WIRELESS COMMUNICATION PLATFORM IQRF AND INTERNET OF THINGS

Zdeňka Kuchtová

Doctoral Degree Programme (3th), FEEC BUT

E-mail: klimesova@feec.vutbr.cz

Supervised by: Jaroslav Kadlec

E-mail: kadlecja@feec.vutbr.cz

Abstract: This paper describes possibilities of measured data transfer from sensors via wireless communication platform IQRF to the internet gateway and finally to the cloud data storage based on Microsoft Azure technology. Data are transferred through Microsoft Azure IoT Hub and filtered by Stream service and stored to SQL server. Stored data are visualized with Microsoft Power BI visualization service.

Keywords: IQRF, DPA, IoT, Communication framework

1. INTRODUCTION

Today, world is going to be connected. All values, that could be measured, are measured by smart sensors and results are stored in cloud. Everything that could by controlled remotely, is controlled. Terms like Internet of Things, Smart City, Smart buildings are quite often used in today’s world. They have at least one common fact, they need communication inside solution and with outside world. For this communication needs wireless or wired solutions are available. Each of them has some advantages and some disadvantages [1, 2, 3]. Within this paper, we will describe communication way from sensor to the cloud and their visualization to the end user. We will focus especially to the wireless communication with sensors.

There is a wide range of the wireless solutions on the market. Some of them are standardized some of them are proprietary, and of course, some of them are trying to be a standard. From the standardized solutions, we could select Wi-Fi, Bluetooth, RFID or ZigBee. From proprietary world MiWi, Z-Wave, IQRF and the others. Like always, all of them has some pros and some cons and are better or worse for different applications. In the rest of the article, we will stay with IQRF [4] communication platform, where a new communication modules and new features are already available.

The article is organized as follows. At the beginning, brief description of IQRF platform, developer tools and communication is presented. Second part contains a case study with IQRF and Microsoft Azure services. At the end of the article is summary and some future visions, acknowledgement and list of used references.

2. IQRF WIRELESS COMMUNICATION PLATFORM

Proprietary wireless communication platform IQRF was designed and developed by Czech company MICRORISC together with Brno University Technology. IQRF works on Sub-1 GHz base, it is a complete solution for design and development of the wireless application.

IQRF is platform based on communication modules. Different types of modules with different peripherals exist. All of them have the same operating system called IQRF OS. This OS was designed especially for fast and easy implementation of wireless communication to broad range of applications and scenarios.

Functions of IQRF OS could be divided into three main parts. The first part contains functions related to wireless communication. They allow to send and receive packets to and from wireless network. The second part are functions for peripherals available on module. Part of them are for SPI communication bus. SPI is the main interface for connection between IQRF module and other systems. The last part are functions for MESH networking.

MESH networking, in IQRF called IQ MESH, is one of the main advantages of IQRF communication platform. Whole MESH network is created with support of IQRF OS. Nodes are fully automatically bonded to network by either user interaction or IQRF OS extension. IQRF OS supports up to 250 routing points in one network. One coordinator can control up to 65000 network nodes. The whole network is possible to extend with the node, which works in one network like node and in a second as network coordinator.

Part of the program and data memory in microcontroller on IQRF module is dedicated for user application. The amount of available space is limited, but is enough for small applications or for definition of communication interfaces to control systems.

IQRF OS could be extended by plugins. Plugin in this meaning is the other user designed software that adds the new functions to OS. For plugins is dedicated other part of the program memory. These plugins could be sold separately for other developers without need to share source code. Block diagram of the most often used IQRF wireless communication module is shown in figure 1.

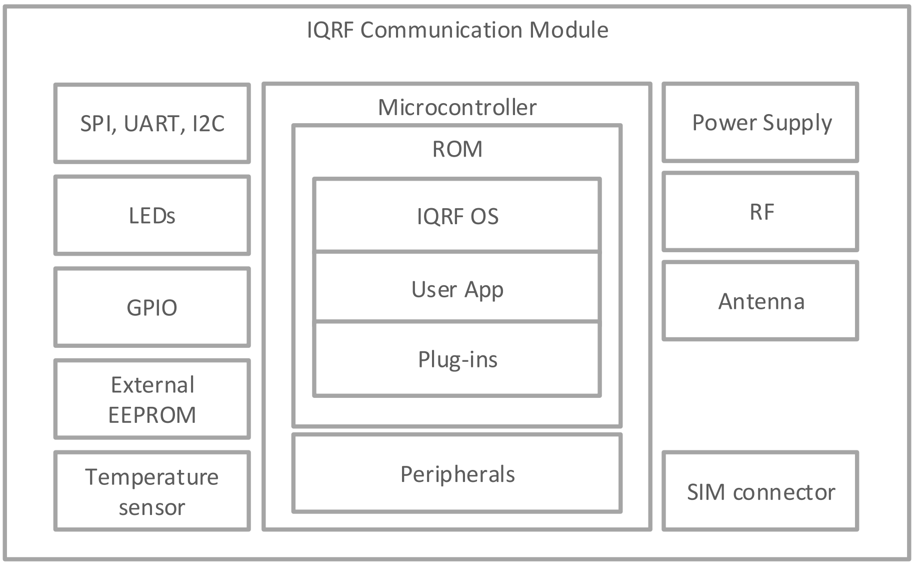


Figure 1: The main part of IQRF communication module.

Every node is addressable by a number defining its private address or by a broadcast message. In case of broadcast messages, no response from nodes is available nor expected. Interesting part of IQRF OS called Fast Response Commands FRC allows reading two bits or one byte from filtered list of nodes by one command. Response time is minimized and depends on network structure.

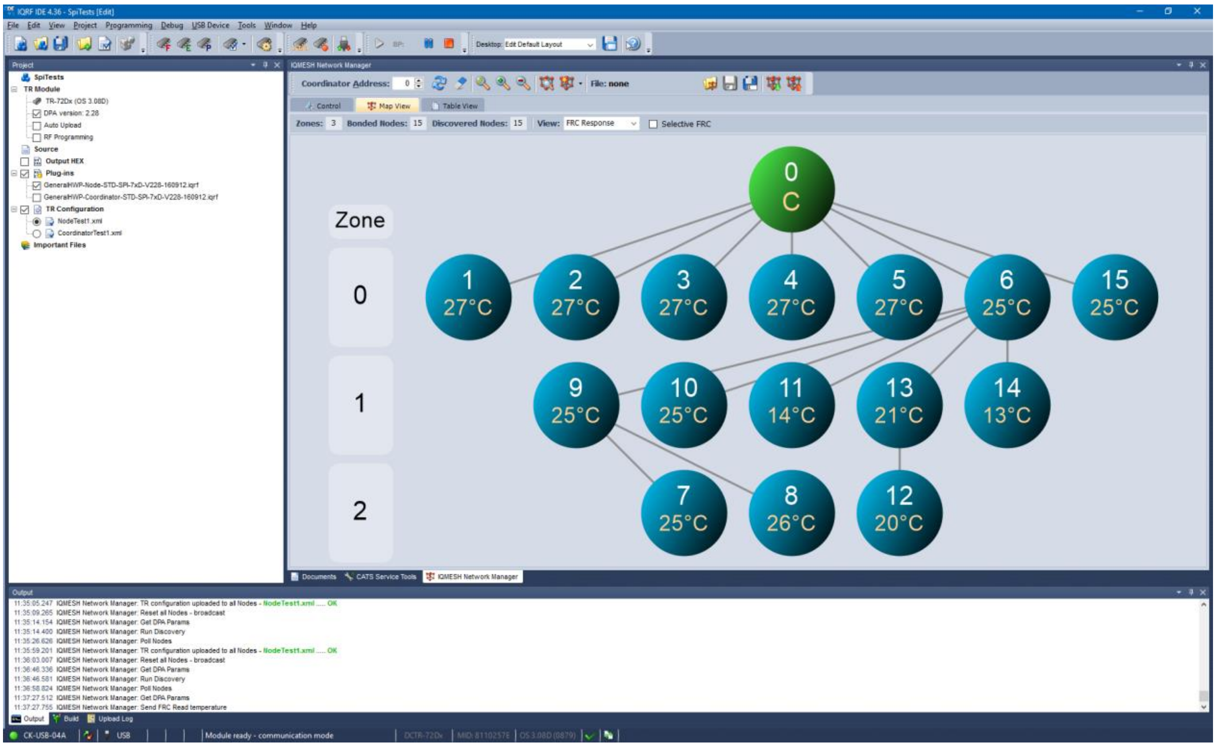
The most useful OS extension is implementation of Direct Peripheral Addressing communication framework (DPA). With this framework, all available module peripherals can be accessed and controlled without the need to write any part of code on module side. Everything is done by a set of commands sent to coordinator via a local interface or in some special occasions directly via a local interface of the node. In the case of necessity, the developer can extend the functionality of DPA via DPA Custom Handler. In this situation framework provides events for each part of message processing. For example, event before a response is sent, or event after resending and so on. Also, some limited access to interrupt function is available.

3. IQRF DEVELOPMENT TOOLS, GATEWAYS AND SUPPORT

Developers' support starts with IQRF IDE. This integrated development environment supports the whole development process within IQRF platform. The developer uses IQRF IDE for preparing software in C, compilation and uploading it to the module. Upload could be done via USB programmer or wirelessly over the wireless network.

Software Communication with module could be tested via integrated terminal. IQRF module connected in programmer works in release mode, sends and receives data on SPI bus. All data are shown in IDE. There is also support for macros definition and repeated request sending.

DPA framework is supported via IQMESH network manager, where is possible to bond modules to the coordinator, set all DPA parameters, analyse network and so on. One part of IQMESH network manager is dedicated to backup and restore the whole module. Network manager shows all information about module connected to the network, like network address, zone, MID, OS version, Hardware profile, all available peripherals and setting and many others. IDE also supports sending DPA command and analyse response. IQRF IDE window is shown in figure 2.



SpiTests
TR Module
TR-72Dx (OS 3.08D)
☑ DPA version: 2.28
☐ Auto Upload
☐ RF Programming
Source
☐ Output HEX
☑ Plug-ins
☑ GeneralHWP-Node-STD-SPI-7xD-V228-160912.iqrf
☐ GeneralHWP-Coordinator-STD-SPI-7xD-V228-160912.iqrf
☑ TR Configuration
● NodeTest1.xml
○ CoordinatorTest1.xml
Important Files

Coordinator Address: 0 File: none

Zones: 3 Bonded Nodes: 15 Discovered Nodes: 15 View: FRC Response ☐ Selective FRC

```
11:35:05.247 IQMESH Network Manager: TR configuration uploaded to all Nodes - NodeTest1.xml ..... OK
11:35:09.265 IQMESH Network Manager: Reset all Nodes - broadcast
11:35:14.154 IQMESH Network Manager: Get DPA Params
11:35:14.400 IQMESH Network Manager: Run Discovery
11:35:26.626 IQMESH Network Manager: Poll Nodes
11:35:59.201 IQMESH Network Manager: TR configuration uploaded to all Nodes - NodeTest1.xml ..... OK
11:36:03.007 IQMESH Network Manager: Reset all Nodes - broadcast
11:36:46.336 IQMESH Network Manager: Get DPA Params
11:36:46.581 IQMESH Network Manager: Run Discovery
11:36:58.824 IQMESH Network Manager: Poll Nodes
11:37:27.512 IQMESH Network Manager: Get DPA Params
11:37:27.755 IQMESH Network Manager: Send FRC Read temperature
```

CK-USB-04A | USB | Module ready - communication mode | DCTR-72Dx | MID: 81102576 | OS 3.08D (0879)

Figure 2: IQRF Integrated Development Environment.

IQRF platform offers a wide range of communication gateways. For remote access to IQRF network Ethernet, Wi-Fi and GSM gateways are available. Gateways based on third party devices with IQRF extensions, like Arduino, Gemalto, chipKIT, Raspberry Pi and many others are also available.

One of IQRF OS extension is DPA. It is part implement directly in the module. Each module is composed of different peripherals (i.e. standard one, prepared by MICRORISC, or users peripherals, prepared by the user). Control of this peripherals in DPA is done by a set of commands, specific for every peripheral, but the structure of command is standardized for the whole DPA. Commands are sent in communication packet with defined protocol, wirelessly or directly to module local interface. This is module point of view. However, on the other side is some control device that needs to communicate with modules. This is exactly the place where DPA communication framework is used.

DPA communication framework prepares packets, sends them and parses response for the higher user or, in this case, programmer comfort. Available are three versions of DPA Communication Framework. One is prepared in Java and is part of IQRF SDK. This framework is suitable where Java is available. It can communicate with IQRF gateways in CDC mode directly via SPI on boards like Raspberry or Gemalto and similar ones. The framework is available with source code via GIT Hub. For systems, where Java is not available is could be used C++ versions of the framework. These are designed and developed directly by MICRORISC.

4. A CASE STUDY

In this use case, we create simple temperature sensor unit. In fact, we would like to measure at least three temperatures. Two of them are temperatures up to 200 °C, temperature below zero are not important for as. Third temperature sensor should measure temperatures up to 800 °C and again zero is limit on the other side. These sensors should show measured values directly on the display and store them to the cloud for remote access. Communication scenario is shown in figure 3.

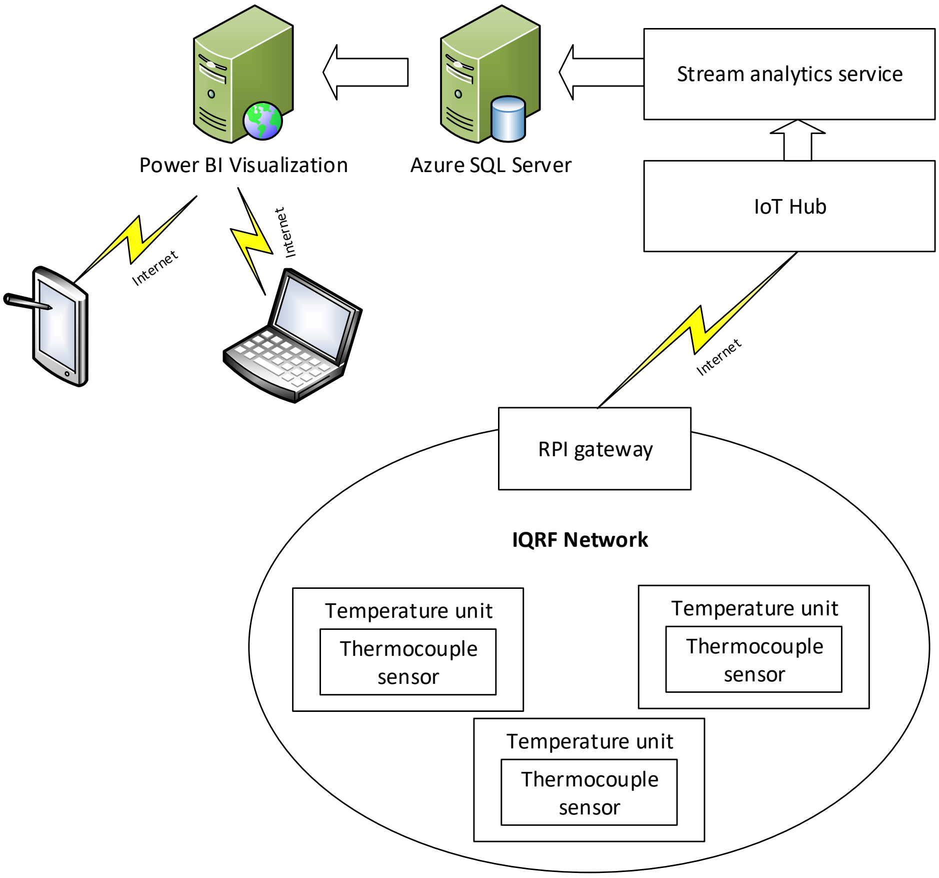


Figure 3: Communication scenario used in the case study.

For temperature sensing we will use thermocouple, type K. Response from thermocouple will be digitalized by Microchip MCP9600 and via I2C bus stored to the PIC16 family microcontroller. Measured temperature will be shown on connected LCD display and through IQRF module sent to network coordinator. Block diagram of temperature sensing unit is in figure 4.

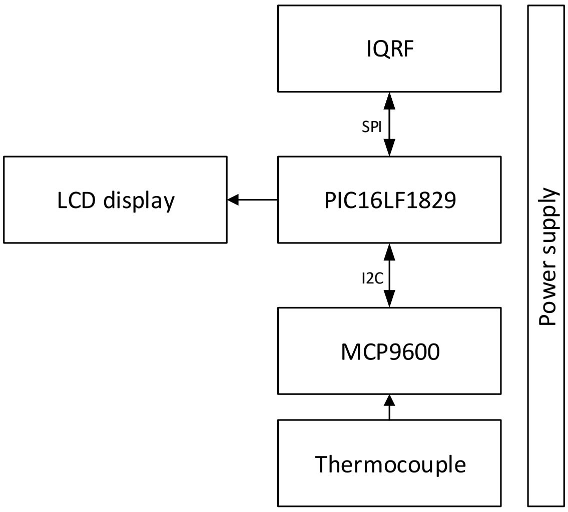


Figure 4: Block diagram of temperature sensor unit.

IQRF network coordinator is part of communication gateway. For this scenario, we prepared gateway based on Raspberry Pi 3 device with Microsoft Windows 10 Core operating system. Software for gateway was prepared in C# for Microsoft .NET framework. Communication with IQRF network is done via .NET IQRF Framework designed by 2KSys company [4]. Gateway sends measured temperatures to Microsoft Azure cloud via IoT Hub services. On the server side data are pre-processed by Azure Stream services and stored to Power BI Interactive Visualization Tools. Result of visualization is shown in figure 5.

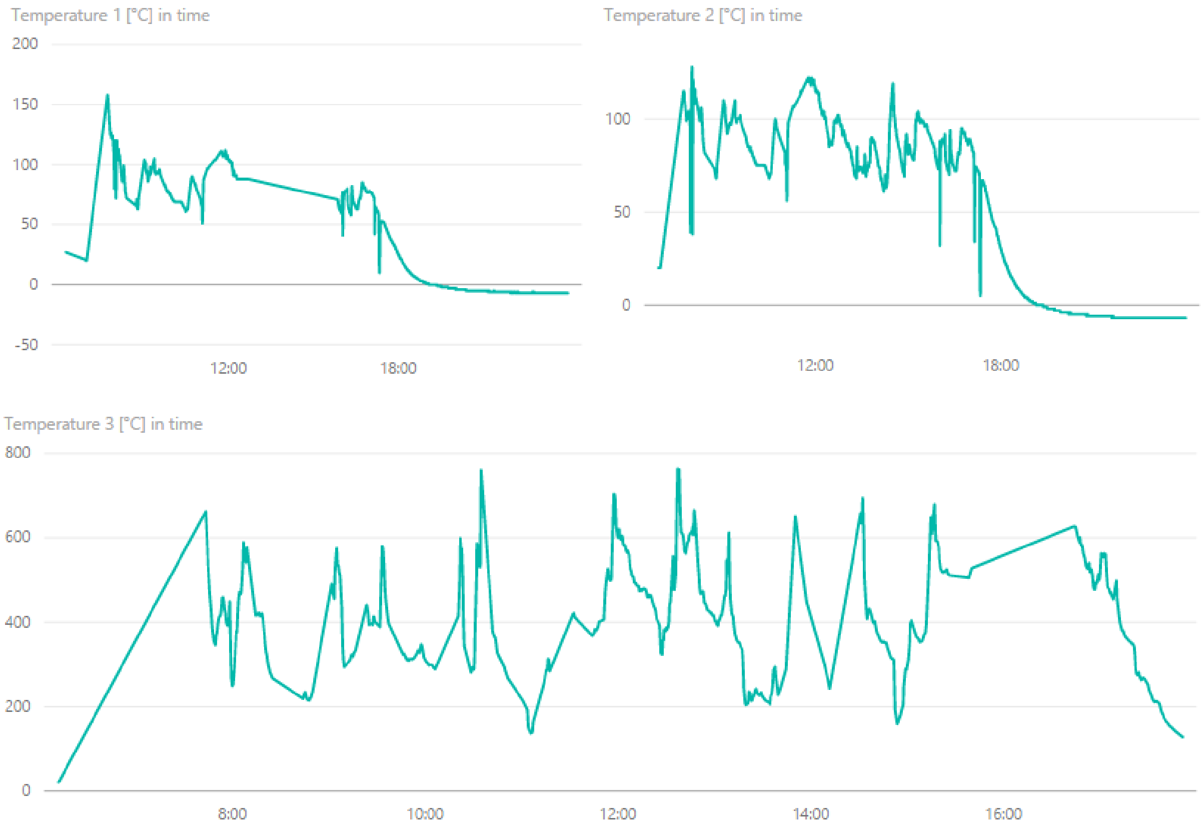


Figure 5: Visualization of measured temperatures by Microsoft Power BI services.

5. SUMMARY AND FUTURE WORK

This paper briefly describes IQRF platform, designed and developed by MICRORISC company and BUT. In the article, the main features of the platform were described, especially IQRF communication modules, development tools, gateways, cloud and DPA communication framework. In the case study, we showed whole temperature measurement solutions application with thermocouple measurement sensors, IQRF communication and Microsoft Azure storage and Power BI visualization.

In the future, we assume to stay in touch with MIRCORISC developers and work on new scenarios and applications for IQRF. We are also involved in communication framework development and gateways testing.

6. ACKNOWLEDGEMENT

Research described in this paper was financed by the National Sustainability Program under grant LO1401. For the research, infrastructure of the SIX Center was used.

7. REFERENCES

[1] O. Bello; S. Zeadally, "Intelligent Device-to-Device Communication in the Internet of Things," in *IEEE Systems Journal* , vol.PP, no.99, pp.1-11 doi: 10.1109/JSYST.2014.2298837.

[2] D. Guinard, V. Trifa, F. Mattern, E. Wilde, 'From the Internet of Things to the Web of Things: Resource Oriented Architecture and Best Practices'. In: Dieter Uckelmann, Mark Harrison, Florian Michahelles (Eds.): Architecting the Internet of Things. Springer, pp. 97–129, New York Dordrecht Heidelberg London, 2011.

[3] V. Sulc, R. Kuchta, and R. Vrba, "IQRF smart house—a case study," in Proceedings of the 3rd International Conference on Advances in Mesh Networks (MESH '10), pp. 103–108, July 2010.

[4] Dpa Framework. *2KSys s. r. o. Web* [online]. Brno: 2KSys s. r. o., 2015 [cit. 2017-02-10]. Available: http://www.2ksys.cz/Products/DpaCfDemoFree/cs.

[5] MICRORISC. IQRF - wireless technology. MICRORISC s. r. o., 2015 [cit. 2017-02-10]. Available: www.iqrf.org.