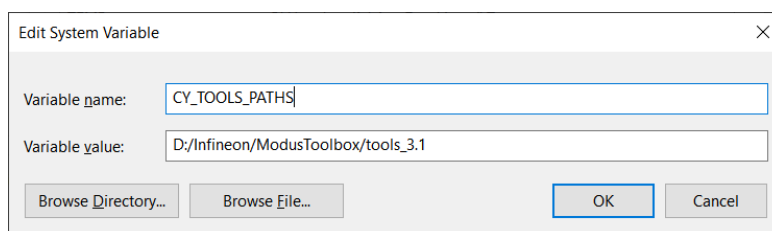


Please proceed accordingly:

- A. If the variable does not exist, then in “Environment Variables” window click **New**, then in new pop-up window “New System Variable” at “Variable name” type CY_TOOLS_PATHS and in “Variable value” put your ModusToolBox path or browse using **Browse Directory**. Then click **OK** as well as click **OK** in “Environment Variables” window.



- B. If the variable exist, then in “Environment Variables” window click **Edit**, then in new pop-up window “Edit System Variable” check if “Variable name” matches to CY_TOOLS_PATHS and “Variable value” matches to your ModusToolBox path or browse using **Browse Directory**. Then Click **OK** as well as click **OK** in “Environment Variables” window.



Important: Very important – use **forward slashes**. / - ok; \ - not ok



After successful installation you should be able to see the tools installed in your start menu under “ModusToolbox 3.1 (Current user)”.

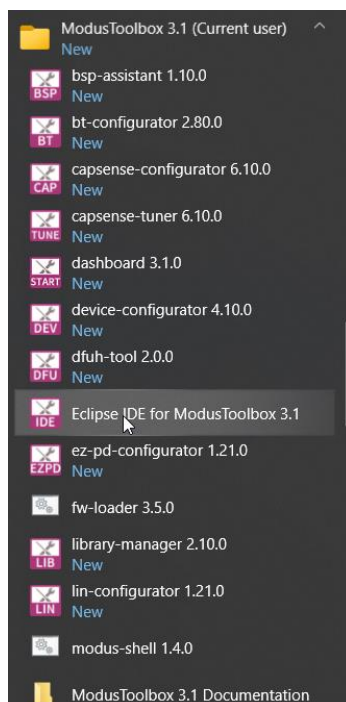
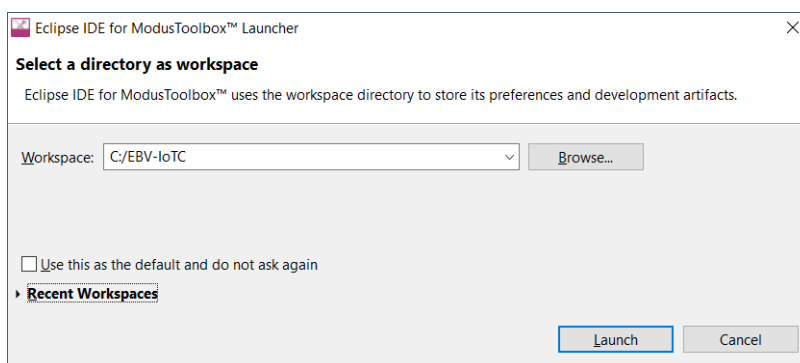


Figure 2: Start menu with ModusToolbox installed

If clicked on **Eclipse IDE for ModusToolBox 3.1** you will see a “Launcher” window popping out.



Task accomplished: If you can run “Eclipse IDE for ModusToolbox™ Launcher” and a workspace selection window is opening, you are good to proceed to a next step.

2.2.2 Installing ModusToolbox™ Programmer



Note: For the “Full” version of the Infineon IoTConnect Cloud connected solution ModusToolbox™ Programmer is not required and you can skip to Sub-section 2.2.3 *Running Python (“modus-shell”)*

Please refer to the Infineon’s **ModusToolbox™ Programmer Software** pages for application download and “ModusToolbox™ Programming GUI user guide” ([link](#)) for installing instructions.



After successful installation you should be able to see the tools installed in your Windows **Start menu** under:

- A. **Recently added** → **mtb-programmer x.x**, or
- B. **Infineon technologies** → **mtb-programmer x.x**

where x.x denotes version of the installed tool.

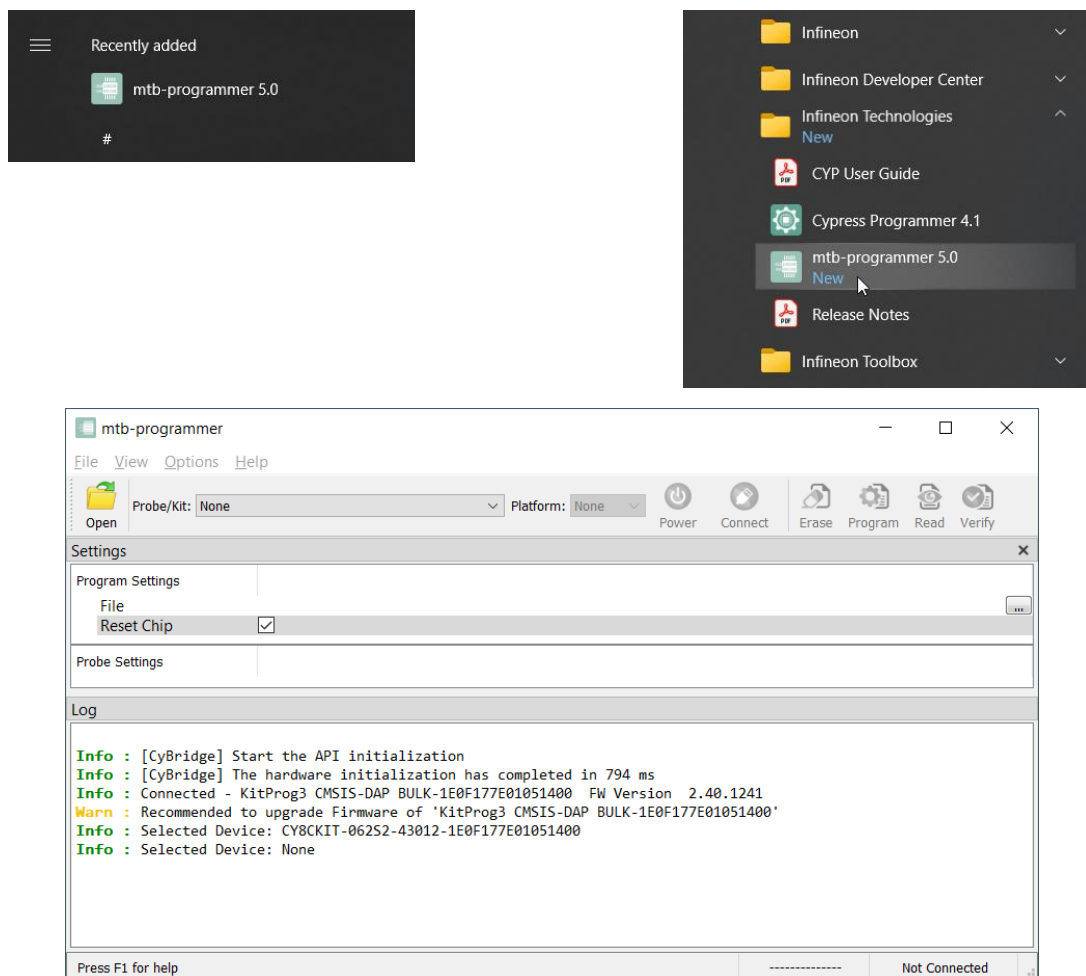


Figure 3: ModusToolbox™ Programmer



Task accomplished: If you can run “mtb-programmer” as depicted in *Figure 3: ModusToolbox™ Programmer* you are good to proceed to a next step.

2.2.3 Running Python (“modus-shell”)

Python is required for OPTIGA™ Trust M provisioning.



Note: If you have Python already installed you can proceed to a step 2.2.4 *Installing “optigatrust” Python module.*



If you do not have Python nor ModusToolbox™ (as one of the prerequisites of the “Full” version of the workshop) installed, you can either:

- Install Python from the “python.org” pages ([link](#)), or
- Install ModusToolbox™ (as in Sub-section 2.2.1 *Installing ModusToolbox*) which comes with “modus-shell” Python instance as part of the installation

Having ModusToolbox™ installed, the easiest way of running “modus-shell” is to look for it under Windows search by typing in “modus”... . The application should soon be listed on top as shown on *Figure 4*. Click on the “App” to run the application.

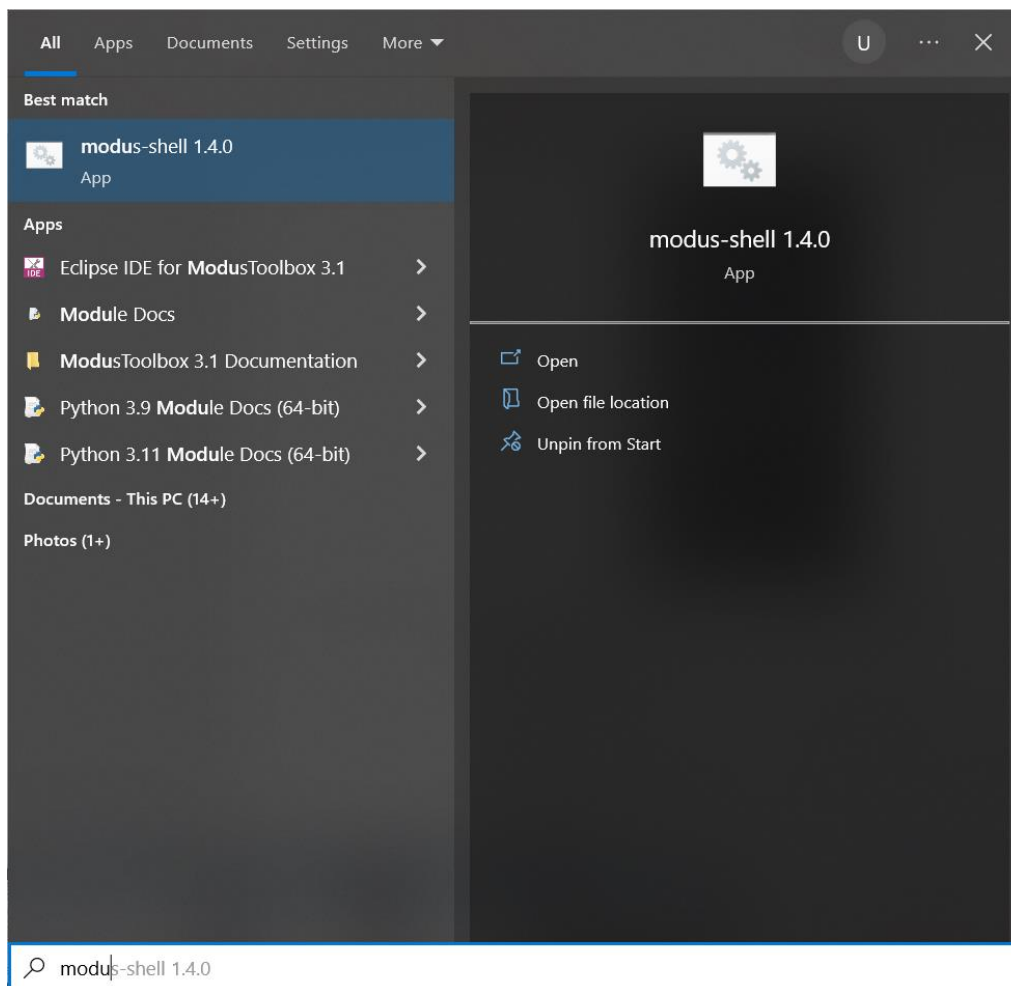


Figure 4: Starting “modus-shell”

Now run either your instance of Python or “modus-shell” Python instance. Please check whether “optigatrust” Python module is installed. Type “optigatrust.exe” into the terminal. If your replay is “bash: optigatrust.exe: command not found”, please proceed to Sub-section 2.2.3 **Error! Not a valid bookmark self-reference.** otherwise proceed to Sub-section 2.2.5 *Installing other Python modules*.

```

user@computer_name ~
$ optigatrust.exe
bash: optigatrust.exe: command not found

user@computer_name ~
$
  
```

2.2.4 Installing “optigatrust” Python module

Please install the “optigatrust” Python module by executing “pip install optigatrust”. Installation is successful if “Successfully installed...” message appears.



```

user@computer_name ~
$ pip install optigatrust
Collecting optigatrust
  Downloading optigatrust-1.3.7-py3-none-any.whl (493 kB)
    |#####| 493 kB 2.2 MB/s
Requirement already satisfied: cryptography in d:\infineon\modustoolbox\tools_3.1\python\lib\site-
packages (from optigatrust) (36.0.1)
Requirement already satisfied: click in d:\infineon\modustoolbox\tools_3.1\python\lib\site-packages
(from optigatrust) (8.0.4)
Collecting jinja2
  Downloading Jinja2-3.1.2-py3-none-any.whl (133 kB)
    |#####| 133 kB 6.8 MB/s
Collecting pyserial
  Using cached pyserial-3.5-py2.py3-none-any.whl (90 kB)
Collecting asn1crypto
  Downloading asn1crypto-1.5.1-py2.py3-none-any.whl (105 kB)
    |#####| 105 kB 6.4 MB/s
Requirement already satisfied: colorama in d:\infineon\modustoolbox\tools_3.1\python\lib\site-
packages (from click->optigatrust) (0.4.5)
Requirement already satisfied: cffi>=1.12 in d:\infineon\modustoolbox\tools_3.1\python\lib\site-
packages (from cryptography->optigatrust) (1.15.1)
Requirement already satisfied: pycparser in d:\infineon\modustoolbox\tools_3.1\python\lib\site-
packages (from cffi>=1.12->cryptography->optigatrust) (2.21)
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-2.1.2-cp38-cp38-win_amd64.whl (16 kB)
Installing collected packages: MarkupSafe, pyserial, jinja2, asn1crypto, optigatrust
Successfully installed MarkupSafe-2.1.2 asn1crypto-1.5.1 jinja2-3.1.2 optigatrust-1.3.7 pyserial-3.5
WARNING: You are using pip version 21.1.3; however, version 23.0.1 is available.
You should consider upgrading via the 'D:\Infineon\ModusToolbox\tools_3.1\python\python.exe -m pip
install --upgrade pip' command.

user@computer_name ~
$
  
```

Quick check for proper “optigatrust” Python module installation is to run “optigatrust” or “optigatrust.exe” command. We should see a replay with short usage instructions.

```

user@computer_name ~
$ optigatrust.exe
Usage: optigatrust [OPTIONS] COMMAND [ARGS]...

Options:
  --version  Show the version and exit.
  --help     Show this message and exit.

Commands:
  create-keys  Generate a keypair
  object       Manages objects data and metadata
  update       Use protected update feature
  update-wizard  Guide through the protected update preparation for a...

user@computer_name ~
$
  
```

2.2.5 Installing other Python modules

To run all the steps related to secure element interfacing using Python scripts, we need to add OpenSSL, QrCode and image related modules. Please install the modules using following commands:

- “pip install pyopenssl”
- “pip install pillow”
- “pip install qrcode”



```

user@computer_name ~
$ pip list
Package            Version
-----
appdirs             1.4.4
.
openssl-python     0.1.1
optigatrust         1.3.7
Pillow              9.5.0
.
pyOpenSSL           23.1.1
.
qrcode              7.4.2
.
WARNING: You are using pip version 21.1.3; however, version 23.1.2 is available.
You should consider upgrading via the 'D:\Infineon\ModusToolbox\tools_3.1\python\python.exe -m pip
install --upgrade pip' command.

user@computer_name ~
$

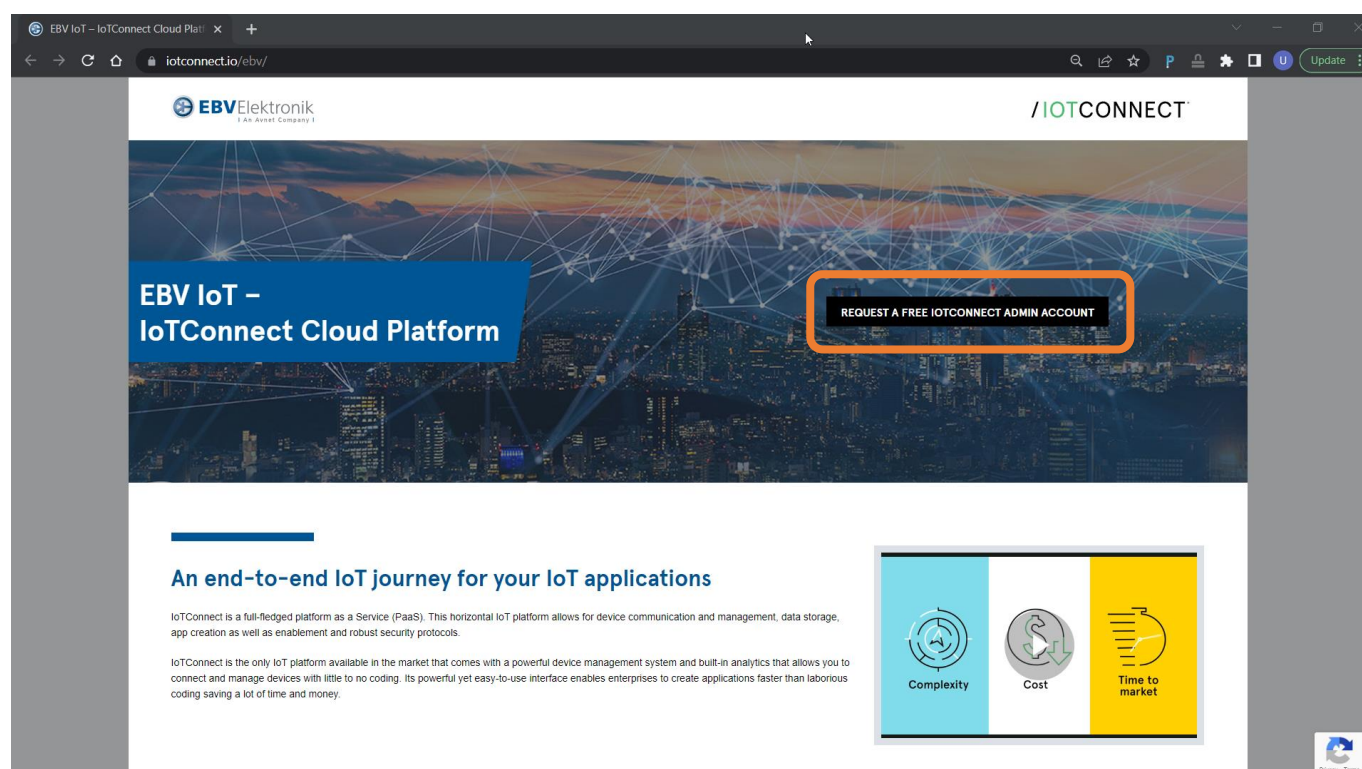
```



Task accomplished: If your “pip list” is listing “optigatrust”, “pyopenssl”, “pillow”, and “qrcode” you are fulfilling Python related requirements.

2.3. Create IoTConnect cloud account

As a first step in every cloud platform it is to create your cloud access account. Use your favourite browser to access EBV IoTConnect website at <https://iotconnect.io/ebv> and click **Request a Free IoTConnect Account**.



Enter your details and make sure your email address is valid then click **Submit** button:

Request a Free IoTConnect Admin Account

FIRST NAME *

LAST NAME *

EMAIL *

PASSWORD *

COMPANY NAME *

COUNTRY CODE * PHONE *

ADDRESS *

COUNTRY *
Select Country

STATE *
Select State

TIMEZONE *
(UTC+01:00) Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna

CITY NAME *

POSTAL CODE *

SUBMIT



Important: Password must contain at least 1 upper case and special character.

If the provided information is correct, a confirmation popup invites you to check your emails for an email with a link to validate your email address. Press **OK** button: Usually, the browser returns you to the beginning of the page.

iotconnect.io says

Your details have been saved successfully. Please check your inbox to verify your email address and complete registration process.

OK

Please check your mailbox. You should be receiving email with a content like one bellow. Click on **Verify Email Now**.

Hello Uros Mali,

Welcome to IoTConnect!

Please click below link to verify your email address and complete the registration process for your free IoTConnect account.

[Verify Email Now](#)

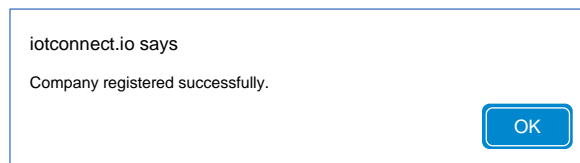
Best Regards,
The IoTConnect team

ssc@ebv.com

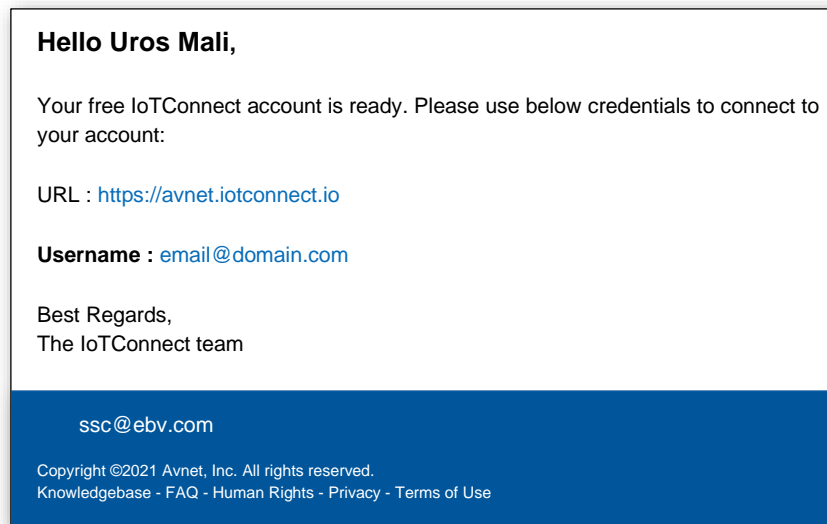
Copyright ©2021 Avnet, Inc. All rights reserved.
Knowledgebase - FAQ - Human Rights - Privacy - Terms of Use



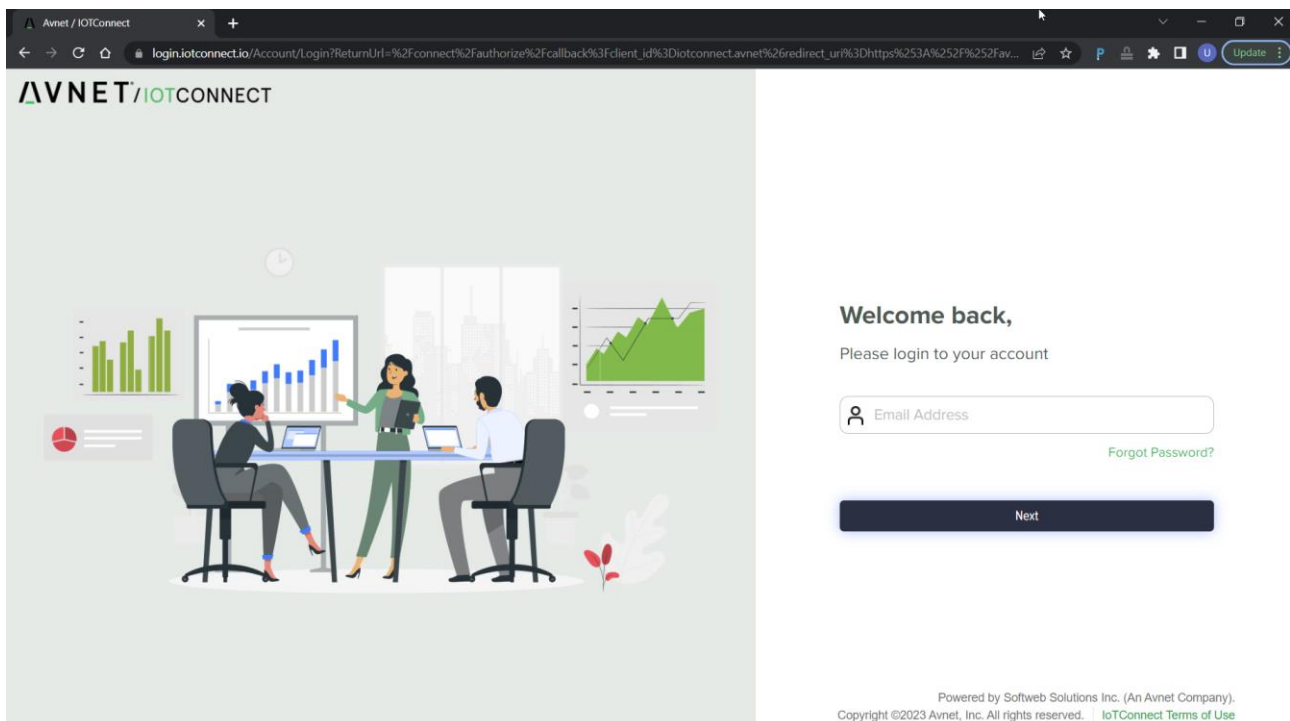
The webpage loads again and process the creation of your account in the background. Once complete, below popup should open to confirm your IoTConnect account has been created successfully. Click **OK** button:



Please check your mailbox. You should now get confirmation email with a link to the cloud platform.



Login to the cloud using email and password you provided during registration process.



If it happens you forgot your password, you can reset the password by clicking **Forgot Password?** at login page.



Reset your password by providing new one and confirmation of the same password and click **Submit**.

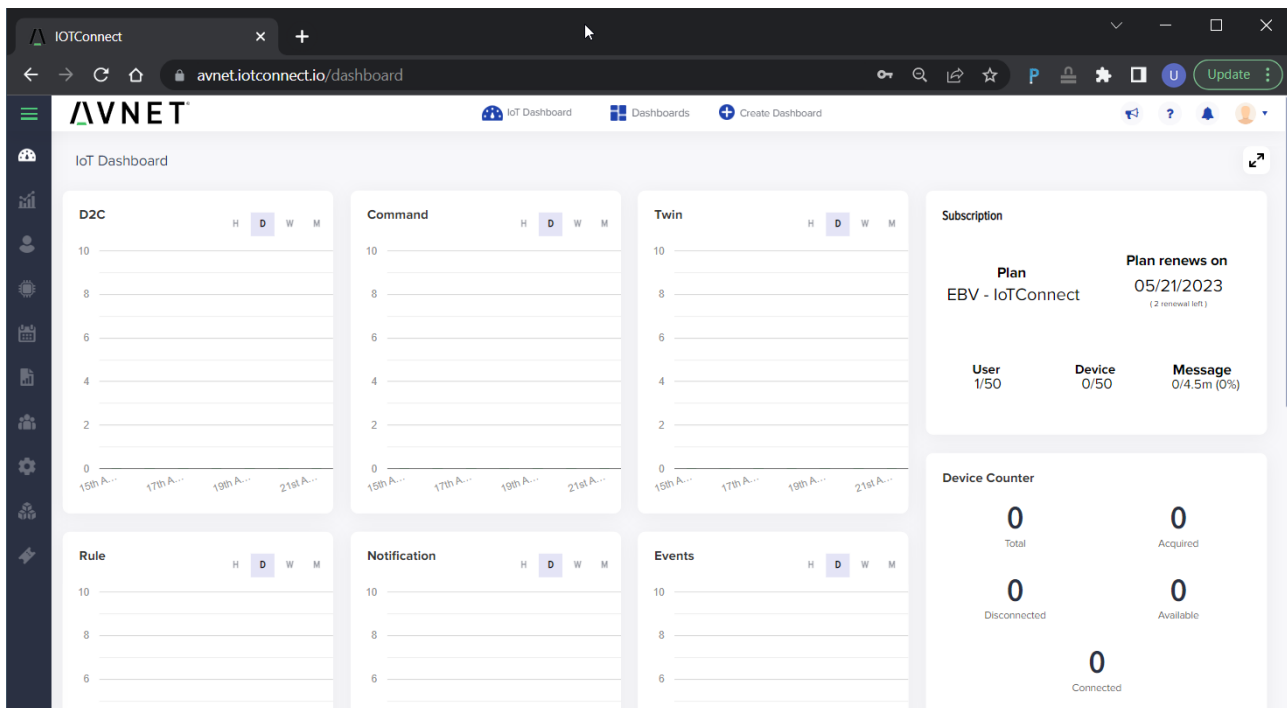


Important: Password must contain at least 1 upper case and special character.

You should be now seeing “Welcome back” page.



After a successful login, the IoTConnect backend will land to the Dashboard page showing information related to device activities (device to cloud messages, command, twin properties, rule, notification, events, etc) and the current subscription plan:



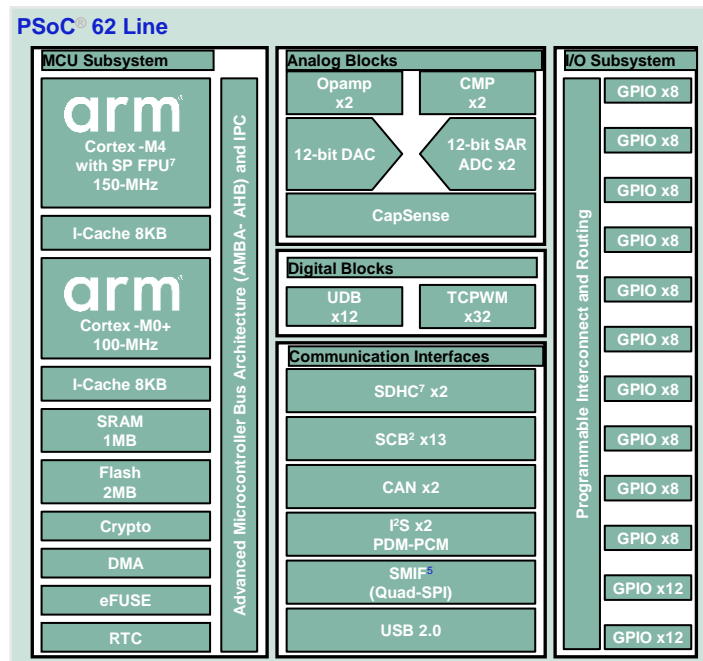
Task accomplished: Congratulations, you have just created your free Avnet IoTConnect cloud account. Save your IoTConnect login credentials securely for your next use.



3. Main system components

3.1. PSoC 62

The PSoC[®] MCU is a scalable and reconfigurable platform architecture that supports a family of programmable embedded system controllers with Arm[®] Cortex[®] CPUs (single and multi-core). The PSoC 62 product family, based on the PSoC 6 MCU platform, is a combination of a dual-core microcontroller with built-in programmable peripherals. It incorporates integrated low-power flash technology, digital programmable logic, high-performance analog-to-digital and digital-to-analog conversion, low-power comparators, touch sensing, serial memory interface with encryption, and standard communication and timing peripherals.



3.1.1 Main features

32-bit Dual CPU Subsystem

- 150-MHz Arm[®] Cortex[®]-M4F (CM4) CPU with single-cycle multiply, floating point, and memory protection unit (MPU)
- 100-MHz Cortex-M0+ (CM0+) CPU with single-cycle multiply and MPU
- User-selectable core logic operation at either 1.1 V or 0.9 V
- Active CPU current slope with 1.1-V core operation
- Active CPU current slope with 0.9-V core operation
- Two DMA controllers with 16 channels each

Memory Subsystem

- 1-MB application flash, 32-KB auxiliary flash (AUXflash), and 32-KB supervisory flash (Sflash); read-while-write (RWW) support. Two 8-KB flash caches, one for each CPU
- 288-KB SRAM with power and data retention control
- One-time-programmable (OTP) 1-Kb eFuse array

Low-Power 1.7-V to 3.6-V Operation

- Six power modes for fine-grained power management
- Deep Sleep mode with SRAM retention
- On-chip Single-In Multiple Out (SIMO) DC-DC Buck converter
- Backup domain and real-time clock

Flexible Clocking Options

- On-chip crystal oscillators
- Phase-locked loop (PLL) for multiplying clock frequency
- Internal main oscillator (IMO)
- Ultra-low-power internal low-speed oscillator (ILO)
- Frequency locked loop (FLL) for multiplying IMO frequency



Quad-SPI (QSPI)/Serial Memory Interface (SMIF)

- Execute-In-Place (XIP) from external Quad SPI Flash
- On-the-fly encryption and decryption
- 4-KB cache for greater XIP performance with lower power
- Supports single, dual, quad, dual-quad, and octal interfaces

Serial Communication

- Nine run-time configurable serial communication blocks (SCBs)
 - Eight SCBs: configurable as SPI, I2C, or UARTs
 - One Deep Sleep SCB: configurable as SPI or I2C
- USB full-speed device interface

Audio Subsystem

- Two PDM channels and one I2S channel with TDM mode

Timing and Pulse-Width Modulation

- Thirty-two timer/counter pulse-width modulators (TCPWMs)
- Center-aligned, Edge, and Pseudo-random modes
- Comparator-based triggering of Kill signals

Programmable Analog

- 12-bit 1-Msps SAR ADC with differential and single-ended modes and 16-channel sequencer with result averaging
- One 12-bit voltage mode DAC
- Two low-power comparators available in Deep Sleep and Hibernate modes
- Two opamps with low-operation modes
- Built-in temp sensor connected to ADC

Up to 100 Programmable GPIOs

- Two Smart I/O ports (16 I/Os) enable Boolean operations on GPIO pins; available during system Deep Sleep
- Programmable drive modes, strengths, and slew rates
- Six overvoltage-tolerant (OVT) pins

LCD

- LCD segment direct block support up to 61 segments and up to 8 commons
- Operates in Active, Sleep, and Deep Sleep modes

Capacitive Sensing

- CapSense Sigma-Delta (CSD) provides best-in-class SNR, liquid tolerance, and proximity sensing
- Enables dynamic usage of both self and mutual sensing
- Automatic hardware tuning (SmartSense™)

Security Built into Platform Architecture

- ROM-based root of trust via uninterruptible Secure Boot
- Step-wise authentication of execution images
- Secure execution of code in execute-only mode for protected routines

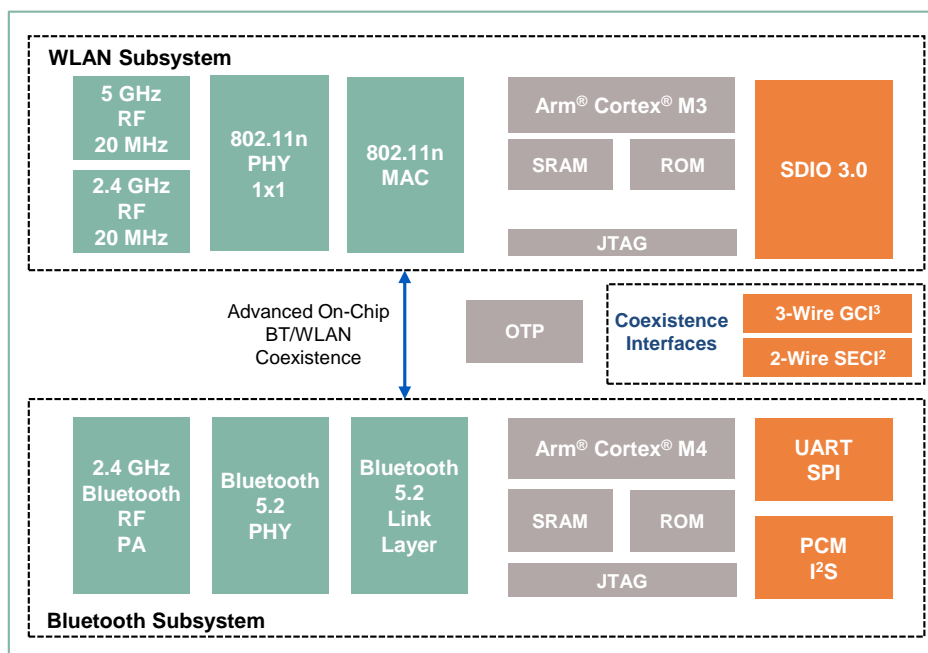




Note: Please refer to “PSoC 6 MCU: CY8C62x6, CY8C62x7 Architecture Technical Reference Manual (TRM)” document for all the details related to PSoC 62 series of MCUs ([link](#)).

3.2. AIROC™ CYW43012 WiFi/BT combo

The Cypress CYW43012 single-chip device integrates a IEEE 802.11a/b/g/n compliant 802.11ac-friendly MAC/baseband/radio and Bluetooth 5.0 + EDR (enhanced data rate). It provides a small form-factor solution with minimal external components to drive down cost for mass volumes and allows for handheld device flexibility in size, form, and function. Comprehensive power management circuitry and software ensure the system can meet the needs of highly mobile devices that require minimal power consumption and reliable operation.



IEEE 802.11x Key Features

- Full IEEE 802.11a/b/g/n compatibility with enhanced performance.
- 802.11ac friendly, MCS8 (256-QAM) for 20 MHz channels in 5 GHz band.
- Single spatial stream with PHY data rates of up to 72.2 Mbps with 802.11n (MCS7) and 78 Mbps with 802.11ac (MCS8).
- 20 MHz channels with optional SGI support for MCS0-MCS7.
- IEEE 802.11ac explicit beamformer support.
- TX and RX low-density parity check (LDPC) support for improved range and power efficiency.
- Receive space-time block coding (STBC)
- On-chip power amplifier/low-noise amplifier for both bands.
- Embedded IPV6 network stack for use with WICED SDK.
- MCS8 in 20 MHz channels (5 GHz band) proven to be interoperable with 802.11ac access points.
- Support for front-end modules (FEMS).
- Supports RF front-end architecture with a single dual-band antenna shared between Bluetooth and WLAN.
- Shared Bluetooth and WLAN receive signal path.
- Supports standard SDIO v2.0 and SDIO v3.0 (SDR40 at 80 MHz and DDR40 at 40 MHz).
- Backward compatible with SDIO v2.0 host interfaces.
- Integrated ARM processor with on-chip RAM and ROM minimizes the need to wake-up the applications processor for standard WLAN functions.



Bluetooth Key Features

- Complies with Bluetooth Core Specification Version 5.0 with
 - provisions for supporting future specifications.
 - QDID: 104548
 - Declaration ID: D035924
- Bluetooth 5.0 compliant with 2 Mbps GFSK data rate for BLE.
- All optional Bluetooth 4.2 features supported.
- Bluetooth Class 1 or Class 2 transmitter operation.
- Supports BDR (1Mbps), EDR (2/3Mbps), BLE (1/2Mbps).
- Host controller interface (HCI) using a high-speed UART interface.
- PCM for audio data.
- Ultra low TX o/p power mode to enable use cases like proximity pairing etc.
- Embedded Bluetooth host stack in ROM.
- Low power consumption improves battery life of handheld devices.
- Supports extended synchronous connections (eSCO), for enhanced voice quality by allowing for retransmission of dropped packets.
- Supports multiple simultaneous Advanced Audio Distribution Profiles (A2DP) for stereo sound.
- Adaptive frequency hopping (AFH) for reducing radio frequency interference.

General Features

- Supports battery voltage range from 3.2V to 4.6V supplies with internal switching regulator.
- Programmable dynamic power management.
- 6144 bits of OTP for storing board parameters.
- 40 GPIOs:
 - 16 WLAN
 - 4 Bluetooth (more available if I2S/PCM/JTAG are not used)
 - 20 shared
- Security:
 - WPA, WPA2 (Personal) with security improvements, WPA3 (Personal) support for powerful encryption and authentication
 - AES and TKIP in hardware for faster data encryption and IEEE 802.11i compatibility
- Worldwide regulatory support: Global products supported with worldwide homologated design.
- Packages:
 - 251-pin WLCSP package (3.76 mm x 4.43 mm, 0.2 mm pitch)
 - 300-ball FCBGA package (9mm x 9mm, 0.4mm pitch)
 - 106-ball WLBGA package (3.76mm x 4.43mm, 0.35mm pitch)



Note: Please refer to “CYW43012” product pages for all the details related to CYW43012 wireless combo ([link](#)).

3.3. XENSIV™ PAS CO2 sensor

The XENSIV™ PAS CO2 sensor is a real carbon dioxide (CO2) sensor in an unprecedented small form factor. Designed on the basis of the photoacoustic spectroscopy (PAS) concept, the sensor saves more than 75 percent space compared to existing commercial real CO2 sensors. Its direct ppm readings, tape & reel packing, SMD capability and simple design allow for a quicker and easier integration into customers' systems in low and high-volume applications alike.



Key features

- Small form factor (14 x 13.8 x 7.5 mm³)
- High accuracy (± 30 ppm $\pm 3\%$ of reading) and robust performance
- SMD package delivered in tape and reel
- Advanced compensation and self-calibration algorithms
- Three interface options: UART, I2C, PWM
- Various configuration options

Key applications

- HVAC (heating, ventilation and air conditioning) systems
- Smart home & building appliances like air purifiers and IoT devices
- Air quality monitor



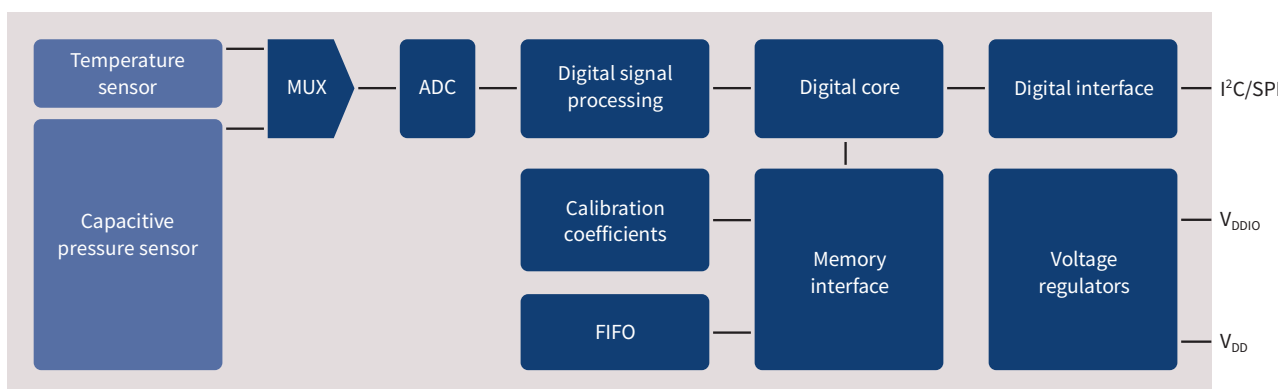
Note: Please refer to “PAS CO₂” product pages for all the details related to CYW43012 wireless combo ([link](#)).

3.4. Xensiv™ DPS310 – digital barometric pressure sensor

The DPS310 is a miniaturized Digital Barometric Air Pressure Sensor with a high accuracy level and low current consumption. The DPS310 is both a pressure and temperature sensor. The pressure sensor element is based on a capacitive principle which guarantees high precision during temperature changes. The small package makes the DPS310 ideal for mobile applications and wearable devices.

The DPS310's internal signal processor converts the output from the pressure and temperature sensor elements to 24-bit results. Each pressure sensor has been calibrated individually and contains calibration coefficients. The coefficients are used in the application to convert the measurement results to true pressure and temperature values.

The sensor has a FIFO that can store the latest 32 measurements. Since the host processor can remain in a sleep mode for a longer period between readouts, a FIFO can reduce the system power consumption.



Sensor measurements and calibration coefficients are available via the serial I2C/SPI interface.

Key features

- Operation range
 - Pressure: 300 ... 1200 hPa
 - Temperature: -40 ... 85°C
- Pressure level precision
 - ± 0.006 hPa (or ± 5 cm) (high-precision mode)
- Pressure sensor relative accuracy
 - ± 0.06 hPa (or ± 0.5 m)
- Temperature accuracy
 - $\pm 0.5^\circ\text{C}$
- Pressure temperature sensitivity
 - < 0.5 Pa/K
- Measurement time
 - Low-power mode: 3 ms



- Average current consumption
 - Low power: 3 μ A (1 measurement/sec.)
 - Standby: < 1 μ A
- Supply voltage
 - VDDIO: 1.2 ... 3.6 V
 - VDD: 1.7 ... 3.6 V
- Operating modes
 - Command (manual)
 - Background (automatic)
- Standby
- Interface
 - I2C and SPI (both with optional interrupt)
- Package dimensions
 - 8-pin LGA
 - 2.0 x 2.5 x 1.0



Note: Please refer to “DPS310” product pages for all the details related to the product ([link](#)).

3.5. OPTIGA™ Trust M

Secured cloud service provisioning – the easy way!

The OPTIGA™ Trust M – OPTIGA – is a high-end security solution that provides an anchor of trust for connecting IoT devices to the cloud, giving every IoT device its own unique identity. This pre-personalized turnkey solution offers secured, zero-touch onboarding and the high performance needed for quick cloud access. It offers a wide range of security features, making it ideal for industrial and building automation applications, smart homes and connected consumer devices. The turnkey set-up with full system integration minimizes design, integration and deployment effort.



Open Source Host Code for OPTIGA™ Trust M available NOW!

Get host code and documentation (github.com/Infineon/optiga-trust-m)

Customers benefit from a direct communication line to developers and will immediately and directly be informed of new versions, features and bug fixes. Be it the integration of standard open-source crypto software libraries or the integration of the host code into other systems – easily possible now. The host code is licensed under the MIT License.

Summary of Features

- High-end CC EAL6+ (high) certified security controller
 - ECC: NIST curves up to P-521, Brainpool r1 curve up to 512
 - RSA® up to 2048
 - AES key up to 256, HMAC up to SHA-512
 - TLS v1.2 PRF and HKDF up to SHA-512
 - TRNG/DRNG › I2C interface with shielded connection
- Hibernate mode for zero power consumption
- USON-10 package (3 x 3 mm)
- Standard and extended temperature ranges: -40 to + 105°C
- Up to 10 kB user memory
 - Protected updates
 - Usage counters



- Dynamic object (e.g. credentials) locking
- Configurable device security monitor
- Lifetime of 20 years for industrial and infrastructure applications
- Cryptographic ToolBox commands for SHA-256, ECC and RSA® Feature, AES, HMAC and Key derivation
- MIT licensed software framework on GitHub github.com/Infineon/optiga-trust-m
- OPTIGA™ Trust M's development process is certified according to the security standard IEC62443-4-1 for industrial automation and control systems, acting as an enabler to achieve component level certification according to IEC62443-4-2.

Features apply to latest product version.



Note: Please refer to “OPTIGA™ Trust M ” product pages for all the details related to product (*link*).



4. System setup using evaluation hardware

4.1. The PSoC Pioneer Kit

The PSoC™ 62S2 Wi-Fi BT Pioneer Kit (CY8CKIT-062S2-43012) is a low-cost hardware platform that enables design and debug of the PSoC™ 62 MCU and the Murata 1LV Module (CYW43012 Wi-Fi + Bluetooth Combo Chip).



Figure 5: The PSoC™ 62S2 Wi-Fi BT Pioneer Kit (CY8CKIT-062S2-43012)



Note: Please refer to “CY8CKIT-062S2-43012 Pioneer Kit” product pages for all the details related to the kit ([link](#)).

4.2. EBV IoT – Infineon PAS CO2 Evaluation Shield

The EBV-IoT – Infineon PAS CO2 Evaluation Shield (the shield) can be used in a standalone operation as well as through Arduino headers. Simple block diagram is depicted on Figure 8. Following please find brief overview of components of the shield.

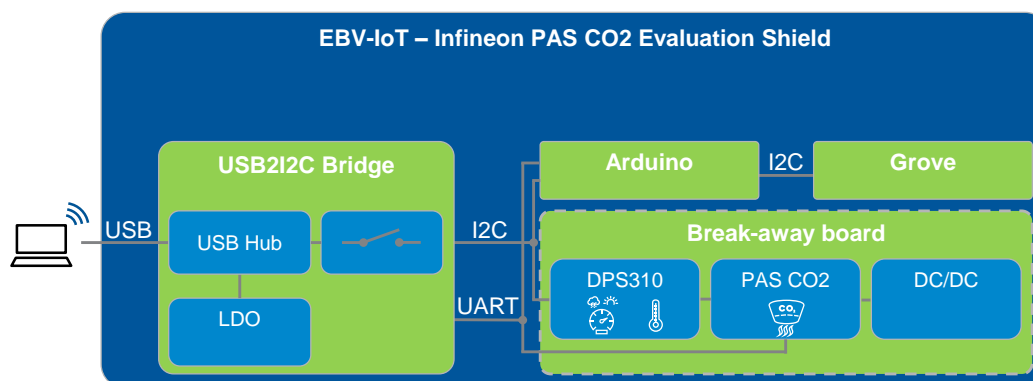


Figure 6: Block diagram

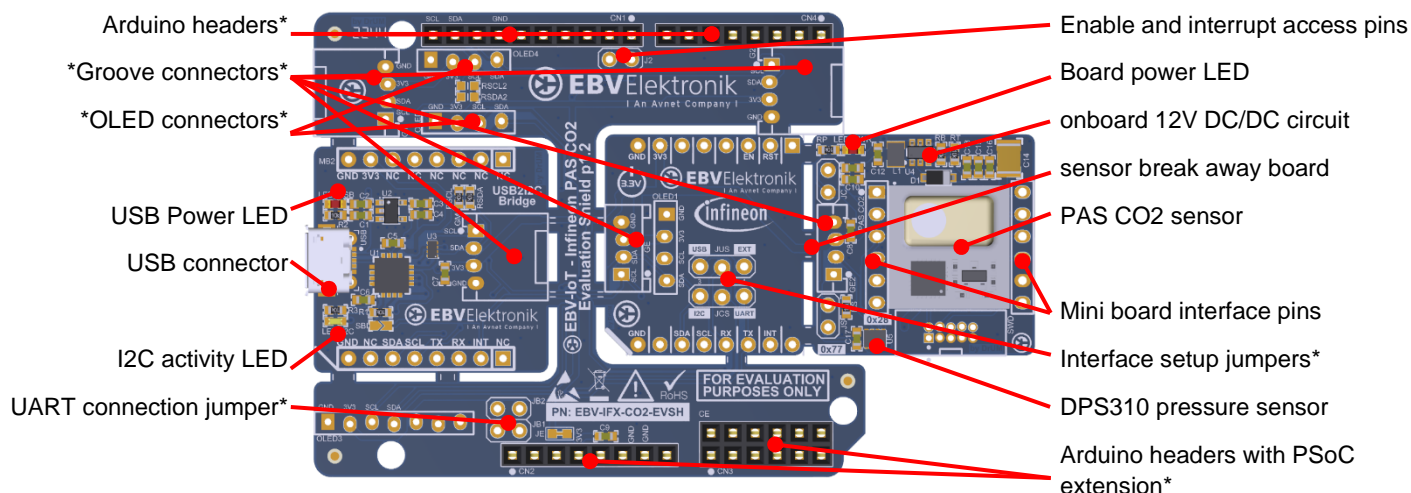
The core are **Infineon PAS CO2** and **DPS310** environmental and barometric pressure and sensors. Please refer to products pages for more information. The shield is Arduino compatible using **Arduino headers**. On the board there is **USB connector** used for powering the board and PC communication.

The onboard LEDs show basic status details. **USB power LED** indicates the board is powered up over USB. The **Board power LED** indicates sensor power on status. **I2C activity LED** indicates I2C communication status.

Multiple **jumper**s (JB1, JB2, JCS, JUS, JS) offer sensor interface mode selection as well as selection using either on-board USB bridge interface or external serial interfaces. Solder bridges (SB1, SB2, SB6, SB7) and “cut” bridges (JU, JE, JEN, JSCL, JSDA) can be used to “hard-wire” configuration using no



jumpers. Please refer to shield's quick start guide for more information. Additional **Grove** and **OLED connectors** placeholders allow using Grove I2C extension or OLED displays.



* optional configuration (not populated in default configuration)

Figure 7: The EBV-IoT – Infineon PAS CO2 Evaluation Shield overview

4.3. EBV-IoT – OPTIGA Trust M Evaluation Shield

The EBV-IoT – Infineon OPTIGA Trust M Evaluation Shield (the shield) can be used in a standalone operation as well as through Arduino headers. Simple block diagram is depicted on Figure 8. Following please find brief overview of components of the shield.

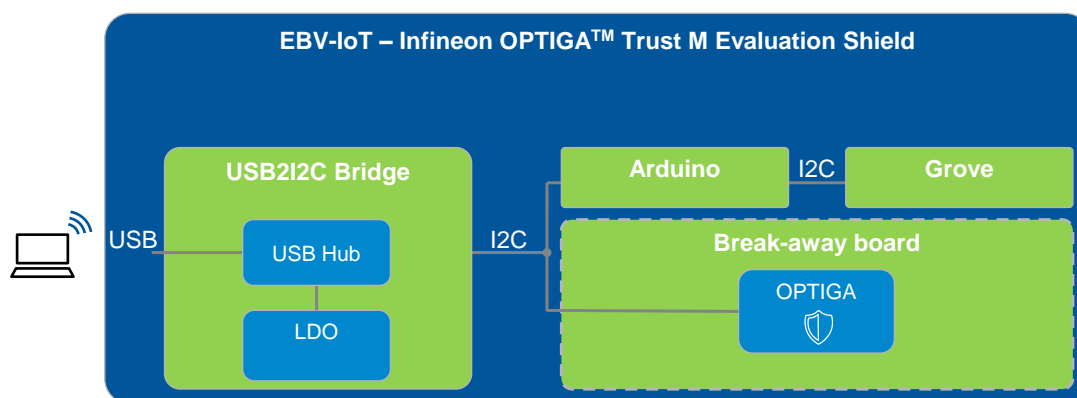


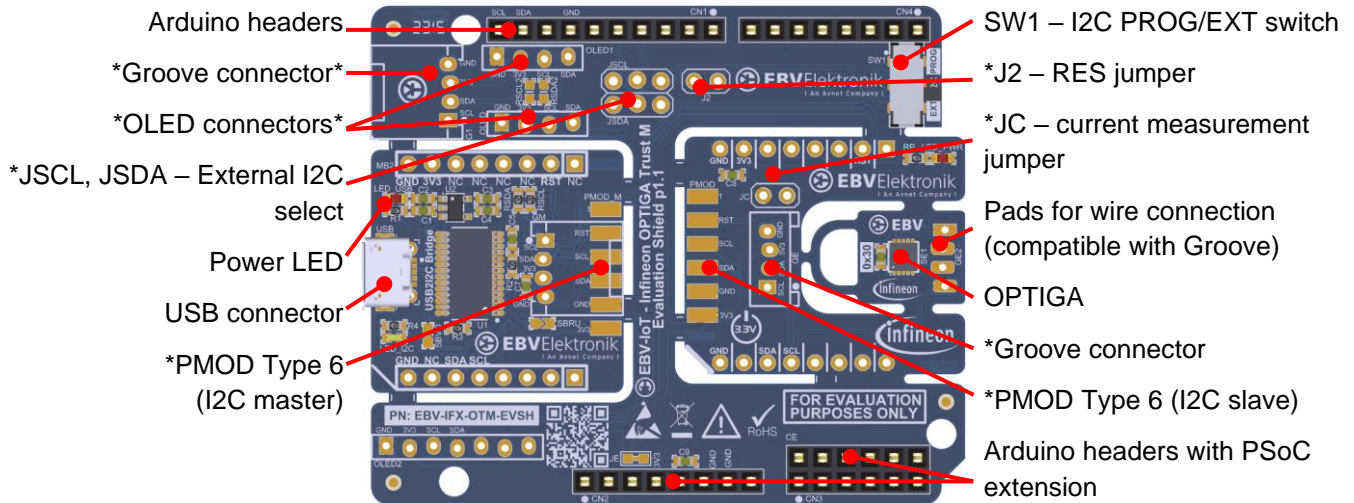
Figure 8: Block diagram

The core of the shield is **Infineon OPTIGA™ Trust M** secure element. Please refer to manufacturer product pages for more information. The shield is Arduino compatible using **Arduino headers**. On the board there is **USB connector** used for powering the board and PC communication.

The onboard LEDs show basic status details. **USB power LED** indicates the board is powered up over USB. The **Board power LED** indicates secure element power on status. **I2C activity LED** indicates I2C communication status.

Multiple **jumpers** (J2, JC, JSDA, JSCL) offer additional hardware configuration. Solder bridges (SBRU, SBVO, SB2, SB6, SB7) and “cut” bridges (JU, JE, JB, JBR, JBSCL, JBSDA) can be used to “hard-wire” configuration using no jumpers. Please refer to shield's quick start guide for more information. Additional **Grove** and **OLED connector** placeholders allow using Grove I2C extension or OLED displays.





* optional configuration

Figure 9: The EBV-IoT – Infineon OPTIGA Trust M Evaluation Shield overview



5. Hardware configuration

There are multiple hardware configurations possible for the shields used during the workshop.

5.1. PAS CO2 hardware setup



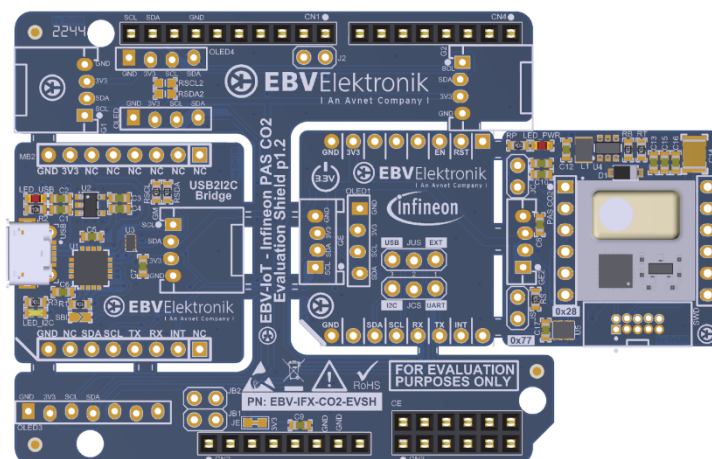
Note: Please refer to “EBV IoT – Infineon PAS CO2 Evaluation Shield - Quick start guide” document for all the details related to the shield

Following please find the configuration setups required for demo to operate.

5.1.1 Out-of-box setup

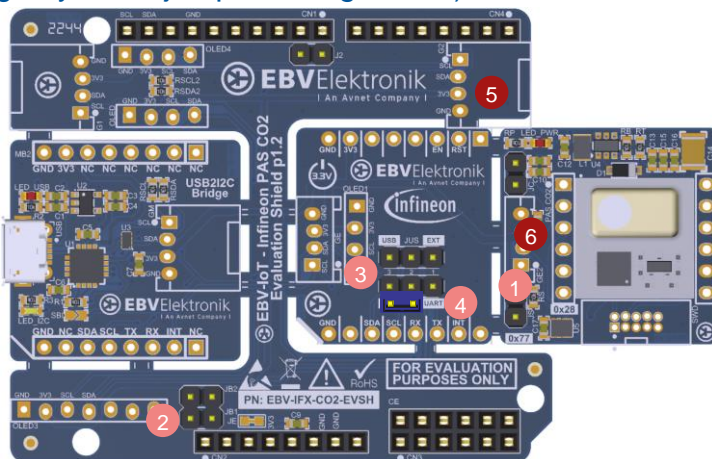
Shipped configuration is preset to support I2C over USB2I2C bridge or external I2C communication.

- Configured to be used in I2C mode
- I2C master can be either
 - integrated USB2I2C bridge or
 - external I2C master using external MCU/MPU platform
- Comes with no through-hole jumpers populated



5.1.2 I2C interface configuration (after using any other jumper configuration)

1. JS - open
2. JB1, JB2 – open (make sure SB1 and SB2 solder bridges located on bottom side are open as well)
3. JUS – no jumper
4. JCS – 2-3 closed (I2C)
5. Make sure SB3 and SB4 solder bridges located on bottom side are open
6. Make sure JSCL and JSDA lines located on bottom side are closed



5.2. OPTIGA shield hardware setup



Note: Please refer to “EBV IoT – Infineon OPTIGA Trust M Evaluation Shield - Quick start guide” document for all the details related to the shield.



5.2.1 Out-of-box setup

Shipped configuration is preset to support I2C over USB2I2C bridge or external I2C communication set by using SW1 switch. Default external I2C pins are set to pins 9 and 10 on CN1. Please make sure the SW1 switch is set to EXT position.

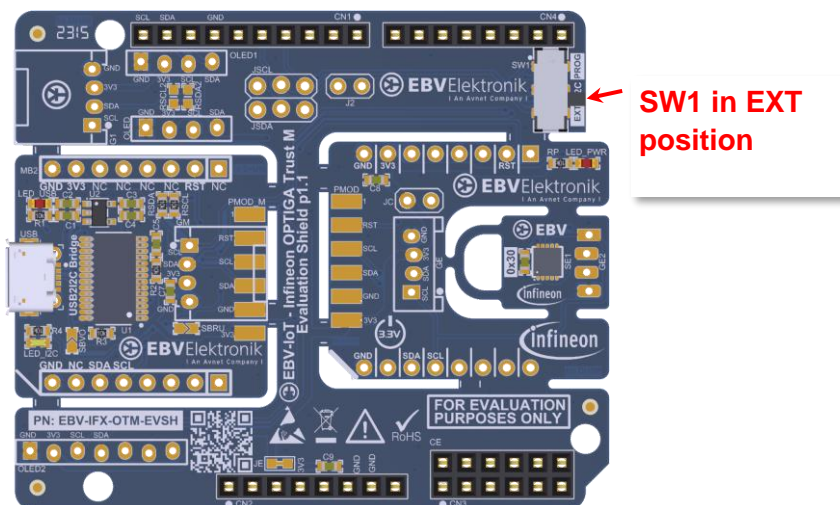


Figure 10: The secure element shield out-of-box configuration

5.2.2 I2C interface configuration (after using any other jumper configuration)

To use default I2C interface configuration on pins 9 and 10 on CN1 set jumpers as shown on Figure 11.

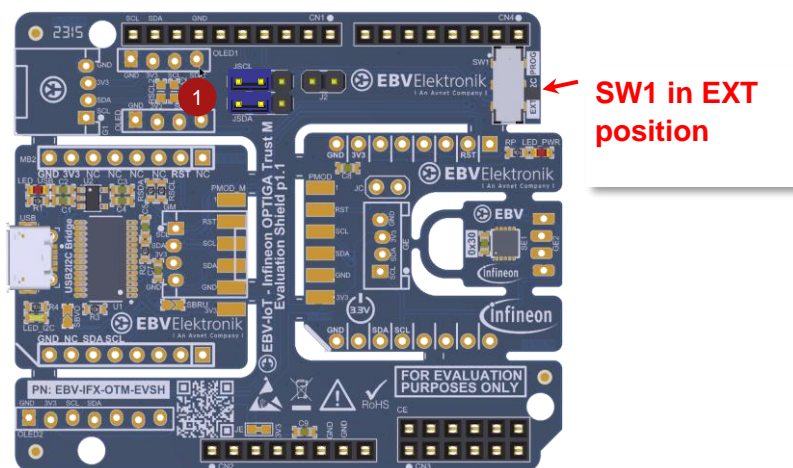


Figure 11: The secure element shield I2C pins configuration



6. EBV IoTConnect – Solution Accelerator Software

IoTConnect is a Solution Accelerator Software-as-a-Service that leverages existing cloud infrastructure and its native services such as storage and compute to create a unified IoT experience for solution designers to easily enable secure device management and develop solutions 50%+ faster than traditional IoT platforms.

IoTConnect is designed to help Original Equipment Manufacturers (OEMs) develop connected solutions by simplifying the process and decreasing time to market by more than 50%. OEMs can utilize IoTConnect to overcome market pressures with a reliable, secure and scalable solution that will simplify overall system design and lower the operations and maintenance costs.

IoTConnect comes with an easy-to-use interface, powerful device and entity management systems, as well as built-in analytics allowing OEMs to connect and manage devices with little to no coding.

IoTConnect provides businesses with the infrastructure required to connect their assets, collect data and analyze it to improve decision-making. This is achieved by facilitating device communication and management while adhering to industry-grade security protocols.

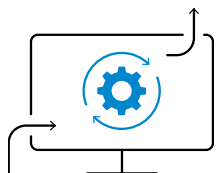
/IOTCONNECT®

Features

IoTConnect is created to bring together all the key elements of an IoT platform to allow OEMs to seamlessly connect devices to the cloud and quickly gain insights from the collected data to make better business decisions. IoTConnect deploys applications that are customized for various businesses and provides benefits such as improved ROI, reduced downtime and a better user experience.

Easy configuration and management

- Device registration without any hassle or programming knowledge
- Multi-tenant architecture with user and roles management
- Device management
- Asset management
- Software management
- Remote troubleshooting
- Over-the-air (OTA) firmware updates



Easy integration and connections

- Cross-device communication support
Supports different industry-grade protocols
- Web services and APIs to simplify exchange of information
- Microservices-based architecture
- Service SDK



Multi-layer security

- Enterprise-grade, end-to-end security
- Device identity management
- Access Security using Device Identity Management, OAuth 2.0
- Protocol transmission over TLS
- Role-based access control



Edge and predictive analytics

- Edge analytics to make quick decisions
- Near real-time monitoring with live graphical charts
- Extensive real-time streaming analytics
- Machine learning and statistical algorithms
- Connectors for data visualization applications



Real-time alerts and notifications

- Create smart rules for autonomous actions
- Fully customizable workflows and notifications
- Detect anomalies and get alerts about potential problems



Note: Evaluating the Avnet IoTConnect cloud is completely free of charge for first three months with no restrictions on the platform features. No credit card is required to perform the registration process.



7. Task 1: PSoC programming and debugging – Hello world



Quick start: Please proceed to step 9 Task 3: OPTIGA™ provisioning for IoTConnect Cloud



Important: ModusToolbox™ has to be installed prior running the task. Please refer to sub-section **2.2.1 Installing ModusToolbox**.

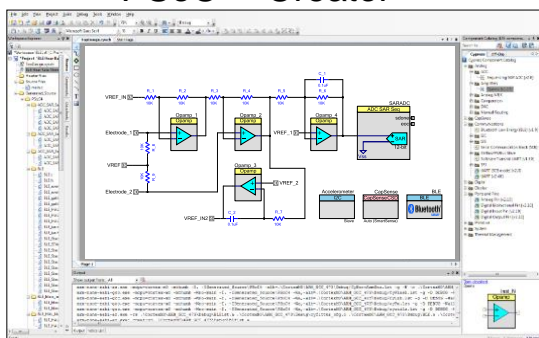
7.1. Overview

In this task we will create a simple “Hello world” project to get familiar with the toolchain.

7.2. ModusToolbox Overview

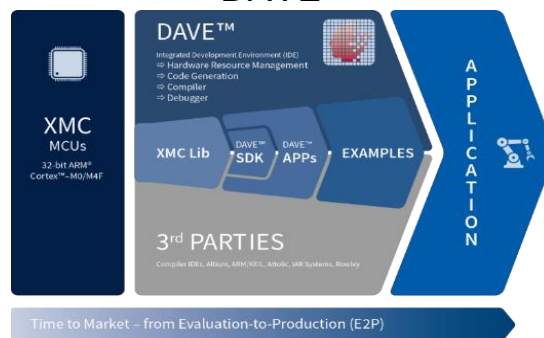
ModusToolbox™ Software is a modern, extensible development environment supporting a wide range of Infineon microcontroller devices, including **PSoC™ Arm® Cortex® Microcontrollers**, **TRAVEO™ T2G Arm® Cortex® Microcontroller**, **XMC™ Industrial Microcontrollers**, **AIROC™ Wi-Fi devices**, **AIROC™ Bluetooth® devices**, and **USB-C Power Delivery Microcontrollers**.

PSoC™ Creator



- › Software IDE for PSoC™ 3, PSoC™ 4, PSoC™ 5
- › Schematic-based capture tool enables custom analog front end and programmable digital development
- › Component-based design with graphical configuration tools

DAVE



Free Eclipse-based code development platform/IDE offering code repository, graphical system design methods, and automatic code generator

Guides XMC™ microcontroller user along the entire process – from evaluation to production (E2P).

XMC™ Lib DAVE™ generated code is tested and released for and use with 3rd party tool.

ModusToolbox™

7.3. Step by step guide



To Do: The task does not involve any software development. Instead, it is only required to process the steps in the right order and to understand their underlying effects.

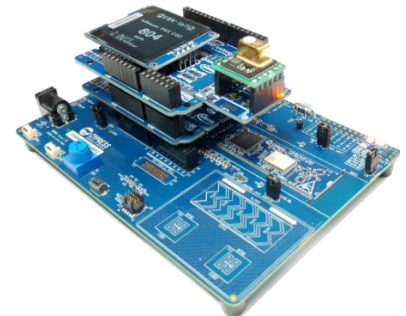


7.3.1 HW setup

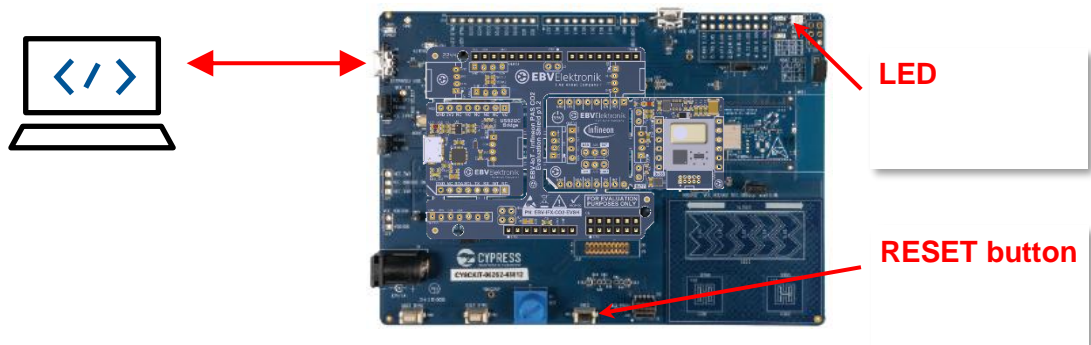
In this task we will actually use only PSoC Pioneer Kit, but we recommend using complete stack of boards as will be required in later tasks. Please check section **5 Hardware configuration** for details.

For convenience, stack the Arduino compatible shields in the following order from bottom to top:

- The PSoC™ 62S2 Wi-Fi BT Pioneer Kit (CY8CKIT-062S2-43012) – at the bottom
- EBV-IoT – Infineon PAS CO2 Evaluation Shield
- EBV-IoT – Infineon OPTIGA Trust M Evaluation Shield
- 128x128 pixels OLED display (optional)

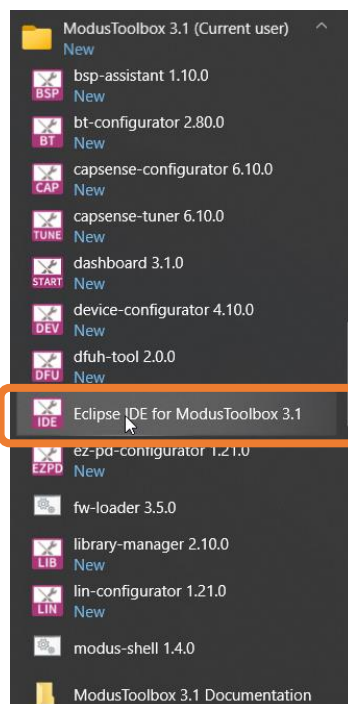


Then only plug your laptop to the PSoC Kit USB programming interface as shown below:



7.3.2 Launch Eclipse IDE for ModusToolbox

To launch the Eclipse IDE start **Eclipse IDE for ModusToolbox 3.1** from Windows start menu under “**ModusToolbox 3.1 (Current user)**”

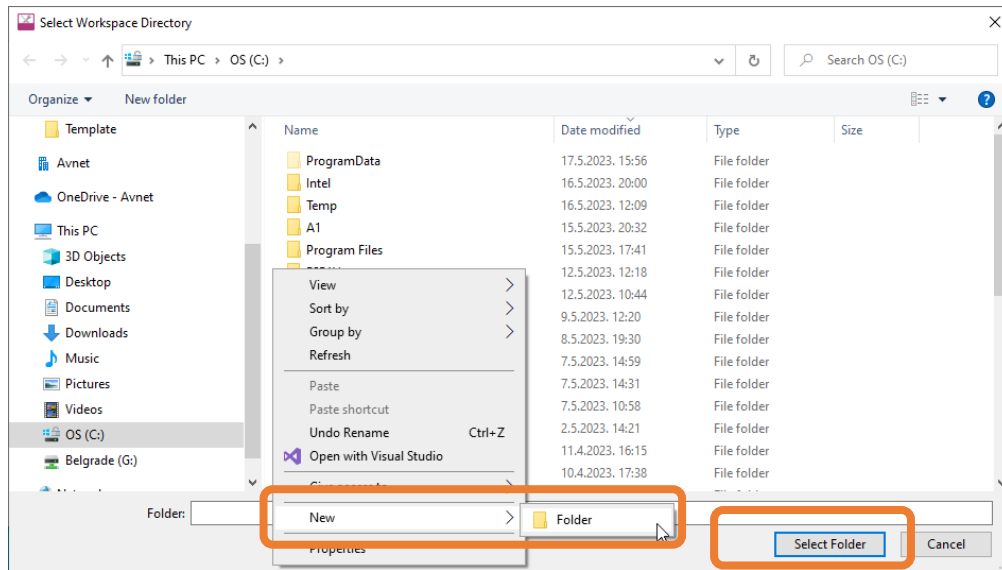


Important: It is **very important** to keep your **Workspace** and **Project** names as **short** as possible. Otherwise, the complete path length may become too long and compilation of some later project may not succeed.

At “**Select a directory as workspace**” prompt you can either (select one option from the list bellow)

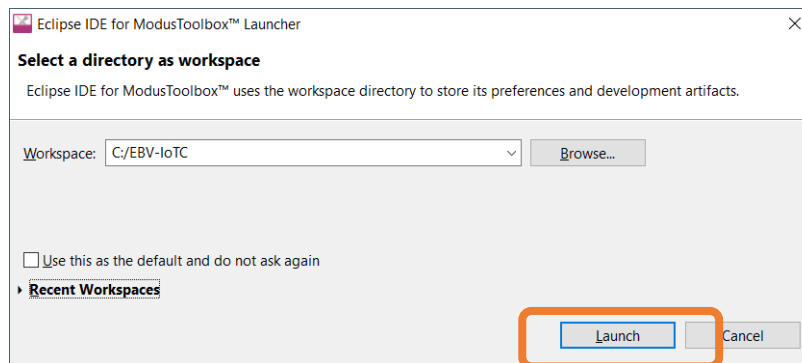


- A. browse for your existing workspace directory clicking **Browse**,
- B. create new directory by clicking **Browse** and then inside the File explorer right click and go for **New→Folder**, name the folder, select newly created folder, and click **Select Folder**,



- C. Type in **any valid path** for your workspace and the tool will create it at **Launch** if missing,
- D. or search among recent workspaces listed when clicking **Recent workspaces**.

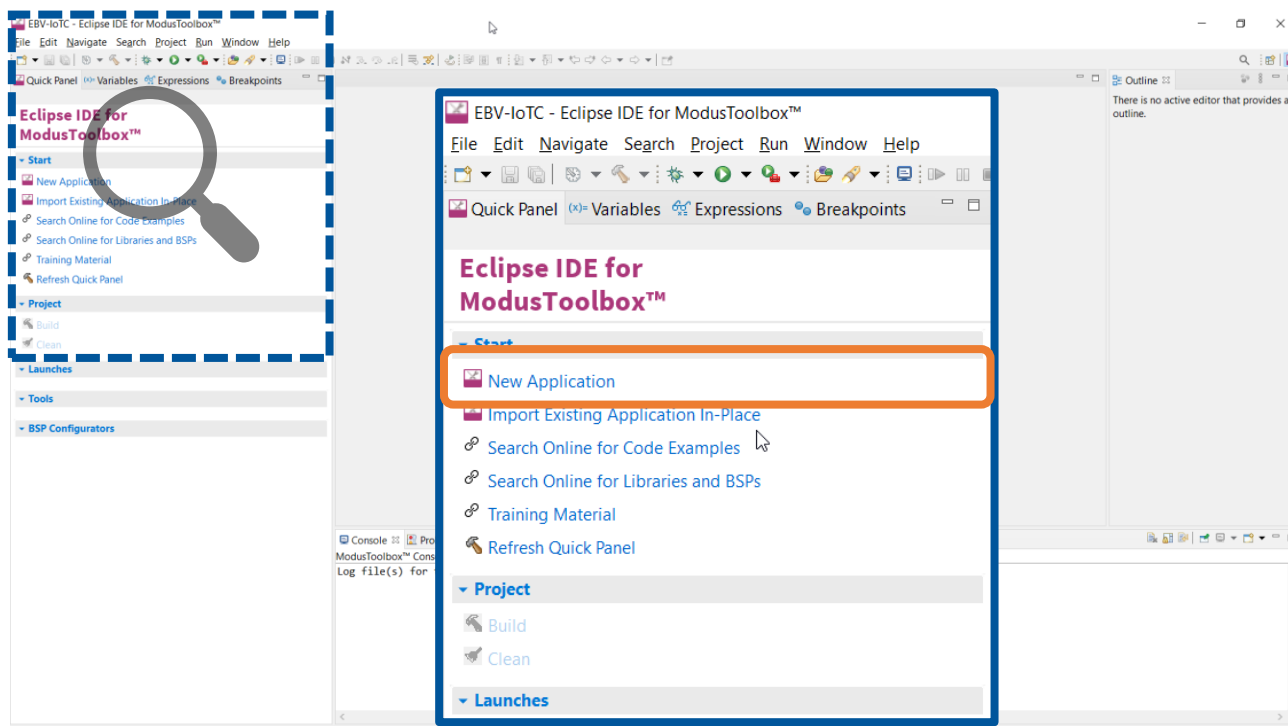
Select Workspace Directory and click **Launch**



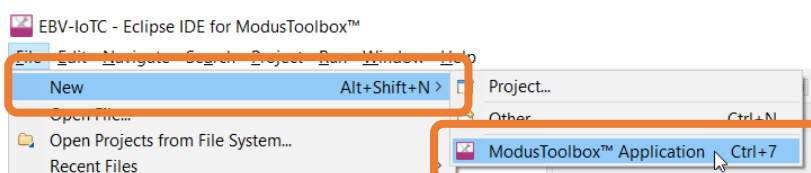
7.3.3 Create a project

When creating a new workspace or if no project were previously added to the workspace, you will see an empty workspace as shown below. Click the **New Application** link in the Eclipse IDE “Quick Panel”.

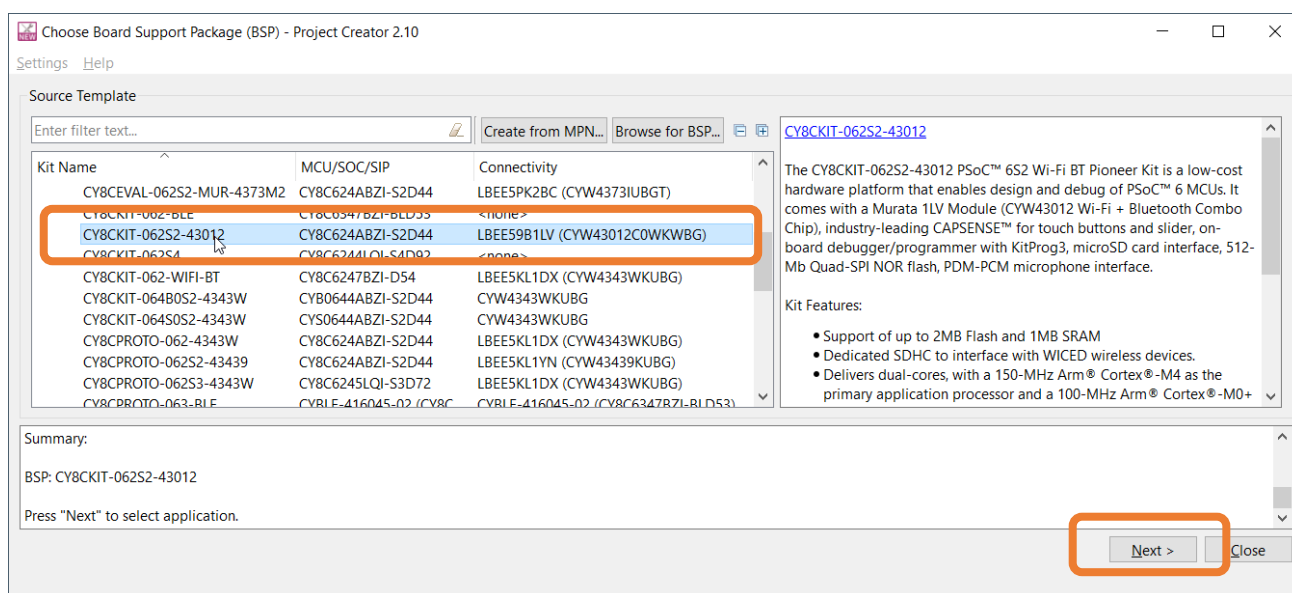




Alternatively, you can use **File → New → ModusToolbox™** Application or press “CTRL+7”.



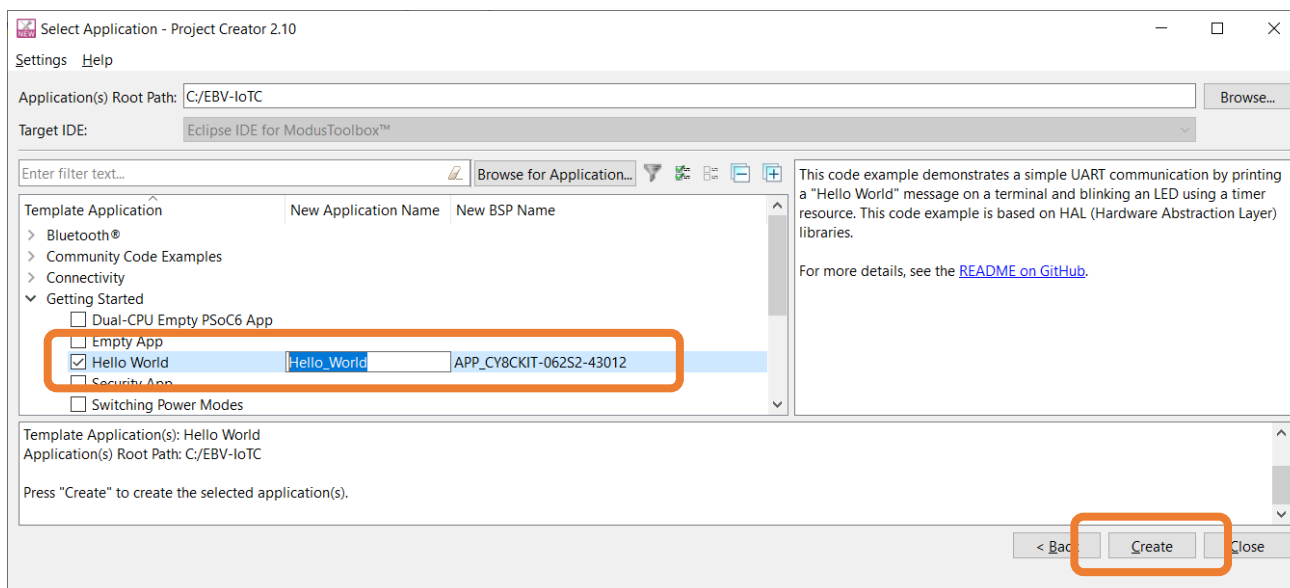
These commands launch the “**Project Creator**” tool with Board Support Package (BSP) selection, which displays a list of boards, showing the Kit Name, MCU, and Connectivity Device (if applicable). As you select each of the kits shown, the description for that kit displays on the right. Depending on the settings for your system, you may see different categories, including PSoC™ 4, PSoC™ 6, and AIROC™ Bluetooth® BSPs. For this example, select the **CY8CKIT-062S2-43012** kit and click **Next**.



Next, “**Select application**” window is opened listing various applications available for the selected kit. As you select an application, a description displays on the right. You can select multiple applications for the selected BSP by enabling the check box next to the applicable applications. In the Project Creator - Select Application dialog, choose the example by enabling the checkbox. Go to “Getting Started” and



select check box next to the **"Hello World"** application. Optionally, you can change the suggested New Application Name. Click Create to complete the application creation process.



When you select a supported kit, the example is reconfigured automatically to work with the kit. To work with a different supported kit later, use the Library Manager to choose the BSP for the supported kit. You can use the Library Manager to select or update the BSP and firmware libraries used in this application. To access the Library Manager, click the link from the Quick Panel.

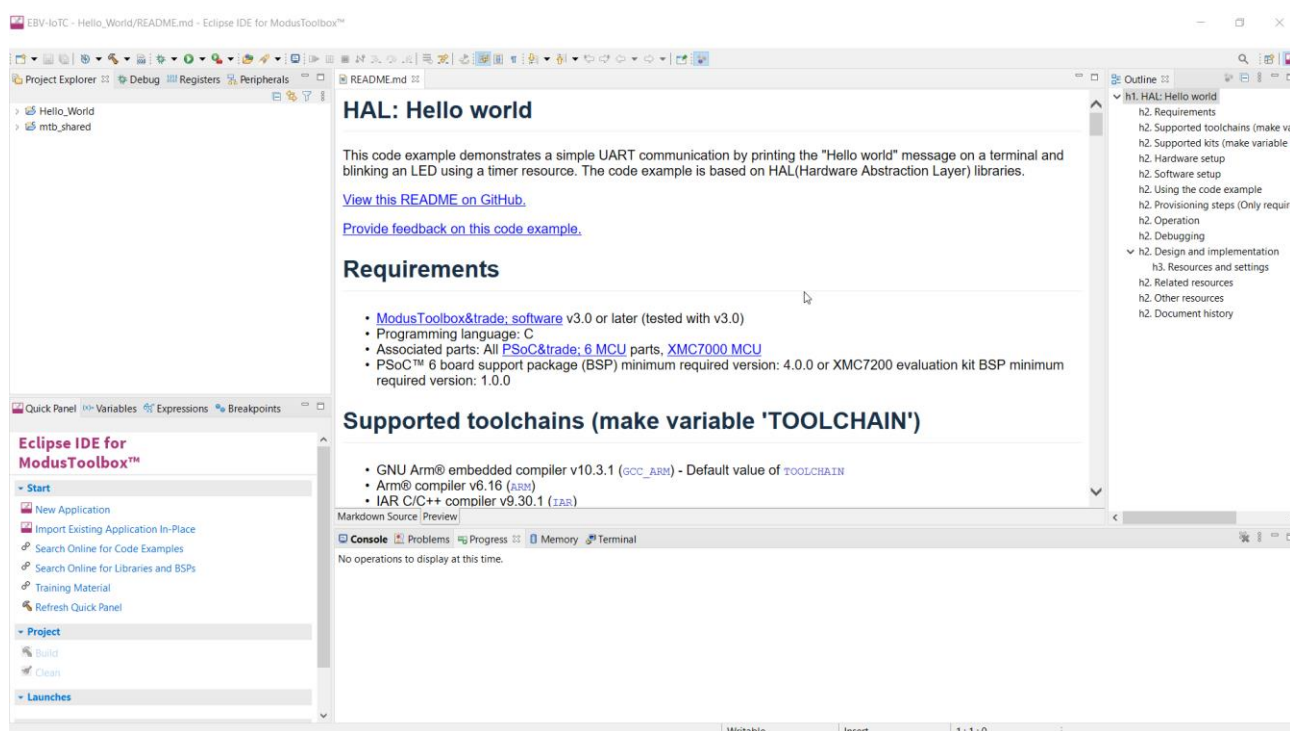
You can also just start the application creation process again and select a different kit.

If you want to use the application for a kit not listed here, you may need to update the source files. If the kit does not have the required resources, the application may not work.



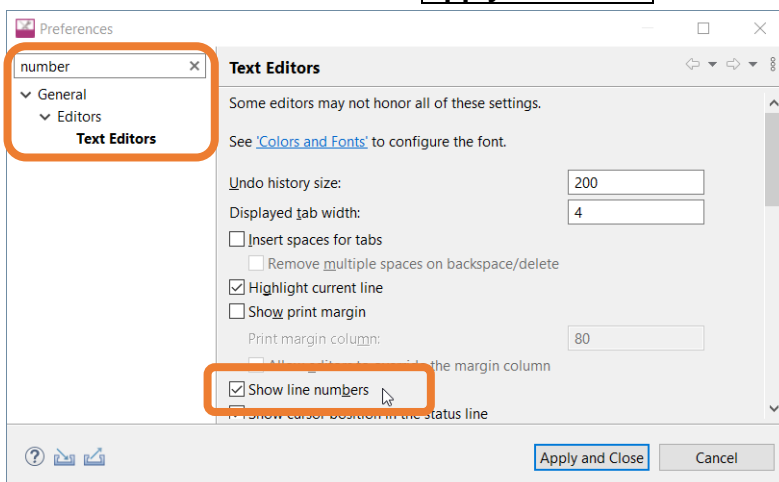
Note: For more details about using this tool, refer to the **ModusToolbox™ Project Creator user guide**.

After several moments, the application opens with the *Hello_World* in the **"Project Explorer"**, and the *README.md* file opens in the file viewer.





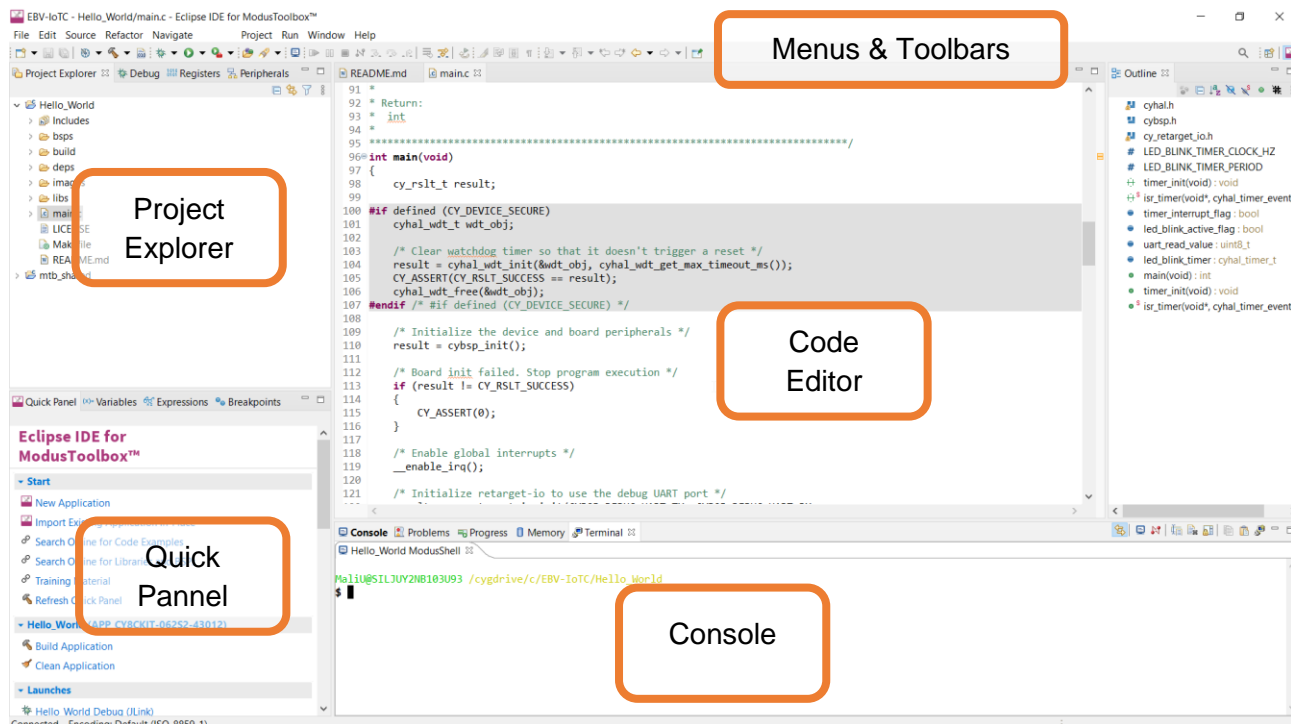
Tip: It is convenient to have line numbers shown in the IDE. Please go for **Window→Preferences** now search for “number” or go for **General→Editors→Text Editors** and **check** “Show line numbers” and click **Apply and Close**.



7.4. IDE overview

The IDE is based on the Eclipse IDE. It uses several plugins, including the Eclipse C/C++ Development Tools (CDT) plugin. For more information about Eclipse, refer to the **Eclipse Workbench User Guide**. We also provide a document called the **Eclipse IDE survival guide**, which provides tips and hints for how to use the Eclipse IDE.

The IDE contains Eclipse standard menus and toolbars, plus various panes such as the Project Explorer, Code Editor, and Console. One difference from the standard Eclipse IDE is the "ModusToolbox™ Perspective". This perspective provides the "Quick Panel", and adds tabs to the **Project Explorer**. "Perspective" is an Eclipse term for the initial set and layout of views in the IDE.



Tip: If you switch to a different perspective, you can restore the ModusToolbox™ Perspective by clicking the ModusToolbox™ icon button in the upper-right corner. You can also select **Perspective → Open Perspective** from the **Window** menu. To restore the ModusToolbox™



Perspective to the original layout, select **Perspective → Reset Perspective** from the **Window** menu.

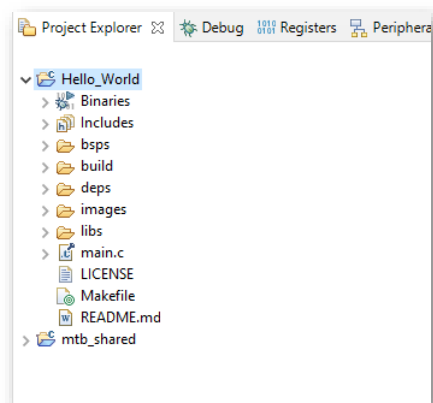
The following describe different parts of the IDE:

- **Menus and toolbars** – Use the various menus and toolbars to access build/program/debug commands for your application. Many of these are covered in the *Eclipse Workbench User Guide*.
- **Project Explorer** – Use the Project Explorer to find and open files in your application. See *Project Explorer* for more information.
- **Quick Panel** – Use this tab to access appropriate commands, based on what you select in the Project Explorer. More in *Quick Panel* sub-section.
- **Code Editor** – Use the Code Editor to edit various source files in your application.
- **Console** – Use these tools to review messages and access the integrated terminal.

7.4.1 Project Explorer

In the Eclipse IDE, after creating an application, the **Project Explorer** contains one or more related project folders. The following images show our “Hello_world” PSoC™ MCU application.

Both PSoC™ MCU application and an AIROC™ Bluetooth® application types of applications contain a similar project structure. Each contains the main application source code, and a Makefile. Note that PSoC™ MCU applications contain a libs directory, while AIROC™ Bluetooth® applications have a wiced_btstack project with shared SDK, BSPs, and libraries for all applications in a workspace.

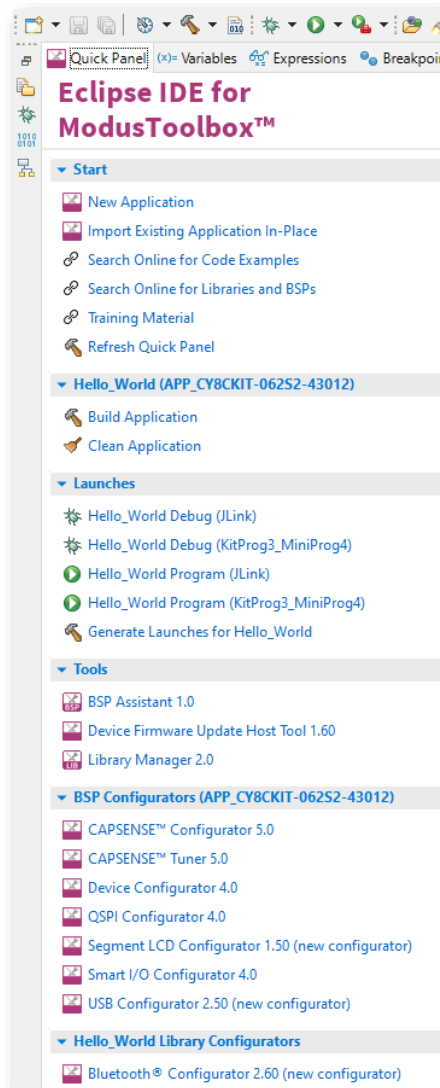


7.4.2 Quick Panel

As stated previously, the “**Quick Panel**” is part of the “ModusToolbox™ Perspective”. It provides quick access to commands and documentation based on what you have selected in the “**Project Explorer**”.

The Quick Panel contains links to various commands and documentation, organized as follows:

- **Start** – This contains the New Application link to create new applications, and links to find Code Examples, Libraries, BSPs, and training material.
- **Selected <app-name> project** – This contains different project-related links based on the project that is selected in the Project Explorer, as well as the type of application. Links here include: Build and Clean the application.
- **Launches** – This contains various Launch Configurations, based on the selected application project and device, which can be used to program the device and launch the debugger. This area is only populated if you have the top project in your application selected (<app-name>).
- **Tools** – This contains links to the various tools available for the selected project. For more information.
- **Documentation** – This may contain several documents in HTML format, which are included as part of the chosen BSP.

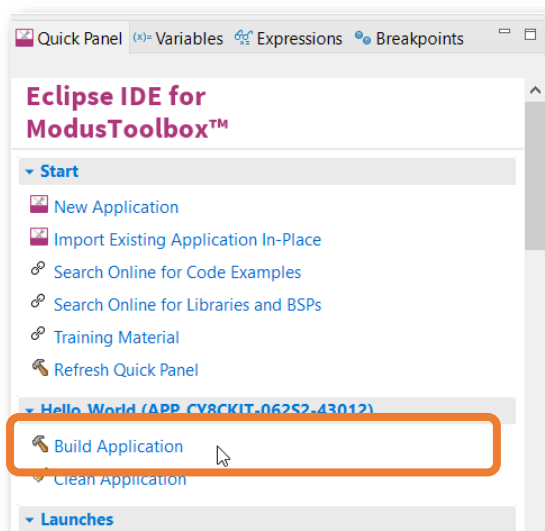


7.4.3 Build application

After loading an application, build it to generate the necessary files. Select a project. Then, in the **Quick Panel**, click the **Build Application** link. The following images show the **Quick Panel** for a typical PSoC™ MCU application.

It may take some time to compile, but it only takes so much time for the first compile or after “Clean Application” execution. In the **Console** you can follow messages flow until receiving something like:

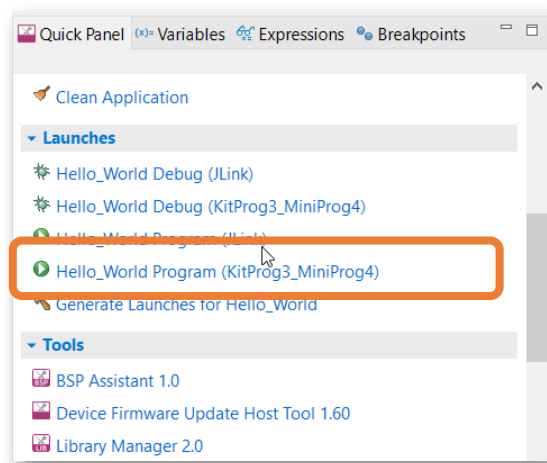
14:08:31 Build Finished. 0 errors, 0 warnings.
(took 53s.74ms)



7.4.4 Program application

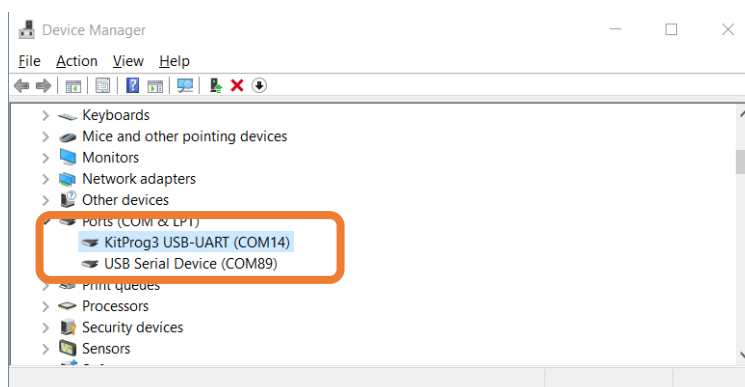
In the **Project Explorer**, select the desired project. Then, in the **Quick Panel**, click the **Hello_World Program (KitProg3_MiniProg4)** link for a PSoC™ MCU application.

In the **Console** you should be seeing some messages including “Erasing” and “Programming” progress bar and the orange LED on the PSoC Pioneer Kit blinking.



7.4.5 Accessing the device through serial terminal

The “Hello_world” application also sends some data to serial communication interface we can read using any serial communication terminal. Please use device manager to get your device serial port number information like the one depicted below:



Use your favorite serial communication terminal for serial communication monitoring. We use TeraTerm or you can use ModusToolbox integrated serial communication terminal. Use following settings:

- Port number: as retrieved from “Device manager”
- Speed: 115200
- Data: 8 bit
- Parity: none
- Stop bits: 1
- Flow control: none



For the “Hello_World” example you should be seeing messages as follows (reset board if needed):

```
Terminal – COM14
File Edit Setup Control Window Help
***** HAL: Hello World! Example *****

Hello World!!!

For more projects, visit our code examples repositories:

https://github.com/Infineon/Code-Examples-for-ModusToolbox-Software

Press 'Enter' key to pause or resume blinking the user LED

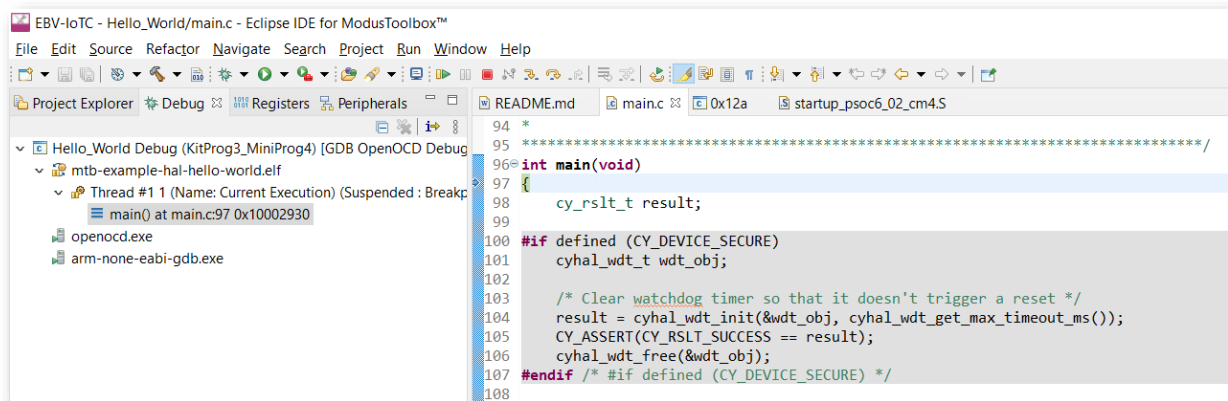
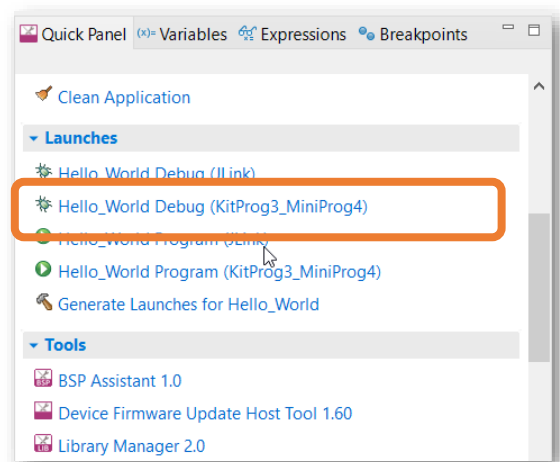
LED blinking paused
```


7.4.6 IDE integrated serial communication terminal

7.4.7 Debug application

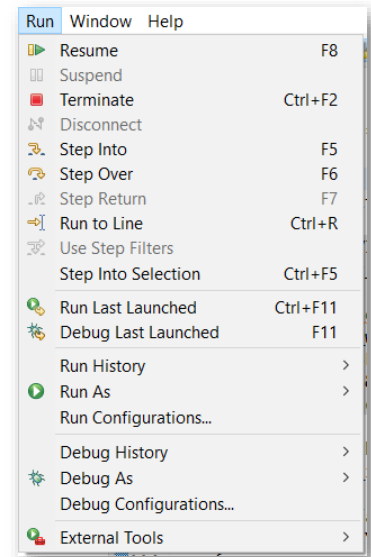
One of the most advanced features of the toolchain is definitely step-by-step debugging. In the **Project Explorer**, select the desired project. Then, in the **Quick Panel**, click the **Hello_World Debug (KitProg3_MiniProg4)** link for a PSoC™ MCU application

The process is very similar to “Build” process: in the **Console** you should be seeing some messages including “Erasing” and “Programming” progress bar, but now your application execution will halt at “main” function entrance.




Now you could do step by step code execution or eventually run it as in the “Program” example. Please use icons  in the “Toolbar” menu or check “Run” for all the options. Eventually you can use following shortcuts:

- F8 – Run or Resume
- F5 – Step Into
- F6 – Step Over
- CTRL + F2 – Terminate
- CTRL + R – Run to Line



Tip: The shortcuts above are very handy, especially if you use debugging for longer period of time.



Note: You may noticed the perspective changed to “Debug” perspective which is more convenient for application debugging. You can change perspective using **Window→Perspective→Open Perspective→Other** and use any other perspective or switch to **ModusToolbox (default)** perspective. You can also select perspective from the **Toolbar** at the far right hand side using icon .



Task accomplished: Congratulations, you have just run your first ModusToolbox application and learn step-by-step debugging.



8. Task 2: PSoC programming and debugging – Adding sensors

8.1. Overview

In the task we will extend the “Hello_World” application with reading Infineon Xensiv sensors through I2C buss and display the values in the terminal. We will copy existing project, include additional libraries for PAS CO2 and DPS310 sensors, and change the “main.c” file content.

8.2. Step by step guide



To Do: The task does not involve any software development. Instead, it is only required to process the steps in the right order and to understand their underlying effects.

8.2.1 HW setup

Hardware setup shall remain same as in the previous task as shown in sub-section **7.3.1 HW setup**.

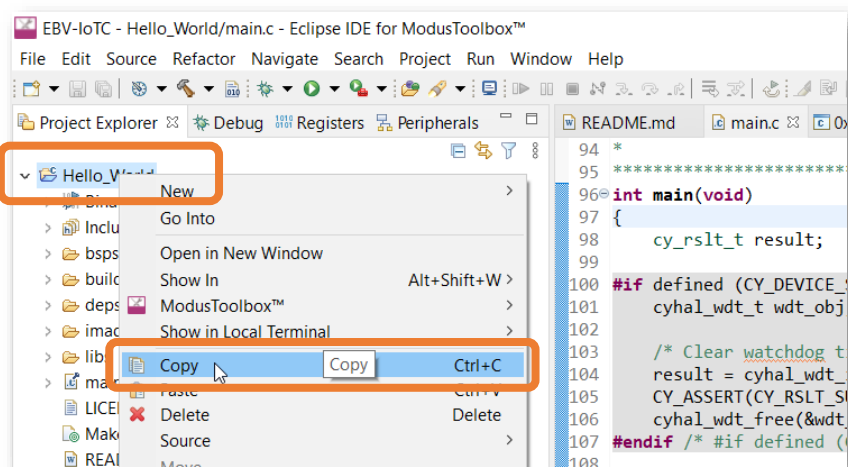
8.2.2 Create a project

Assuming your Eclipse IDE for ModusToolbox is still open from the previous task.

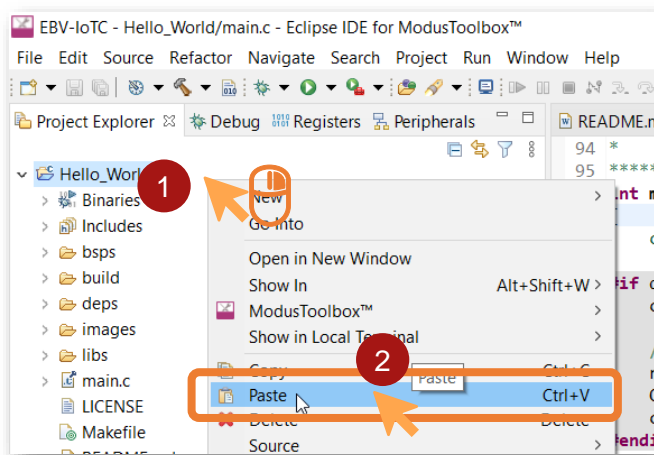


Important: **Terminate** previous **debug** session (if still running) and switch perspective to **ModuToolbox (default)** perspective.

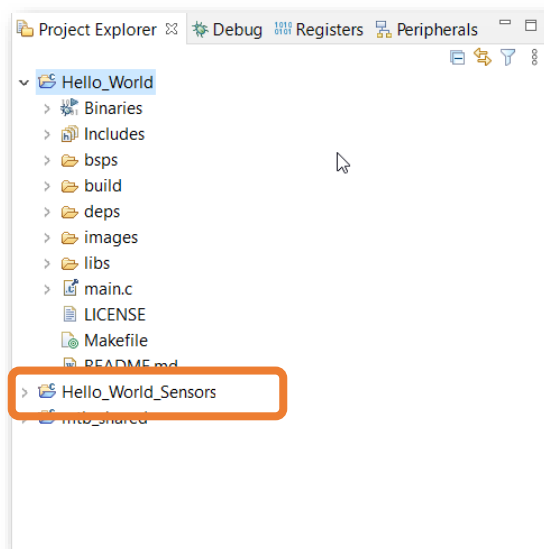
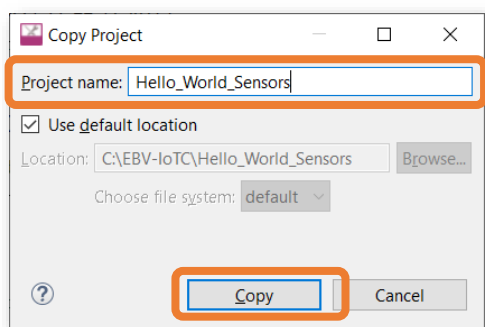
In the **Project Explorer** **right click** on “Hello_World” project and click **Copy**.



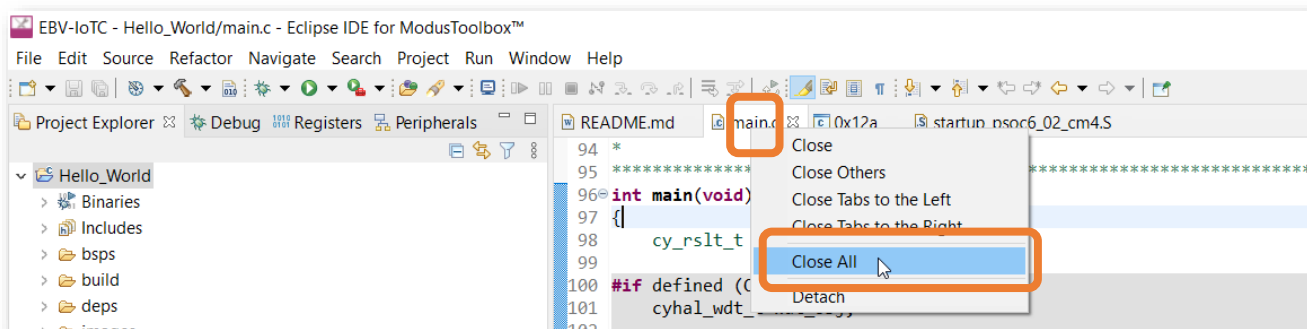
Now **right click** anywhere in empty space inside **Project Explorer** and click **Paste**.



You will be prompted for new **Project name**. We will use “Hello_World_Sensors” but can be any name you like until it is kept short. Then click **Copy**. In the **Project Explorer** you shall now see “Hello_World_Sensors” project.



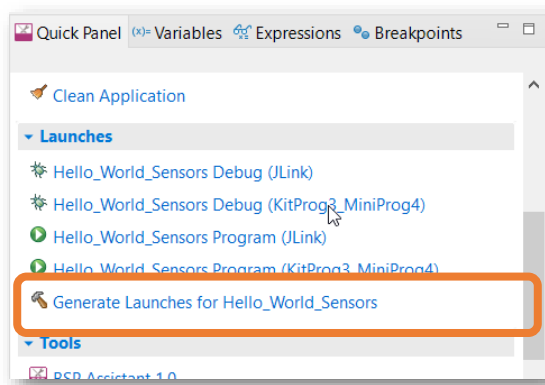
It is strongly recommended to close all the files on **Code Editor**. **Right click** on any of the file names in Code Editor and click **Close All**.



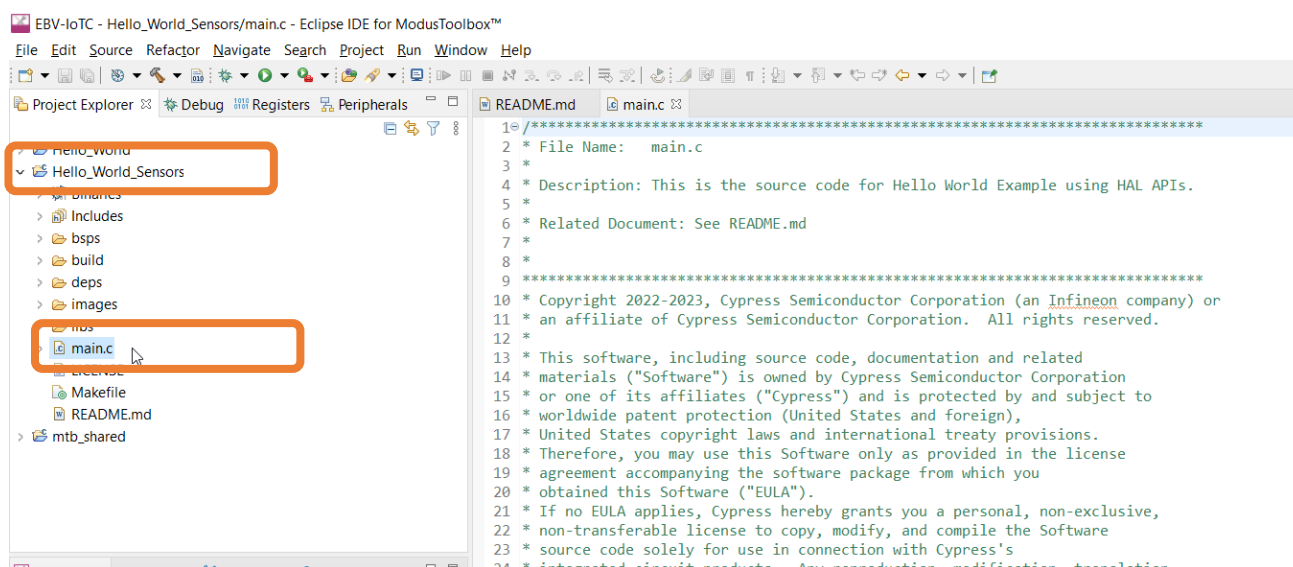
In the **Project Explorer** select “Hello_World_Sensors” project, go to **Quick Panel** to look for **Generate Launches for Hello_World_Sensors** and click to create launches for the new project. You should now



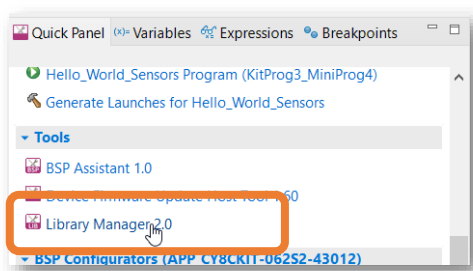
see updated launches in the **Quick Panel** as well as “HAL: Hello world”. “README.md” file is being opened in the **Code Editor**.



Now open the “Hello_World_Sensors” project and double click “main.c”.

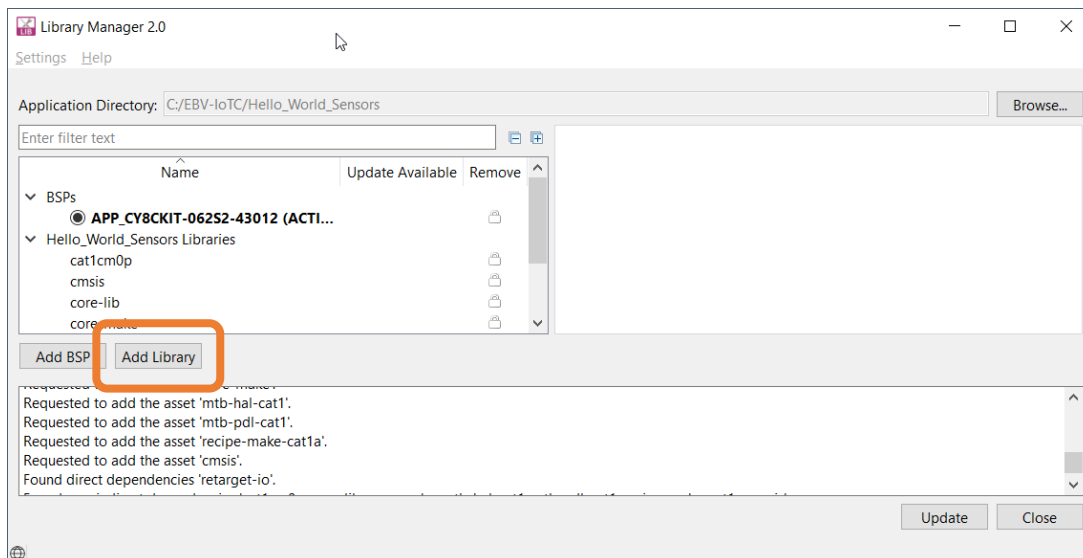


Next please click on **Library Manager 2.0** located in **Quick Panel**:

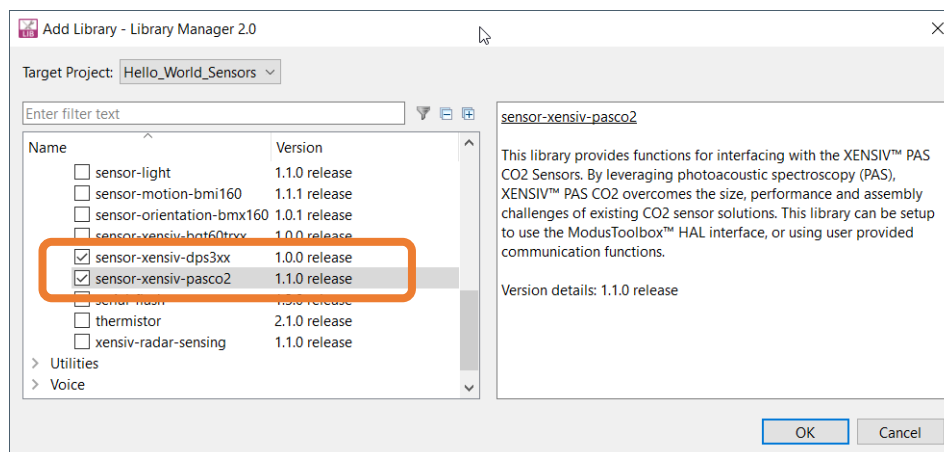


When opened click **Add Library**

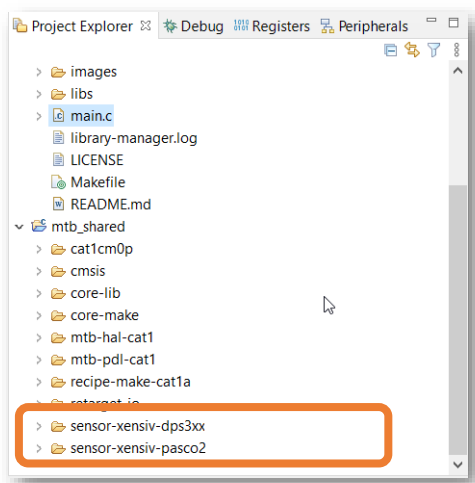




In the **Add Library – Library Manager 2.0** please look for “**sensor-xensiv-dps3xx**” and “**sensor-xensiv-pasco2**” libraries and check the checkbox in front to get like following and click **OK**:



At **Library Manager 2.0** window please click **Update** and wait to retrieve all the needed libraries. It may take some time... When **Close** gets enabled click **Close**. You should notice added libraries in the “mtb_shared” folder in **Project Explorer**.

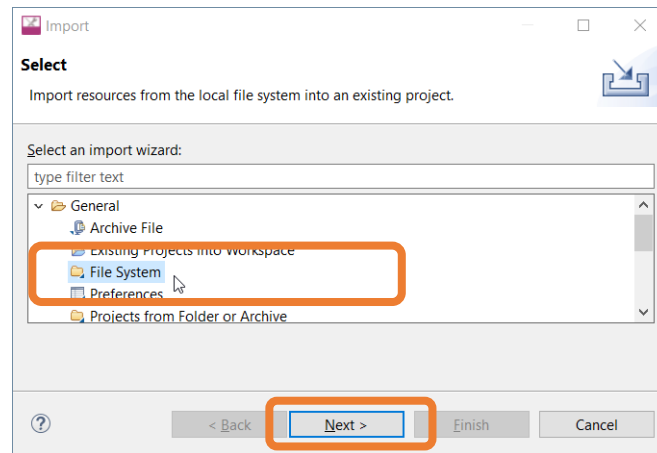


Since the “main.c” did not change at all and to add sensor readings, we have to add sensor reading functionality to the code. To keep it easy we provided the updated “main.c” file in

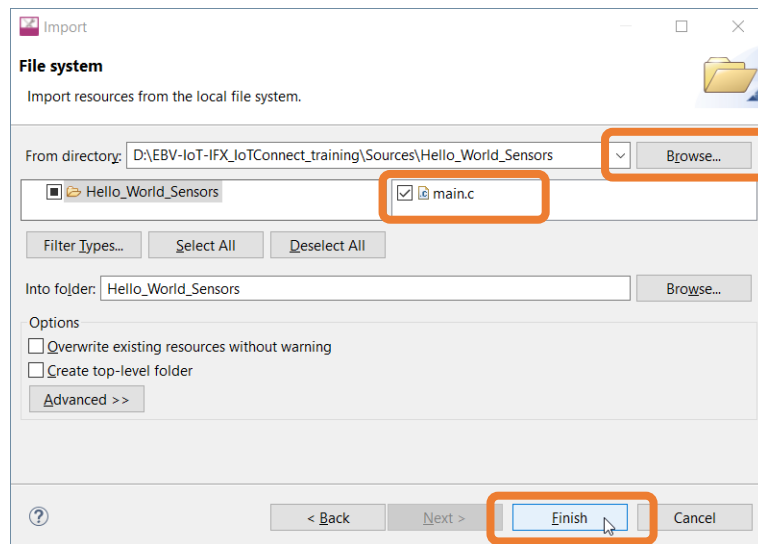


“Sources/Hello_World_Sensor” folder in the training material. There are multiple options to replace the content in the “main.c” file:

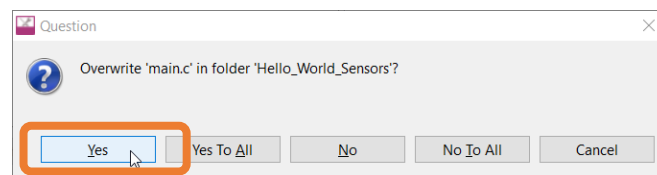
- A. **Right click** on “Hello_World_Sensors” project at **Project Explorer** and select **Import**
At Import prompt select **General**→**File System** and click **Next**



Browse for the Sources→Hello_World_Sensors in the training material and use **Select Folder**.
Tick “main.c” and click **Finish**.

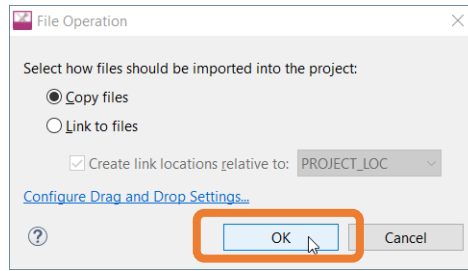


When prompted for “Overwrite ‘main.c’ ... ?” click **Yes** or **Yes To All**.

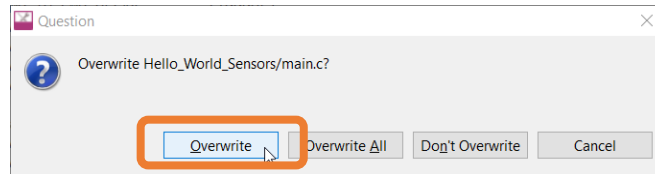


- B. In **Windows Explorer** replace the existing “main.c” with new “main.c”, i.e. **Copy-Paste** new “main.c” file to “{Your workspace}/Hello_World_Sensors” and confirm **Replace the file in the destination**.
C. **Drag and drop** the new file from **Windows Explorer** to “Hello_World_Sensors” project at **Project Explorer**, when prompted select **Copy files**, and click **OK**.



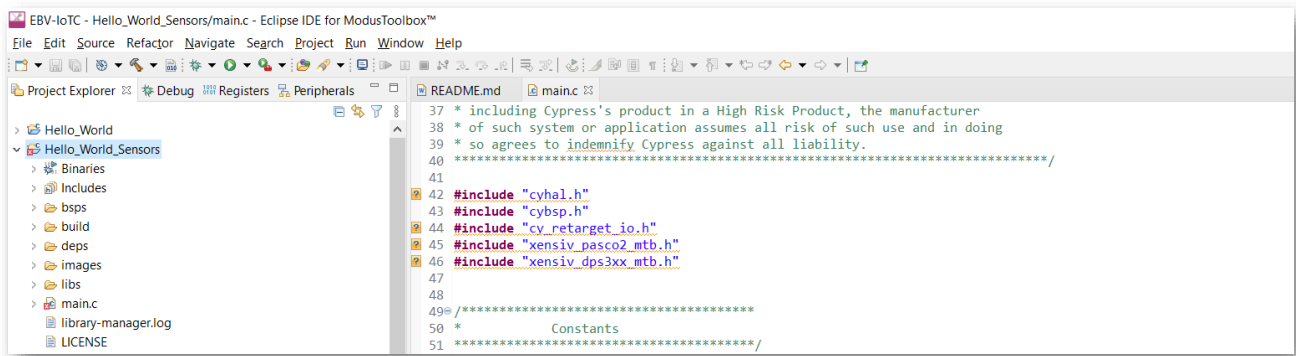


When prompted for “Overwrite ‘main.c’ ... ?” click **Overwrite** or **Overwrite All**.

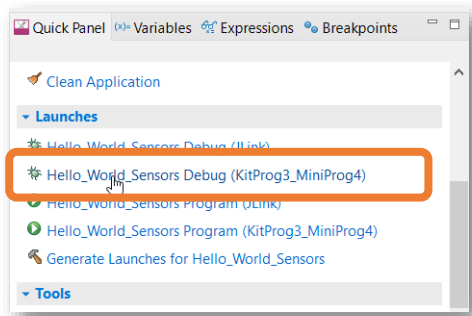


- D. Copy-Paste: **Copy** the content of the “{training material}/Sources/Hello_World_Sensors/main.c” file content and **Paste – make sure to overwrite all content –** it to “{your workspace}/Hello_World_Sensors/main.c” file using your favorite text file editor.

If all goes as expected, in the “main.c” file in **Code Editor** you should be seeing two “xensiv” includes in lines 45 and 46 as shown bellow:



Now in **Quick Panel** click **Hello_World_Sensors Debug (KitProg3_MiniProg4)**, wait for project to compile, build, program...



then click **Run→Resume** or press **F8** and check serial communication terminal... you should be seeing messages like bellow:



```

Terminal – COM14
File Edit Setup Control Window Help
*****
EBV-IoT - PAS C02 Demo
PSoC 6 MCU, I2C, PAS C02, DPS310, OLED
*****

>> Configuring I2C Master..... Done

Press 'Enter' key to pause or resume blinking the user LED

C02:      :      1816 ppm
Pressure  :      992.38 mbar
Temperature:      26.14 °C

C02:      :      1800 ppm
Pressure  :      992.37 mbar
Temperature:      26.14 °C
  
```

In previous step you learned how to step-by-step debug application. In this step you can explore further by checking included libraries. You can check the code through browsing “mtb_shared” folder in **Project Explorer** or...



Tip: In any of the “.c” or “.h” files you can explore elements like functions, defines or variables by holding “CTRL” button, hovering over the element, and clicking on it. It will take you to either the definition or declaration of the respective element.



Task accomplished: Congratulations, you copied your first “Hello_World” project, added new libraries and extended functionality sensor readouts through I2C bus.



9. Task 3: OPTIGA™ provisioning for IoTConnect Cloud



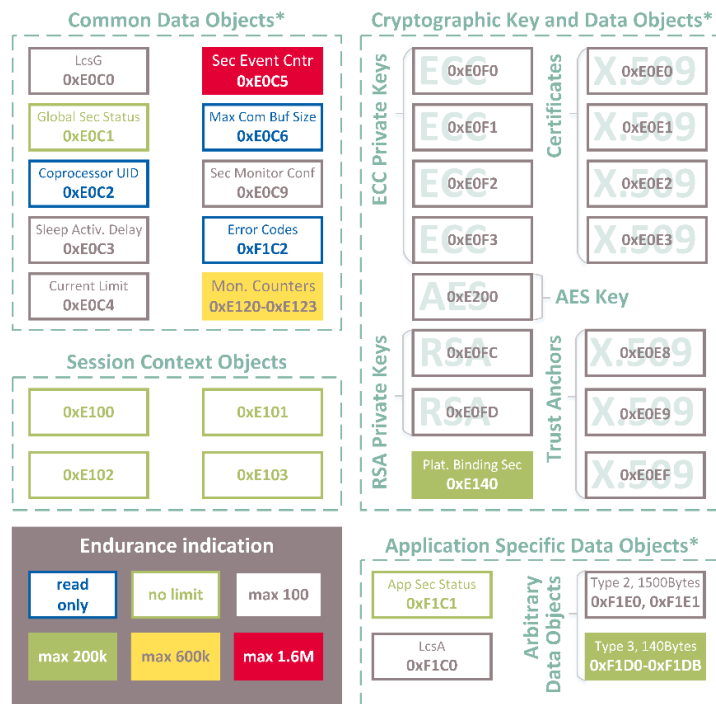
Important: This task requires access to IoTConnect cloud platform using your IoTConnect credentials. Software prerequisites 2.3 *Create IoTConnect cloud account* has to be completed prior.

9.1. Overview

In this task we will learn how to use OPTIGA™ Trust M secure element – OPTIGA – using provided tools by Infineon ([link](#)). We are going to use OPTIGA's unique ID and pre-provisioned certificate with a self-signed certificate to enable mutual TLS authentication with the Avnet IoTConnect cloud.

It is of a great importance for each IoT device getting unique identity linked with unique device certificate. Most of the time knowing the data is coming from the device we expect is more important than eventually preventing the data to be read.

OPTIGA consists of various objects which can be accessed for read/write using its Object IDs – OIDs. Objects can be Common Data such as Coprocessor UID, counters; Cryptographic Keys and Data objects such as unique AES key, ECC/RSA private keys and X.509 Certificates; or Application specific data objects using for various implementations.



* The content is either preconfigured by Infineon or is part of the personalization offer

Figure 12: OPTIGA objects map

In our implementation we use following objects:

- Coprocessor UID (0xE0C2) – for unique device ID extraction
- X.509 certificate (0xE0E0) – as a main certificate
- Application specific data object 1 (0xF1D2) – to store “ENV” value
- Application specific data object 2 (0xF1D3) – to store “CPID” value

Using secure element such as OPTIGA provides the OEMs with following benefits:

- Optiga pre-provisioned – avoiding additional provisioning step during production
- Anti-cloning protection (we only allow known devices to be accessing our system).
- There is no third-party in the personalization process – only trusted authorities involved





Note: A self-signed certificate is a type of digital certificate that is issued by the entity itself rather than a trusted third-party certificate authority (CA). In the context of computer security and encryption, a digital certificate is used to verify the authenticity and integrity of a website, server, or application.

When a website or server presents a digital certificate signed by a trusted CA, it provides assurance to users that the entity they are communicating with is legitimate and can be trusted. The CA acts as a trusted third party that verifies the identity of the entity and signs the certificate to confirm its authenticity.

In the case of a self-signed certificate, the entity generates its own certificate without involving a CA. This means that the certificate is not backed by a trusted third party and has not undergone the same level of verification. Consequently, self-signed certificates are not automatically trusted by web browsers or other software that relies on certificates for secure communications.

Self-signed certificates are commonly used in development or testing environments, where the purpose is not to establish trust with external users but rather to enable secure communication within a controlled environment. They can also be used for personal or private networks where the users are aware of the self-signed nature of the certificate.

9.2. Step by step guide



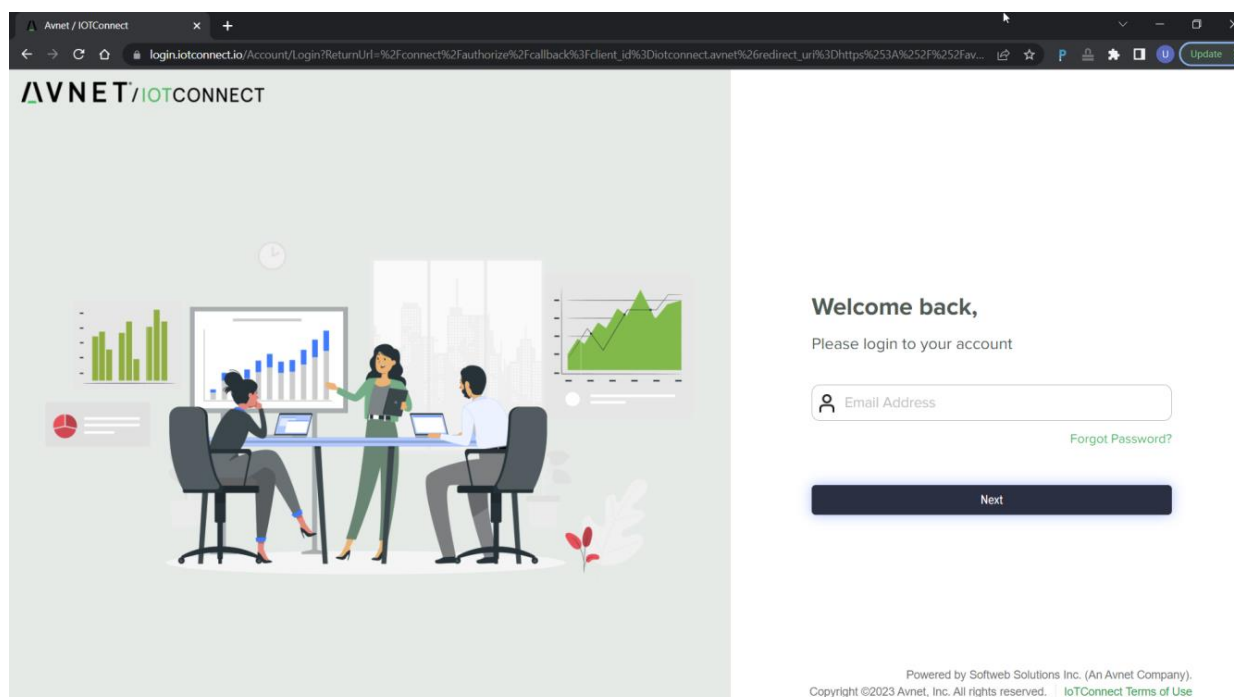
To Do: The task does not involve any software development. Instead, it is only required to process the steps in the right order and to understand their underlying effects.

To provision the OPTIGA™ Trust M secure element, it is necessary to get specific detail/s related to your IoTConnect cloud account:

1. Company Identifier.

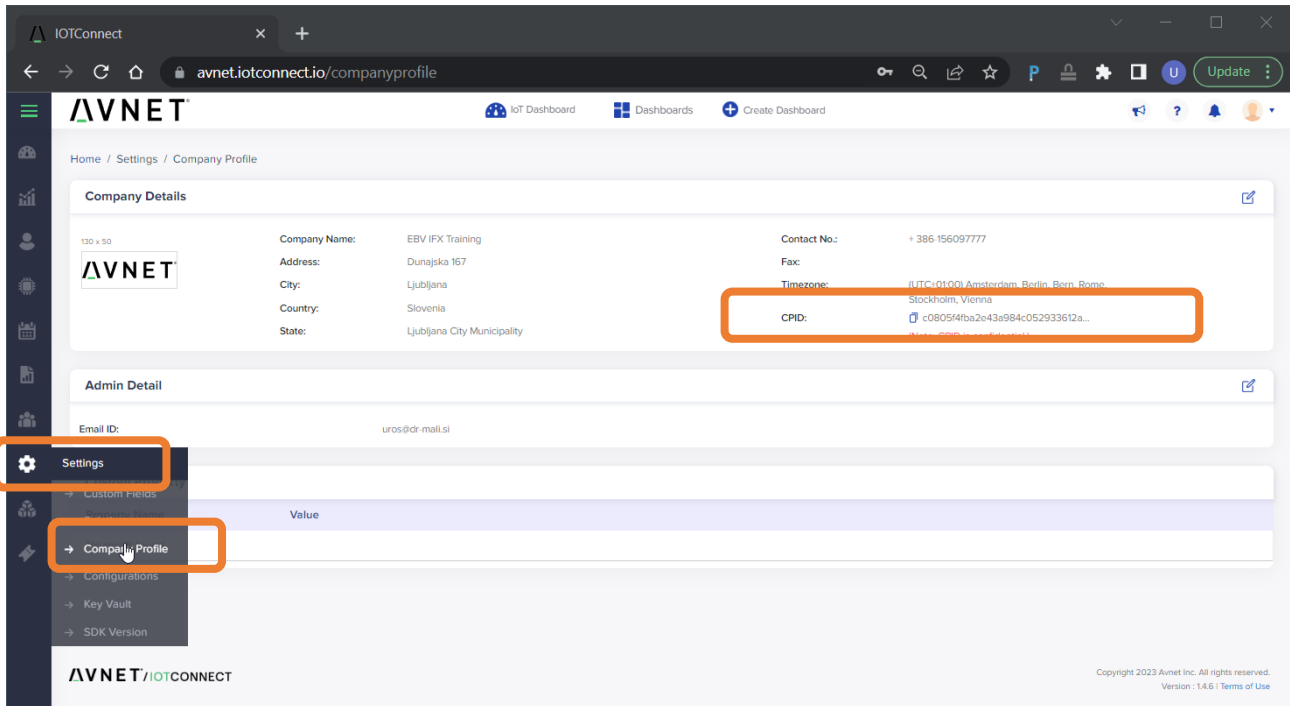
9.2.1 Retrieve IoTConnect cloud account details

Please login IoTConnect cloud using your IoTConnect cloud credentials - <https://avnet.iotconnect.io>.



From the left hand side menu, click **Settings → Company Profile** and notice Company Name and CPID (Company Identifier). Those details will be required when configure the secure element in the next step:






The screenshot shows the IoTConnect web interface. The 'Company Details' section contains the following information:

Field	Value
Company Name	EBV IFX Training
Address	Dunajska 167
City	Ljubljana
Country	Slovenia
State	Ljubljana City Municipality
Contact No.	+386 156097777
Fax	
Timezone	(UTC+01:00) Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna
CPID	c0805f4fba2e43a984c052933612a...

The 'Admin Detail' section shows the Email ID: uros@dr-mali.si.

The 'Settings' menu is visible on the left, with 'Company Profile' highlighted.

You can click  button to easily copy your CPID.

9.2.2 Edit OPTIGA programming script

In provided workshop material please locate “EBVIoT-OptigaTrustM-program.py” script file located in “Scripts” folder and open it using your favorite Python script editor. Eventually any text editor would just work fine.

Locate line starting with “CPID” and replace the value between the single quotes with CPID value retrieved from previous step.

```

EBVIoT-OptigaTrustM-program.py
File Edit Format View Help
1  #!/usr/bin/env python
2  import optigatrust as optiga
3  from optigatrust import objects
4
5  from OpenSSL.crypto import load_certificate, FILETYPE_PEM
6  import base64
7
8  from optigatrust import port
9  import json
10
11 ans = ""
12 ENV = b'avnepoc'
13 CPID = b'xx805f4fba2e43a984c052933612adf1'
14
15 #UID
  
```

Save the script.

9.2.3 Program the secure element



To Do: We will program the secure element using prepared scripts. To get more into the secure element itself we recommend checking following:

- »EBV-IoT - Infineon OPTIGA Trust M Evaluation Shield - Quick start guide« available in documentation folder
- Infineon »python-optiga-trust« github pages ([link](#))
- OPTIGA Trust M github page ([link](#))

Connect the “EBV-IoT – Infineon OPTIGA Trust M Evaluation Shield” to a PC using micro USB cable (Type A / Micro B or Type C / Micro B). Both RED LEDs should be on as shown on Figure 13. SW1 switch should be in PROG position.





Note: You can connect OPTIGA shield to your PC while remaining stacked. Just make sure the USB is plugged to the OPTIGA shield.



Important: To access secure element on I2C bus from the USB2I2C interface SW1 switch should be in PROG position.

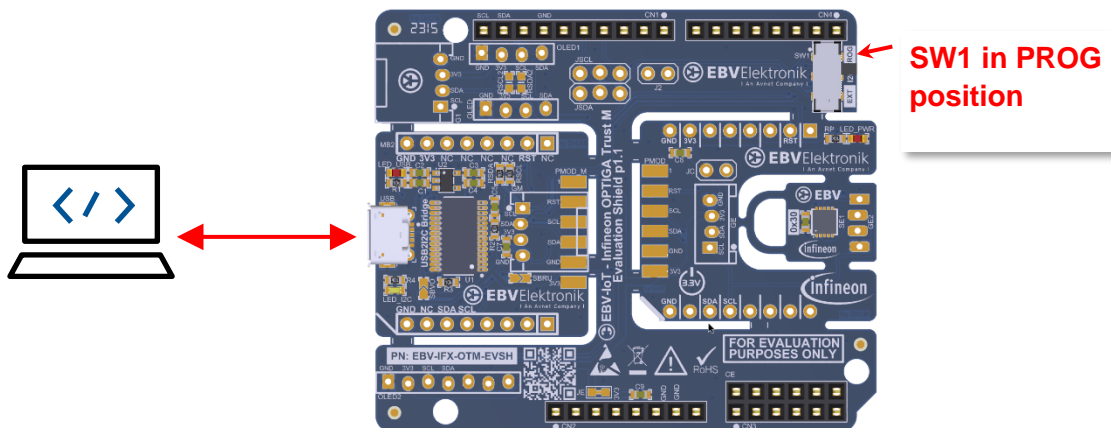


Figure 13: Connecting the OPTIGA shield to a PC

Next please run “modus-shell” or your instance of Python.



Note: We use “modus-shell” as a part of ModusToolbox™ toolchain. You can use Python as you toolchain of choice for following steps, too.

Go to a location of the previously edited Python script using command “cd {path}” where {path} is the location of the script. Use “TAB” key for autocompletion. Next please run the script using command “./EBVIoT-OptigaTrustM-program.py”

```

/cygdrive/d/EBVTraining/Scripts
user@computer_name ~
$ cd d:EBVTraining/Scripts

user@computer_name /cygdrive/d/EBVTraining/Scripts
$ ./EBVIoT-OptigaTrustM-program.py
Loaded: liboptigatrust-libusb-win-amd64.dll

Unique Id:
zRYzTQEAHABFAAAKCRtcABUASQB0gBA

Device Certificate Thumbprint:
a26c9a15818dfb5f305fe8d378668eb7f2fc1d85

Device Certificate:
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----

Saving certificate file: device_zRYzTQEAHABFAAAKCRtcABUASQB0gBA.pem

user@computer_name /cygdrive/d/EBVTraining/Scripts
$
    
```



Tip: You will need device’s Unique ID for the future reference. To copy any value from the editor, use mouse to mark the value to be copied and use either of the options:

- Press “Enter”
- Use mouse right click

Or you can extract the value from the certificate name at any stage later on.





Note: There is certificate – “pem” file generated and saved into the “../Script” folder which you will need in the following steps.

9.2.3.1 Troubleshooting

Most common mistake when interfacing OPTIGA is either the shield not properly connected to PC or OPTIGA not detected on the I2C bus. There are two possible reasons for the I2C bus detection issue:

1. The shield switch “SW1” is in a position “EXT” → please put the switch “SW1” to position “PROG” as shown on *Figure 13: Connecting the OPTIGA shield to a PC*.
2. The I2C bus is taken by external I2C master, or the pins are not in “open-collector” mode i.e. pins are externally held to low or high. → please release the pins. Disconnecting the shield from any other Arduino interface usually solves the issue.

If you see “Error: 0x102” it is very likely OPTIGA not seen on I2C bus, while only “Error: 0x202” listed, indicates an issue with shield to PC connection.

```

user@computer_name ~
$ optigatrust.exe object --id 0xe0e0
Error: 0x102

!!!Error in opening serial port : 2Error: 0x202
Could not find module 'D:\Infineon\ModusToolbox\tools_3.0\python\lib\site-
packages\optigatrust\csrc\lib\liboptigatrust-i2c-win-amd64.dll' (or one of its dependencies). Try
using the full path with constructor syntax.
libusb: Failed to connect
uart: Failed to connect
i2c: Failed to find library liboptigatrust-i2c-win-amd64.dll in
D:\Infineon\ModusToolbox\tools_3.0\python\lib\site-packages\optigatrust\csrc\lib

user@computer_name ~
$
  
```

9.2.4 Get QR code for seamless device onboarding

For one of the later steps of adding device to an end user account one of the most convenient ways is to use QR code to scan for provisioning. Like programming the secure element, we will run another script for QR code generation. Please run the script using command “./GetOptigaQR.py”

```

/cygdrive/d/EBVTraining/Scripts
user@computer_name ~
$ cd d:EBVTraining/Scripts

user@computer_name /cygdrive/d/EBVTraining/Scripts
$ ./GetOptigaQR.py
Loaded: liboptigatrust-libusb-win-amd64.dll

Unique Id:
zRYzTQEAHABFAAAKCRtcABUASQB0gBA

user@computer_name /cygdrive/d/EBVTraining/Scripts
$
  
```

After running the script, the QR code should be opened using any image viewer as well as the “qrcode001.png” file is left in the “Script” folder of the training material for your use in later steps.



Task accomplished: Congratulations, you have just programmed Optiga, extracted certificate, and created QR code for further use. You will need “Unique ID” and “Thumbprint” values in next step.



10. Task 4: Create a device in IoTConnect Cloud



Important: This task requires access to IoTConnect cloud platform using your IoTConnect credentials and secure element programmed. 2.3 Avnet IoTConnect cloud platform account and Task 3: OPTIGA™ provisioning for IoTConnect Cloud have to be completed prior respectively.

10.1. Overview

In the previous task, the OPTIGA™ Trust M secure element has been programmed with a »Company ID« and on top of that we use certificate »thumbprint« to connect and authenticate to your IoTConnect cloud account. The device resource must also be created in the cloud before the device is allowed to connect.

A certificate thumbprint, also called a fingerprint, is a hash of a certificate, computed over all certificate data and its signature. Thumbprints are used as unique identifiers for certificates, in applications when making trust decisions, in configuration files, and displayed in interfaces.

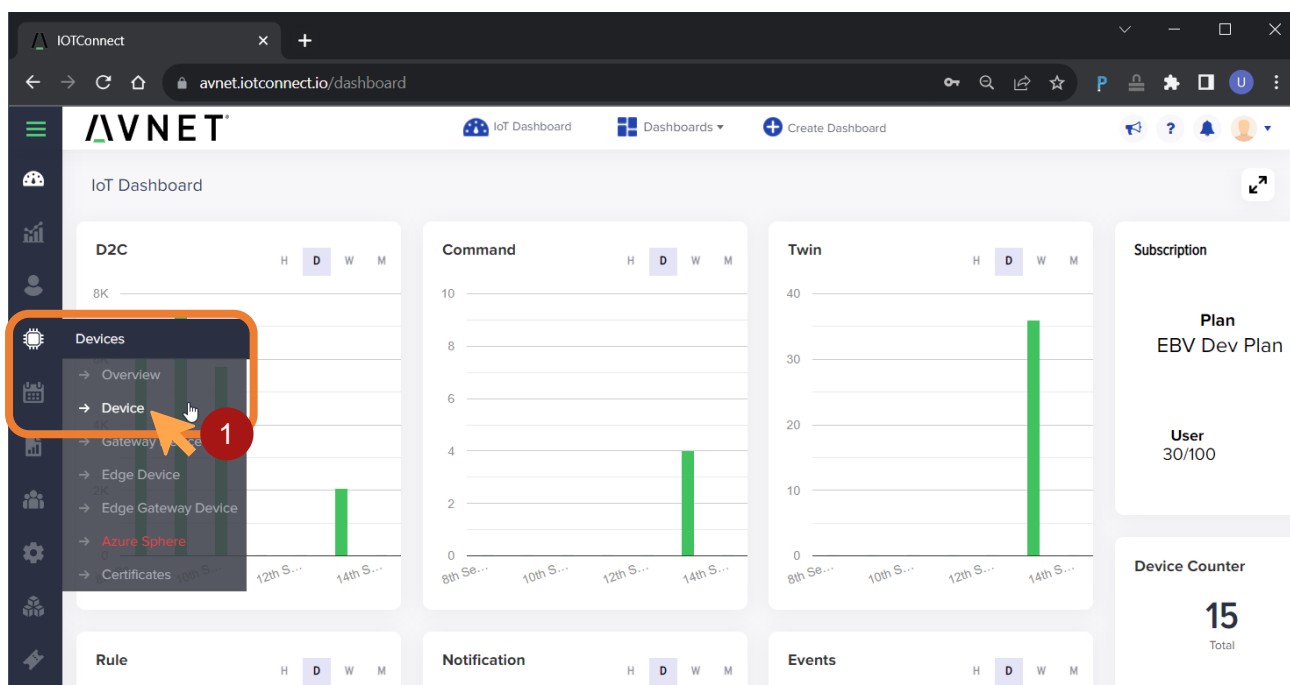
10.2. Step by step guide



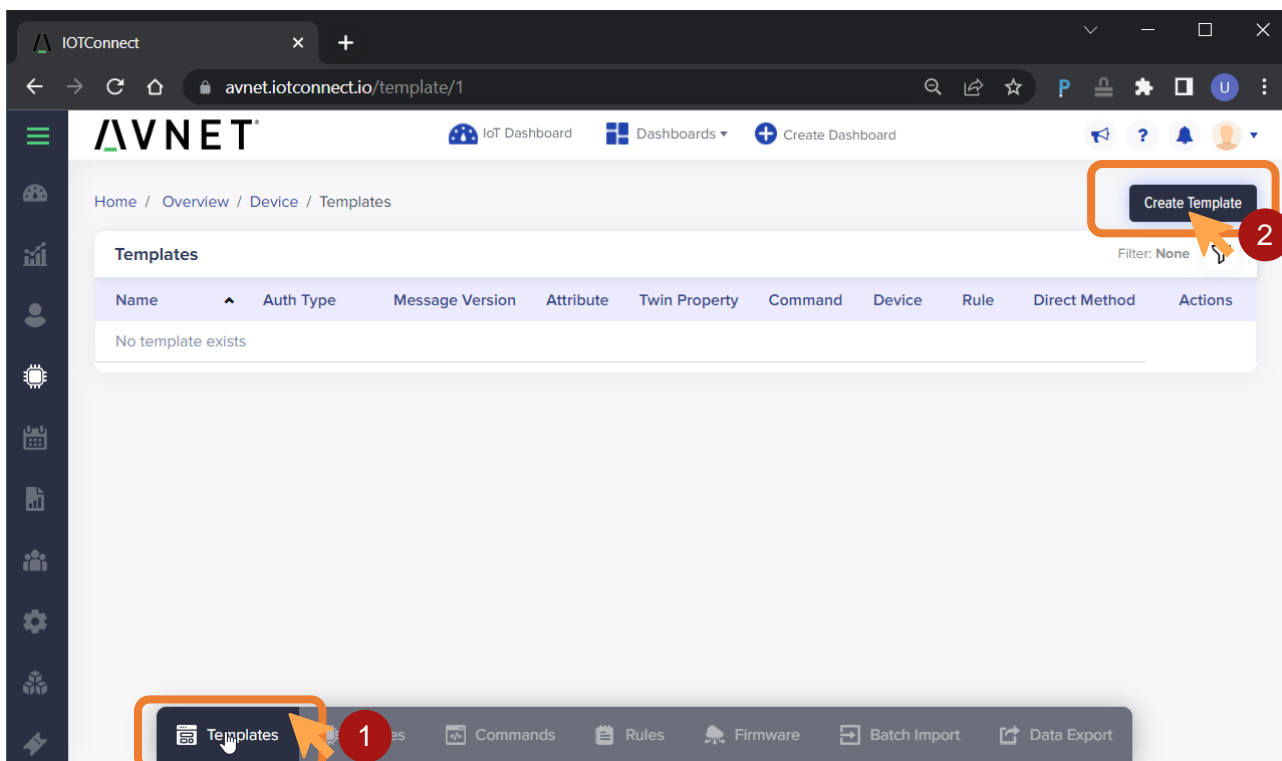
To Do: The task does not involve any software development. Instead, it is only required to process the steps in the right order and to understand their underlying effects.

Before creating a new device, it is necessary to create the corresponding device template. A template defines the JSON data structure that a device can report. Typically, it is used to provide a description of the sensors present on the board and its corresponding data types. Adding the device template needs to be done just one time.

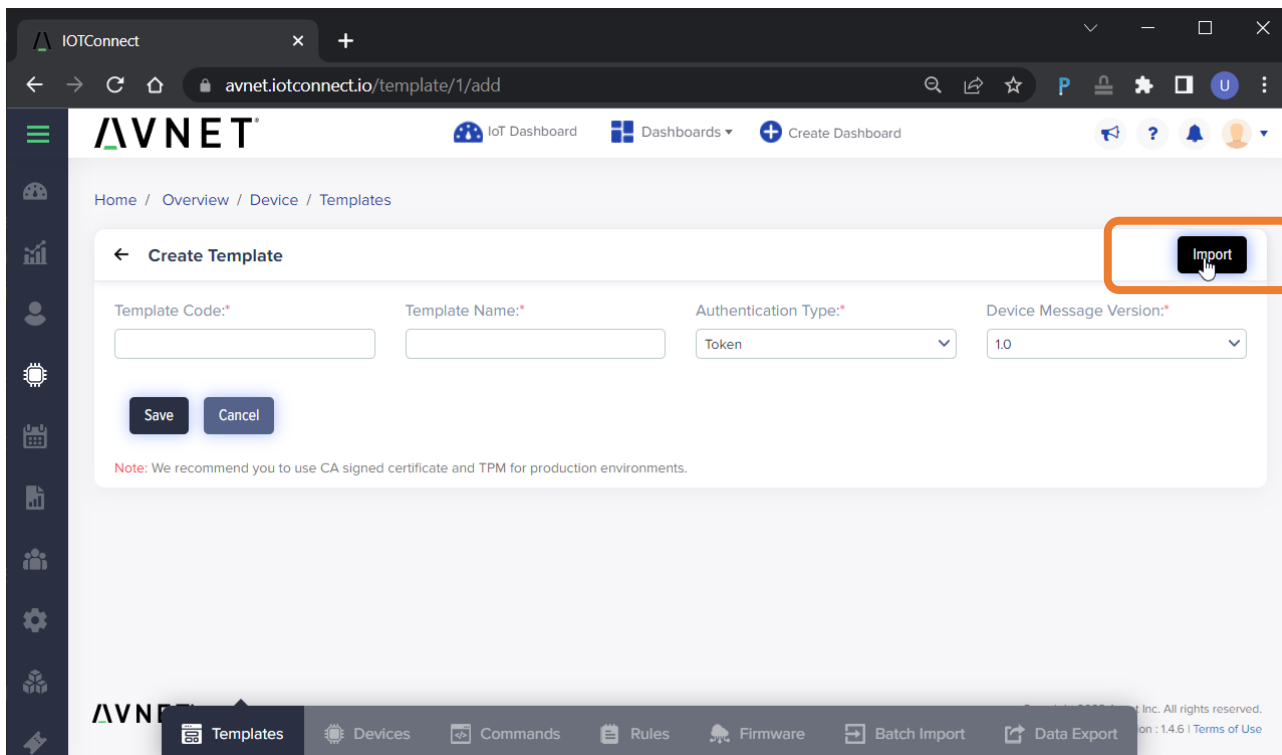
Please go to **Devices/Device** page as depicted below:



At the bottom menu click **Templates** and then click **Create Template** button:



It is possible to define the device template manually, nevertheless a faster solution is to click **Import** button and select the template.



At "IMPORT" pop up window **Browse** for "EBV-IoT Xensiv Demo v1.1_template.JSON" file from workshop material and finally click **Save** button:



IMPORT ✕

1. All keys are case-sensitive.
2. Supported *authType* : 1 - Token, 2 - x509, 4 - TPM, 5 - Symmetric Key
3. a. Supported type (datatype) : **number & string**
 b. Supported type (datatype) for 2.1 : **bit, boolean, date, datetime, decimal, integer, latlong, long, object, string & time**
4. You can set only one command as **OTA command**
5. Attribute Color should be in Hex Color Code format (Example : **#000000**).
6. Json property "responseTimeout" in Direct Method is in second

[Download Sample Template File](#)

Upload File:

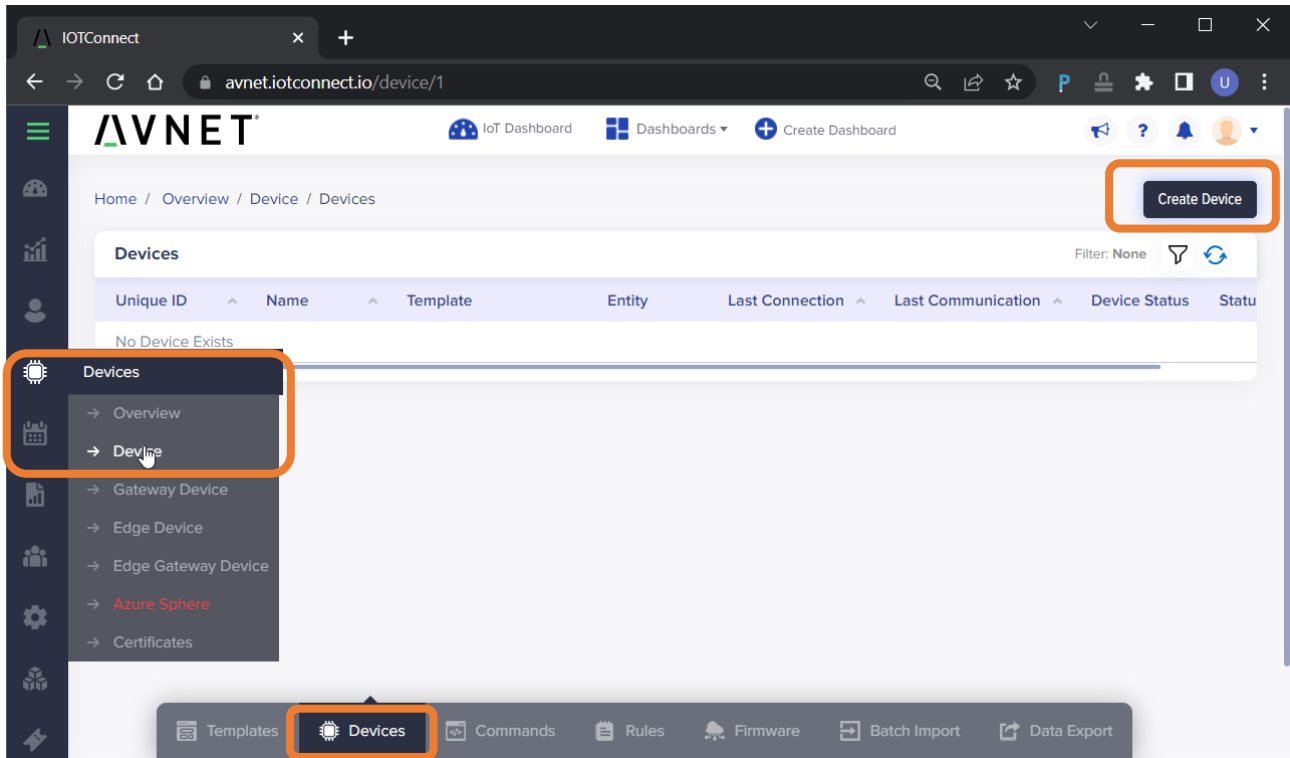
EBV-IoT - Infineon Xensiv Demo v1.2_template.JSON Browse

Save
Cancel

You should be seeing template details as shown on figure below as well as the template should now be listed under **Templates**.

Now browse to devices list using either left hand side menu **Devices → Device menu** or directly clicking on **Devices** at the bottom menu. As no devices were created so far we see no device on the list. Click **Create Device** button located on the upper right hand side.





The new device will be defined as following:

- **Unique Id:** each device must have Unique ID associated with. We use first 31 characters from "base64" representation of OPTIGA's Coprocessor UID



Important: Use "Unique Id" extracted during secure element programming in subsection **9.2.3 Program the secure element**.

- **Display name:** a friendly name to reference the device (can be anything)
- **Entity:** select the default entity that was created (usually only one value possible for your instance of IoTConnect cloud)
- **Template:** select the "EBV-IoT - Infineon Xensiv Demo v1.2" that has just been imported
- **Notes:** can be anything
- **Device certificate:** Browse for device's "pem" certificate



Important: Use device "pem" certificate extracted during secure element programming in subsection **9.2.3 Program the secure element**.



The screenshot shows the 'Create Device' form in the IoTConnect web interface. The form contains the following fields and options:

- Unique Id.***: A text field containing 'zRYzkwEAHAFAAAKAScXAAoAKQBCgBA' (Callout 1).
- Display Name.***: A text field containing 'Uros Xensiv Demo' (Callout 2).
- Entity.***: A dropdown menu showing '- EBV IFX Training' (Callout 3).
- Template.***: A dropdown menu showing 'EBV IoT - Infineon Xensiv Demo v1.2' (Callout 4).
- Device certificate.***: Radio buttons for 'Use my certificate' (selected) and 'Auto-generated'.
- Notes**: A text area for additional information.
- Certificate Authority**: A dropdown menu with 'Select Certificate'.
- Device Certificate.***: A text field with a 'Browse' button (Callout 5).
- Buttons**: 'Save' (Callout 6), 'View', and 'Cancel' buttons at the bottom.

Click **Save** button.

You should be seeing “Device created successfully” and device listed in the device view:

The screenshot shows the 'Devices' view in the IoTConnect web interface. A green banner at the bottom indicates 'Device created successfully'. The table below lists the created device:

Unique ID	Name	Template	Entity	Last Connection	Last Communication	Device Status	State
zRYzTQEA...	Uros Xensiv D...	Self Signed 1.0	EBV-IoT ...	EBV IFX Tr...	NA	NA	DISCONNECTED

Showing 1-1 of 1 entry



Task accomplished: Congratulations, the device has now been properly setup and is ready to authenticate to the IoTConnect cloud.



11. Task 5: Getting hardware ready for IoTConnect cloud



Important: This task requires device created in your instance of IoTConnect cloud platform. *Task 4: Create a device in IoTConnect Cloud* has to be completed prior.

11.1. Overview

In this task we will learn how to import existing solution, apply minor code change, and adjust WiFi settings.



Note: All the settings related to your IoTConnect cloud instance are provisioned in the secure element.

11.2. Step by step guide



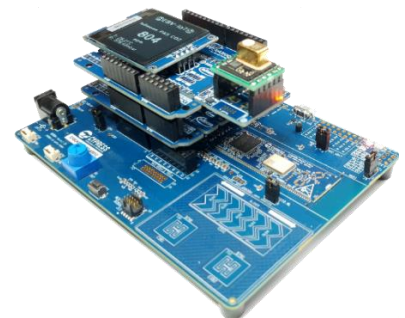
To Do: The task does not involve any software development. Instead, it is only required to process the steps in the right order and to understand their underlying effects.

11.2.1 HW setup

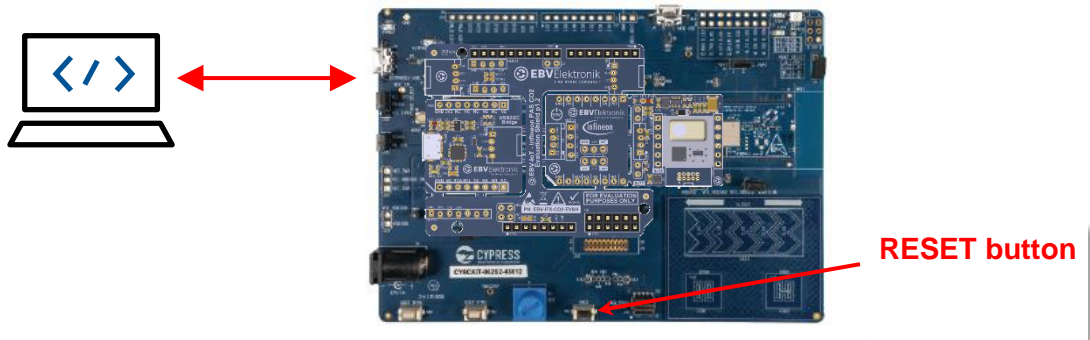
Before continuing, make sure that **EBV-IoT – Infineon OPTIGA Trust M evaluation shield** I2C switch is on **EXT** position. Make also sure that **EBV-IoT – Infineon PAS CO2 Evaluation Shield** is set to I2C mode. Please check section **5 Hardware configuration** for details.

For convenience, stack the Arduino compatible shields in the following order from bottom to top:

- The PSoC™ 62S2 Wi-Fi BT Pioneer Kit (CY8CKIT-062S2-43012) – at the bottom
- EBV-IoT – Infineon PAS CO2 Evaluation Shield
- EBV-IoT – Infineon OPTIGA Trust M Evaluation Shield
- 128x128 pixels OLED display (optional)



Then only plug your laptop to the PSoC Kit USB programming interface as shown below:



Important: **Terminate** previous **debug** session (if still running) and switch perspective to **ModusToolbox (default)** perspective.



Quick Start: Please jump to section **12 Task 6: Flashing device firmware** using “hex” firmware file.



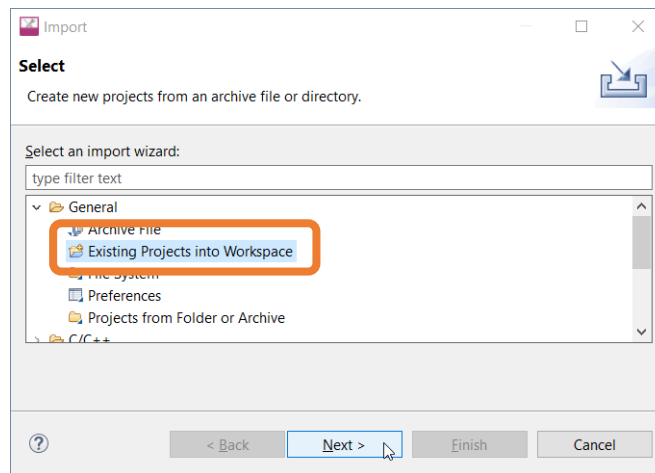
11.2.2 Import project

In the Elclipse IDE for ModusToolbox in your existing workspace we will import existing project to simplify the process of application development. Most of the time this is the case how you start developing your project and added functionalities of your choice.

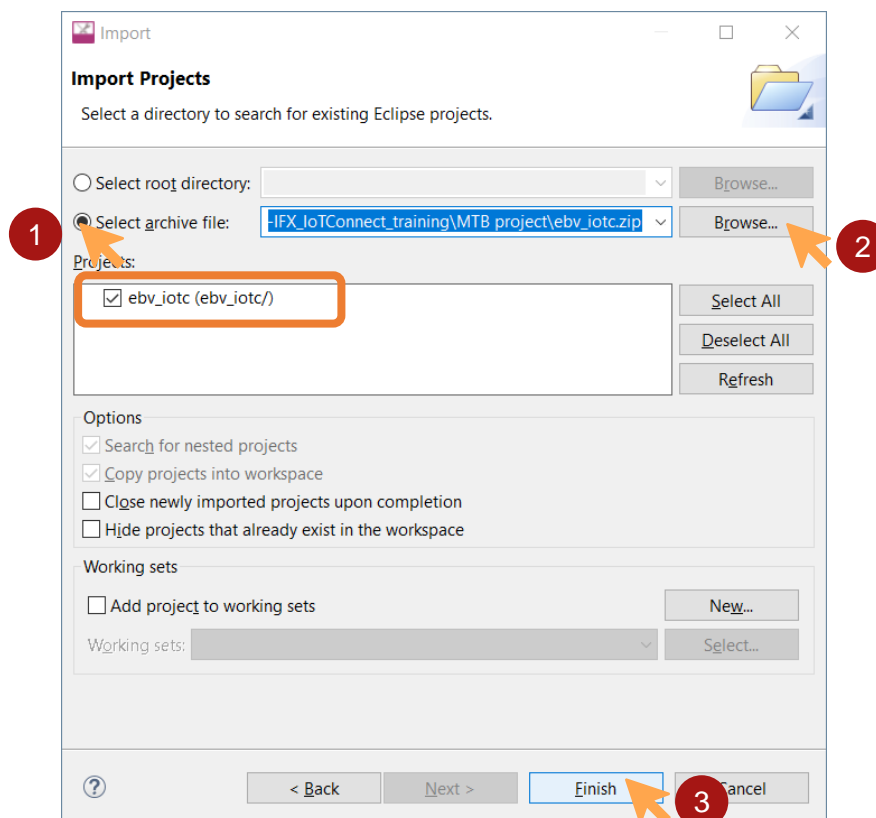
There are multiple ways of importing projects:

- A. Go to **Toolbar** menu and select **File->Import**.
- B. Or **Right click** on empty space of **Project Explorer** and select **Import**.

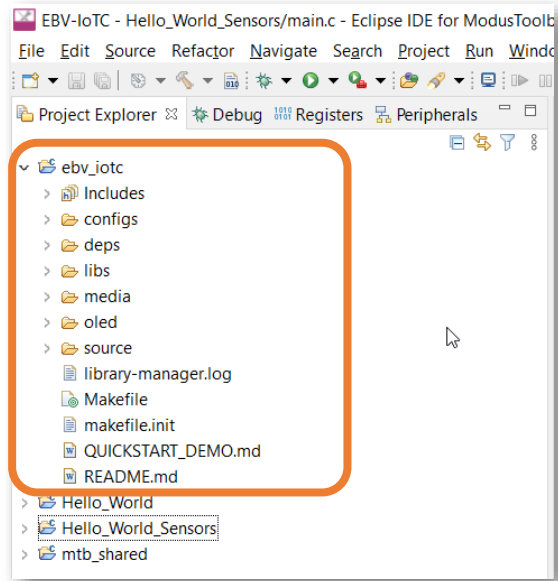
When **Import – Select** window prompted select **General→Existing Projects into Workspace** and click **Next**:



At **Import – Import Projects** window please select **Select archive file:** and click **Browse**. Search for **“ebv-iotc.zip”** located in training material **“MTB project”** folder. Make sure the **“ebv-iotc”** project is ticked in **Projects:** area. Finally, click **Finish**.



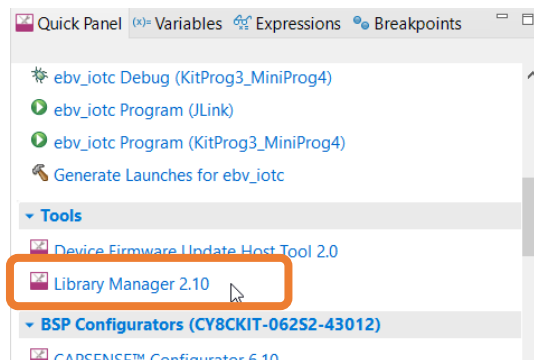
You should be seeing newly imported project in Project Explorer:



Important: We imported a project and most importantly now is to update library set. It is of a great importance to follow steps of this sub-section.

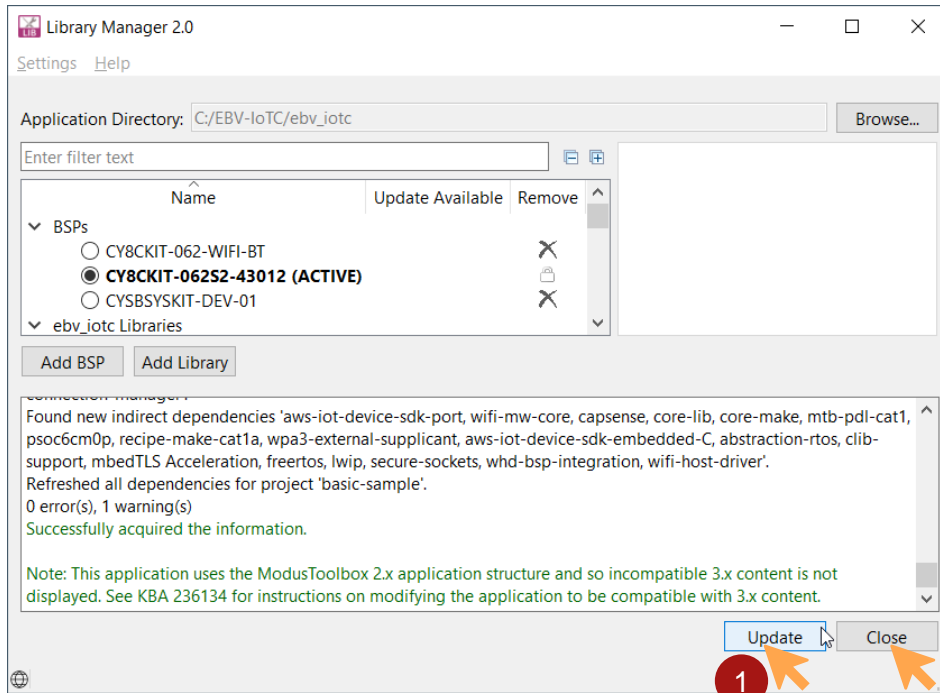
There are multiple ways to update libraries (proceed using only one of the options):

- A. Browse for “main.c” file in **Project Explorer** under “ebv-iotc/source” and open it (**double click** or **right click** → **Open**). In **Quick Panel** click **Library Manager 2.1**

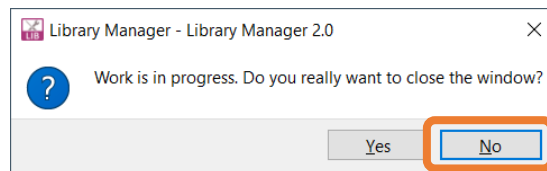


In the **Library Manager 2.1** just click **Update**, wait for the update to finish, wait for **Update** button to get enabled and click **Close**. The update process might take a while...

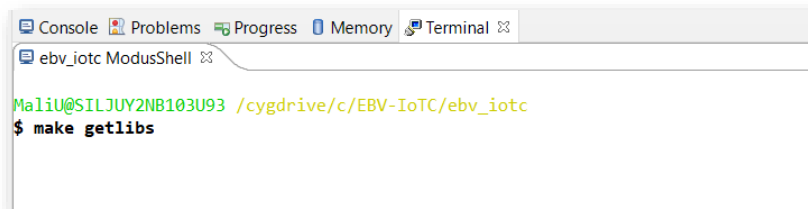




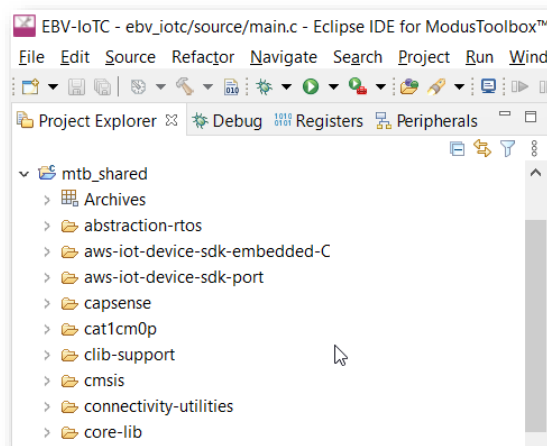
If you click **Close** before update process finishes, you will be prompted for confirmation. Click **No** and wait for Update button to get enabled and click **Close**.



B. Alternatively, in the “ebv-iotc ModusShell” terminal execute “make getlibs” command.



In the **Project Explorer** under “mtb_shared” you should see many newly added libraries:

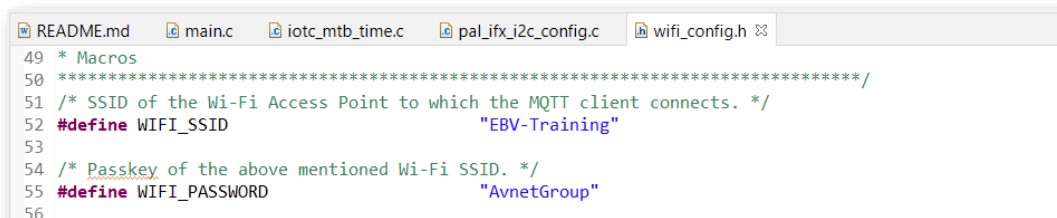


11.2.3 Configure WiFi credentials



Note: WiFi credentials can be changed also using serial communication terminal at later steps or at any time later.

In “**ebv-iotc→configs**” folder locate and open “**wifi_config.h**”. In lines 52 and 55 you can see **WIFI_SSID** and **WIFI_PASSWORD** definitions, respectively. Change the values to match your WiFi access point settings.



```

49 * Macros
50 *****/
51 /* SSID of the Wi-Fi Access Point to which the MQTT client connects. */
52 #define WIFI_SSID "EBV-Training"
53
54 /* Passkey of the above mentioned Wi-Fi SSID. */
55 #define WIFI_PASSWORD "AvnetGroup"
56
  
```

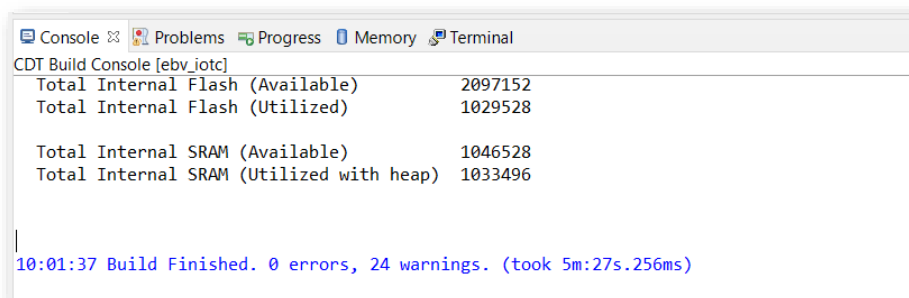
11.2.4 Build

Finally go back to “**ebv-iotc**” project in **Project Explorer** and execute **Build Application** from **Quick Panel**. In the console you should be seeing message flow with all the steps of build process. This step takes some time...



Note: Due to number of included packages it takes some time to compile, but it only takes so much time for the first compile or after “Clean Application” execution. In the **Console** you can follow messages flow until receiving something like:

10:01:37 Build Finished. 0 errors, 24 warnings. (took 5m:27s.256ms)



```

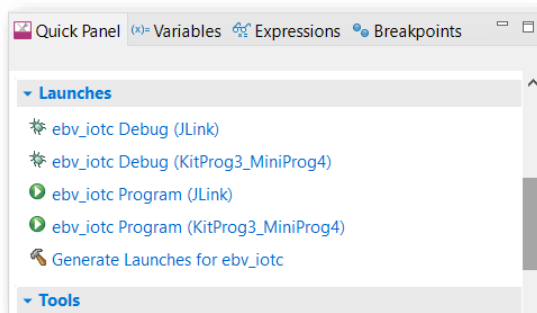
CDT Build Console [ebv_iotc]
Total Internal Flash (Available)      2097152
Total Internal Flash (Utilized)       1029528

Total Internal SRAM (Available)       1046528
Total Internal SRAM (Utilized with heap) 1033496

10:01:37 Build Finished. 0 errors, 24 warnings. (took 5m:27s.256ms)
  
```

11.2.5 Program and run the device application

Now you can run application using **ebv-iotc Debug (KitProg3_MiniProg4)** launch at **Quick Panel**.



Click **Run→Resume** or press **F8** and check serial communication terminal...



Task accomplished: Congratulations, the device has now been flashed using ModusToolbox™ debugging functionality. Proceed to Section 13 *Task 7: Accessing the device through serial terminal.*



12. Task 6: Flashing device firmware



Tip: For the “Full” version of the Infineon IoTConnect Cloud connected solution please proceed to Section 13 Task 7: Accessing the device through serial terminal

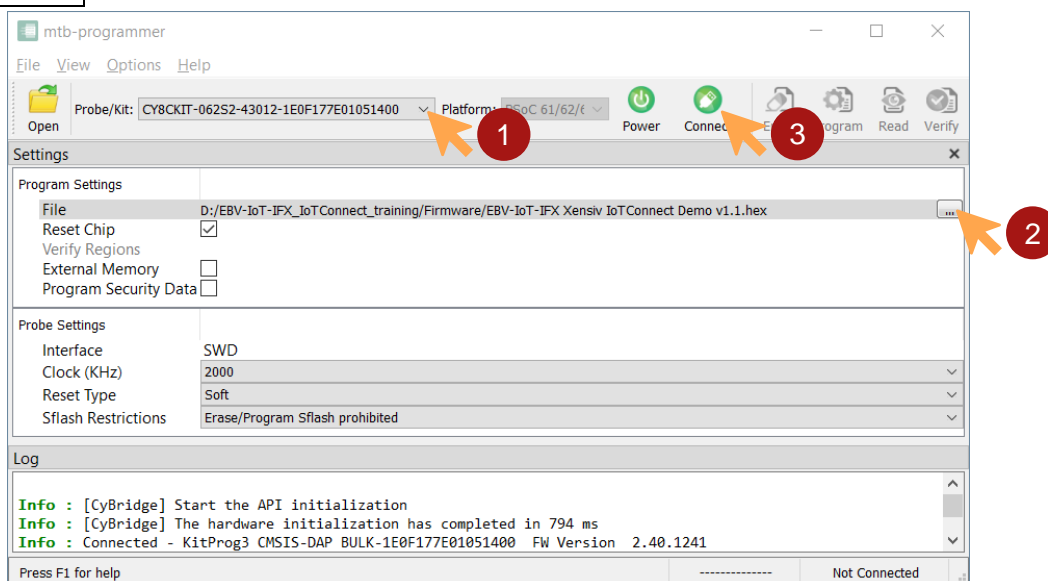
Please refer to the Infineon’s **ModusToolbox™ Programmer Software** pages for application download and “ModusToolbox™ Programming GUI user guide” ([link](#)) for installing instructions or use the installation and guide provided in the training material.

Assuming the PSoC 62 Pioneer Kit is properly connected to your PC using micro USB cable, proceed with launching **ModusToolbox™ programmer** in your Windows **Start menu** under:

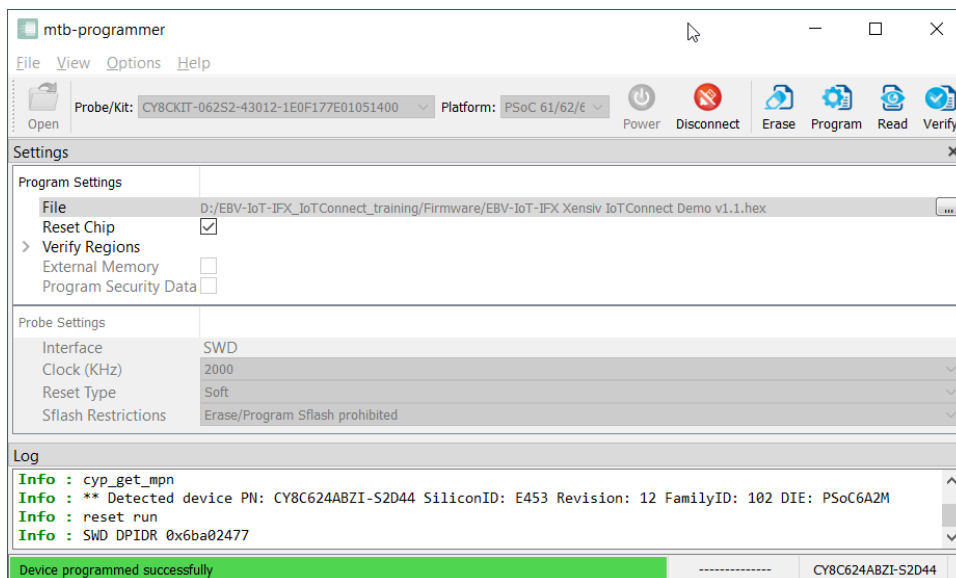
- A. **Recently added** → **mtb-programmer x.x**, or
- B. **Infineon technologies** → **mtb-programmer x.x**

where x.x denotes version of the installed tool.

1. Select “CY8CKIT-062S2-43012” from the drop-down menu. Usually, the tool recognizes connected kit automatically.
2. Click on [...] and browse for valid “hex” type firmware file located in “Firmware” folder of provided training material.
3. Click on **Connect**.



When “Program” icon gets enabled, click **Program** and wait for “Info : ** Programming Finished **” message or “Device programmed successfully” message is displayed.



Now click Disconnect.

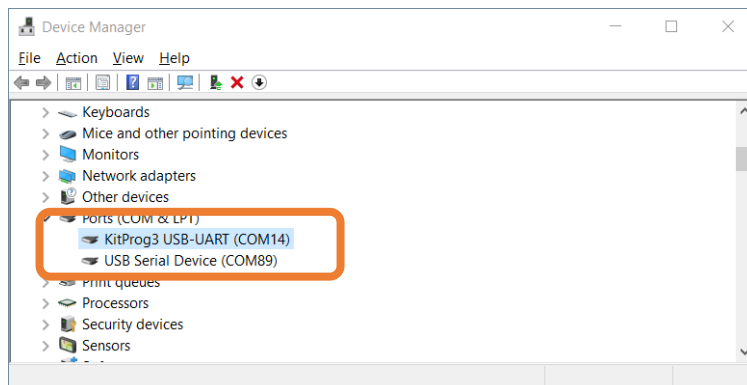


Task accomplished: Congratulations, you have now successfully flashed the device.



13. Task 7: Accessing the device through serial terminal

Please use device manager to get your device serial port number information like the one depicted below:



Use your favorite serial communication terminal for serial communication monitoring. We use TeraTerm or you can use ModusToolbox integrated serial communication terminal. Use following settings:

- Port number: as retrieved from "Device manager"
- Speed: 115200
- Data: 8 bit
- Parity: none
- Stop bits: 1
- Flow control: none

When device is connected and after resetting you should see flow of data in your terminal, like the one below:

```

Terminal – COM14
File Edit Setup Control Window Help
=====
EBV-IoT - Infineon Xensiv & Avnet IoTConnect Demo v1.02.02
=====
(00:00:00.000) [optiga_trust_helpers]: OPTIGA Trust: Starting initialization...
(00:00:00.079) [optiga_trust_helpers]: optiga_util_write_data: successful
(00:00:00.079) [optiga_trust_helpers]: OPTIGA Trust initialization successful.
(00:00:00.162) [optiga_trust_task]: Your certificate is:
-----BEGIN CERTIFICATE-----
MIICxTCCAkugAwIBAgIECsw3ADAKBgqhkhjOPQQDAzByMQswCQYDVQQGEwJERTeh
MB8GA1UECgwYYSw5maW5lb24gVGVjaG5vbG9naWVzIEFHRMRMEQYDVQQLDAPUFRJ
R0EoVE0pMSswKQYDVQQDDCJJbmZpbmVvbiBPUFRJR0EoVE0pIFRydXN0IE0gQ0Eg
MzA2MB4XDTEyMDcwNDANNDgyM1oXDTQyMDcwNDANNDgyM1owDTELMAkGA1UEAwWC
IiIwWATBgcqhkhjOPQIBBgqhkhjOPQMBBwNCAAS2XMfg180WaEb2oxsKERej9B+A
5T1X0Qoa7r0FRrhy5j2gHjcYUU2yEea+bY3vHBzQzB24E3HQzP1Py/yk5r-fio4IB
MjCCAS4wYAYIKwYBBQUHAQEEDBSMFAGCCsGAQUFBzAChkRodHRwcovL3BraS5p
bmZpbmVvbi5jb20vT3B0aWdhVHJ1c3RlY2NDQTMwNi9PcHRpZ2FucnVzdEVjY0NB
MzA2LmNyDDAdBgNVHQ4EFgQUAXtoQYtJPpFXRIa7k4V9s5jX/5FYwDgYDVR0PAQH/
=====

```

You can see message flow with various content including sensor values.

13.1.1 Setting WiFi credentials using serial terminal

In this step we will set WiFi credentials if the one provided in the source code does not match. Within terminal tool press either "x" or "X" at any time. You should be seeing a menu with few options:



```

Terminal – COM14
File Edit Setup Control Window Help
=====
Enter credential data using following input
=====
H followed by 'ENTER' to show this menu
S{WiFi SSID} followed by 'ENTER' - SET WiFi SSID
P{WiFi Password} followed by 'ENTER' - SET WiFi Password
U followed by 'ENTER' to display UID
Q followed by 'ENTER' to display QR code web link
D followed by 'ENTER' to use default values
R followed by 'ENTER' to review credentials
W followed by 'ENTER' to write credentials to EEPROM and EXIT
L{x} followed by 'ENTER' to set printf debug level {x} 2-8
M{lat,lon} followed by 'ENTER' to set latitude/longitude
E followed by 'ENTER' to EXIT!
=====
  
```

To change WiFi credentials, please use following commands:

- “S{WiFi SSID}” followed by ‘ENTER’
- “P{WiFi Password}” followed by ‘ENTER’
- “W” followed by ‘ENTER’ to write credentials to EEPROM and EXIT

```

Terminal – COM14
File Edit Setup Control Window Help
SMYWiFi↵
OK.
PMYPassword↵
OK.
W↵
Saving data...OK.
  
```

When you notice “Successfully connected to Wi-Fi network ‘{WiFi SSID}’” or you notice “Publishing data” you are ready to go for next step.



Note: During entering credential data it may happen some messages override messages on the screen. Please do not mind and finish typing, and press ‘ENTER’. You should be seeing “OK” message and valid enter.



Tip: For a WiFi access point you can use your PC and create “Mobile hotspot” located next to WiFi settings in your task bar.

Additionally, you can set “level” of printed messages. Default level is “L6” which prints additional messages on terminal. Following please find all the “level” settings:

- L2: Setting print log message level for Critical error, caller function crashes the application,
- L3: Setting print log message level for Error recovered,
- L4: Setting print log message level for Notice on possible error,
- L5: Setting print log message level for Information describing expected outcome,
- L6: Default - Setting print log message level for Additional information useful for debugging,
- L7: Setting print log message level for Less important information useful for debugging,
- L8: Setting print log message level for all levels.




```
Terminal – COM14
File Edit Setup Control Window Help
L5.
Setting print log message level for Information describing expected outcome
OK
L6.
Setting print log message level for Additional information useful for debugging
OK
E.
EXIT & no saving... OK.
```

When “level” set, type E followed by ‘Enter’.

For demonstration purposes you can also set position (latitude/longitude) of the device. When in the menu use command “M{latitude,longitude}” followed by ‘Enter’, “R” followed by ‘Enter’ to review values and “W” followed by ‘Enter’ to store values in the EEPROM.

```
Terminal – COM14
File Edit Setup Control Window Help
M48.17189112731593, 11.801411796690346
OK
R
WiFi SSID: EBV-Training
WiFi Password: AvnetGroup
EBV LOGs level: 8
Your position:
https://www.google.com/maps/search/?api=1&query=48.17189112731593,11.801411796690346
```



Tip: To set a position you can directly copy values from the Google Maps adding “M” prior like e.g. above: “M48.17189112731593, 11.801411796690346”



Task accomplished: Congratulations, the device has now been properly setup and is ready to check device connecting to the IoTConnect cloud.



14. Task 8: Connecting the demo to IoTConnect Cloud



Important: This task requires the device being programmed with IoTConnect enabled application. *Task 5: Getting hardware ready for IoTConnect cloud* has to be completed prior the task.

14.1. Overview

Actually, your device should already be connected to the cloud. We will walk through the cloud platform back-end for more details.

14.2. Step by step guide

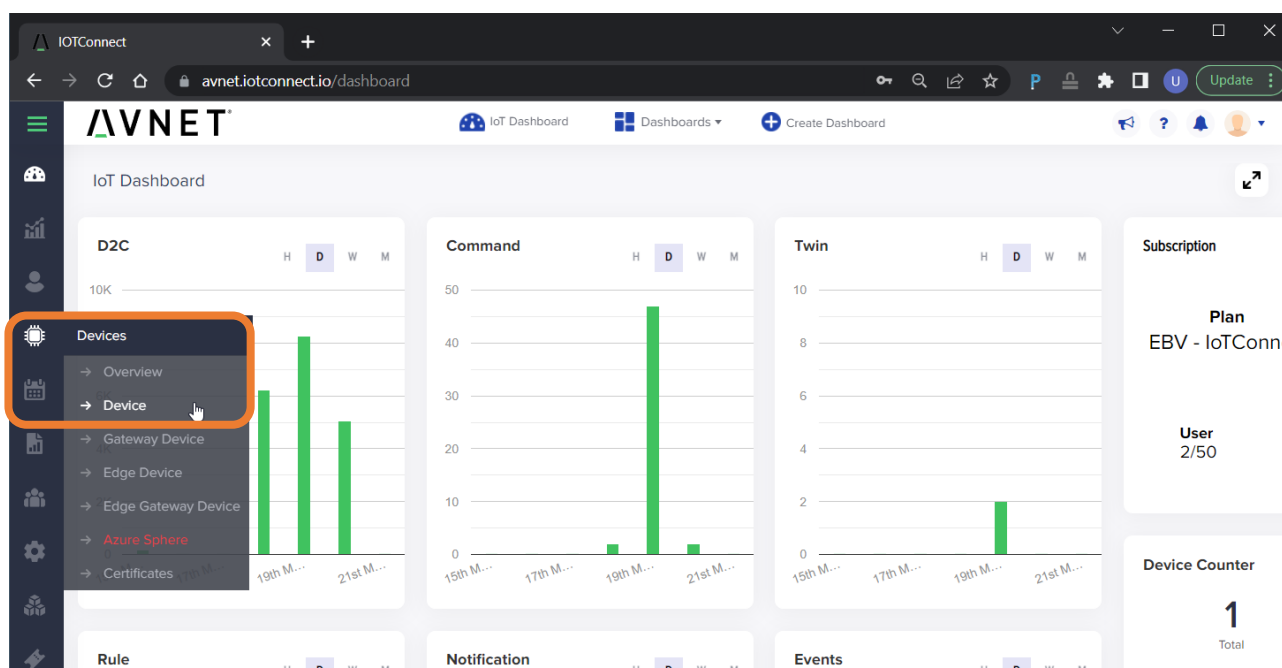


To Do: The task does not involve any software development. Instead, it is only required to process the steps in the right order and to understand their underlying effects.

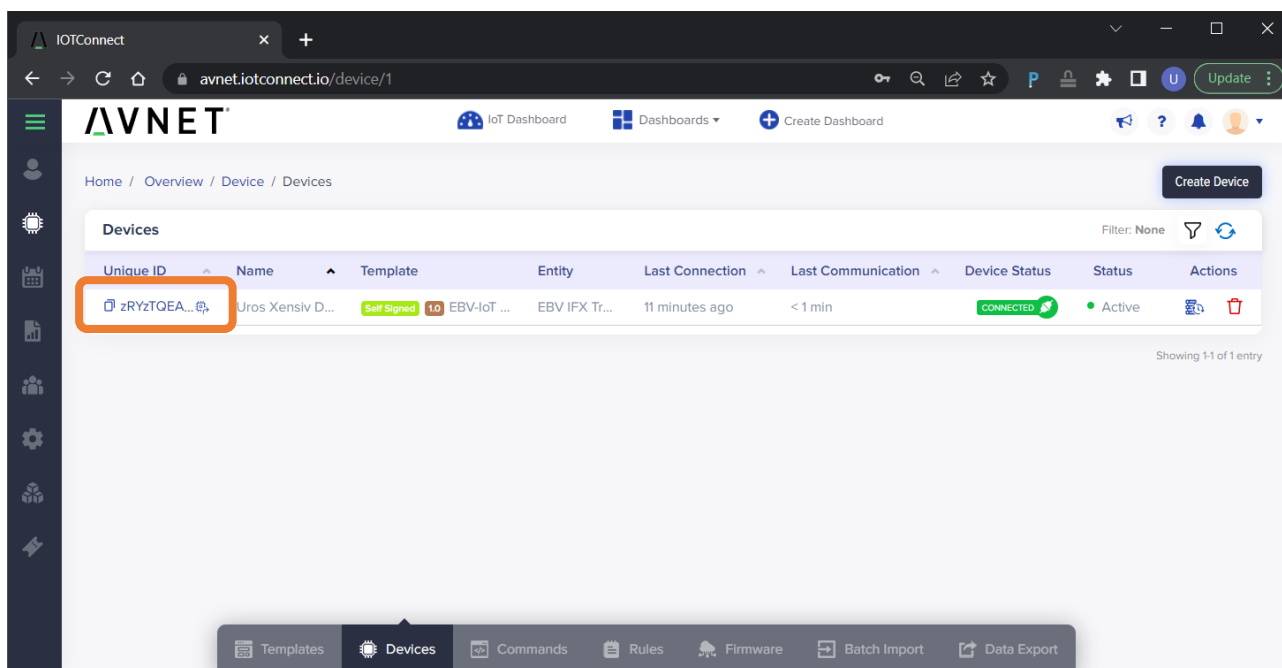
Once device fully programmed, open the serial communication terminal. The device firmware should automatically reset, and the terminal output should show a successful connection to the IoTConnect cloud:

```
Terminal - COM14
File Edit Setup Control Window Help
(00:00:14.709) [wifi_connection_task]: Successfully connected to Wi-Fi network 'Admin'
(00:00:14.711) [wifi_connection_task]: IPv4 Address Assigned: 192.168.1.16
(00:00:16.685) [sensor_task]: -----
(00:00:16.685) [sensor_task]: CO2:           :      438 ppm
(00:00:16.689) [sensor_task]: Pressure    : 1002.50 mbar
(00:00:16.695) [sensor_task]: Temperature: 28.82 *C
(00:00:16.701) [sensor_task]: -----
Obtaining network time.....
Time received from NTP. Time now: 2023-05-21T08:36:05.000Z!
(00:00:16.715) [wifi_connection_task]: Initializing IoTConnect SDK...
Discovery response parsing successful.
Sync response parsing successful.
CPID: c080***
ENV: avnetpoc
MQTT connection successful.
```

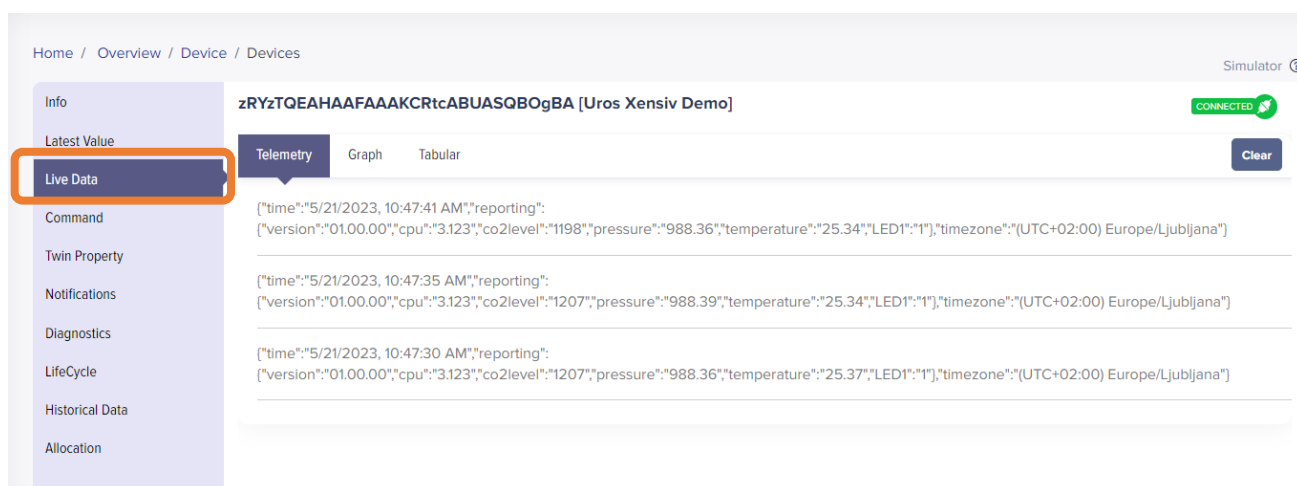
From the IoTConnect cloud backend, go to **Device** list



click your device **Unique Id** in the **Devices** menu to access its content.

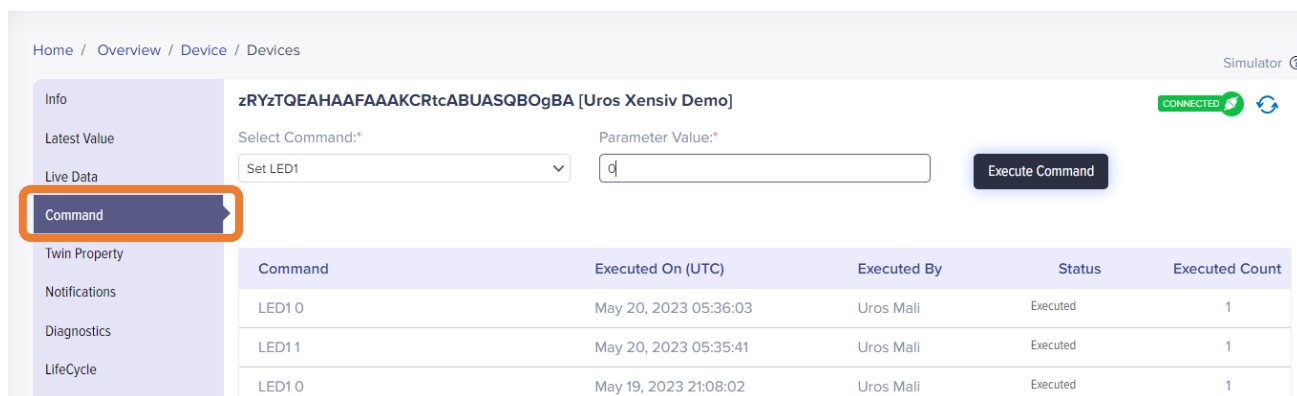


Then click **Live Data** tab to see the incoming raw JSON stream from the device:



Next, click **Command** tab to send a command to the device. The template defined 1 “Set board LED” command, allowing to set the status of user LEDs on the board.

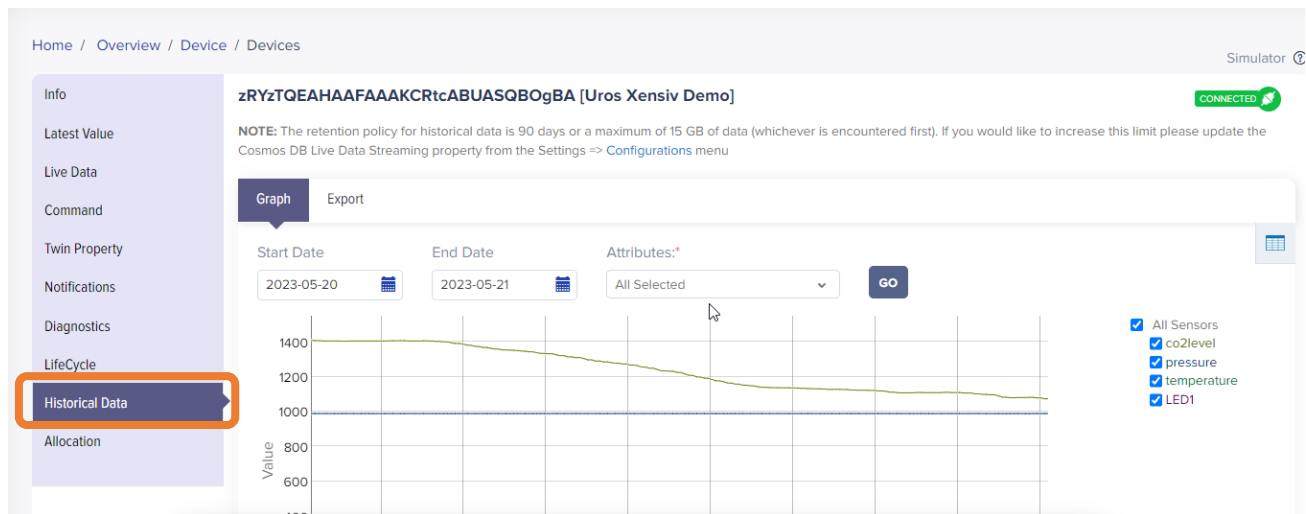
Select “Set LED1” command with parameter **0** or **1** to light the LED on or off, then press the **Execute Command** button.



The corresponding board LED should change its status accordingly within a second.



Click **Historical Data** tab to see all the data that was log for this device. The default retention policy allows for 90 days or a maximum of 15 GB of data (whichever is encountered first). Select **Attributes→All** and click **Go**. You should see chart with values starting to display.



Any area of the graph can be zoomed in by simply selecting the desired portion of the graph. Then double click the graph to zoom out and get back to the specified date interval. The date interval **Start Date** and **End Date** can be used to render extended time interval.

Enabling or disabling specific sensor data rendering is possible by clicking the corresponding sensor checkbox on the right side.



Task accomplished: The device is securely connected to the IoTConnect cloud using TLS certificate based mutual authentication with custom PKI. Also, the device private key is stored inside the secure element to ensure highest security level. Device access and remote configuration is easily managed through the IoTConnect backend portal.



15. Task 9: Remote access from EBV Mobile App

15.1. Overview

EBV Elektronik has designed a Mobile Application compatible with IoTConnect that allows the end user to access his own account at the IoTConnect OEM cloud space, visualize data and interact with the device. The source code for this mobile application is provided for free as an example and can be found in the provided training material.

From the previous tasks, device has been onboarded to the IoTConnect OEM cloud account. The device secure element is used as a unique identifier for the device in the cloud and attests the authenticity of the device. It also allows a better control of the OEM supply chains as devices can be populated in the OEM cloud early on during production stage. Overbuilding or cloning devices becomes impossible thus efficiently protecting OEM's revenue and reputation.

The IoTConnect cloud topology allows to have devices assigned to entities within the OEM cloud account. This feature is used to support an unlimited number of end-users and devices. Therefore, each end-user will have a dedicated entity, ensuring end-user data and information privacy with a login to this entity.

The EBV Mobile App makes the process completely transparent and greatly simplifies the workload for both OEM and end-users.

15.2. Step by step guide



To Do: The task does not involve any software development. Instead, it is only required to process the steps in the right order and to understand their underlying effects.

15.2.1 Install the EBV Mobile App on your smartphone

The mobile application is available for **iOS** and **Android** platforms. Open a browser on your smartphone to one of the links below with respect to your smartphone operating system:



iOS:

https://install.appcenter.ms/orgs/softwebsolutions/apps/ebv_iot-ios/distribution_groups/dev

Android:

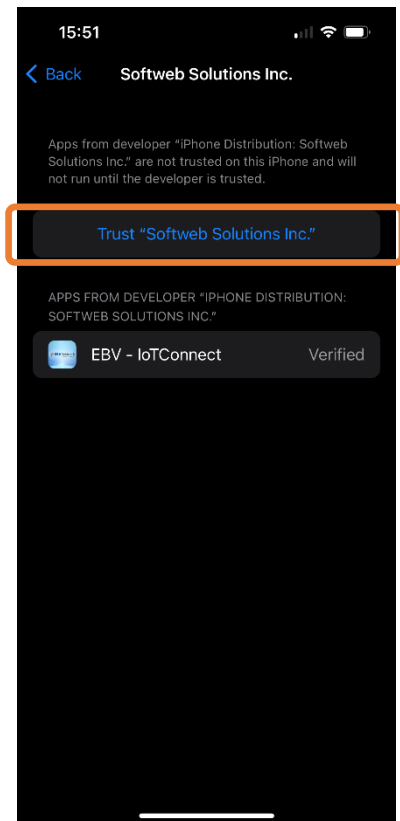
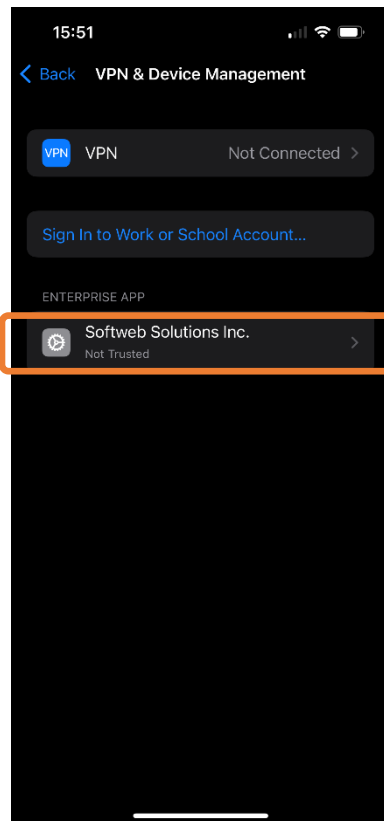
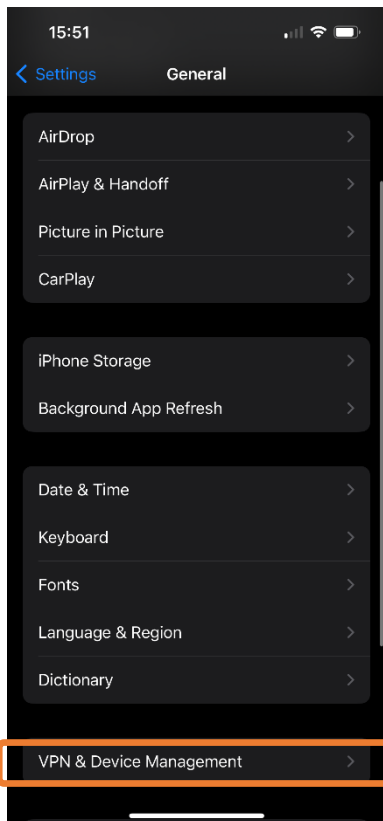
https://install.appcenter.ms/orgs/softwebsolutions/apps/ebv_iot-android/distribution_groups/dev

Click **Download** to download the application and **Install** it on your phone.

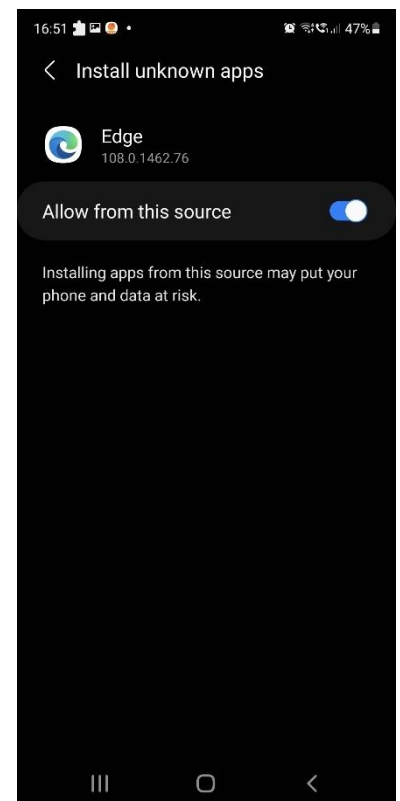
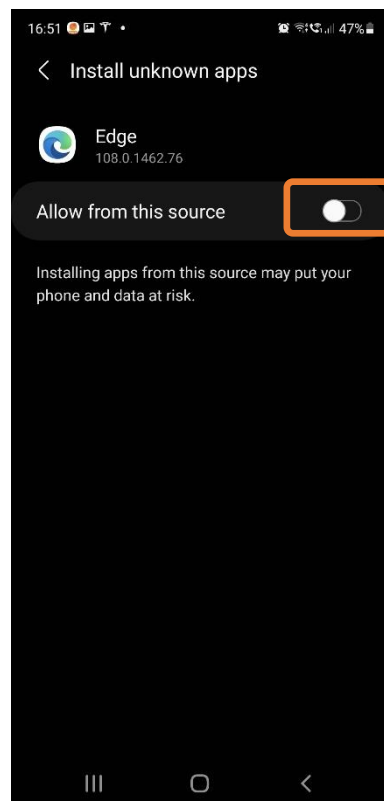
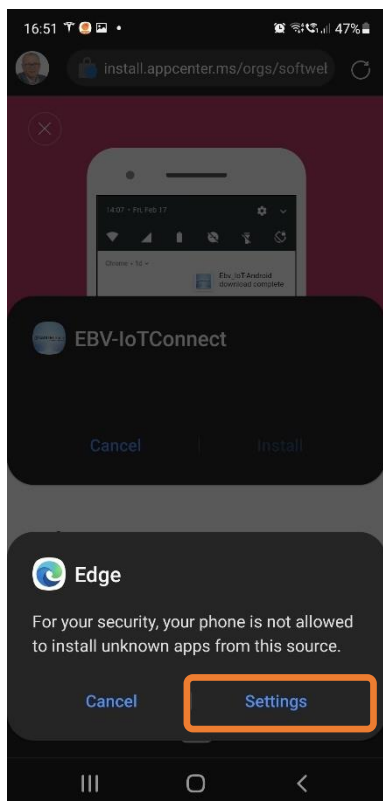
Since the app is not yet published on the application store, user will have to trust the application manually before running the app.



For iOS: go to **Settings** → **General** → **VPN & Device Management**, then select **Softweb Solutions Inc.** and click **Trust "Softweb Solutions Inc."**.

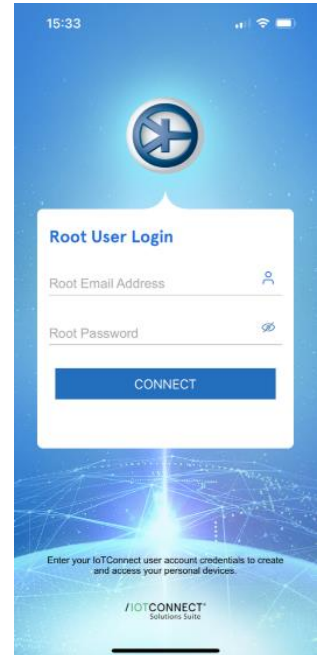


For Android: the settings can be directly accessed via a browser popup, click **Settings** and then enable **Allow from this source**:



15.2.2 Open the app and connect to your IoTConnect OEM root account

The first step is to run the mobile app to connect to the IoTConnect OEM root cloud account. This step would typically not happen on a production application as the app would embed credentials with limited accesses to allow connection to the OEM cloud account, to create end user account and to claim devices. This app is provided as a demonstrator; therefore, it is designed to allow connecting to any IoTConnect OEM cloud account as long as root user credentials are provided. The root user credentials were obtained when registering for IoTConnect access in sub-section **2.3 Create IoTConnect cloud account** of this document or check for “**Owner**” role in your users list at your instance of IoTConnect cloud.



Enter **Root Email Address** and **Root Password**.

Click **Connect**.



Tip: You can continue logging in using **root account** where all created devices from your instance of the IoTConnect cloud will be displayed. Skip following step and continue at sub-section **15.2.5 Device list and menu overview**

15.2.3 End-user signs up to the OEM cloud account



Note: In a real production application, below screenshot would be the first screen seen by end-user where he would need to provide his user cloud account credentials or sign-up as a new user.

Click **Sign-Up**.

The following sign-up form will expect at least the following 4 mandatory fields to be provided:

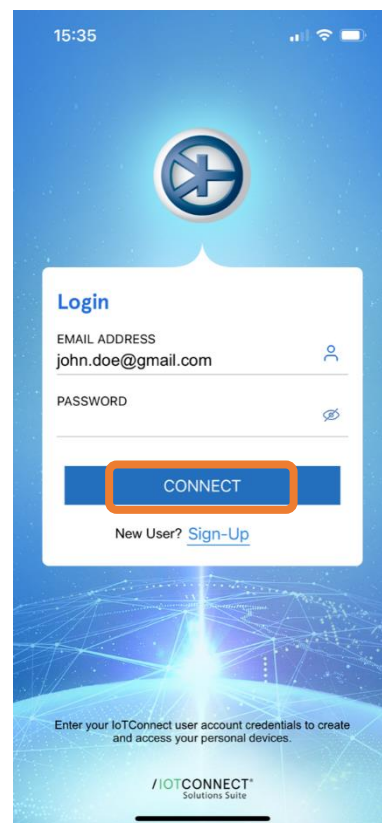
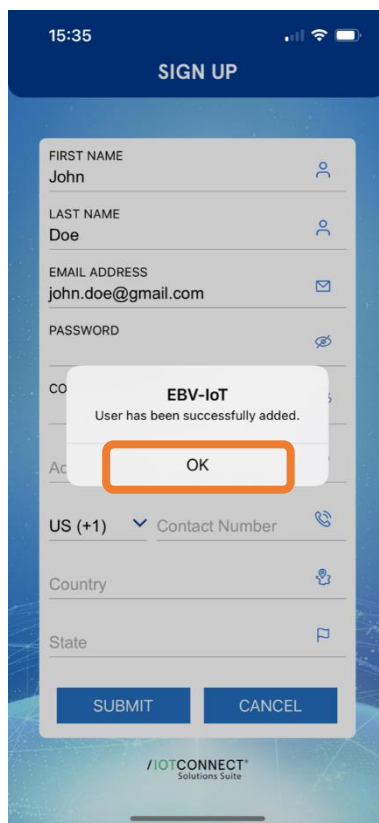
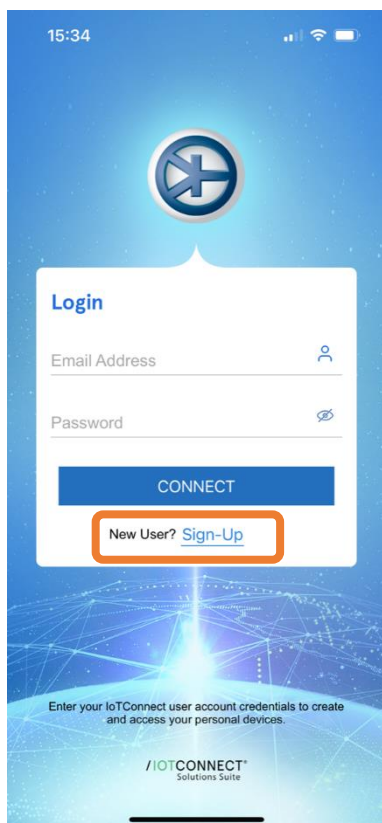
- User first name.
- User last name.
- User email address.
- User password.



Note: At the time of writing this document, the app will not perform any email verification, but it is recommended to choose a valid email address, so credentials can be retrieved if lost.

Click **Submit**.





After clicking the confirmation popup, user is sent back to the login screen where he can enter his credentials.

Click **Connect**.

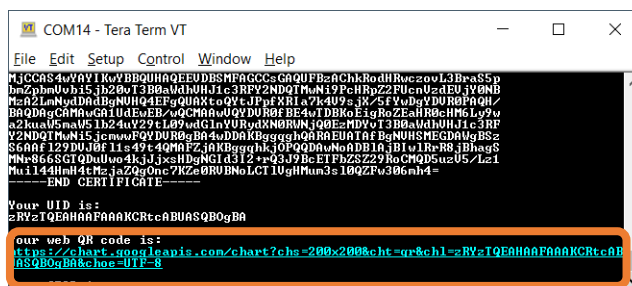
15.2.4 Add a device to the end-user cloud account

In the IoTConnect cloud ecosystem, devices are not created at this stage since device resource creation happened during production stage. Instead, device ownership is simply transferred from the OEM root entity to the end-user entity. The end-user entity is internally identified using the end-user email address to guarantee uniqueness.

From IoTConnect perspective, devices are uniquely identified using the unique ID of the secure element. While this information could be manually entered by the end-user, it is much easier to use a QR code.



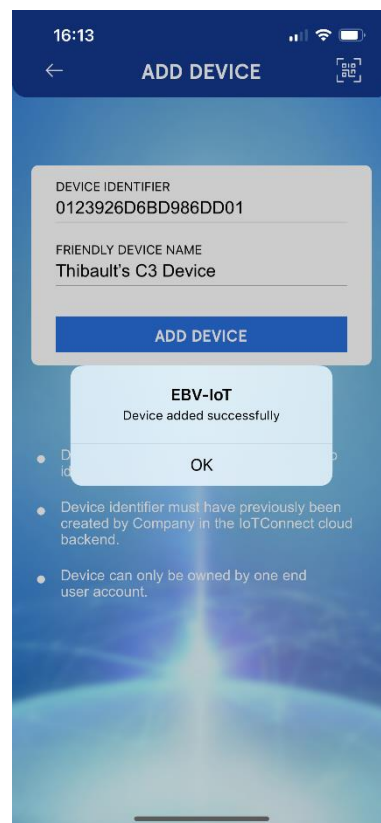
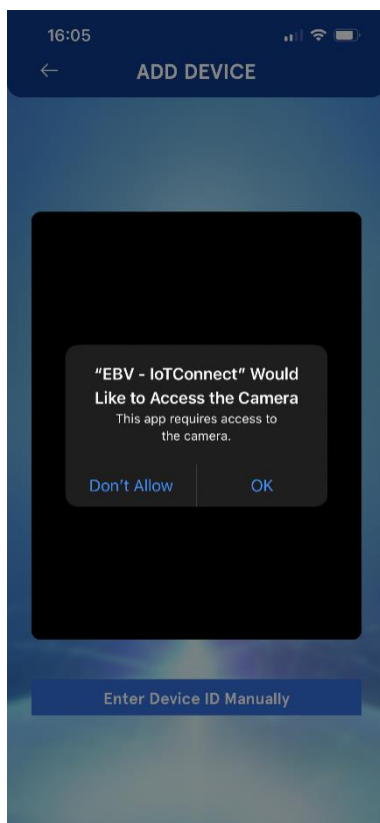
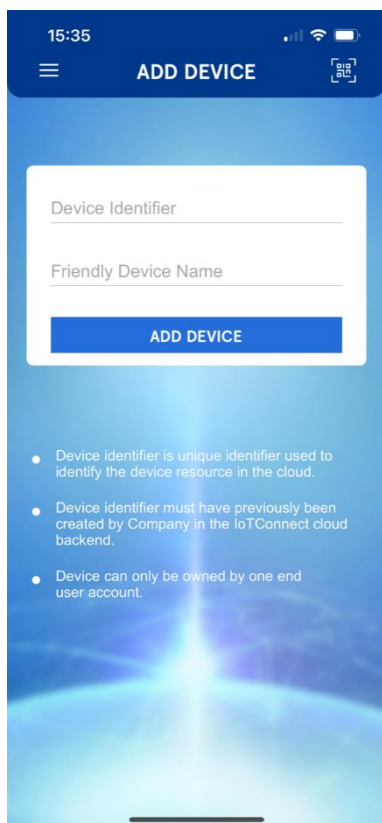
Note: The QR code has been generated in the sub-section **9.2.4 Get QR code for seamless device onboarding** and should be there in the “Scripts” folder. If for whatever reason do not have access to the QR code use alternative approach. The HTTPS line with device QR code is printed in the terminal output after device reset or retrieve the link through terminal access as described in **13 Task 7: Accessing the device** through serial terminal. Double click the link or use the link to render the QR code in a browser.



Click **QR code icon** from the on top right corner, **OK** to allow access to the camera, and scan the device QR code.



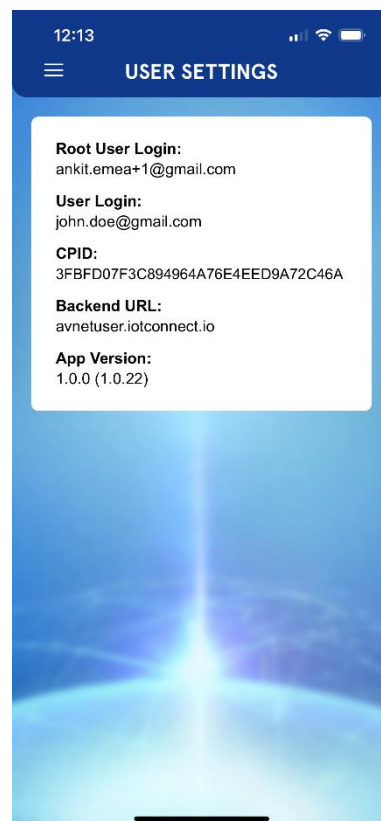
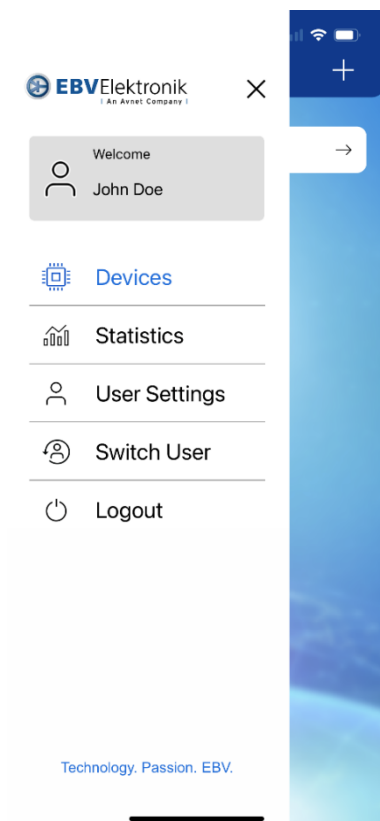
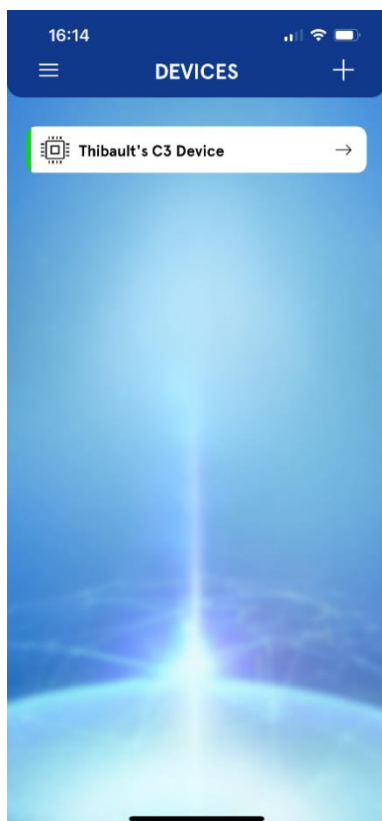
Enter a device friendly name, then click **Add Device**.



15.2.5 Device list and menu overview

The end-user device is now visible in the EBV Mobile App and can be accessed by clicking the device friendly name.

Additionally, the top left button provides access to the app main menu:



The app main menu provides useful information and actions to the end-user:

- **Devices**: access the device list view.
- **Statistics**: display some metrics about the current activity on the OEM cloud.
- **User Settings**: provides a summary of the login information that is used to connect the EBV Mobile App to the IoTConnect OEM cloud.
- **Switch User**: allow switching between user accounts without losing current user account setup.
- **Logout**: clear both root and user credentials. This would essentially reset the app to a similar state after a fresh install.



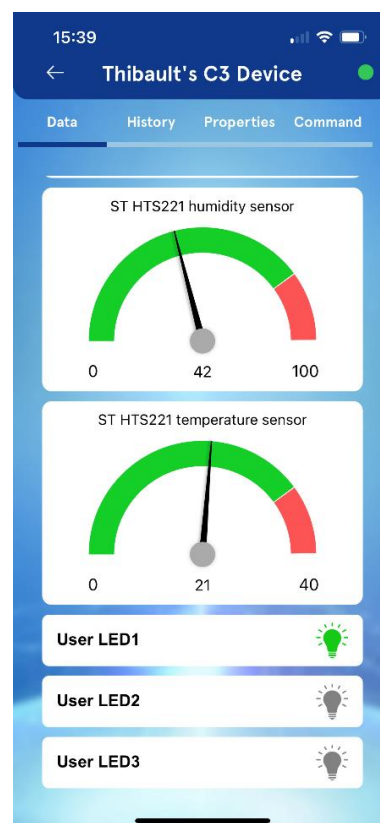
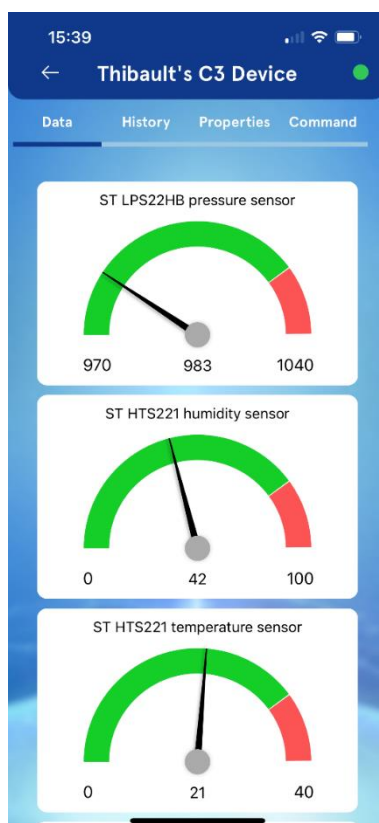
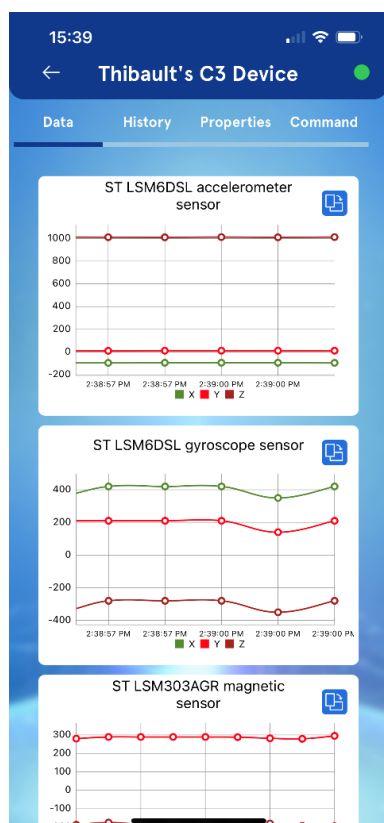
Note: The menu options are quite simple and limited as this app is only intended as a demonstrator app.

15.2.6 Interacting with the device

From the device list screen, just click on the device friendly name to access its content. The data carried out by the device is represented by a set of widgets, such as graphs, gauges, sliders, LEDs and so on. The widget type used to represent the data is defined in the device template file that was uploaded to the IoTConnect cloud backend in the previous tasks.



Note: The look and feel of the widgets representation may vary among various implementations.



LED widgets are interactive and can be clicked to toggle ON or OFF the corresponding device LED.

The upper tabs provide useful information about the selected device:

- **Data**: shows the dynamical view with widgets to render device data.
- **History**: display past user data in a single graph view, where individual sensor data can be enabled or disabled.



- **Properties:** display properties latest status as reported by the device. In this hands-on only the firmware version of the device is reported as “fw” property.
- **Command:** ability to remotely execute commands on the device. This feature is transparently used when pressing the LED widget to toggle the corresponding device LED.



Task accomplished: Congratulations! Device can now be fully operated by the end-user from the EBV Mobile App.



Thank you!

We hope you enjoyed the training. We would like to kindly request just five minutes of your time to participate in a survey related to the training you attended. Your feedback is incredibly valuable to us as we strive to improve our training programs and provide the best possible experience for future participants.

The survey consists of few short questions and is designed to gather insights on your training experience, any suggestions you may have, and areas where we can enhance our services. Your responses will remain confidential.

Please scan or click on the following QR code access the survey:



Thank you in advance for your time and valuable feedback. Should you have any questions or require further assistance, please feel free to reach out to me. Your participation is genuinely appreciated.

For more information, please contact your local EBV elektronik representative or at ssc@ebv.com.



Revision history

Revision	Date	Notes
0.1	April 18 th , 2023	First draft
1.0	May 23 rd , 2023	Initial release
1.1	September 8 th , 2023	Minor corrections, removed source changes, update for MTB 3.1,
1.2	September 14 th , 2023	Separated Prerequisites chapter
1.3	November 30 th , 2023	“Full” vs. “Quick start” version separation/paths added
1.4	January 25 th , 2024	Updated template (certificate type and position)



EBV European Headquarters

EBV Elektronik GmbH & Co. KG | D-85586 Poing | Im Technologiepark 2-8 | Phone: +49 (0)8121 774-0 | www.ebv.com

EBV Regional Offices | Status July 2022

AUSTRIA

1120 Wien
Grünbergstraße 15/1, 4. Stock
Phone: +43 1 89152 0
Fax: +43 1 89152 30

BELGIUM

1831 Diegem
De Kleetlaan 3
Phone: +32 2 716001 0
Fax: +32 2 72081 52

BULGARIA

1505 Sofia
48 Sitnyakovo Blvd., Serdika
offices, 10th floor, Unit 1006
Phone: +359 2 9264 337
Fax: +359 2 9264 133

CZECH REPUBLIC

18600 Prague
Amazon Court, Karolinska 661/4
Phone: +420 2 34091 011
Fax: +420 2 34091 010

DENMARK

Elkjærvej 19, 1 sal
DK-8230 Åbyhøj
Phone: +45 8 6250 466
Fax: +45 8 6250 660

ESTONIA

80042 Pärnu
Suur-Jõe 63
Phone: +372 5 8864 446

FINLAND

02240 Espoo
Klovinpellontie 1-3, 6th floor
Phone: +358 9 2705279 0
Fax: +358 9 27095498

FRANCE

91300 Massy Cedex (Paris)
Le Copernic bât B
12 rue Jean Bart
Phone: +33 1 644729 29

35700 Rennes
16, Rue de Jouanet
Phone: +33 2 998300 51
Fax: +33 2 998300 60

67400 Illkirch Graffenstaden
35 Rue Gruninger
Phone: +33 3 904005 92
Fax: +33 3 886511 25

31500 Toulouse
8 chemin de la terrasse
Parc de la plaine
Phone: +33 5 610084 61
Fax: +33 5 610084 74

69693 Venissieux (Lyon)
Parc Club du Moulin à Vent
33, Av. du Dr. Georges Lévy
Phone: +33 4 727802 78
Fax: +33 4 780080 81

GERMANY

85609 Aschheim-Dornach
Einsteinring 1
Phone: +49 89 388 882 0
Fax: +49 89 388 882 020

10553 Berlin
Kaiserin-Augusta-Allee 14
Phone: +49 30 747005 0
Fax: +49 30 747005 55

31275 Lehrte
Gaußstr. 10
Phone: +49 5139 8087 0
Fax: +49 5139 8087 70

59439 Holzwickede
Wilhelmstraße 1
Phone: +49 2301 94390 0
Fax: +49 2301 94390 30

41564 Kaarst
An der Gumpgesbrücke 7
Phone: +49 2131 9677 0
Fax: +49 2131 9677 30

71229 Leonberg
Neue Ramtelstraße 4
Phone: +49 7152 3009 0
Fax: +49 7152 759 58

90471 Nürnberg
Lina-Ammon-Straße 19B
Phone: +49 911 817669 0
Fax: +49 911 817669 20

04435 Schkeuditz
Frankfurter Straße 2
Phone: +49 34204 4511 0
Fax: +49 34204 4511 99

78048 VS-Villingen
Marie-Curie-Straße 14
Phone: +49 7721 99857 0
Fax: +49 7721 99857 70

65205 Wiesbaden
Borsigstraße 36
Phone: +49 6122 8088 0
Fax: +49 6122 8088 99

HUNGARY

1117 Budapest
Budafoki út 91-93, West Irodaház
Phone: +36 1 43672 29
Fax: +36 1 43672 20

ISRAEL

4581500 Bnei Dror
Tirosh 1
Phone: +972 9 77802 60
Fax: +972 3 76011 15

ITALY

20095 Cusano Milanino (MI)
Via Alessandro Manzoni, 44
Phone: +39 02 660962 90
Fax: +39 02 660170 20

50019 Sesto Fiorentino (FI)
Via Lucchese, 84/B
Phone: +39 05 543693 07
Fax: +39 05 542652 40

41126 Modena (MO)
Via Scaglia Est, 31
Phone: +39 059 292 4211
Fax: +39 059 292 9486

00155 Roma (RM)
Via de Settebagni, 390
Phone: +39 06 4063 665/789
Fax: +39 06 4063 777

35030 Sarmeola di Rubano (PD)
Piazza Adelaide Lonigo, 8/11
Phone: +39 049 89747 01
Fax: +39 049 89747 26

10144 Torino (TO)
Via Treviso, 16
Phone: +39 011 26256 90
Fax: +39 011 26256 91

IRELAND

Fitzwilliam Hall
Fitzwilliam Place
Dublin 2
D02 T292
Phone: +353 1 4097 802
Fax: +353 1 4568 544

NETHERLANDS

Zonnebaan 9
3542 EA Utrecht
Phone: +31 346 5830 10
Fax: +31 346 5830 25

NORWAY

1181 Oslo
Brannfjellveien 11
Phone: +47 22 67 17 80
Fax: +47 22 67 17 89

POLAND

80-838 Gdansk
Targ Rybny 11/12
Phone: +48 58 30781 00

P02-676 Warszawa
Postępu 14
Phone: +48 22 209 88 05

50-062 Wrocław
Pl. Solny 16
Phone: +48 71 34229 44
Fax: +48 71 34229 10

PORTUGAL

4400-676 Vila Nova de Gaia Unipessoal
LDA / Edifício Tower Plaza
Rotunda Eng. Edgar Cardoso, 23 - 14th G
Phone: +351 22 092026 0
Fax: +351 22 092026 1

ROMANIA

020334 Bucharest
4C Gara Herastrau Street
Building B, 2nd Floor - 2nd District
Phone: +40 21 52816 12
Fax: +40 21 52816 01

RUSSIA

620028 Ekaterinburg
Tatischeva Street 49A
Phone: +7 343 31140 4
Fax: +7 343 31140 46

127486 Moscow
Korovinskoye Shosse 10,
Build 2, Off. 28
Phone: +7 495 730317 0
Fax: +7 495 730317 1

197374 St. Petersburg
Atlantic City, Savushkina str 126,
lit B, premises 59-H, office 17-2
Phone: +7 812 635706 3
Fax: +7 812 635706 4

SERBIA

11070 Novi Beograd
Milentija Popovica 5B
Phone: +381 11 40499 01
Fax: +381 11 40499 00

SLOVAKIA

82109 Bratislava
Turčianska 2 Green Point Offices
Phone: +421 2 321114 1
Fax: +421 2 321114 0

SLOVENIA

1000 Ljubljana
Dunajska cesta 167
Phone: +386 1 5609 778
Fax: +386 1 5609 877

SOUTH AFRICA

7700 Rondebosch, Cape Town
Belmont Office Park, Belmont Road
1st Floor, Unit 0030
Phone: +27 21 402194 0
Fax: +27 21 4196256

3629 Westville
Forest Square, 11 Derby Place
Suite 4, Bauhinia Building
Phone: +27 31 27926 00
Fax: +27 31 27926 24

2128 Rivonia, Sandton
Johannesburg
33 Riley Road
Pinewood Office Park
Building 13, Ground Floor
Phone: +27 11 23619 00
Fax: +27 11 23619 13

SPAIN

08014 Barcelona
c/Tarragona 149 - 157 Planta 19^a
Phone: +34 93 47332 00
Fax: +34 93 47363 89

39005 Santander (Cantabria)
Racing n° 5 bajo
Phone: +34 94 22367 55
Phone: +34 94 23745 81

28760 Tres Cantos (Madrid)
c/Ronda de Poniente 14 - 2^a planta
Phone: +34 91 80432 56
Fax: +34 91 80441 03

SWEDEN

16440 Kista
Isafjordsgatan 32B, Floor 6
Phone: +46 859 47023 0
Fax: +46 859 47023 1

SWITZERLAND

8953 Dietikon
Bernstrasse 394
Phone: +41 44 74561 61
Fax: +41 44 74561 00

TURKEY

06520 Ankara
Armada Is Merkezi
Eskisehir Yolu No: 6, Kat: 14
Ofis No: 1406, Sogutozu
Phone: +90 312 2956 361
Fax: +90 216 528831 1

34774 Ümraniye / Istanbul
Tatlisu Mahallesi Pakdil Sokak 7
Phone: +90 216 528831 0
Fax: +90 216 528831 1

35580 Izmir
Folkart Towers
Manas Blv. No 39 B Blok
Kat: 31 Ofis: 3121
Phone: +90 232 390 9196
Fax: +90 216 528831 1

UKRAINE

03040 Kiev
Vasilivskaya str. 14
off. 422-423
Phone: +380 44 496222 6
Fax: +380 44 496222 7

UNITED KINGDOM

Maidenhead (South)
Berkshire, SL6 7RJ
2, The Switchback
Gardner Road
Phone: +44 16 28778556
Fax: +44 16 28783811

Manchester (North)
M22 5WB
Manchester International Office Centre
Suite 3E (MIOC) Styal Road
Phone: +44 16 149934 34
Fax: +44 16 149934 74

