

Thinking about the wireless IoT application? Take advantage of the IoT Starter Kit!

We used the new [IoT Starter Kit](#) to get the sensor data and turn on / off relays. The application can be easily complemented by additional HW and SW extensions.

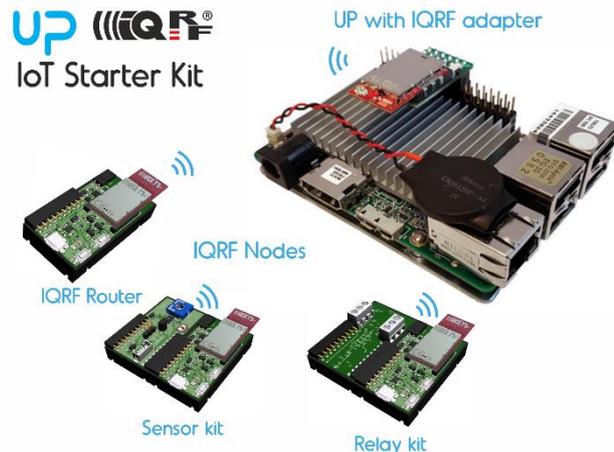
Because all necessary software and services are already ready for it and made available to the public, the development of the wireless IoT application is easy and fast. Tutorials are also available for the kit, which will take us through all the basic steps.

UP board

AAEON's popular **UP** developer board forms the interface between **IQRF** and **Microsoft Azure** web services. It is a very interesting hardware product of excellent quality, durability and equipment. Details can be found [here](#).

The **Gateway** provides communication with the **IQRF** network, it can read data thanks to the scheduler regularly and sends it via the **MQTT** channel to the ready **IoT Hub** in **Microsoft Azure**. Serving MQTT channels, formatting JSON messages and all the communication controls the IQRF Daemon, which is available for this development kit.

To set up the MQTT channel, we set up the name of the IoT Hub in Microsoft Azure in the *MqttMessaging.json* configuration file and the name of our GW that we added here as a device.



```
{
  "Name": "MqttMessaging2",
  "Enabled": true,
  "Properties": {
    "BrokerAddr": "ssl://YOURIOTHUB.azure-devices.net:8883",
    "ClientId": "YOURDEVICE",
    "Persistence": 1,
    "Qos": 1,
    "TopicRequest": "devices/YOURDEVICE/messages/devicebound/#",
    "TopicResponse": "devices/YOURDEVICE/messages/events/",
    "User": "YOURIOTHUB.azure-devices.net/YOURDEVICE",
    "Password": "SharedAccessSignature sr=YOURIOTHUB.azure-devices.net%2Fdevices%2FYOURDEVICE&sig=aUV7SV1YUtFtO3 NoXtPxtnqoI4YH4xO9A8cAAwt8%2BQ%3D&se=1492996556",
    "EnabledSSL": true,
    "KeepAliveInterval": 20,
    "ConnectTimeout": 5,
  }
}
```

```

    "MinReconnect": 1,
    "MaxReconnect": 64
  }
}

```

In the *Scheduler.json* configuration file, we have set parameters for reading sensory values. Parameters in the *pnum* and *pcmd* variables are based on the *Custom DPA Handler*, which is uploaded in the appropriate IQRF node for sensor data readings.

```

{
  "Tasks": [
  ],
  "TasksJson": [
  {
    "time": "*/30 * * * * *",
    "service": "BaseServiceForMQTT2",
    "message": {
      "ctype": "dpa",
      "type": "raw-hdp",
      "msgid": "1",
      "timeout": 500,
      "nadr": "01",
      "pnum": "20",
      "pcmd": "31",
      "hwpid": "ffff",
      "req-data": "",
      "request": ".",
      "request-ts": "",
      "confirmation": ".",
      "confirmation-ts": "",
      "response": ".",
      "response-ts": ""
    }
  }
  ]
}

```

More detailed information about installing and configuring services can be found [here](#).

[IQRF wireless mesh network](#)

The **IQRF** platform is the world leader in wireless secure mesh networks. It was founded in MICRORISC company more than 14 years ago and, besides scientific awards, has many successful IoT projects.

In the **IoT Starter Kit** you will find a **sensor** kit for sensory data collection (external thermometer, light sensor for measuring the level of light, potentiometer for voltage measurement) and **relays** for switching other devices on / off.

Custom DPA Handlers are available to ensure easy communication with relays and sensors.

In the *Custom-DPA-Handler-DDC-core.c* file, you find the commented source code where, after a quick analysis, you find out what peripherals and commands to use to communicate with the sensor or relay kit.

```
----- DDC-SE-01
Request: PNUM = 0x20, PCMD = 0x31
Response:
PData[0]=Temperature at [C]
PData[1]=Photoresistor value 0x00-0xFF
PData[2]=Potentiometer value 0x00-0xFF
returns ERROR_FAIL when error reading temperature

FRC=0xC0, returns temperature at C, 127 for 0 C, 0x80 for error reading temperature
FRC=0xC1, returns photoresistor value, returns 1 instead of 0
FRC=0xC2, returns potentiometer value, returns 1 instead of 0

----- DDC-RE-01
Request: PNUM = 0x20, PCMD = 0x32
PData[0]=Relay1Ctrl
PData[1]=Relay2Ctrl

Response:
PData[0]=Relay1State
PData[1]=Relay2State

RelayCtrl: 0=Switch the Relay off, 1=Switch the Relay on, Other=Do not control the Relay
RelayState: 0=Relay was off, 1=Relay was on
```

A description of the DPA protocol, that simplifies wireless communication, is [here](#).

IQRF Videotutorials

- [Build IQRF network with IQRF OS 4.0](#)
- [Custom DPA Handlers for IoT Starter Kit](#)
- [Macros for IoT Starter Kit](#)

Microsoft Azure

The MQTT protocol is used to send the collected values from individual IQRF sensors to IoT Hub in MS Azure. Here, the data is processed and sent to the appropriate queries. A [video](#) has been produced for the demonstration to see how data can be displayed in **Power BI**.

To get individual sensory values from the incoming string, we created a JavaScript function and SQL query.

javascript

```
// Convert Hex value to integer.
function main(hexValue) {
    return parseInt(hexValue, 16);
}
```

T-SQL

```

SELECT
*
,UDF.hex2Int(Substring (response,25,2)) AS Temperature
,UDF.hex2Int(Substring (response,28,2)) AS Light
,UDF.hex2Int(Substring (response,31,2)) AS Potentiometer
INTO
DataOUT
FROM
DataIN

```

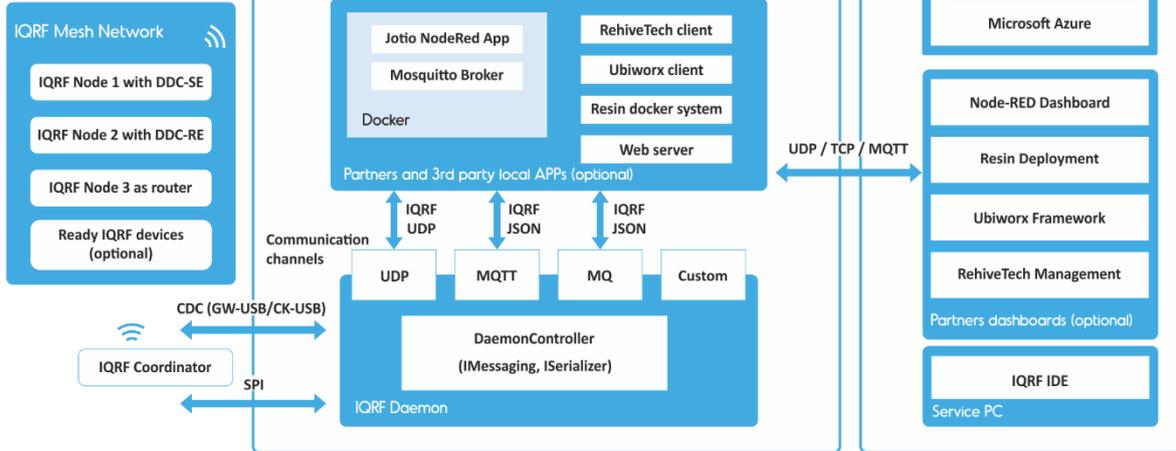
Microsoft Azure also offers other advanced tools to create, for example, a complete application for controlling the sensory network or prediction of future states in monitored objects.



Workshop

The [IQRF Summit 2017](#) will host an extensive workshop for IoT Starter Kit (Prague, from June 7th to June 8th). Summit participants may profit from a significant discount for the purchase of this product and other extensions.

You can buy this product on [IQRF Alliance e-shop](#) or with the help of [IQRF distributors](#).



IQRF Alliance, 777 775 735, alliance@iqrf.org, www.iqrfalliance.org