WÜRTH ELEKTRONIK

ANR007 CALYPSO

IOT APPLICATION BASED ON CALYPSO MODULE

VERSION 1.2

OCTOBER 21, 2019

Revision history

Document version	Notes	Date
1.0	- Initial version	April 2019
1.1	- Updated file name to new AppNote name structure. Updated important notes, legal notice & license terms chapters.	June 2019
1.2	- Updated address of Division Wireless Connectivity & Sensors location in Trier	October 2019

Contents

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. Interacting devices and the data they generate, provide for a range of new applications. Industrial automation, healthcare, home automation, 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 world of machines holds the potential for profound changes in global industry and, therefore, for many aspects of private and social life - to the way we work and live.

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 proposition 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 using a smart home example. Finally, a solution using the Calypso Wi-Fi module is presented.

1.2 The IoT stack

Irrespective of the area of application, an end-to-end IoT solution consists of the following components (figure 1). Explosive growth of IoT market has called for industry standard that define how various layers of the stack interact with each other. However, practical implementation of the same remains a challenge.

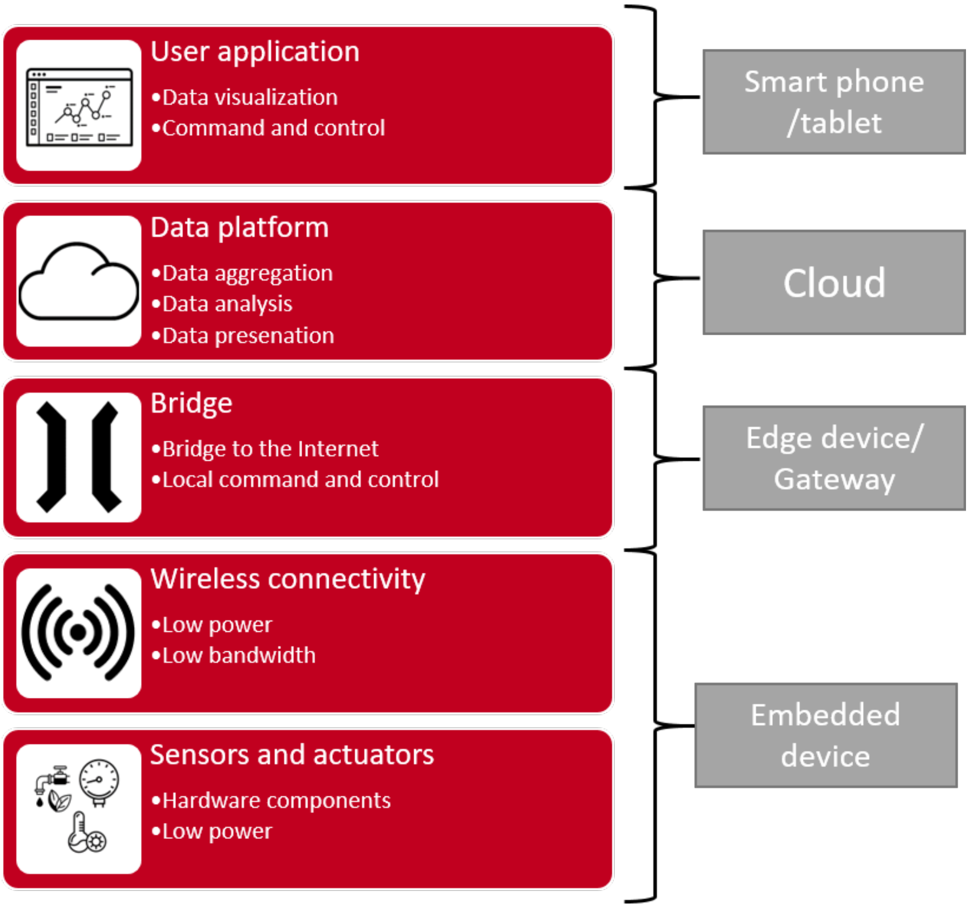


Figure 1: IoT application stack

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

Artificial intelligence, and big data analytics creates a potential for huge advances in productivity, efficiency, and cost savings. The IIoT creates a universe of sensors that enables an accelerated deep learning of existing operations. These data tools allow for rapid contextualization, automatic pattern, and trend detection. Furthering this for manufacturing operations will finally allow for true quantitative capture of qualitative operations.
In the next section a sample IoT application is considered.

1.3 IoT application example

"Smart home" or "connected home" is a typical use case for IoT. Networking individual devices (sensors and actuators) used in a common household can result in an intelligent system which provides several benefits including energy efficiency, cost savings and general well-being.

1.3.1 Requirements

A typical smart home application has the following key features.

- Heating ventilation and air conditioning (HVAC).
- Access control (door and windows).
- Security and surveillance.
- Lighting control.
- Control Home appliances or consumer electronics.
- Energy management.

The preferred ways of interaction with humans would be,

- Computer via web-interface.
- Mobile application.
- Smart voice assistants.

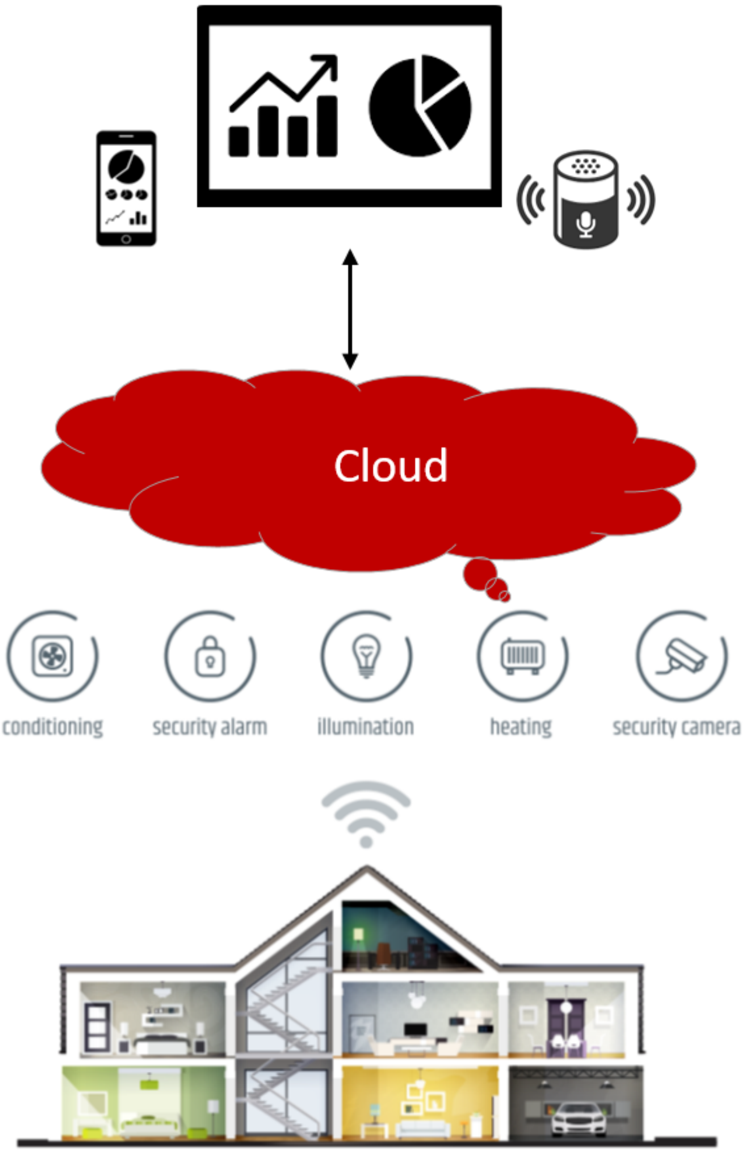


Figure 2: Smart home example

1.3.2 System architecture

Each of the above mentioned tasks require 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.

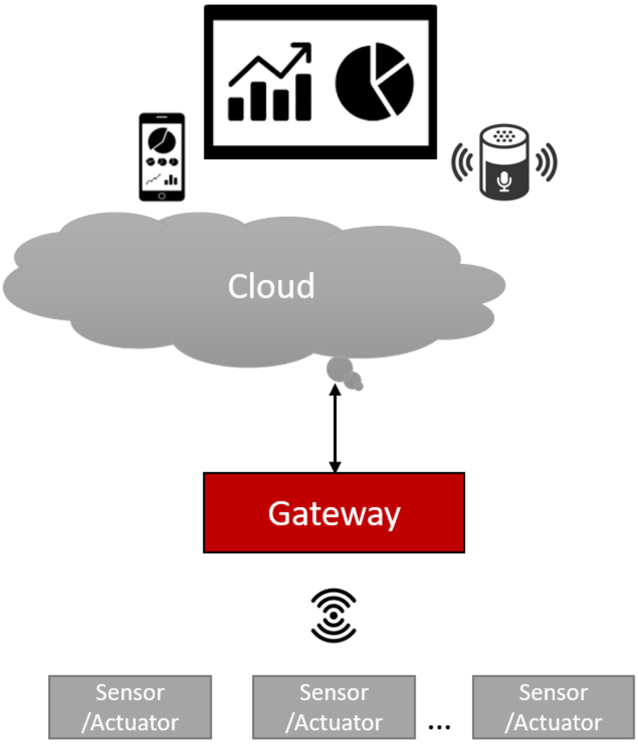


Figure 3: System architecture

1.4 System design using Calypso

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 the perfect building block for an IoT application. Figure 4 illustrates the design that realizes the architecture described in 1.3.2.

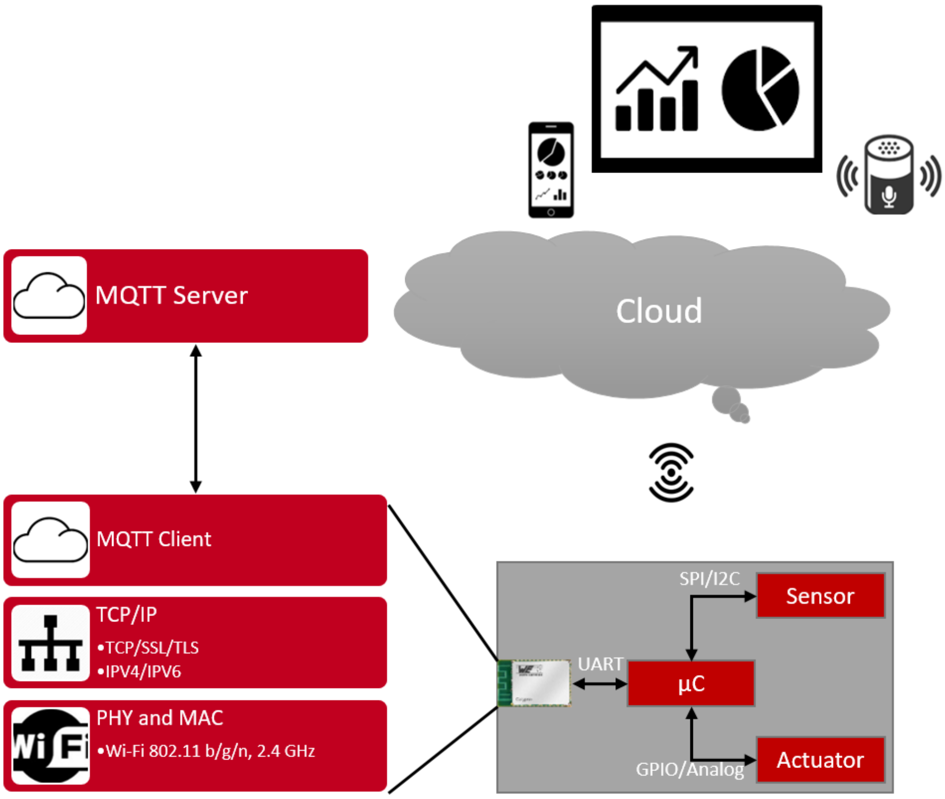


Figure 4: System design

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 AP).

Further, the MQTT-client implemented on-board the Calypso can be configured to connect to a broker/sever running in the network. Most of the commonly used cloud platforms today use 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 either use 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 μA) 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, SSL/TLS allow authentication as well as end to end encryption of data. Any modern application require 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.

1.5 IoT application demo

Rest of this application note is intended to demonstrate the capabilities of the Calypso Wi-Fi module in an IoT application example. Chapter 2 gives a brief introduction to the MQTT protocol and Chapter 3 provides a step-by-step demo of data transfer between a MQTT client and a MQTT broker using the Calypso evaluation board.

2 MQTT

MQTT - MQ Telemetry Transport is a light weight 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 makes 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 [1]. 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).

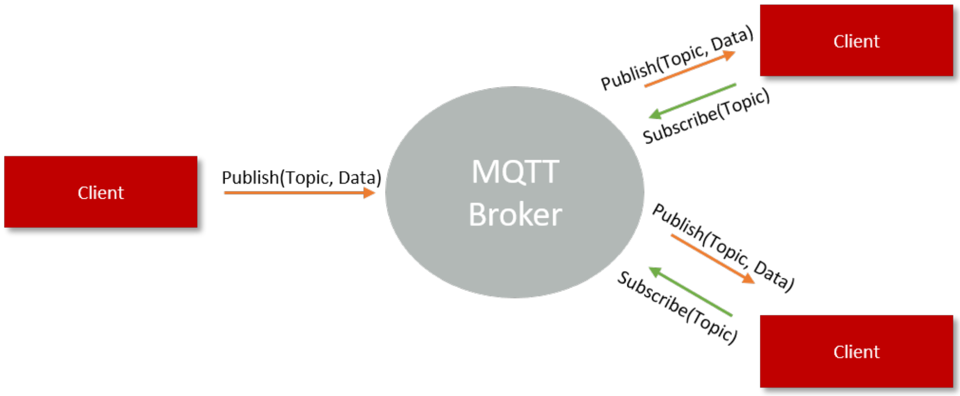


Figure 5: Publish/Subscribe mechanism

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/ligth_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 turns 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 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 connect and allow or disallow a client connection. However, the user has to run MQTT over TLS/SSL to enable end-to-end encryption and advanced client authentication.

3 IoT demo using Calypso evaluation board

In this chapter, a demonstration of a smart home IoT application is presented using the Calypso evaluation board. The intention here is to go through the process step-by-step so as to enable the user to implement the same without much effort. Instead of using a specific cloud platform, a more generic approach is taken here by using an open source Mosquitto broker. However, the same concept can be extended to work with any cloud based MQTT broker.

3.1 Prerequisites

The following hardware is necessary to go though this demo.

1. One Calypso evaluation board.
2. An IEEE 802.11 b/g/n compatible access point in the 2.4 GHz band.
3. Computer with a serial terminal emulator like Tera Term.
4. Raspberry Pi 3 Model B or Raspberry Pi 3 Model B+ with latest version of raspbian OS installed [6].
5. Internet connectivity for Raspberry Pi to download and install packages.
6. Smartphone with MQTT app installed. MyMQTT [3] and MQTTTool [4] applications on android and iOS platforms respectively are used in this example.

3.2 Demo set-up

The main objective of this demo is to create a quick prototype of the system designed in section `1.4`.

- **Host processor:** For simplicity, a PC/Laptop running a serial UART emulator acts as the host processor that controls the actions of the Calypso module.
- **Sensors and actuators:** Simulated sensor data is input into the console on the PC. No actual sensors are used here.
- **Wireless connectivity and gateway:** Calypso provides Wi-Fi connectivity as well as MQTT client for information exchange.
- **MQTT broker:** Mosquitto broker running on the Raspberry Pi.
- **User interface:** The user interaction is demonstrated by using MQTT apps installed on smartphones.

The set-up consists of a Wi-Fi network (simulating the smart home network) with an access point and a DHCP-server. The Raspberry Pi, the Calypso and the smartphone are connected to this network. The MQTT-broker is running on the Raspberry Pi while MQTT clients run on the smartphone and the Calypso. With this configuration, data exchange can take place between the smartphone (user interface) and Calypso (sensor/actuator) over Raspberry Pi (cloud platform). In the following sections, step-by-step instructions to achieve the same are presented.

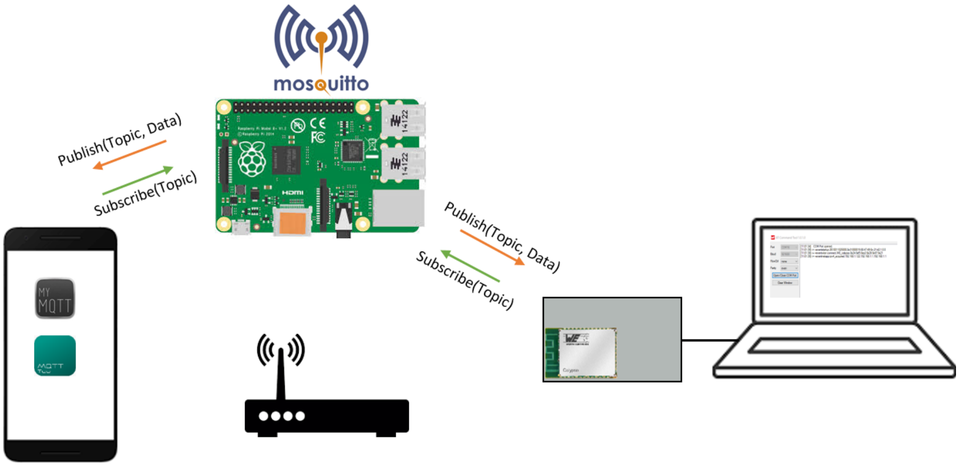


Figure 6: Set up

3.3 Eclipse Mosquitto

Eclipse Mosquitto is an open source light weight message broker that implements MQTT protocol v3.1 and v3.1.1. Mosquitto is a part of eclipse foundation. In this particular use case Mosquitto on Raspberry Pi is used as message broker. Mosquitto is available either as executable and/or source on most of the commonly used platforms [2].

3.4 Installing Eclipse Mosquitto MQTT broker on Raspberry Pi

Go through the following steps to install the MQTT broker on the Raspberry Pi.

1. Make sure that the Raspberry Pi is connected to the internet (see [5]).

2. Update the Raspbian OS by typing in the following commands into the terminal on Raspberry Pi.

```
$sudo apt−get update
$sudo apt−get upgrade
```

3. In order to get the latest version of the Mosquitto repository, the key to the repository has to be imported.

```
$sudo wget http://repo.mosquitto.org/debian/mosquitto−repo.gpg.key
$sudo apt−key add mosquitto−repo.gpg.key
```

4. Make the repository available to apt.

```
$cd /etc/apt/sources. list .d/
```

5. Now get the correct repository based on the corresponding Debian distribution.

```
$sudo wget http://repo.mosquitto.org/debian/mosquitto−stretch.list
```

6. Updating the apt information to access the correct repository.

```
$sudo apt−get update
```

7. Install mosquitto.

```
$sudo apt−get install  mosquitto
```

8. Reboot the system after installation.

3.5 Setting-up Wi-Fi network

An access point compatible with IEEE 802.11b/g/n operating in the 2.4 GHz frequency band is necessary for this application. In this particular example, an access point setup with W-PA2, SSID "WE_calypso" and pre-shared key "calypsowlan" is used. The DHCP server on-board the AP is configured to lease IPv4 addresses to the clients. As all network applications are local, no internet connection is necessary.

3.6 Connecting Raspberry Pi to the network

First, the Raspberry Pi is connected to the AP by selecting the correct SSID and entering the pre-shared key [5]. Check the IP address acquired by entering the following command,

```
 $ifconfig

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet  192.168.1.126 netmask 255.255.255.0 broadcast 192.168.1.255
inet6  fe80::fe36:49e4:3851:ea47 prefixlen  64   scopeid 0x20<link>
ether  b8:27:eb:ad:ec:72  txqueuelen 1000  (Ethernet)
RX packets 27 bytes 3167 (3.0 KiB)
RX errors 0   dropped 0  overruns 0   frame 0
TX packets 87  bytes 13191 (12.8 KiB)
TX errors 0   dropped 0 overruns 0  carrier  0    collisions  0
```

3.7 Connecting Calypso to the network

3.7.1 Connecting Calypso to PC

1. Connect the Calypso evaluation boards to the laptop/PC via USB.

2. The power LED indicates that supply voltage is active.

The FTDI driver for the converter IC on the evaluation board has to be installed and/or updated to the latest version. On correct driver installation, the evaluation board appears as a virtual COM port.

3. Open an instance of the serial port emulator with COM port settings 921600 baud, 8e1 for the Wi-Fi module connected to the PC via USB.
4. On pressing the Reset button, the start-up message appears on the terminal with the product article number, chipID, MAC address and the current software version. It has to be noted that the MAC address is unique to every module.

```
+eventstartup:2610011025000,0x31000019,c8:fd:19:05:54:b4,1.0.0
```

3.7.2 Connect to an access point

1. In this example an access point with the following settings is used.

   SSID : WE_calypso

   Security method : WPA2_PSK

   Key : calypsowlan

2. Type in the following command into the terminal to connect to the access point.

```
AT+wlanconnect=WE_calypso,,WPA_WPA2,calypsowlan,,,
OK
+eventwlan:connect,WE_calypso,0x24:0xf5:0xa2:0x28:0x97:0x21
+eventnetapp:ipv4_acquired,192.168.1.169,192.168.1.1,192.168.1.1
```

3. The above log indicate a successful Wi-Fi connect and subsequent IP acquisition. The Wi-Fi connection process typically takes a few seconds to complete.

3.8 Connecting smartphone to the network

Make sure that the smartphone/tablet is also connected to the same Wi-Fi network.

In the current example, the addresses assigned are as shown below,

	IP Address	Role
Raspberry Pi	192.168.1.126	MQTT broker
Calypso evaluation board	192.168.1.169	MQTT client
Smartphone	192.168.1.161	MQTT client

Table 1: IP addresses and roles

3.9 Starting MQTT broker on Raspberry Pi

Once mosquitto is installed as describe in section 3.4, the broker start as a background process on boot-up. The following commands can be used to start and stop the background process

```
$sudo /etc/ init .d/mosquitto stop
$sudo /etc/ init .d/mosquitto start
```

The broker can be started in verbose mode to get information regarding MQTT events.

```
$sudo mosquitto −v
```

On starting the Mosquitto broker, a server listens for connections on port 1883.

3.10 Starting MQTT client on Calypso

1. Firstly, an MQTT client is created using the following command. In this case the client ID is chosen to be "kitchen_temp".

   ```
   AT+mqttcreate=kitchen_temp,ip4,192.168.1.126,1883,,,,,,,v3_1,0,1
   +mqttcreate:0
   OK
   ```

2. The next step is to establish a connection with the broker using,

   ```
   AT+mqttconnect=0
   +eventmqtt:operation,connack,0
   OK
   ```

   If the broker is opened in verbose mode, the following logs indicate a successful connection,

   ```
   61548680537: New connection from 192.168.1.169 on port 1883.
   1548680537: New client connected from 192.168.1.169 as kitchen_temp (c1, k25
         ).
   1548680537: Sending CONNACK to kitchen_temp (0, 0)
   1548680562: Received PINGREQ from kitchen_temp
   1548680562: Sending PINGRESP to kitchen_temp
   ```

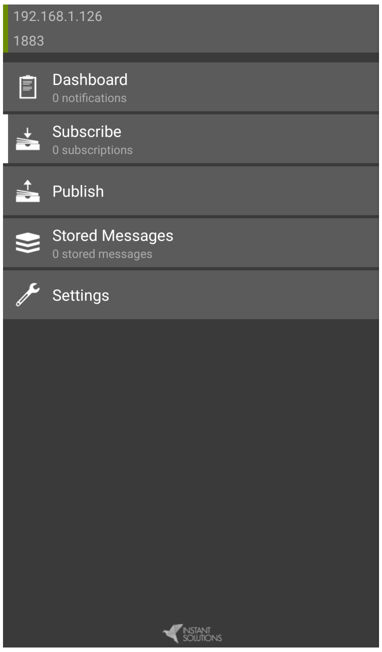
3.11 Starting MQTT client on smartphone

1. Open the smartphone application and enter the broker address and port.

2. Initiate a connection to the broker.

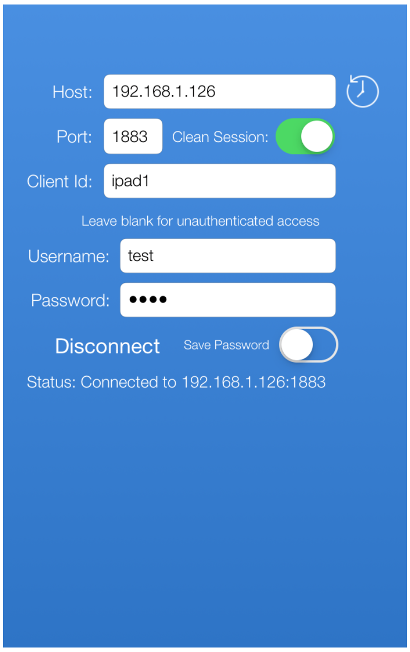
   If the broker is opened in verbose mode, the following logs indicate a successful connection,

```
1548680731: New client connected from 192.168.1.161 as root.1548680731721 (
    c1, k60).
1548680731: Sending CONNACK to root.1548680731721 (0, 0)
```



192.168.1.126
1883

Figure 7: Connect on android device



Host: 192.168.1.126

Port: 1883 Clean Session: on

Client Id: ipad1

Leave blank for unauthenticated access

Username: test

Password: ••••

Save Password

Status: Connected to 192.168.1.126:1883

Figure 8: Connect on iOS device

3.12 Data exchange

The final step in the process is to exchange data between the two clients using the previously explained publish/subscribe mechanism. In order to simulate a smart home application, the Calypso sends simulated temperature data under the topic "home/kitchen/temperature". The smartphone on the other hand controls a simulated cooling system by publishing under the topic "home/kitchen/cooler". Both the module have to subscribe to the other topic to receive messages.

3.12.1 Subscribe from Calypso evaluation board

The Calypso evaluation board can be made to subscribe to a topic by sending the following command,

```
AT+mqttsubscribe=0,1,home/kitchen/cooler,QOS0,
OK
```

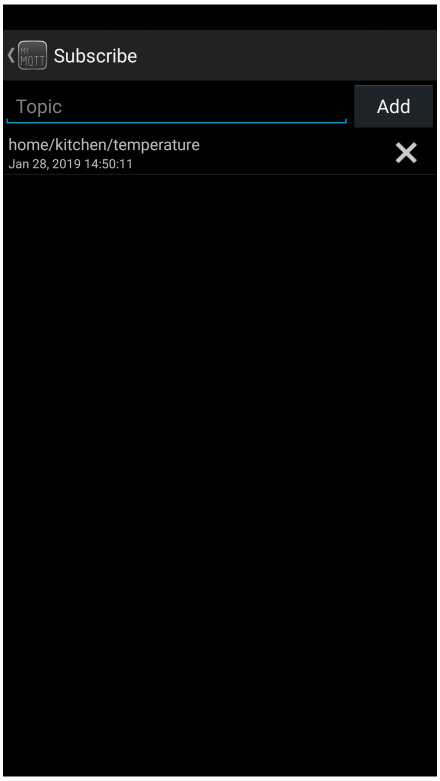
If the broker is opened in verbose mode, the following logs indicate a successful subscribe,

```
1548682694: Received SUBSCRIBE from kitchen_temp
1548682694:         home/kitchen/cooler (QoS 0)
1548682694: kitchen_temp 0 home/kitchen/cooler
1548682694: Sending SUBACK to kitchen_temp
```

3.12.2 Subscribe from smartphone

Subscribe to the topic "home/kitchen/temperature" from the smartphone app.
If the broker is opened in verbose mode, the following logs indicate a successful subscribe,

```
1548682694: Received SUBSCRIBE from kitchen_temp
1548682694:         home/kitchen/cooler (QoS 0)
1548682694: kitchen_temp 0 home/kitchen/cooler
1548682694: Sending SUBACK to kitchen_temp
```



home/kitchen/temperature
Jan 28, 2019 14:50:11

Figure 9: Subscribe on android device

3.12.3 Publish from Calypso evaluation board

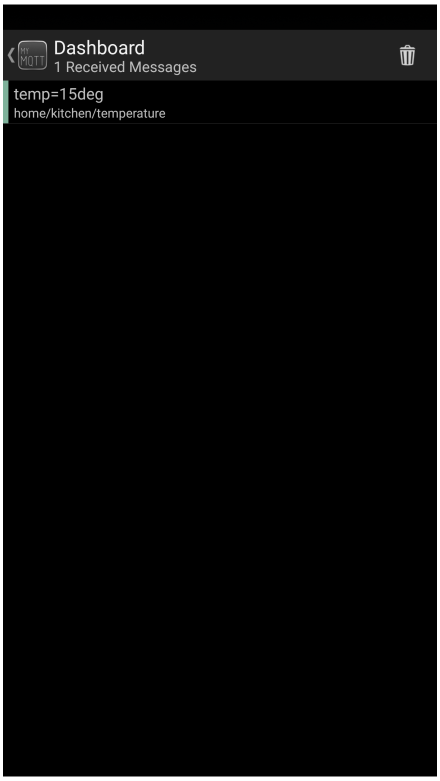
The Calypso evaluation board can be made to publish a string say "temp=15deg" ("dGVtcD0xNWRlZw==" base64 encoded) by sending the command,

```
AT+mqttpublish=0,home/kitchen/temperature,QOS0,0,16,dGVtcD0xNWRlZw==
OK
```

If the broker is opened in verbose mode, the following logs indicate a successful publish,

```
1548683799: Received PUBLISH from kitchen_temp (d0, q0, r0, m0, 'home/kitchen/
    temperature', ... (10 bytes))
1548683799: Sending PUBLISH to root.1548683772471 (d0, q0, r0, m0, 'home/kitchen
    /temperature', ... (10 bytes))
```

Smartphone being subscribed to the topic receives a notification of the same.



temp=15deg
home/kitchen/temperature

Figure 10: Receive on android device

3.12.4 Publish from smartphone

On initiating a publish from the smartphone on the topic "home/kitchen/cooler" with data "switch_off", the following message appears on the Calypso.

```
+eventmqtt:recv,home/kitchen/cooler,qos0,0,0,1,16,c3dpdGNoX29mZg==
```

Here the publish is notified as an event in the command interface of the Calypso.
If the broker is opened in verbose mode, the following logs indicate a successful publish,

```
1548683524: Received PUBLISH from root.1548683393664 (d0, q0, r0, m0, 'home/
    kitchen/cooler', ... (10 bytes))
1548683524: Sending PUBLISH to kitchen_temp (d0, q0, r0, m0, 'home/kitchen/cooler'
    , ... (10 bytes))
```

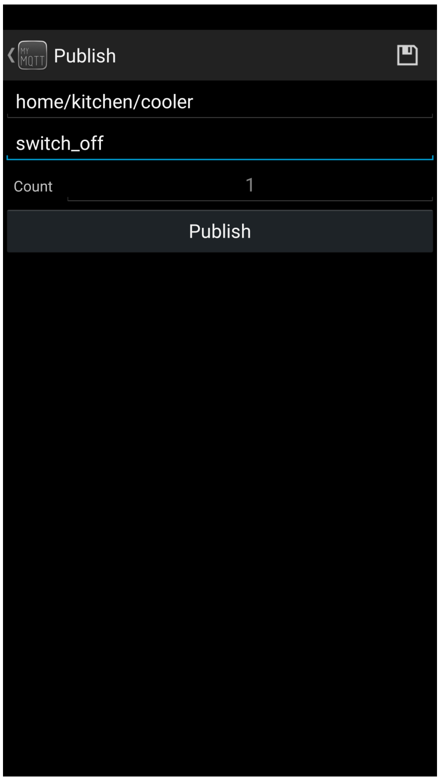
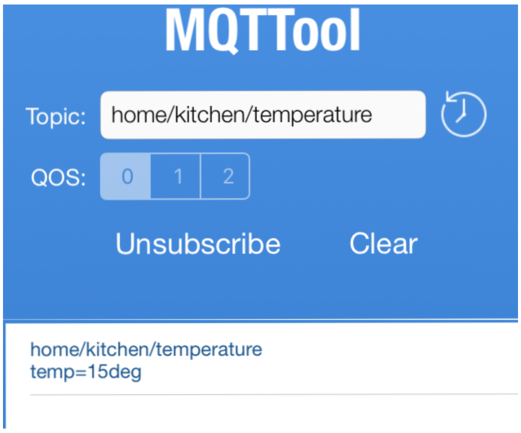


Figure 11: Publish from android device



home/kitchen/temperature
temp=15deg

Figure 12: Publish/Subscribe iOS device

3.13 Further enhancements

This example is intended to build a quick prototype to demonstrate the system design. However, it can be easily extended to include more advanced features. Here is a list of few enhancements.

- Use user authentication at the MQTT broker.
- TLS instead of TCP to enable end-to-end-encryption.
- Implement sleep-modes for battery operated applications.

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.

In this application note, a simple IoT technology stack that represents the principle behind any IoT solution is presented. Further, with the help of a smart home example, a solution using the Calypso Wi-Fi module is described. Finally, a step-by-step demo of the proposed solution is presented. This application note illustrates the ease of integration of the Calypso into any embedded application.

References

[1] *MQTT v3.1.1 now as OASIS Standard.* 2014. [Online] *http://mqtt.org/2014/11/mqtt-v3-1-1-now-an-oasis-standard*.

[2] *Eclipse Mosquitto.* [Online] *https://mosquitto.org/download/*.

[3] *MyMQTT android application.* [Online] *https://play.google.com/store/apps/details?id=at.tripwire.mqtt.client*.

[4] *MQTTTool iOS application.* [Online] *https://itunes.apple.com/us/app/mqttool/id1085976398?mt=8*.

[5] *Wireless connectivity in Raspberry Pi Desktop.* [Online] *https://www.raspberrypi.org/documentation/configuration/wireless/desktop.md*

[6] *Raspbian download page.* [Online] *https://www.raspberrypi.org/downloads/raspbian/*

5 Important notes

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

5.1 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.

5.2 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.

5.3 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.

5.4 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.

5.5 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 5.1 and 5.2 remains unaffected. All wireless connectivity module driver software ¨wireless connectivity SDK¨ and it's source codes as well as all PC software tools are not subject to the Product Change Notification information process.

5.6 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.

5.7 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.

5.8 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*.

6 Legal notice

6.1 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.

6.2 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 in-formation 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. KGand its representatives against any damages arising out of the use of any Würth Elektronik eiSos GmbH & Co. KG components in safety-critical applications.

6.3 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.

6.4 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.

7 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.

7.1 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.

7.2 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.

7.3 Ownership

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

7.4 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.

7.5 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.

7.6 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.

7.7 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.

7.8 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.

7.9 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

List of Tables

WÜRTH ELEKTRONIK

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