



ANR023 CALYPSO

SECURE CLOUD CONNECTIVITY USING CALYPSO Wi-Fi MODULE

VERSION 1.0

JUNE 10, 2021

Revision history

Document version	Notes	Date
1.0	<ul style="list-style-type: none">• Initial version	June 2021

Contents

1 IoT application	4
1.1 Introduction	4
1.2 The IoT stack	4
1.3 IIoT application example	6
1.4 System design using Calypso and cloud platforms	7
1.4.1 Embedded design	8
1.4.2 Cloud platform design	9
1.5 Building a prototype application	11
2 MQTT	12
2.1 Publish/Subscribe	12
2.2 Topics/Subscriptions	12
2.3 QoS	13
2.4 Message persistence	13
2.5 Last Will and Testament	13
2.6 Security	13
2.7 MQTT on cloud platforms	13
3 Sensor-to-cloud application prototyping	14
3.1 WE FeatherWings	14
3.2 Prerequisites	15
3.3 Set-up	15
3.4 Installation of tools	16
3.5 Configure and run the code	16
3.6 Sensor data in the cloud	18
4 Summary	20
5 References	21
6 Important notes	22
6.1 General customer responsibility	22
6.2 Customer responsibility related to specific, in particular safety-relevant applications	22
6.3 Best care and attention	22
6.4 Customer support for product specifications	22
6.5 Product improvements	23
6.6 Product life cycle	23
6.7 Property rights	23
6.8 General terms and conditions	23
7 Legal notice	24
7.1 Exclusion of liability	24
7.2 Suitability in customer applications	24
7.3 Trademarks	24
7.4 Usage restriction	24
8 License terms	26
8.1 Limited license	26
8.2 Usage and obligations	26

8.3	Ownership	27
8.4	Firmware update(s)	27
8.5	Disclaimer of warranty	27
8.6	Limitation of liability	27
8.7	Applicable law and jurisdiction	28
8.8	Severability clause	28
8.9	Miscellaneous	28

1 IoT application

1.1 Introduction

The Internet of Things (IoT) can be broadly defined as an umbrella term for a range of technologies that enable devices to connect and interact with each other. Interconnected devices generating data provide for a range of new applications. Industrial automation, healthcare, smart home, smart cities, smart grids and smart farming are some of the areas in which IoT provides substantial benefits.

Dubbed the "fourth industrial revolution" or Industry 4.0, the Industrial IoT (IIoT) is the digitization of industrial assets and processes that connects products, machines, services, locations/sites to workers, managers, suppliers, and partners. Closer networking of the digital world with the physical world of machines helps achieve higher productivity, safety, efficiency and sustainability.

The core task of any IoT solution is to get data from the field to the cloud where analysis of the same generates the desired value addition for the application. This application note aims to propose an elegant solution to achieve this task based on Calypso Wi-Fi module.

This chapter begins by describing the parts of a typical IoT application. Further, an application scenario is discussed with an example. Finally, a sensor-to-cloud IoT solution is presented using tools from Würth Elektronik eiSos. Specifically, a step-by-step description is presented to get sensor data over Wi-Fi into the most popularly used cloud platforms like Amazon AWS and the Microsoft Azure.

1.2 The IoT stack

Irrespective of the area of application, an end-to-end IoT solution consists of the following components (figure 1).

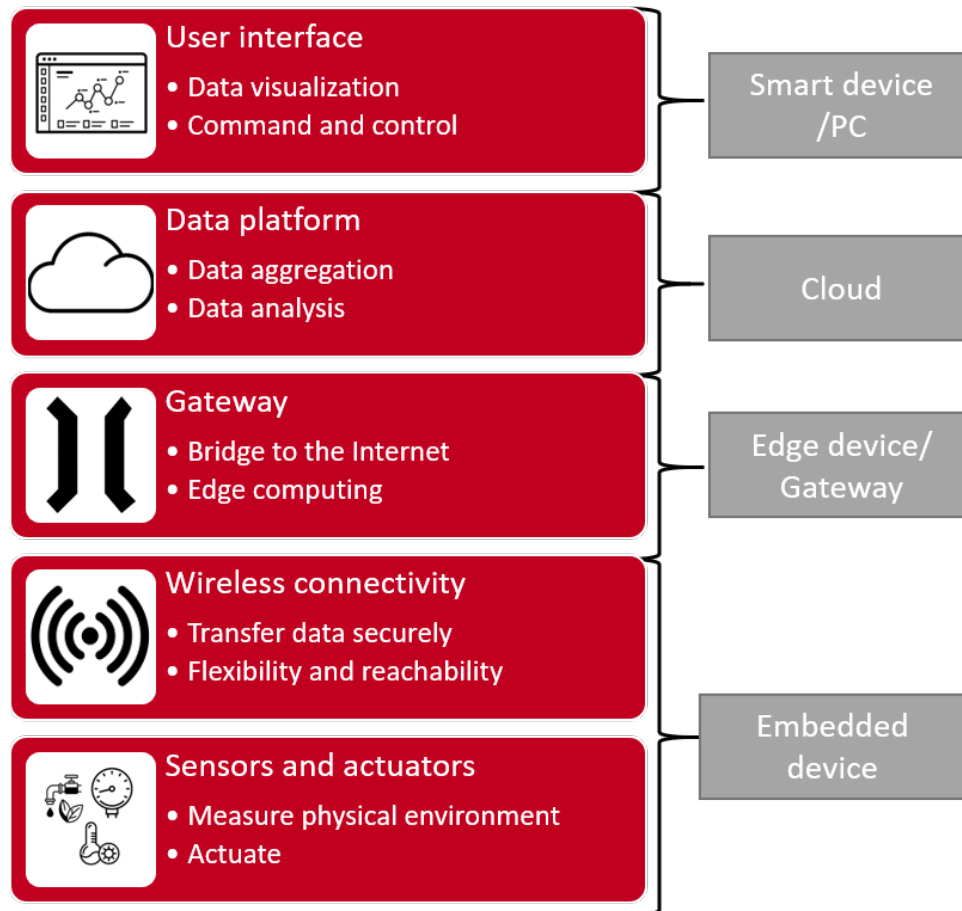


Figure 1: IoT application stack

- **Sensors and actuators:** This is a part of the system that directly interfaces with the physical environment. Sensors measure the state of the environment and interpret the same as analog or digital data. On the other hand, actuators activate a physical change in the measured environment. Advances in the field of electronics in general and semiconductors in particular has led to availability of a wide range of sensors and actuators which are highly efficient and yet very compact.
- **Wireless connectivity:** Sensors and actuators are typically installed in devices with limited access to the digital world. Consider, for example, a temperature sensor mounted inside an industrial boiler. Wireless connectivity provides in addition to many advantages the reachability necessary for such applications. A wide variety of standards as well proprietary wireless connectivity solutions are available today. A number of factors including range, throughput, spectrum regulations, local statutory requirements and power budget determine the choice of wireless connectivity solution.

Modern embedded designs usually combine the above components into a single device (node) interacting with a gateway.

- **Gateway device:** A gateway device acts as a bridge between the physical and the digital worlds. It interprets the multitude of wireless connectivity protocols, collects the data and forwards the same in a format understood by entities above. In certain applications, the gateway device also performs basic analytics like threshold detection.
- **Data platform:** This is the platform where the data is finally stored and presented for

further analysis. Options here can range from a local database to a cloud server with redundancies. A typical platform consists of the components as shown in figure 2. The data platform enables the use of technologies like Artificial Intelligence (AI) and Machine Learning (ML) to perform advanced data analytics that generates value additions to the application.

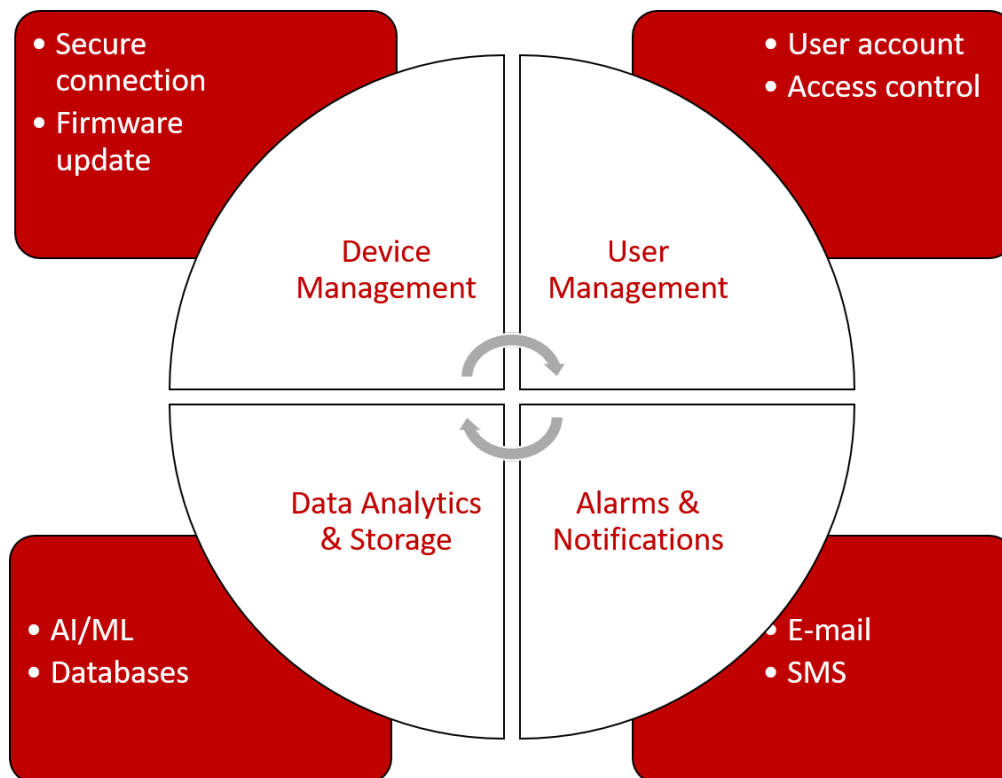


Figure 2: Components of a data platform

- **User interface:** This is the interface between the human users and the digital world. Here the status of the observed environment is presented in a human readable format. The user can take the necessary actions by interacting with this application

1.3 IIoT application example

The IIoT creates a universe of sensors that enables an accelerated deep learning of existing operations that allows rapid contextualization, automatic pattern, and trend detection. This leads to true quantitative capture of qualitative operations resulting in better quality, efficiency, higher safety, lower costs and several other benefits. This has led to the use of IoT in several application use cases.

Remote monitoring and control of production/storage environment (temperature, humidity, pressure etc.) is one of the essential tasks in the industry. Maintaining optimal conditions and automating this process has been a challenge. This use case is considered in this application note and an IoT based approach to solve this problem is presented here.

The above-mentioned task requires sensors/actuators to interact with the environment, a wireless connectivity method, a gateway to collect the data, a cloud platform to store data and a user interface to enable human interaction. The architecture of such a system is as shown in figure 3.

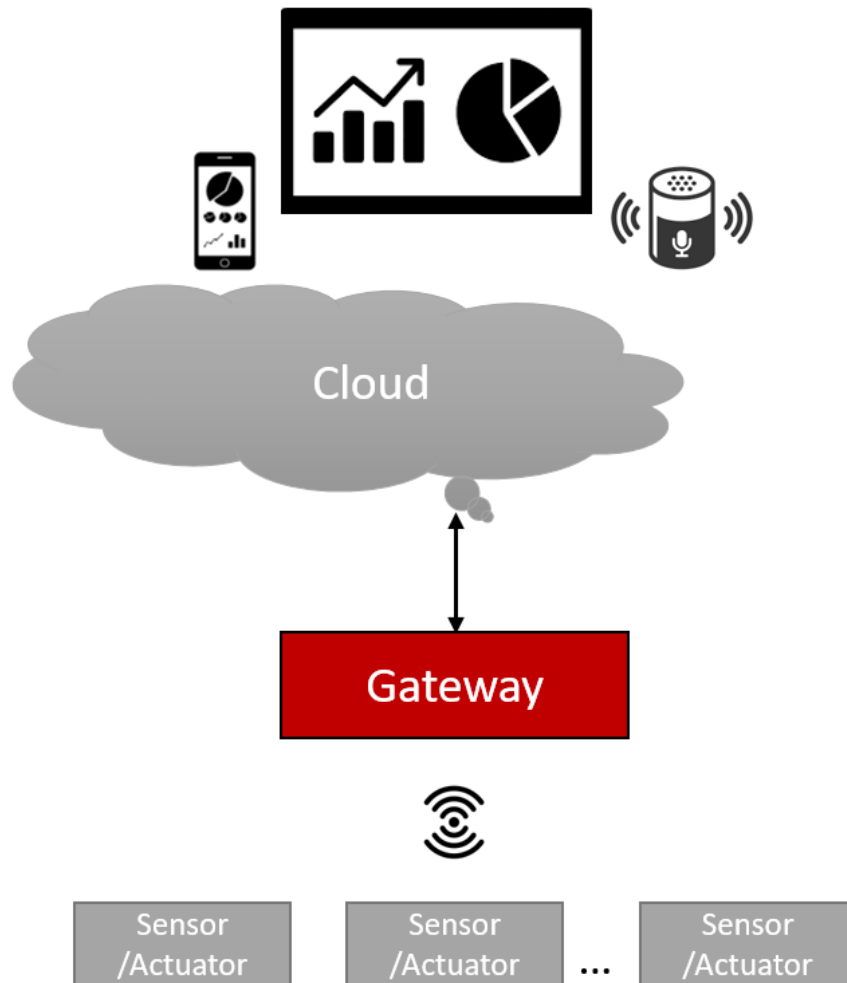


Figure 3: System architecture

1.4 System design using Calypso and cloud platforms

Figure 4 illustrates the design that realizes the architecture described in 1.3. In this example, sensor data is transferred to the cloud platforms using the Wi-Fi connectivity where it is stored and processed.

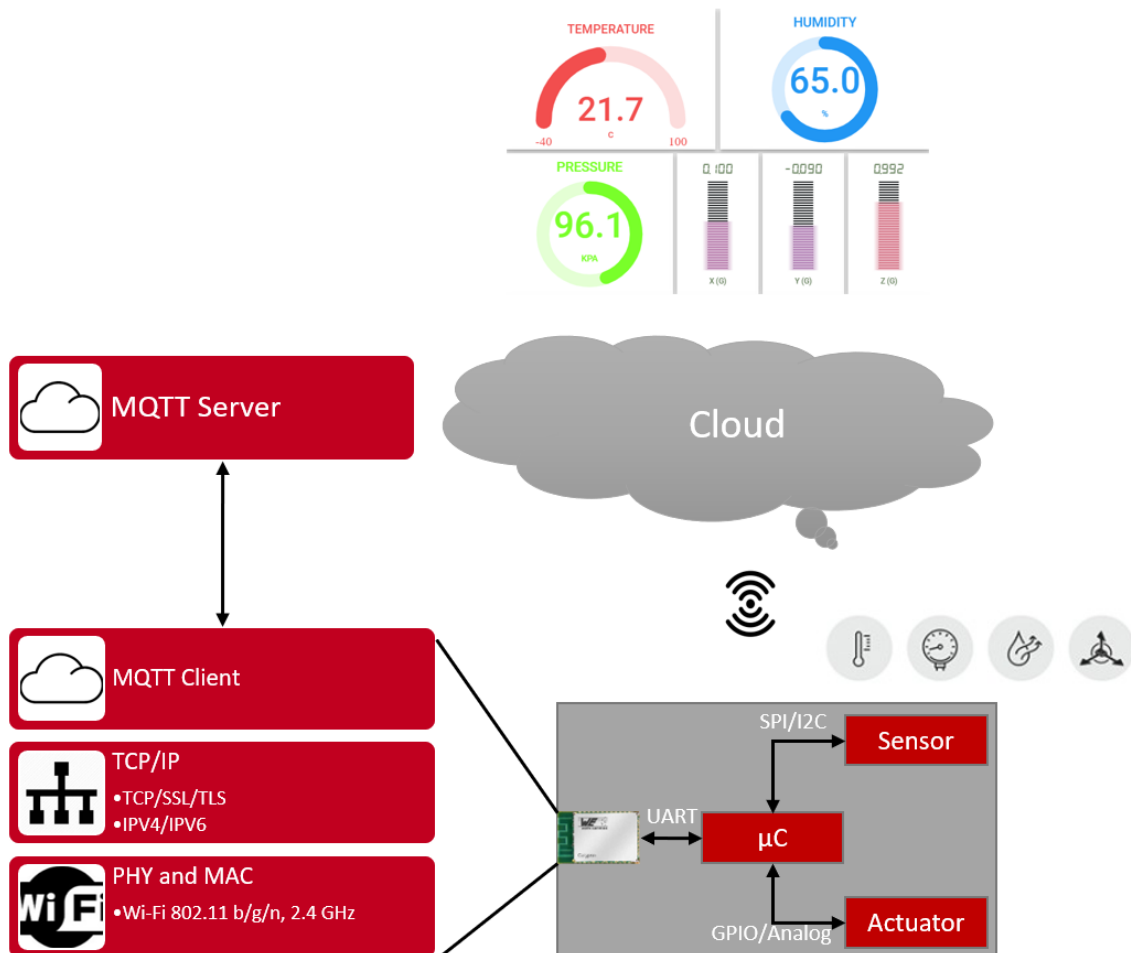


Figure 4: System design

1.4.1 Embedded design

The Calypso is a compact Wi-Fi radio module based on IEEE 802.11 b/g/n (2.4 GHz). With an on-board TCP/IP stack and MQTT protocol implemented out-of-the-box, Calypso acts as the perfect building block for an IoT application.

At the hardware level, a host microcontroller connects and controls sensors/actuators over standard interfaces like I2C, SPI, GPIO or analog. The Calypso uses UART as the primary interface to the Host. Being Wi-Fi compliant, the module can be configured to connect to the local infrastructure network (via an Access Point).

Further, the MQTT-client implemented on-board the Calypso can be configured to connect to a broker/server running in the network. Most of the commonly used cloud platforms today use a type of secure MQTT protocol to connect devices and exchange data. Hence, Calypso provides a direct communication link to the cloud without the need of a gateway device in between.

Here are a few advantages provided by this design approach:

- **Easy hardware integration:** Calypso's compact form factor, edge castellated connection and smart antenna selection allows easy integration into any hardware design. The smart antenna configuration enables the user to use either the on-board PCB antenna or an external antenna. With UART as the standard interface, the Calypso can be interfaced with most of the standard host microcontrollers.

- **Connection to existing network:** Wi-Fi is one of the most commonly used wireless connectivity technologies. Today, most homes already have Wi-Fi infrastructure in use. Hence, integration into the existing wireless network is easier and does not require an extra bridge device.
- **Provisioning into the network:** Any device on deployment needs to be configured to connect to the local network. The Calypso offers a provisioning mode, which allows the user to connect via a smartphone/tablet and configure the device via web pages running on-board.
- **Easy software integration:** Calypso comes with an intuitive AT-style command interface over UART. This allows the host microcontroller to send ASCII based commands to the module to initiate the necessary actions. Additionally, a fully featured TCP/IP stack with several network applications implemented allows the host controller to delegate the complete network handling to the Calypso module.
- **Low-power operation:** Often, the devices deployed in IoT applications are battery operated. Calypso's low power mode allows the module to consume very less current ($< 10 \mu\text{A}$) when the device is not in use. However, features like fast connect and auto connect ensure that the module is up and running in a very short time after wake-up.
- **Security:** One of the major challenges in designing any IoT application is security. Calypso deals with this challenge at several levels. The module itself has a secure bootloader implemented to detect firmware tampering. Wi-Fi compliance ensures conformity to standard Wi-Fi security feature like WPA2 and WPA2 enterprise. Further, at the transport layer, TLS allows authentication as well as end-to-end encryption of data. Any modern application requires secure storage to store sensitive information like credentials, certificate-key pairs etc. Calypso also has an encrypted file system with limited space to enable secure storage. Thus, Calypso provides a very good basis for secure IoT application development.

1.4.2 Cloud platform design

Hundreds of cloud platforms exists in the market today. In this example, the two of the most popular platforms, Amazon Web Services (AWS) and Microsoft Azure are considered. Most cloud platforms are made up of cloud services, which offer specialized functionalities like,

- Device management
- User management
- Data storage
- Data streaming
- Data visualization
- AI/ML
- Security
- Networking
- Billing and cost management

These services act as building blocks for any IoT application. The challenge here is to pick and choose the right services and combine them into a secure, flexible and scalable application that serves the desired purpose.

In this application note, two such combinations are considered keeping in mind the most frequently occurring use cases.

- **Amazon AWS:** Figure 5 shows a sample application using Amazon AWS. In this example, the devices are provisioned to connect securely to the Amazon AWS IoT core. The incoming data is then processed using the AWS Lambda function. Using the thresholds stored in the Dynamo DB, the lambda function detects the thresholds and sends notifications to the user using the notification service (Amazon SNS). Further, the data is streamed to an S3 bucket using the Kinesis Firehose streaming service. The stored data can be used for visualization using Amazon Quicksight service.

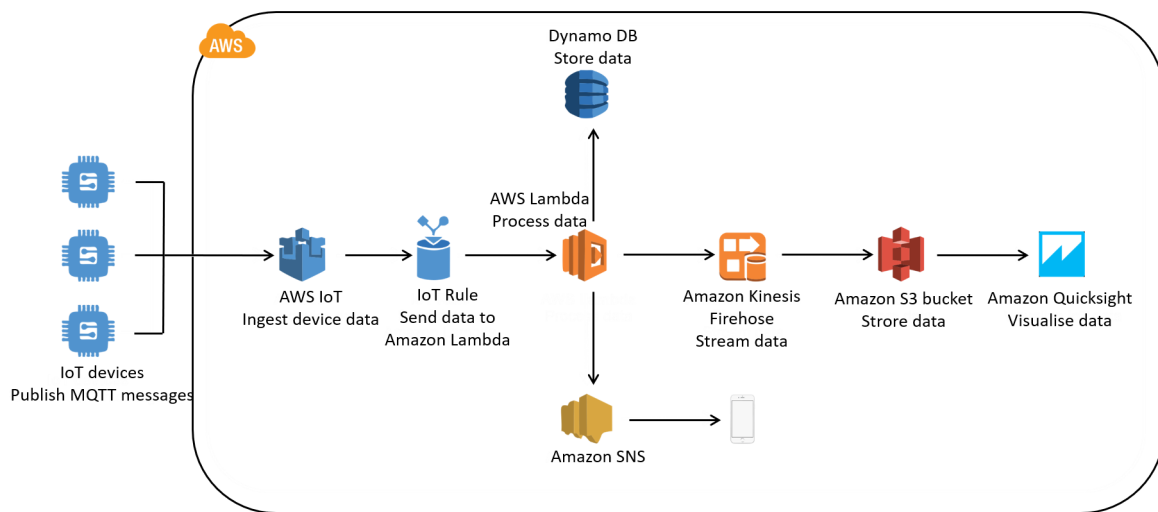


Figure 5: AWS example

- **Microsoft Azure:** Microsoft's Azure IoT hub provides the device management necessary to securely connect IoT devices to the cloud. Once the data is in the cloud it can follow one or a combination of the following paths,
 - Streamed into PowerBI for analytics using the stream analytics service
 - Visualized using a web application
 - Stored in an SQL database and processed using Excel

Figure 6 shows a sample application using Microsoft Azure.

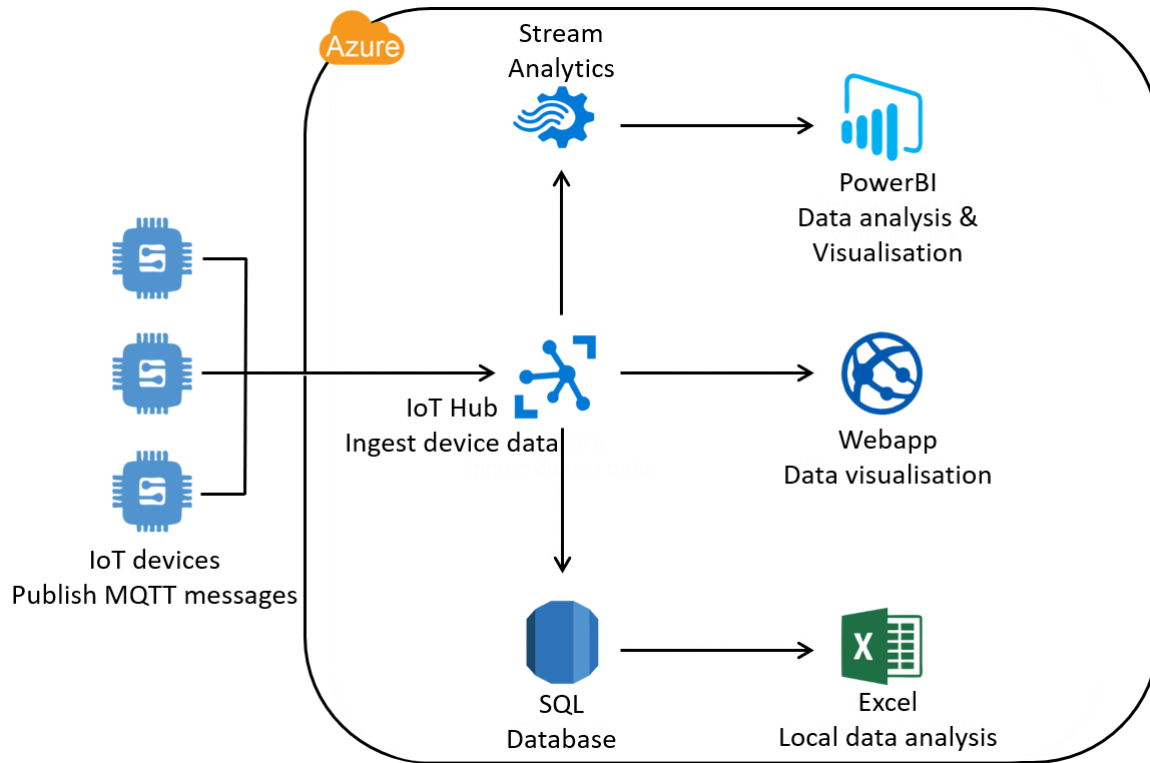


Figure 6: Azure example

1.5 Building a prototype application

Rest of this application note is intended to describe the process of building a prototype for an end-to-end IIoT solution (from sensor to cloud) using tools from Würth Elektronik eiSos. Chapter 2 gives a brief introduction to the MQTT protocol and chapter 3 provides a step-by-step description for building a prototype IoT application including the hardware, firmware and the cloud components.

2 MQTT

MQTT - Message Queuing Telemetry Transport is a lightweight application layer protocol based on a publish/subscribe messaging mechanism. This protocol was designed for resource constrained and unreliable networks with limited bandwidth and high latency. These characteristics make MQTT suitable for low-power, low-bandwidth IoT applications. Inherently, the MQTT protocol offers some degree of assurance of delivery thereby offering the robustness necessary for industrial machine-to-machine communication.

MQTT was originally developed by IBM and the version 3.1.1 is an OASIS standard that is open and royalty-free [3]. It is based on client-server architecture where every client connects to a server (broker) over TCP resulting in a star topology. Once connected, the clients send and receive messages using the publish/subscribe mechanism.

2.1 Publish/Subscribe

Data transfer in MQTT takes place based on publish/subscribe mechanism. The clients connected to the broker can publish messages under certain topics. The clients can also subscribe to topics that are of interest to them. When a client publishes a message on a topic, the broker forwards the message to any client that has subscribed to the topic. This mechanism enables bi-directional communication with an extremely low overhead (2-byte header).

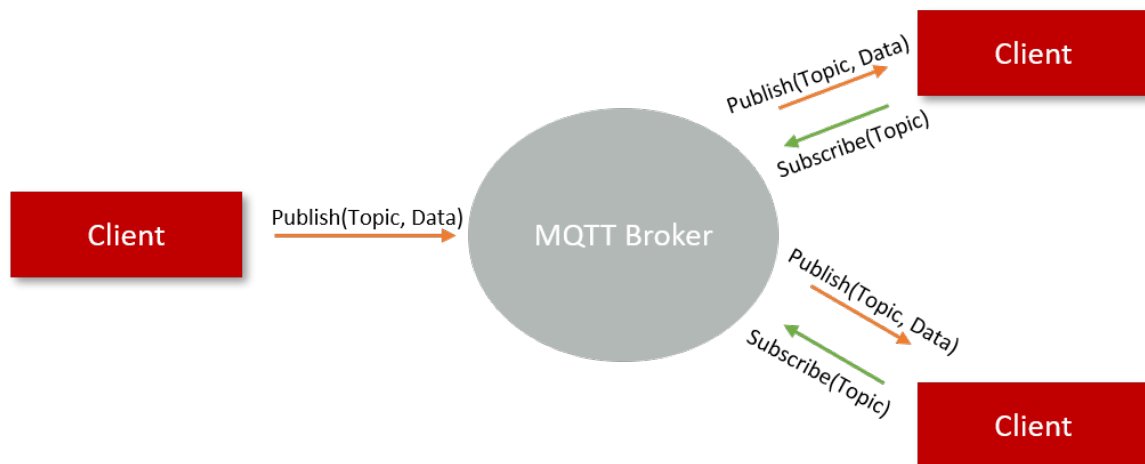


Figure 7: Publish/Subscribe mechanism

2.2 Topics/Subscriptions

Messages in MQTT are always published on topics. A hierarchy can be created in topics using the '/' character. For example, the status of smart light in the living room can have a topic "home/lighting/living_room/light_index".

Clients can create subscription on a topic by explicitly mentioning the topics or by using wildcard characters. There are two wildcards available, '+' and '#'. '+' matches any topics on a single hierarchical level whereas '#' matches any number of levels. For example, subscribing to "home/lighting/+/light_index" would result in getting status change messages of

all lights with "light_index" from all rooms of the house. On the other hand subscribing to "home/lighting/#" results in messages with all lights (all light indices) from all rooms.

This feature makes MQTT modular and highly scalable. Inserting a new node to an existing network does not require a lot of configuration.

2.3 QoS

Based on the requirement of the application one of the following levels of Quality-of-Service (QoS) can be chosen.

- **QoS level 0:** The broker/client delivers the message only once without acknowledgement of reception. The reliability in this case is same as that of the underlying TCP.
- **QoS level 1:** The broker/client delivers the message at least once. In this case an acknowledgement of received packet is done. This case however does not handle duplicate packets.
- **QoS level 2:** The broker/client delivers the message exactly once using a four-step handshake. This in turn offers higher reliability at the cost of lower throughput.

2.4 Message persistence

The publisher can specify if a message published to a topic has to be retained. If marked as retained, the broker retains the message and sends it to all new subscriptions. This acts as the "last known good" mechanism where nodes that come into network do not have to wait long to get the first message.

2.5 Last Will and Testament

This mechanism enables the client to publish one last message to all subscribed clients when it abruptly disconnects from the network. Clients can send a last will and testament message to the broker at any point. If the broker detects that a client has gone offline without a disconnect message, it sends the LWT message on the specified topic. This mechanism is very helpful to detect node failures in due to battery failure or networks outages.

2.6 Security

MQTT offers basic authentication where the client has to send a username and password with the connect message. The broker can authenticate the connection and allow or disallow a client. However, the user has to run MQTT over TLS/SSL to enable end-to-end encryption and advanced client authentication.

2.7 MQTT on cloud platforms

Most cloud platforms support the MQTT protocol although with subtle variations. Both Amazon AWS and Microsoft Azure support only the secure version of MQTT with authentication and end-to-end encryption built in. Please refer to [1] and [2] for more details.

3 Sensor-to-cloud application prototyping

In this chapter, a prototype of a sensor-to-cloud IoT application is presented using the Würth Elektronik eiSos's prototyping tools. The intention here is to go through the process step-by-step to enable the user to replicate the same without much effort. Connection to the two most commonly used cloud platforms, Amazon AWS and Microsoft Azure are presented. However, the same concept can be extended to work with any cloud platform that supports MQTT.

This chapter begins with a short introduction to the Würth Elektronik eiSos FeatherWing line of evaluation boards, which will be used for building this prototype application followed by the detailed description of the prototype application.

3.1 WE FeatherWings

Würth Elektronik eiSos's FeatherWings are a range of development boards that are open source and fully compatible with the Feather form factor. Through these development boards, WE brings a range of wireless connectivity modules, sensors and power modules to the Feather ecosystem.

Adafruit Feather is a complete line of development boards from Adafruit and other developers that are both standalone and stackable. They can be powered by LiPo batteries for on-the-go use or by their micro-USB plugs for stationary projects. Feathers are flexible, portable, and as light as their namesake.

FeatherWings are stacking boards and add functionality and room for prototyping. At its core, the Adafruit Feather is a complete ecosystem of products - and the best way to get your project flying by supercharging your prototyping.

In addition to the hardware, Würth Elektronik eiSos provides a software development kit (SDK) with examples to support all the WE FeatherWings. Here are the salient features of the WE FeatherWing SDK.

- The SDK is open-source and well documented.
- It uses a popular open-source tool chain including an IDE.
- The examples are written in Arduino-styled C/C++ for quick prototyping.
- The core components of the SDK are written in pure C to enable easy porting to any microcontroller platform.
- Development platform independent (Windows, Linux or MAC).
- Modular structure of the software stack makes it easy to integrate into any project.

The SDK can be accessed on Github at <https://github.com/WurthElektronik/FeatherWings>. More information regarding WE FeatherWings can be found under, <https://www.we-online.com/featherwings>

In the following section, a prototype sensor-to-cloud solution is built using some of these FeatherWings.

3.2 Prerequisites

The following hardware is necessary to go through this demo.

1. Adafruit Feather M0 express development board.
2. WE Sensors FeatherWing, WE Calypso Wi-Fi FeatherWing.
3. An IEEE 802.11 b/g/n compatible access point with internet access.
4. A computer to build and modify the code. For the following example, a Windows10 PC is considered.

3.3 Set-up

The main objective here is to create a quick prototype of the system designed in section 1.4. The prototype consists of the following components,

- **Host Microcontroller:** For this prototype the Adafruit Feather M0 express to interface with the sensors, Wi-Fi module and the cloud. At the Feather M0's heart is an ATSAM21G18 ARM Cortex M0 processor, clocked at 48 MHz and at 3.3V logic, the same one used in the new Arduino Zero. This chip has 256K of FLASH 32K of RAM (16x as much). This chip comes with built in USB so it has USB-to-Serial program and debug capability built in with no need for an FTDI-like chip. To make it easy to use for portable projects, a connector is added for any 3.7V Lithium polymer battery and built in battery charging.
- **Sensors:** The Würth Elektronik eiSos Sensor FeatherWing development board is used to monitor the environment. It consists of the following four sensors,
 - WSEN-PADS - Absolute pressure sensor (2511020213301)
 - WSEN-ITDS - 3-axis acceleration sensor (2533020201601)
 - WSEN-TIDS - Temperature sensor (2521020222501)
 - WSEN-HIDS - Humidity sensor (2525020210001)

All four sensors are connected over the shared I²C bus and hence can be connected to any of the Feather microcontroller boards.

- **Wireless connectivity and gateway:** Calypso FeatherWing provides Wi-Fi connectivity as well as MQTT client for cloud connection. The on-board Calypso radio module offers Wi-Fi connectivity based on IEEE 802.11 b/g/n with a fully featured TCP/IP (IPv4 and IPv6) stack. With out-of-the-box support to commonly used network applications like SNTP, HTTP(S), MQTT(S) Calypso offers an easy and secure solution to any IoT application.

It has an AT-style command interface on the standard UART and hence can be connected to any of the Feather microcontroller boards. Advanced security features such as secure boot, secure storage and secure sockets makes this a good basis for secure IoT application.

- **Cloud data platform:** In this example, one of the two of the most commonly used cloud platforms are supported.

- Amazon AWS
- Microsoft Azure

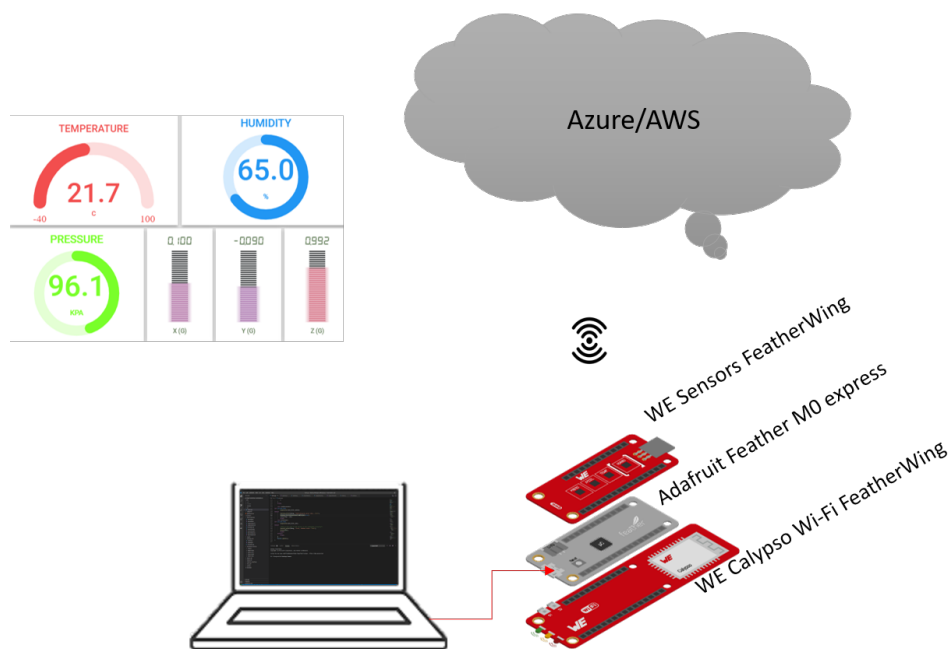


Figure 8: System set-up

3.4 Installation of tools

- Install Visual Studio Code on the computer of your choice following the instructions under <https://code.visualstudio.com/docs>
- Follow the instructions under <https://platformio.org/platformio-ide> to install PlatformIO IDE extension.
- Download or clone the sensor-to-cloud example from <https://github.com/WurthElektronik/FeatherWings>
- Follow the instructions in the respective links to configure the cloud platform
 - **Amazon AWS:**
<https://github.com/WurthElektronik/FeatherWings/Sensor2CloudConnectivity/aws>
 - **Microsoft Azure:**
<https://github.com/WurthElektronik/FeatherWings/Sensor2CloudConnectivity/azure>

3.5 Configure and run the code

1. Prerequisites for running this example:
 - A Wi-Fi access point with WPA2 personal security and a known password and SSID.
 - For Azure, the IoT-Hub has to be configured as described in the previous section.
 - For AWS, the IoT Core has to be configured as described in the previous section.

- ```
// WiFi access point parameters
#define WL_FI_SSID "AP"
#define WL_FI_PASSWORD "pw"
```

- ```
#define AZURE_CONNECTION 1
#define AWS_CONNECTION 0
```

- ```
#define MQTT_CLIENT_ID "we-iot-device"
#define MQTT_SERVER_ADDRESS "we-exampleHub.azure-devices.net"
#define MQTT_PORT 8883
#define MQTT_TOPIC "devices/we-iot-device/messages/events/"
#define MQTT_USER_NAME "example_for_documentation.azure-devices.net/
we-iot-device"
#define MQTT_PASSWORD
"SharedAccessSignature_
"sr=we-iotHub.azure-devices.net%2Fdevices%2Fwe-iot-device&sig=
"example_for_documentation=1111111111"
```

- ```

/*MQTT settings – AWS*/
#define MQTT_CLIENT_ID "we-iot-device-t1"
#define MQTT_SERVER_ADDRESS "example_for_documentation.iot.
amazonaws.com"
#define MQTT_PORT 8883
#define MQTT_TOPIC "test"
#define MQTT_CERT_PATH "cert"
#define MQTT_PRIV_KEY_PATH "key"
#define MQTT_USER_NAME "calypso"
#define MQTT_PASSWORD "secret"

#define MQTT_CERTIFICATE "-----BEGIN_CERTIFICATE-----\n\
11111111111111111111111111111111111111111111111111111111111111111111\
n\
example_for_documentation\n\
-----END_CERTIFICATE-----"

#define MQTT_PRIV_KEY "-----BEGIN_RSA_PRIVATE_KEY-----\n\
11111111111111111111111111111111111111111111111111111111111111111111\
n\
example_for_documentation\n\
-----END_RSA_PRIVATE_KEY-----"

#endif

```

6. Configure SNTP server parameters in "main.cpp".

```
// SNTP settings
#define SNTP_TIMEZONE "+60"
#define SNTP_SERVER "0.de.pool.ntp.org"
```

7. Build and upload the code from the PlatformIO tab as shown in the figure 9.

8. After successful upload, click on "Monitor" in the PlatformIO extension tab to view the debug logs in the serial terminal.

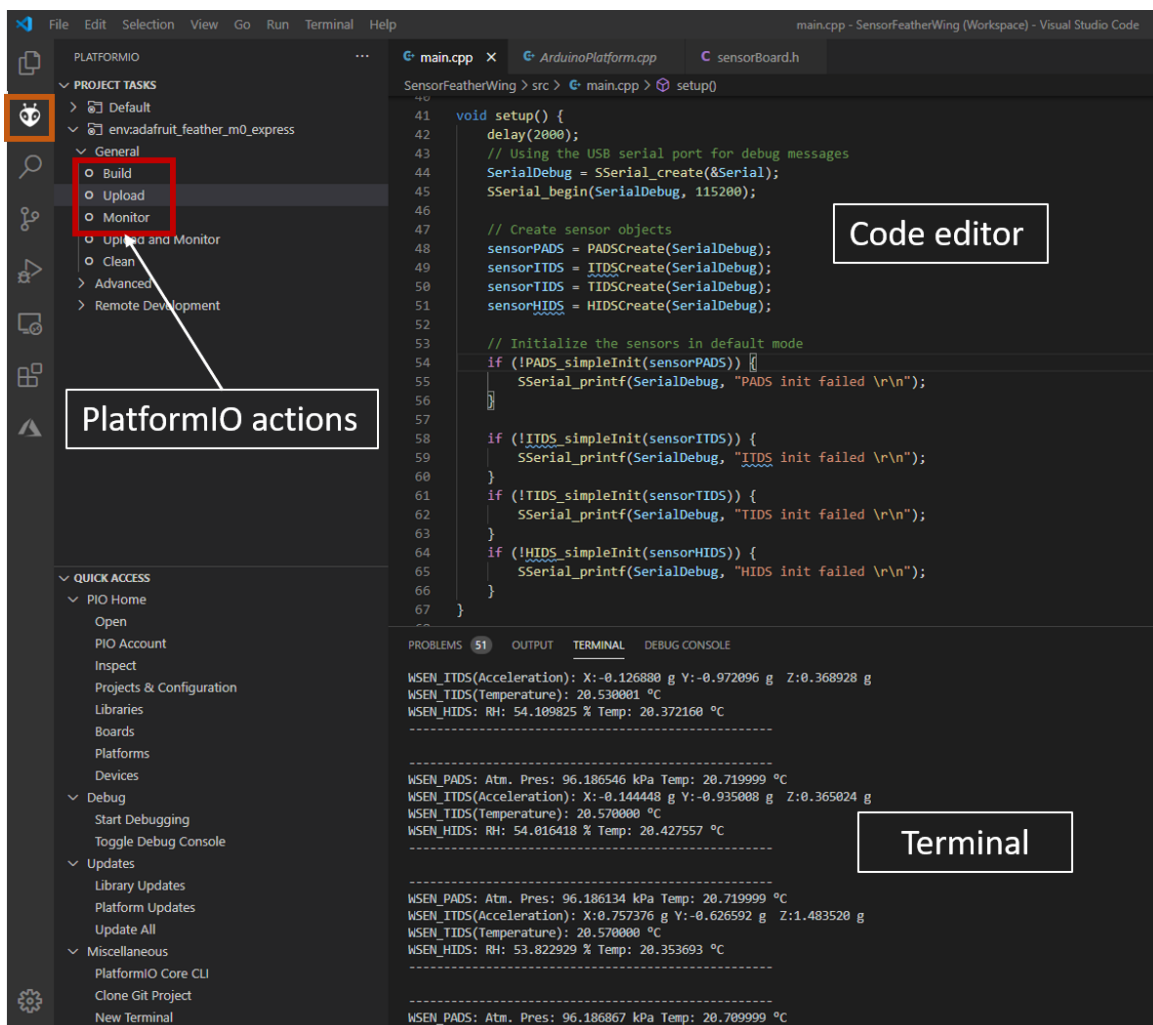


Figure 9: VS code

3.6 Sensor data in the cloud

On successful upload the following actions are performed by the application,

- The Calypso Wi-Fi module connects to the configured Wi-Fi network.
- A secure MQTT connection is made with the configured cloud.
- Sensor data is periodically read and transmitted to the cloud.

Please refer to the instruction in the links below to process the sensor data further.

- **Amazon AWS:**

<https://github.com/WurthElektronik/FeatherWings/Sensor2CloudConnectivity/aws>

- **Microsoft Azure:**

<https://github.com/WurthElektronik/FeatherWings/Sensor2CloudConnectivity/azure>

A sample snippet of the sensor data is as shown below,

```
{{ "messageId":1},{ "PADS_T":22.81,"PADS_P":94.766},{ "ITDS_X":-0.08,"ITDS_Y":-1.019,"ITDS_Z":0.158},{ "TIDS_T":22.87},{ "HIDS_T":23.2,"HIDS_RH":54.19}}
```

4 Summary

Designing an IoT solution brings with it a number of challenges. Being multifaceted, IoT applications demands a lot of competence from hardware design to UI development. Under such situations, it is prudent to take a modular approach. This means breaking down the architecture into functionally independent blocks. One such essential building-block for an IoT application is wireless connectivity. Integrating a radio module such as Calypso enables the designer to completely delegate the task of wireless connectivity, which saves a lot of effort enabling quicker time-to-market.

At the other end, it is a completely different approach necessary to arrive at a cloud solution. Flexibility and security are the key parameters that needs to be kept in mind while designing a cloud solution.

In this application note, a simple IoT technology stack that represents the principle behind any IoT solution is presented. Further, with the help of an example, a solution using the Calypso Wi-Fi module and AWS/Azure cloud platforms is described. Finally, a step-by-step prototype of the proposed solution is presented.

5 References

- [1] MQTT Amazon AWS. <https://docs.aws.amazon.com/iot/latest/developerguide/mqtt.html>.
- [2] MQTT Microsoft Azure. <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-mqtt-support>.
- [3] MQTT specification (V3.1.1). <https://mqtt.org/mqtt-specification/>.

6 Important notes

The following conditions apply to all goods within the wireless connectivity product range of Würth Elektronik eiSos GmbH & Co. KG:

6.1 General customer responsibility

Some goods within the product range of Würth Elektronik eiSos GmbH & Co. KG contain statements regarding general suitability for certain application areas. These statements about suitability are based on our knowledge and experience of typical requirements concerning the areas, serve as general guidance and cannot be estimated as binding statements about the suitability for a customer application. The responsibility for the applicability and use in a particular customer design is always solely within the authority of the customer. Due to this fact, it is up to the customer to evaluate, where appropriate to investigate and to decide whether the device with the specific product characteristics described in the product specification is valid and suitable for the respective customer application or not. Accordingly, the customer is cautioned to verify that the documentation is current before placing orders.

6.2 Customer responsibility related to specific, in particular safety-relevant applications

It has to be clearly pointed out that the possibility of a malfunction of electronic components or failure before the end of the usual lifetime cannot be completely eliminated in the current state of the art, even if the products are operated within the range of the specifications. The same statement is valid for all software sourcecode and firmware parts contained in or used with or for products in the wireless connectivity and sensor product range of Würth Elektronik eiSos GmbH & Co. KG. In certain customer applications requiring a high level of safety and especially in customer applications in which the malfunction or failure of an electronic component could endanger human life or health, it must be ensured by most advanced technological aid of suitable design of the customer application that no injury or damage is caused to third parties in the event of malfunction or failure of an electronic component.

6.3 Best care and attention

Any product-specific data sheets, manuals, application notes, PCN's, warnings and cautions must be strictly observed in the most recent versions and matching to the products firmware revisions. This documents can be downloaded from the product specific sections on the wireless connectivity homepage.

6.4 Customer support for product specifications

Some products within the product range may contain substances, which are subject to restrictions in certain jurisdictions in order to serve specific technical requirements. Necessary information is available on request. In this case, the field sales engineer or the internal sales person in charge should be contacted who will be happy to support in this matter.

6.5 Product improvements

Due to constant product improvement, product specifications may change from time to time. As a standard reporting procedure of the Product Change Notification (PCN) according to the JEDEC-Standard, we inform about major changes. In case of further queries regarding the PCN, the field sales engineer, the internal sales person or the technical support team in charge should be contacted. The basic responsibility of the customer as per section 6.1 and 6.2 remains unaffected. All wireless connectivity module driver software "wireless connectivity SDK" and its source codes as well as all PC software tools are not subject to the Product Change Notification information process.

6.6 Product life cycle

Due to technical progress and economical evaluation we also reserve the right to discontinue production and delivery of products. As a standard reporting procedure of the Product Termination Notification (PTN) according to the JEDEC-Standard we will inform at an early stage about inevitable product discontinuance. According to this, we cannot ensure that all products within our product range will always be available. Therefore, it needs to be verified with the field sales engineer or the internal sales person in charge about the current product availability expectancy before or when the product for application design-in disposal is considered. The approach named above does not apply in the case of individual agreements deviating from the foregoing for customer-specific products.

6.7 Property rights

All the rights for contractual products produced by Würth Elektronik eiSos GmbH & Co. KG on the basis of ideas, development contracts as well as models or templates that are subject to copyright, patent or commercial protection supplied to the customer will remain with Würth Elektronik eiSos GmbH & Co. KG. Würth Elektronik eiSos GmbH & Co. KG does not warrant or represent that any license, either expressed or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, application, or process in which Würth Elektronik eiSos GmbH & Co. KG components or services are used.

6.8 General terms and conditions

Unless otherwise agreed in individual contracts, all orders are subject to the current version of the "General Terms and Conditions of Würth Elektronik eiSos Group", last version available at www.we-online.com.

7 Legal notice

7.1 Exclusion of liability

Würth Elektronik eiSos GmbH & Co. KG considers the information in this document to be correct at the time of publication. However, Würth Elektronik eiSos GmbH & Co. KG reserves the right to modify the information such as technical specifications or functions of its products or discontinue the production of these products or the support of one of these products without any written announcement or notification to customers. The customer must make sure that the information used corresponds to the latest published information. Würth Elektronik eiSos GmbH & Co. KG does not assume any liability for the use of its products. Würth Elektronik eiSos GmbH & Co. KG does not grant licenses for its patent rights or for any other of its intellectual property rights or third-party rights.

Notwithstanding anything above, Würth Elektronik eiSos GmbH & Co. KG makes no representations and/or warranties of any kind for the provided information related to their accuracy, correctness, completeness, usage of the products and/or usability for customer applications. Information published by Würth Elektronik eiSos GmbH & Co. KG regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof.

7.2 Suitability in customer applications

The customer bears the responsibility for compliance of systems or units, in which Würth Elektronik eiSos GmbH & Co. KG products are integrated, with applicable legal regulations. Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of Würth Elektronik eiSos GmbH & Co. KG components in its applications, notwithstanding any applications-related information or support that may be provided by Würth Elektronik eiSos GmbH & Co. KG. Customer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences lessen the likelihood of failures that might cause harm and take appropriate remedial actions. The customer will fully indemnify Würth Elektronik eiSos GmbH & Co. KG and its representatives against any damages arising out of the use of any Würth Elektronik eiSos GmbH & Co. KG components in safety-critical applications.

7.3 Trademarks

AMBER wireless is a registered trademark of Würth Elektronik eiSos GmbH & Co. KG. All other trademarks, registered trademarks, and product names are the exclusive property of the respective owners.

7.4 Usage restriction

Würth Elektronik eiSos GmbH & Co. KG products have been designed and developed for usage in general electronic equipment only. This product is not authorized for use in equipment where a higher safety standard and reliability standard is especially required or where a failure of the product is reasonably expected to cause severe personal injury or death,

unless the parties have executed an agreement specifically governing such use. Moreover, Würth Elektronik eiSos GmbH & Co. KG products are neither designed nor intended for use in areas such as military, aerospace, aviation, nuclear control, submarine, transportation (automotive control, train control, ship control), transportation signal, disaster prevention, medical, public information network etc. Würth Elektronik eiSos GmbH & Co. KG must be informed about the intent of such usage before the design-in stage. In addition, sufficient reliability evaluation checks for safety must be performed on every electronic component, which is used in electrical circuits that require high safety and reliability function or performance. By using Würth Elektronik eiSos GmbH & Co. KG products, the customer agrees to these terms and conditions.

8 License terms

This License Terms will take effect upon the purchase and usage of the Würth Elektronik eiSos GmbH & Co. KG wireless connectivity products. You hereby agree that this license terms is applicable to the product and the incorporated software, firmware and source codes (collectively, "Software") made available by Würth Elektronik eiSos in any form, including but not limited to binary, executable or source code form.

The software included in any Würth Elektronik eiSos wireless connectivity product is purchased to you on the condition that you accept the terms and conditions of this license terms. You agree to comply with all provisions under this license terms.

8.1 Limited license

Würth Elektronik eiSos hereby grants you a limited, non-exclusive, non-transferable and royalty-free license to use the software and under the conditions that will be set forth in this license terms. You are free to use the provided Software only in connection with one of the products from Würth Elektronik eiSos to the extent described in this license terms. You are entitled to change or alter the source code for the sole purpose of creating an application embedding the Würth Elektronik eiSos wireless connectivity product. The transfer of the source code to third parties is allowed to the sole extent that the source code is used by such third parties in connection with our product or another hardware provided by Würth Elektronik eiSos under strict adherence of this license terms. Würth Elektronik eiSos will not assume any liability for the usage of the incorporated software and the source code. You are not entitled to transfer the source code in any form to third parties without prior written consent of Würth Elektronik eiSos.

You are not allowed to reproduce, translate, reverse engineer, decompile, disassemble or create derivative works of the incorporated Software and the source code in whole or in part. No more extensive rights to use and exploit the products are granted to you.

8.2 Usage and obligations

The responsibility for the applicability and use of the Würth Elektronik eiSos wireless connectivity product with the incorporated Firmware in a particular customer design is always solely within the authority of the customer. Due to this fact, it is up to you to evaluate and investigate, where appropriate, and to decide whether the device with the specific product characteristics described in the product specification is valid and suitable for your respective application or not.

You are responsible for using the Würth Elektronik eiSos wireless connectivity product with the incorporated Firmware in compliance with all applicable product liability and product safety laws. You acknowledge to minimize the risk of loss and harm to individuals and bear the risk for failure leading to personal injury or death due to your usage of the product.

Würth Elektronik eiSos' products with the incorporated Firmware are not authorized for use in safety-critical applications, or where a failure of the product is reasonably expected to cause severe personal injury or death. Moreover, Würth Elektronik eiSos' products with the incorporated Firmware are neither designed nor intended for use in areas such as military, aerospace, aviation, nuclear control, submarine, transportation (automotive control, train control, ship control), transportation signal, disaster prevention, medical, public information network etc. You shall inform Würth Elektronik eiSos about the intent of such usage before design-in stage. In certain customer applications requiring a very high level of safety and in which the malfunction or failure of an electronic component could endanger human life or

health, you must ensure to have all necessary expertise in the safety and regulatory ramifications of your applications. You acknowledge and agree that you are solely responsible for all legal, regulatory and safety-related requirements concerning your products and any use of Würth Elektronik eiSos' products with the incorporated Firmware in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by Würth Elektronik eiSos. YOU SHALL INDEMNIFY WÜRTH ELEKTRONIK EISOS AGAINST ANY DAMAGES ARISING OUT OF THE USE OF WÜRTH ELEKTRONIK EISOS' PRODUCTS WITH THE INCORPORATED FIRMWARE IN SUCH SAFETY-CRITICAL APPLICATIONS.

8.3 Ownership

The incorporated Firmware created by Würth Elektronik eiSos is and will remain the exclusive property of Würth Elektronik eiSos.

8.4 Firmware update(s)

You have the opportunity to request the current and actual Firmware for a bought wireless connectivity Product within the time of warranty. However, Würth Elektronik eiSos has no obligation to update a modules firmware in their production facilities, but can offer this as a service on request. The upload of firmware updates falls within your responsibility, e.g. via ACC or another software for firmware updates. Firmware updates will not be communicated automatically. It is within your responsibility to check the current version of a firmware in the latest version of the product manual on our website. The revision table in the product manual provides all necessary information about firmware updates. There is no right to be provided with binary files, so called "Firmware images", those could be flashed through JTAG, SWD, Spi-Bi-Wire, SPI or similar interfaces.

8.5 Disclaimer of warranty

THE FIRMWARE IS PROVIDED "AS IS". YOU ACKNOWLEDGE THAT WÜRTH ELEKTRONIK EISOS MAKES NO REPRESENTATIONS AND WARRANTIES OF ANY KIND RELATED TO, BUT NOT LIMITED TO THE NON-INFRINGEMENT OF THIRD PARTIES' INTELLECTUAL PROPERTY RIGHTS OR THE MERCHANTABILITY OR FITNESS FOR YOUR INTENDED PURPOSE OR USAGE. WÜRTH ELEKTRONIK EISOS DOES NOT WARRANT OR REPRESENT THAT ANY LICENSE, EITHER EXPRESS OR IMPLIED, IS GRANTED UNDER ANY PATENT RIGHT, COPYRIGHT, MASK WORK RIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT RELATING TO ANY COMBINATION, MACHINE, OR PROCESS IN WHICH THE WÜRTH ELEKTRONIK EISOS' PRODUCT WITH THE INCORPORATED FIRMWARE IS USED. INFORMATION PUBLISHED BY WÜRTH ELEKTRONIK EISOS REGARDING THIRD-PARTY PRODUCTS OR SERVICES DOES NOT CONSTITUTE A LICENSE FROM WÜRTH ELEKTRONIK EISOS TO USE SUCH PRODUCTS OR SERVICES OR A WARRANTY OR ENDORSEMENT THEREOF.

8.6 Limitation of liability

Any liability not expressly provided by Würth Elektronik eiSos shall be disclaimed. You agree to hold us harmless from any third-party claims related to your usage of the Würth Elektronik eiSos' products with the incorporated Firmware, software and source code. Würth

Elektronik eiSos disclaims any liability for any alteration, development created by you or your customers as well as for any combination with other products.

8.7 Applicable law and jurisdiction

Applicable law to this license terms shall be the laws of the Federal Republic of Germany. Any dispute, claim or controversy arising out of or relating to this license terms shall be resolved and finally settled by the court competent for the location of Würth Elektronik eiSos' registered office.

8.8 Severability clause

If a provision of this license terms is or becomes invalid, unenforceable or null and void, this shall not affect the remaining provisions of the terms. The parties shall replace any such provisions with new valid provisions that most closely approximate the purpose of the terms.

8.9 Miscellaneous

Würth Elektronik eiSos reserves the right at any time to change this terms at its own discretion. It is your responsibility to check at Würth Elektronik eiSos homepage for any updates. Your continued usage of the products will be deemed as the acceptance of the change.

We recommend you to be updated about the status of new firmware and software, which is available on our website or in our data sheet and manual, and to implement new software in your device where appropriate.

By ordering a wireless connectivity product, you accept this license terms in all terms.

List of Figures

1	IoT application stack	5
2	Components of a data platform	6
3	System architecture	7
4	System design	8
5	AWS example	10
6	Azure example	11
7	Publish/Subscribe mechanism	12
8	System set-up	16
9	VS code	18



more than you expect



**Internet
of Things**



**Monitoring
& Control**



**Automated Meter
Reading**

Contact:

Würth Elektronik eiSos GmbH & Co. KG
Division Wireless Connectivity & Sensors

Max-Eyth-Straße 1
74638 Waldenburg
Germany

Tel.: +49 651 99355-0
Fax.: +49 651 99355-69
www.we-online.com/wireless-connectivity

