IQRF®

Reliable by definition

# IQRF® Technical Guide

# IQRF overview

IQRF provides a complete platform for **wireless** connectivity. It includes **hardware** (transceivers, gateways, repeaters, accessories, development tools, …), **software**, **protocols**, **standards**, **alliance**, **support**, and **services**.

IQRF core values are **Industrial reliability**, True low power, Ultimate security, **Simple integration** and **Interoperability**.

## Technology characteristic

**IQRF** is a **wireless** (**RF**) technology for the license-free **ISM** bands (currently mainly **868 MHz** and **916 MHz**). It requires **no infrastructure by external providers**, **no license** and **no carrier fees**.

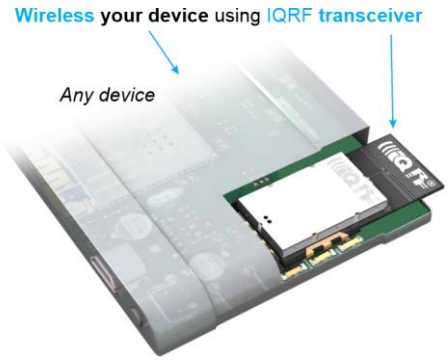
It is focused on **low speed** (20 kb/s), low power (battery operated), and **low data** volume **packet-oriented** bidirectional communication (with packets up to 64 B).

## RF range

- In buildings — Tens of meters
- In open space — Hundreds of meters
- In networks — Up to several kilometers

## Generic usage

IQRF can **make any device wireless**. It can be used **with any electronic equipment**, whenever there is a need for wireless transfer, e.g. remote control or monitoring of remotely acquired data. Typical IQRF usage is **IoT**, e.g. telemetry, industrial control and automation of buildings and cities (lighting, parking, etc.). It excels in street lighting.




## TR transceivers

IQRF is based on the RF **transceiver** (**TR**) which allows any devices to communicate wirelessly with each other. The functionality is quite **generic** and depends only on the **software uploaded** in the TR.

**Two SW layers** are provided with TRs:

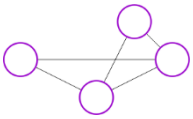
- **OS** (Operating System): **lower layer**, providing powerful support for all **TR resources**.
- **DPA** (Direct Peripheral Access): **optional higher layer** with enhanced **network** functionality.

**Specific functionality** can be achieved by extending with **additional software** that can be **written** in **C** language by the **user's application engineer**.

## Communication topologies

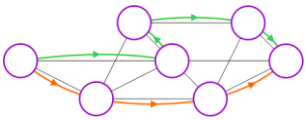
### Point-to-point

Communication can be simple (**point-to-point**, bidirectional) between **two** or **more** devices at the same level (**peer-to-peer**) and within **direct RF range** of each other. However, such a simple IQRF application often means cracking nuts with a steam hammer. IQRF's greatest strength lies in “real” **networks**.



### Mesh network

The most powerful and reliable network type is **mesh**. It enables packet deliveries even between devices **out of the direct RF range**. In mesh, packets are propagated (**routed**) to their destination by **hopping** over other network devices that are in range of each other. The more possible hops between nodes, the more **redundant paths** are available. This allows to bypass obstacles and unpassable areas which makes mesh communication **robust** and **reliable**.



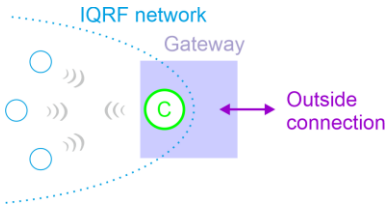
## IQMESH network

The IQRF **mesh** routing protocol is called **IQMESH®**. It is a completely unique approach using sophisticated **routing techniques** including directional flooding algorithm as well as numerous unique and often **patented features**. This ensures an outstandingly **efficient**, **deterministic**, **collision free** and **extremely reliable** traffic.

Routing capability can be provided in two ways:

- In addition to normal operation, **any Node** device can **concurrently route** packets (**in background**) for other Nodes.
- Moreover, single-purpose routing devices (so called dedicated repeaters) are available.

## Gateway




Typically, the IQMESH **Coordinator** is hosted inside a gateway (**GW**). It is a central wireless network device designed as an **interface** between the IQRF network and the rest of the world. It **manages all devices** in the entire IQMESH and provides **communication from/to outside**. It supports various connectivity standards, enabling **Internet** connection and access to any existing **cloud**.

GW is mostly based on a small single-board computer and **Linux** open-source **IQRF GW Daemon**. Most IQMESH applications use a **ready-made** gateway, but your application engineer can create your own **specific** one.

## GW Daemon

IQRF Gateway Daemon is a powerful **open source** SW package allowing to easily create an **IQRF gateway** with **Internet** and **cloud** connectivity from a **Linux** machine (typically **Raspberry Pi, Linux PlugPC, BeagleBone** or similar). It can be connected to a TR via **SPI**, **UART**, or **USB CDC**. The TR implements an IQMESH network **Coordinator** under the **DPA** layer. The Daemon utilizes multiple communication channels **UDP**, **MQTT** and **WebSocket** and can be managed through the IQRF Gateway Webapp.

## Cloud

The easiest but powerful IQRF to Internet connection is a **cloud**. The cloud server provides a transparent bidirectional encrypted channel between end devices in IQRF network and worldwide users. **Any existing third party cloud** (e.g. AWS, IBM Cloud, Azure or InteliGlue) can be used.

# IQRF network application example


---

# Development tools

## Hardware

- Generic **modular kits** for rapid development and solderless prototyping. The **two fundamental ones** are CK-USB-04A (programmer and multifunctional kit) and DK-EVAL-04A (end node kit).
- Tools for easy installation, service and mainenance, either as dedicated HW devices (e.g. IQuip) or implemented by CK-USB-04A (e.g. IQuip).

## Software

- IQRF IDE is a **fundamental** SW for all development and service work on IQRF applications.

# Application support

**Free** application support and a lot of services are available. Examples, reference designs, video tutorials, training, FAE, consultancy, development support, troubleshooting, …

# IQRF Alliance

An association of entities involving in products with IQRF inside. It defines standards for **interoperability** of **different products from different manufacturers**, enables **synergies** between its members and operates a common IQRF Market Place.

# Open IQRF Standard

After two decades on the market, the **IQRF gets standardized**, allowing everyone to use and implement all technical achievements and reliable protocols protected by **dozens of patents** under one **royalty-free license**.

This allows **each manufacturer** to **develop** and **produce IQRF-compatible transmitters** themselves.

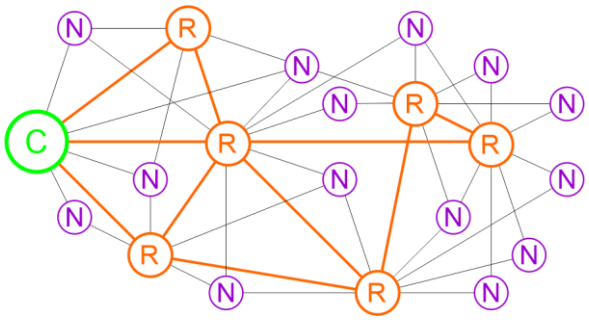
# Networks

## IQMESH

The IQMESH network contains a **Coordinator** and a number of **Nodes** (other network devices equipped with TR). Each Node may (or may not) also perform routing.

There are different types of Nodes:

- Regular end device **non-routing Node**
- Regular **Node** intended for primary end device function but additionally **routing in the background**
- **Node** primarily **dedicated** only **to routing** (called a **repeater**).




C **Coordinator**

R **Routing** Node:
- Either an end device additionally routing in the **background**
- or a **dedicated repeater**

N **Non-routing** end device Node

Routing Nodes (together with the Coordinator) form a **routing structure** (routing backbone, shown in orange and green in the diagram above). It is automaticaly created by invoking a process called Discovery.

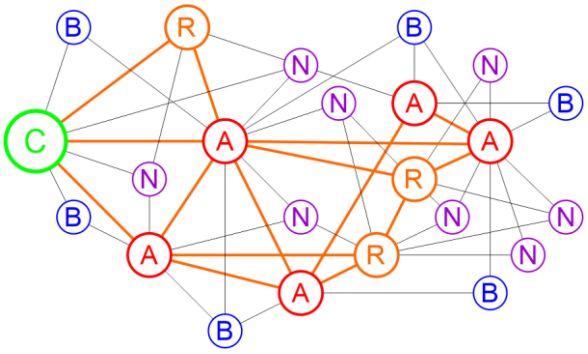
IQMESH is primarily intended for more or less **static** systems where neither routing Nodes nor obstacles move. After possible significant changes in topology (adding, removing, or relocating **routing** Nodes, changes in obstacles, etc.), the network must "**relearn**" the routing paths (by reinvoking Discovery). **Non-routing Nodes** can move **unrestrictedly**, but must stay in direct range with at least one (but preferably more for redundancy) repeater.

## Beaming

In order to operate end devices (typically sensors) with extremely low power consumption, IQRF supports so-called **Beaming** Nodes.

A Beaming sensor (e.g. IQD-SE02-04A) is always in **sleep** mode, except for brief moments when it periodically or upon some event measures and asynchronously unidirectionally sends data to all aggregation repeaters in direct RF range.

An Aggregating repeater (e.g. IQD-REP04) is a routing device (either background routing or a dedicated repeater) that not only provides functions as a conventional routing Node, but also serves as an interface for **collecting data from Beaming sensors**. The aggregated data is then forwarded from it to the Coordinator in the standard synchronous way. Each beaming sensor must be in direct range with at least one (but preferably more for redundancy) aggregating repeater.




C **Coordinator**

A **Aggregating repeater**

R **Routing** Node (non-aggregating):
- Either an end device additionally routing in the **background**
- or a **dedicated repeater**

N **Non-routing** end device Node

B **Beaming** Node

# IQMESH routing

**Performance** and **reliability** of any wireless mesh network especially depend on **routing algorithms**. IQMESH supports **various routing techniques**, numerous unique and often **patented features** and selectable parameters to fit the functionality according to the project-specific needs. The **DPA framework** solves the routing transparently. **Routing structure** is discovered fully **automatically**.

**Pure IQMESH** (except Beaming and Local FRC) is strictly **synchronous**. A routed packet is then propagated through the network according to the TDMA (Time Division Multiple Access) rule. Specifically, in IQMESH, **each routing Node routes in the time slot corresponding to its VRN** address. IQMESH packets are propagated through the whole network natively by the directional flooding algorithm.

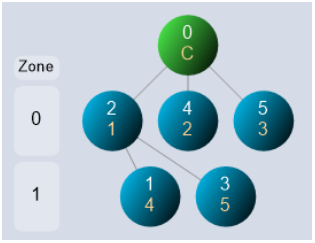
There is great efficiency in simplicity. All results of bonding and Discovery (including the table of VRNs) are stored in only a few internal data arrays in the Coordinator. No vectors or tables are stored in routers, nor are they delivered in packets. Each router knows only its position in the network and thus its dedicated time slot. **Self-healing** (finding an alternative path in case of some broken links) is provided directly (by itself), **all possible paths are routeded at once**, not on subsequent requests after the first attempt failed. This ensures an outstandingly **efficient**, **deterministic**, **collision free** and **extremely reliable** traffic with **high robustness** and **high throughput**. IQRF is **reliable by definition**.

# Discovery (creating the routing structure)

During the inclusion of the Node in the network (so called bonding), the Node gets assigned a **Logical address**. It is intended for **identification** and **addressing** the given Node **by the user**. In all subsequent communication, the user only needs to specify the Logical addresses of the recipients and the packets are delivered automatically.

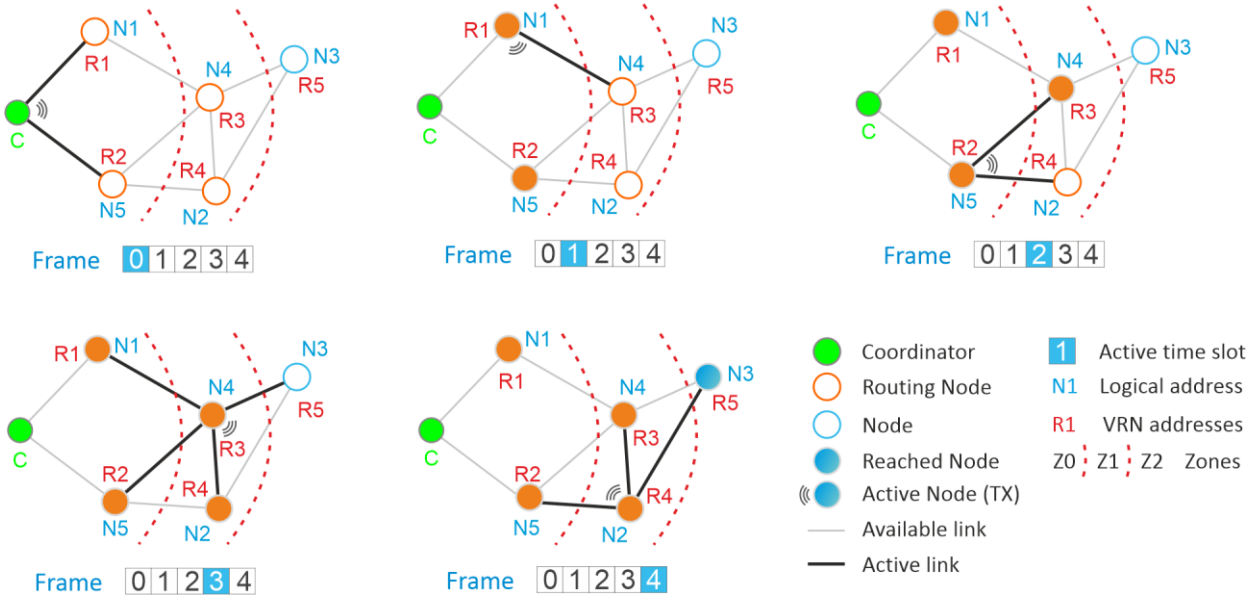
However, for internal routing algorithms, Logical addresses are not suitable, but a different internal arrangement is appropriate. The **routing structure** reflecting the topology of Nodes with routing capability is created automatically (in real time and even in low power mode, if needed) by the Discovery process. Its goal is to find and virtually **re-address** all **routing Nodes** in **ascending order** according to the accessibility („distance“ in hops) from the Coordinator. The resulting addresses are called **VRN**s (Virtual Routing Numbers).

During the Discovery, the discovered Nodes are divided into routing **Zones** (groups of Nodes in the same „distance“ (number of **routing hops** to reach the Node) from the Coordinator. Zone 0 contains all the discovered Nodes in direct RF range with the Coordinator (0 routing hops), etc.
Information about Zones is not needed for common users, but can be useful for installation of end devices or network maintenance to learn about possibilities to improve network performance. E.g., it is convenient to have a sufficient number of routing Nodes in Zone 0. IQRF technology provides tools to vizualize the Zones:




## Packet propagation

A packet is typically propagated from the Coordinator to the Node for control purposes or back for a response or data collection. The forward routing (from the Coordinator to the Node N3 (either routing or not) supported by routing Nodes N1, N5, N4 and N2) using the **directional flooding** algorithm works as follows:

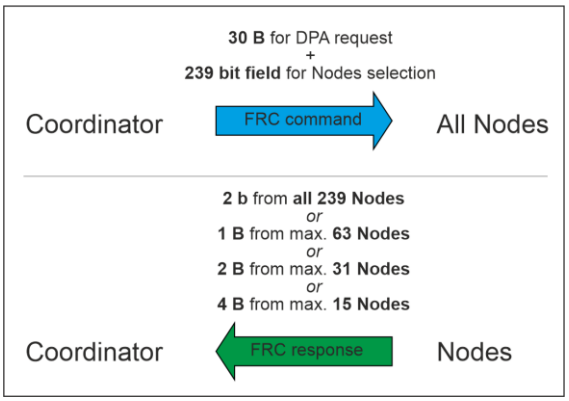


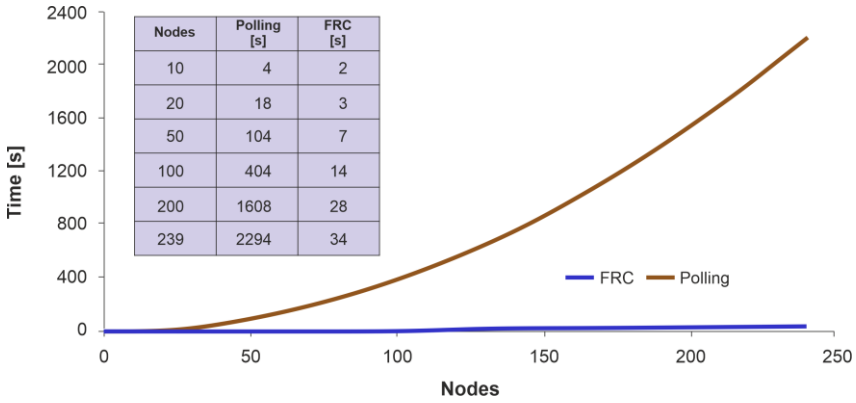

In this example, Node N3 (in Zone 2) received the packet from both routing Nodes (N4 and N2) in Zone 1.

**Video animations** (including self-healing when bypassing obstacles): » Unicast » Broadcast

# FRC – Fast response command

Besides of other sophisticated networking features, IQRF transceivers have implemented the powerful FRC (Fast response command) routines. They allow the **fastest** network **control**, management and **data collection** (even in low power mode, if needed). It is a patented method how to **send a command** from the Coordinator **to all or selected Nodes** and **receive responses** including **small data** collected by individual Nodes in outstandingly short time (e.g., from all **239 Nodes** in **less than 40 s**). FRC is **much faster** (even by orders of magnitude) than **polling** individual Nodes one by one:




| Nodes | Polling [s] | FRC [s] |
|---|---|---|
| 10 | 4 | 2 |
| 20 | 18 | 3 |
| 50 | 104 | 7 |
| 100 | 404 | 14 |
| 200 | 1608 | 28 |
| 239 | 2294 | 34 |

» Video animation

## FRC types

### FRC

FRC **synchronously** sent from the **Coordinator** to all or selected Nodes.

Typical usage:

- **Control:** Fast sending a command to multiple Nodes and checking the results (e.g. an **acknowledged broadcast**).
- **Telemetry:** Fast collecting of small data from multiple Nodes (**sensors**, etc.).

### Local FRC

Local FRC is an FRC **asynchronously** invoked **from a Node** to **control other Nodes** in **direct RF range**. Quickly, without lengthy propagation of the packet through the entire network. Thus, e.g., controllers (pushbuttons, switches, motion sensors, …) can control and monitor actuators (lights, blinds, …) **with no significant delay**.

### Offline FRC

Offline FRC is intended for collection data from Beaming sensors by Aggregating repeaters. It enables to operate sensors with outstandingly low power.

# Network creation

**User-friendliness** and **security** when creating a network is one of the key challenges of wireless technologies.

## Bonding

It is possible to include (**bond**) a Node in IQRF network (even in low power mode, if needed) by four methods.

### Bonding via IQuip

The most user-friendly way of bonding an end Node device equipped with the NFC communication is via IQuip, the **IQRF NFC tool** IQD-NFC-01.

- The Node to be bonded need **not** be **in direct RF range with any other network device**.
- **No action** is needed on the Node side. The Node will obtain all the necessary information from the IQuip (paired with the Coordinator).
- Afterwards, bonding must be finalized by Autonetwork.

### Smart connect

- For Smart connect, **no action** is needed on the Node side.
- Based on IBK (Individual Bonding Key) and MID (Module ID) codes, which are unique, fixed, and stored in each TR from the factory.
- IBK and MID must be imported into the Coordinator in advance (in the IQRF Smart connect code format). It is open and can easily be generated (e.g. in IQRF IDE).
- The easiest way to transfer this into the Coordinator is via the **QR code** using the freely available mobile application IQRF Network Manager for Android.

## Local bonding

- For Local bonding, the Node to be bonded must have the **Access password** (the same as it is used by the Coordinator) specified **in TR configuration**.
- The Node to be bonded must be **in direct RF range with the Coordinator**. After the bonding, the Node can be relocated to its final location. If the Node has **routing enabled**, then the Discovery **must be invoked**.
- An **action** (bond request selected from the DPA menu) is **required** on the Node side.

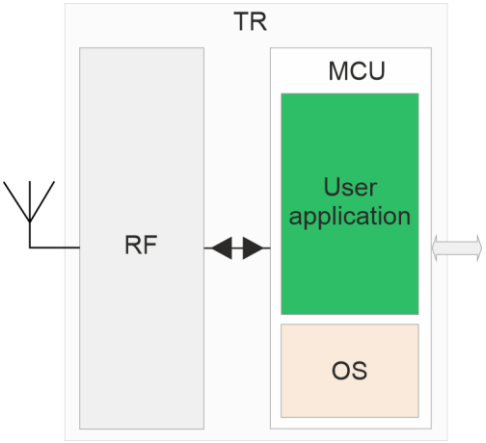
## Autonetwork

IQRF network can even be created (bonding, Discovery) fully automatically by the Autonetwork. It can be launched from **IQRF GW Daemon** or **IQRF IDE**.

- **No action** is needed on the Nodes side.
- All Nodes to be bonded must have the **Access password** (the same as it is used by the Coordinator) specified **in TR configuration**.

# Operating system

IQRF **OS** is the **lower system SW layer** that manages all **TR resources** and provides powerful services for applications.



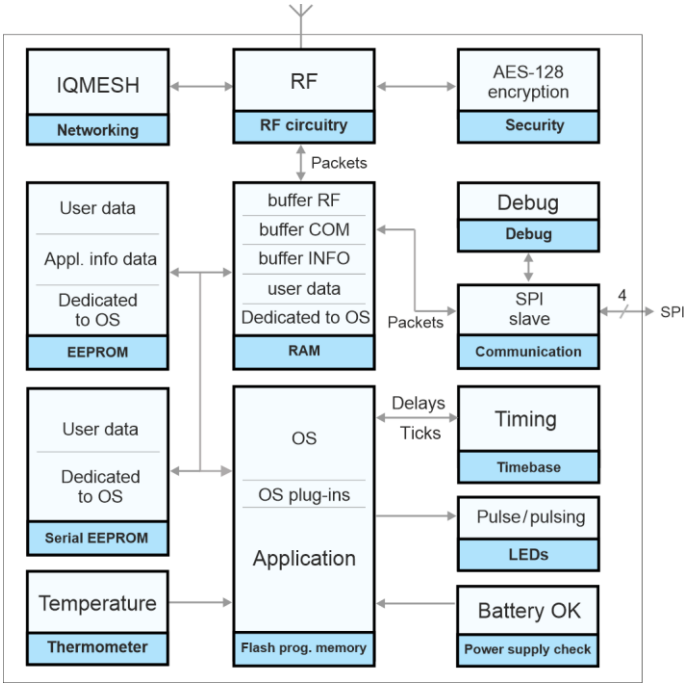

OS is directly **built in the transceivers**, unlike other wireless platforms where a SW solution stack (in the order of 10 thousand of lines) has to be compiled with the user program. This (and many other advantages) make IQRF outstandingly easy to use. **Peer-to-peer** (**non-mesh**) applications can be easily programmed **directly above the OS** (without DPA) in **C language** by a user's software engineer. For example, the code sending a 10 B long RF packet (when the user payload data is already pre-prepared in the buffer dedicated to RF communication (buffer COM, see the diagram below)) to all TRs in RF range is simple and intuitive:

```
PIN = 0;        // Peer-to-peer
DLEN = 10;      // 10 B of payload data will be sent
RFTXpacket();   // Transmit the packet
pulseLEDG();    // Green LED flash indication after transmission
```

In this approach (fully programmed above OS) it is possible to write the whole peer-to-peer application.

## Crucial features

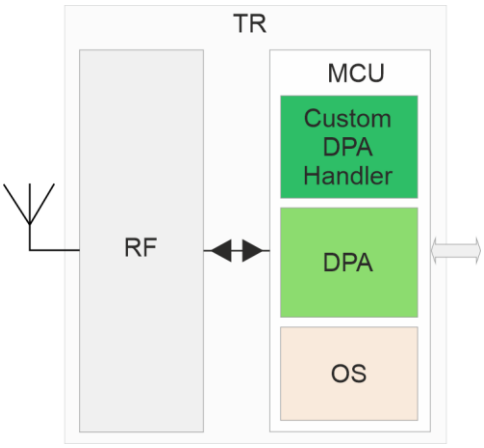
- Versatile **RF** communication
- Light but efficient **license free IQMESH protocol**
- Considerable **flexibility**. RF and other parameters are **configurable** or **programmable**.
- Powerful routines for network creation and maintenance (Bonding, Discovery, …)
- Low power modes: RF (STD, LP, XLP), Deep sleep, Sleep, …
- **Buffer-oriented** communication including serial (**SPI**) connectivity running **in OS background**
- **Access** to RAM and EEPROM **memories** (byte, block and buffer support)
- Industrial standard AES-128 **encryption**
- **Debug**: Break, watch and continue
- Efficient development tools are available.
- IQRF transceivers allow **upgrading** to a higher **OS** version (typically done by the user).
- Application SW or a plug-in can be **uploaded** in TR even **wirelessly** (RFPGM). Thus, **all Nodes** of a given type in the entire network can be **programmed in one stroke**.
- …

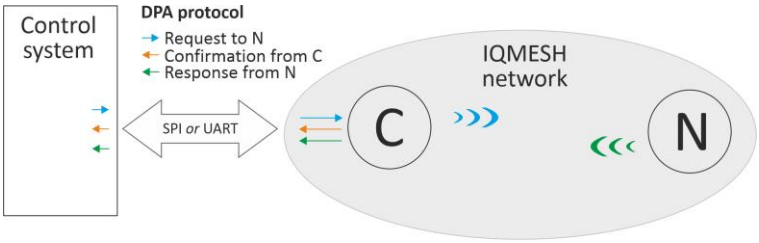



These features are supported by a comprehensive set of optimized functions.

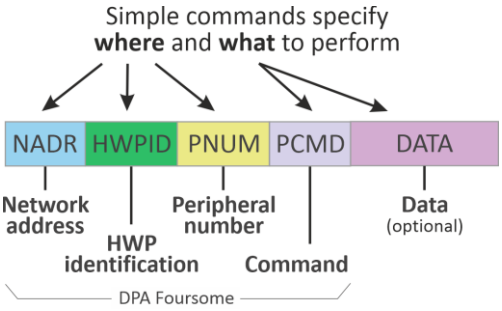
# DPA

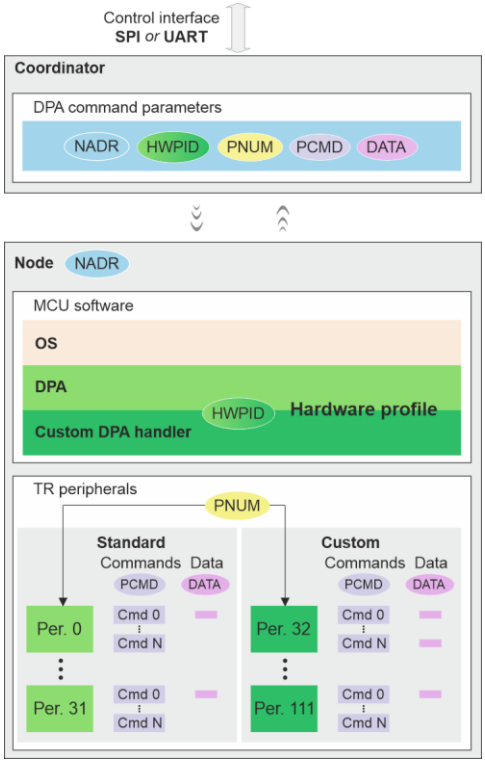
IQRF DPA (Direct Peripheral Access) is the **higher system SW layer** that provides massive support for networks. It is **optional** (but mandatory for IQMESH). The **DPA framework** implements the simple byte-oriented **DPA protocol** above the **IQMESH** protocol which supports network communication and allows to control **services** and **peripherals** of network devices (Coordinator and Nodes). The network is controlled by sending **requests** and receiving **responses** via the **control interface** (using standard wired SPI or UART).




DPA commands use a fixed, simple and intuitive format with 4 mandatory parameters (so-called **DPA Foursome**) and an optional data block.



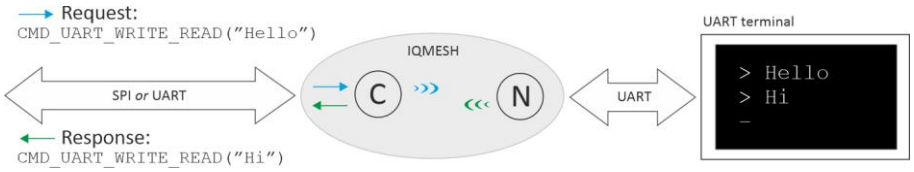

DPA is distributed in the form of a ready-to-use **plug-in** to be uploaded in TR instead of an application program. The user-specific functionality can than be achieved by an extension called **Custom DPA Handler**. It can be written in **C** (similarly as above the OS without DPA) by a user's application engineer. For this, IQRF framework also provides an additional support with the DPA API functions. Development above the DPA is thus simple and rapid. Moreover, many handlers implementing various standard tasks for the respective HW circuitry (so-called **hardware profiles**, **HWP**) are available ready-made, so some applications can be implemented even **without programming**.

DPA also offers a number of additional features for installation and maintenance. **DPA menu** provides uniform and simple control of Node devices. **Backup** and **Restore** enables seamless replacement of a broken Node and restoration a complete network in case of broken Coordinator. Other useful features are Enumeration, several SW service and test tools, firmware upgrade, etc.

DPA enables worldwide **interoperability** of products with IQRF inside. This is ensured by IQRF Alliance. The IQRF Interoperability Standard specifies standard DPA commands, so the compliant devices are not only compatible (working in **one network**) but also **interoperable** (controlled by the **same commands** and understanding the **same data formats**). Thus, products from different producers can plug-and-play work together. Each interoperable product is identified via its HWPID on IQRF Repository.

## Example using UART

UART peripheral is typically used to control an external device connected to the Node device via the HW UART interface. The following diagram shows an example where the Coordinator writes a text "Hello" by UART Write & Read DPA request to the UART peripheral at Node. There is a terminal (external device) connected using UART to the Node. Text "Hello" is then displayed at the terminal and text "Hi" (in this example the terminal automatically answers "Hi" to "Hello") is read back to the Coordinator.



# IQRF features

## RF parameters

- Selectable RF parameters
  - RF **bands**: mainly **868** MHz and **916** MHz
  - RF **channels**: **62** for 868 MHz and **189** for 916 MHz
  - RF **output power**: up to **10 mW**, selectable in 8 steps
  - **Adjustable** receive **sensitivity**. Incoming signal level and quality check and filter, for noise immunity, finding a free channel (e.g. for LBT or FHSS), etc.
- RF **bit rate**: **19.8 kb/s**
- RF **range**: Tens of meters in buildings, hundreds of meters in open space, kilometers in networks

## RF communication

**Bidirectional**, **payload:** up to **64 B**

- Unicast - Packet sent to a single Node, with response.
- Broadcast - Packet sent to all Nodes, without responses.
- Beaming - Data sent from sensors with outstandingly low power (enabling battery lifetime for years).
- **FRC** - Sending a command to Nodes and receive responses with data collected from them in outstandingly short time.
  - FRC - Synchronously sent from the Coordinator to all or selected Nodes.
  - Local FRC - Asynchronously sent from a Node to control selected Nodes in direct RF range
  - Offline FRC - Collection data from Beaming sensors by Aggregating repeaters

## IQMESH features

- Up to **240 devices** in the network (1 Coordinator + 239 Nodes)
- Up to **240 hops**
- Every TR can work either peer-to-peer or in mesh network as a **Coordinator**, **Node** (common or **beaming**), background **router**, or **repeater** (common or **aggregating**)
- Every Node can additionally route packets in background
- Routing in real time:
  - Max. 60 ms per hop for STD network, based on the payload
  - Max. 100 ms per hop for LP network, based on the payload
- Various routing algorithms (Full mesh, Optimized mash, Star, …) to tune throughput, response time, consumption, etc.
- Dynamic timing (selectable number of hops and time slot duration according to needs specific for given application, to increase throughput and reduce power consumption and noise)
- Low latency (**STD**) as well as low power (**STD+LP**) network types supported
- Three independent encryptions by **AES-128**: Bonding, Networking, and User. All network communication is **automatically encrypted**.
- **Autonetwork** – automated network build-up
- **Discovery** functionality to discover/rediscover topology (paths for hopping) in real time

## Other features

- Unique transceiver **architecture** with **built-in operating system** and **DPA framework**.
- **Robust routing** algorithm for really working and **reliable mesh**.
- True **low power**. Receive modes **STD**, **LP**, and **XLP**. Bonding, Discovery and FRC also in LP.
- Powerful **IQRF IDE** and **IQRF GW Daemon** for efficient development and network management.
- **Wireless upload/upgrade** of **all SW** (OS, TR configuration, HWP, Custom handler, application code as well as the content of both EEPROM memories) simultaneously for all devices in a network, remotely over the air (OTA).
- IQRF is a **generic** technology. The user has control over all TR resources and the application. DPA protocol is free and documented. No license and carrier fees.
- Custom modifications of TR functionality are possible on request. For example, modified TR-7xDA-IL transceivers have been developed to adapt tightened laws in Israel.

## IQRF Open standard

The **IQRF gets standardized**, allowing everyone to use and implement all technical achievements and reliable protocols protected by **dozens of patents** under one **royalty-free license**.

# Transceivers

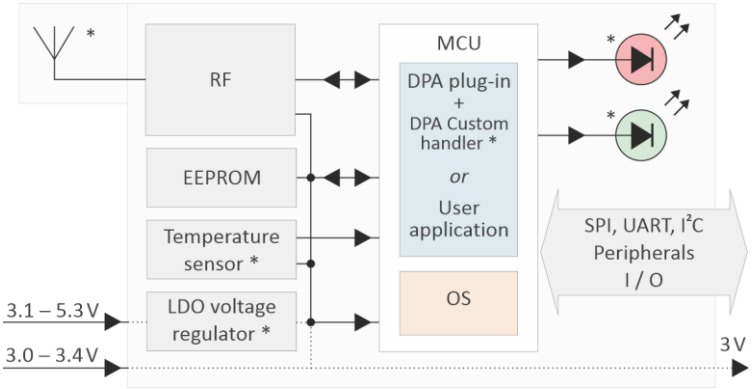
IQRF transceiver (**TR**) is a tiny intelligent electronic board with the complete circuitry needed to implement a wireless **RF connectivity**. It is a key communication component of IQRF technology. The specific functionality is given by an application software inside the MCU.

## Features

- Compact highly integrated design, no external components needed.
- **OS** (Operating System): lower SW layer, providing powerful support for all TR resources.
- **DPA** (Direct Peripheral Access): optional higher SW layer with enhanced network functionality.
- **Programmable** in **C** language.
- License-**free sub-GHz** ISM bands, worldwide (currently **868**, **916**, and 433 MHz).
- RF output power up to **10 mW**, programmable.
- Range per hop: tens of meters in buildings, **hundreds of meters** in open space, up to several kilometers in networks (up to **240 hops** per packet).
- **Extremely low power**: Standby < 300 nA, **receive 13 µA** (XLP).

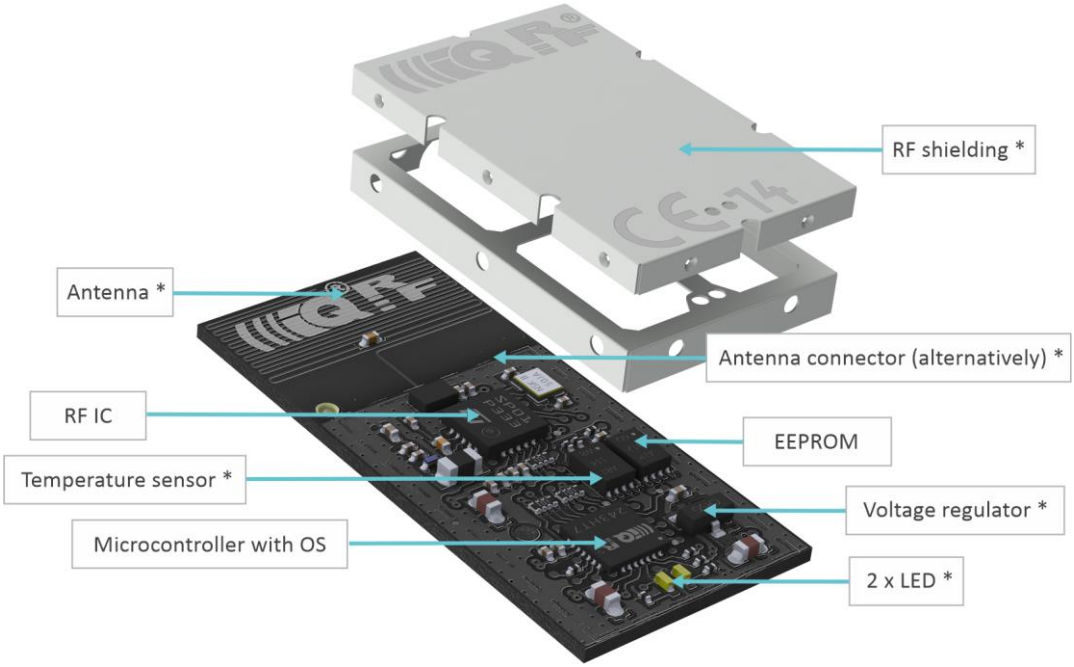
## Peripherals

- Up to **12 I/O** pins and 1 input only pin
- **SPI**, **I²C**, **UART**, and other interfaces
- **A/D** and **D/A** converters, analog **comparator**
- Multiple **PWM** output, programmable HW **timer**
- Extended MCU memories
- **EEPROM** 256 B inside MCU, **serial EEPROM** 256 Kb
- Extended MCU resources (interrupt capability and programmable internal pull-ups on all I/O pins, remappable digital peripherals, …)
- **Temperature** sensor
- Voltage regulator (**LDO**), battery monitoring
- **LEDs**

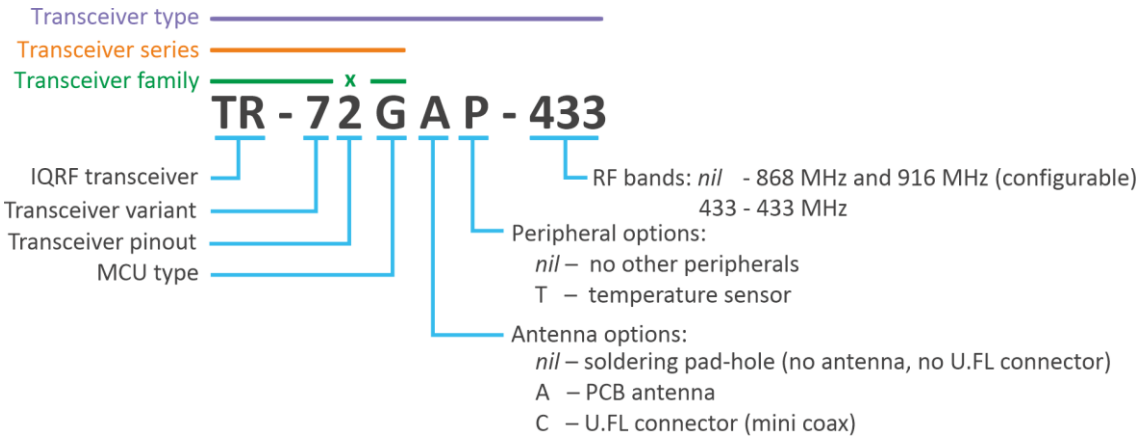



*General illustrations. For individual TR types see respective datasheets.*

** Optional.*



## Transceiver options



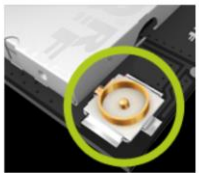

## TR cross table

| Series | | | TR-77D | TR-72G | TR-75G | TR-76G |
|---|---|---|---|---|---|---|
| Mounting | | | SMT | SIM | Vertical | SMT |
| Pins | | | 18 | 8 | 12 | 18 |
| I/O | | | 12 | 6 | 10 | 13 |
| MCU | | | PIC16LF1938 | PIC16LF18877 | | |
| MCU | Flash memory | | 16 K × 14 b | 32 K × 14 b | | |
| MCU | RAM | | 1024 B | 4096 B | | |
| MCU | EEPROM | | 256 B | | | |
| Serial EEPROM | | | 32 KB | | | |
| RF IC | | | Spirit 1 | S2-LP | | |
| RF bands | | | 868 | 868/916 | | |
| RF power (max.) | | | 4 mW | 10 mW | | |
| RF range (in open space) | | | 230 m | 500 m | | |
| SAW filter | | | Yes | – | | |
| LDO voltage regulator | | | – | Yes | – | |
| Supply voltage | | | 3.0 – 3.4 V | 3.1 – 5.3 V | 3.0 – 3.4 V | |
| Supply current | Deep sleep | | 56 nA | < 2 µA | < 300 nA | |
| Supply current | Sleep | | 610 nA | 2.3 µA | < 1 µA | |
| Supply current | Receiving | LP | 250 µA | 190 µA | 190 µA | |
| Supply current | Receiving | XLP | 16.3 µA | 13 µA | 13 µA | |
| Temperature sensor (optional) | | | – | Yes | – | |
| LEDs | | | 2 | | | – |
| A/D inputs | | | 3 | 2 | 3 | |
| Extended MCU resources | | | – | Yes | | |
| Dimensions [mm] | without antenna | | 25.1 × 14.9 | 25.1 × 14.9 | – | 15.2 × 14.9 |
| Dimensions [mm] | with antenna | | 31.8 × 14.9 | 31.8 × 14.9 | 27.4 × 14.9 | 23.3 × 14.9 |

*All parameters specified here are for guidance only, may depend on conditions and are subject to change. For exact parameters refer to IQRF datasheets.*

## Antenna options

Soldering pad-hole

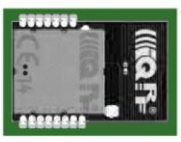


U.FL connector

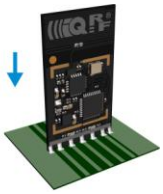
Built-in PCB

## Mounting options

SIM connector



SMT



Vertical soldering

# Transceiver types

**TR-7xG** is a family of new IQRF transceivers. It is the successor (in black) to the mature TR-7xD family (in red), with much larger memories and other improvements. To some extent it can be combined with older TR series in one network.

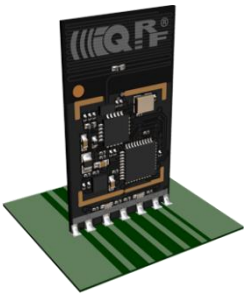
## TR-72G

TR-72GA

TR-72GA

- TR-72G is the TR-72D successor.
- Mounting in **SIM** card connector KON-SIM-02 enables flexible handling.
- 6 I/O pins, more flexible compared to TR-72D.
- **Ideal to start** with IQRF.

## TR-75GA



- TR-75GA is the TR-75DA successor.
- Vertical mounting for the highest RF range.
- Plugging into SIM connector KON-SIM-02 is also possible.
- Ideal for **street lighting** and similar usage.
- Supplied without a shielding can as standard.

## TR-76G

*Scale 1:1*

TR-76G

TR-76GA

- TR-76G is the TR-76D successor.
- The **smallest** form factor (20.2 x 14.9 mm).
- Stamp-hole pads for **SMT** mounting.
- Plugging into SIM connector KON-SIM-02 is also possible.
- 13 I/O pins, more flexible compared to TR-76D.
- The **lowest price** and very low **consumption**.

## TR-77D

TR-77D

TR-77DA

- High **resistivity** against **external interferences** and **noise**.
- **Extended blocking of unwanted signals**.
- **EN 54-25:2008**, **subclause 4.2.4** compliant
- For **fire alarms** and similarly demanding applications
- Stamp-hole pads for **SMT** mounting.
- Plugging into SIM connector KON-SIM-02 is also possible.

## USB transceiver GW-USB-06

- Based on TR-7xD architecture, dongle-like
- Additional MCU communicating with internal TR via SPI
- **IQRF** ↔ **USB** interface, **PC** connectivity, portable
- USB **Custom device** and **3 CDC classes**, supported by IQRF SDK libraries.
- Suitable as:
  - IQRF **local gateway**.
  - Portable IQRF **wireless programmer** (more comfortable than when using CK-USB-04A).

# TR interfaces

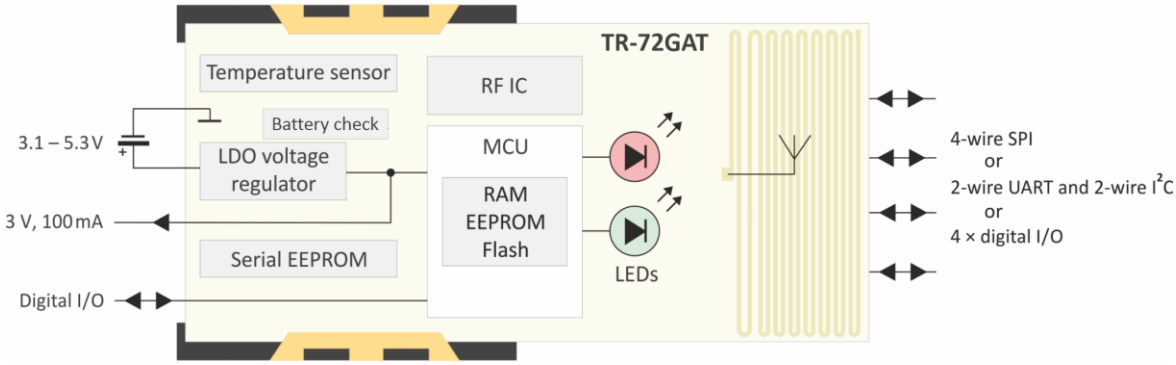
**Simple** interfaces are available **directly** on TR transceivers.

- **Standard communication**
  - Full duplex **SPI Slave** is the basic IQRF wired interface. It is directly supported not only by MCU **HW** module but even by IQRF OS via a set of powerful functions and runs in **OS background**.
  - **UART**, **SPI Master** and **I²C Master/Slave** are supported by MCU **HW** modules running in **MCU background**. Ready-to-use examples are available.
  - **Other** simple standards (e.g. **Dallas 1-wire**) can be implemented in application SW. Ready-to-use examples are available.
  - **USB** is available using a special type of **dongle-like** transceiver GW-USB-06.
- **Proprietary communication** is also possible. **Any** simple wired **protocol** can be implemented in application SW.
- **HW peripherals**

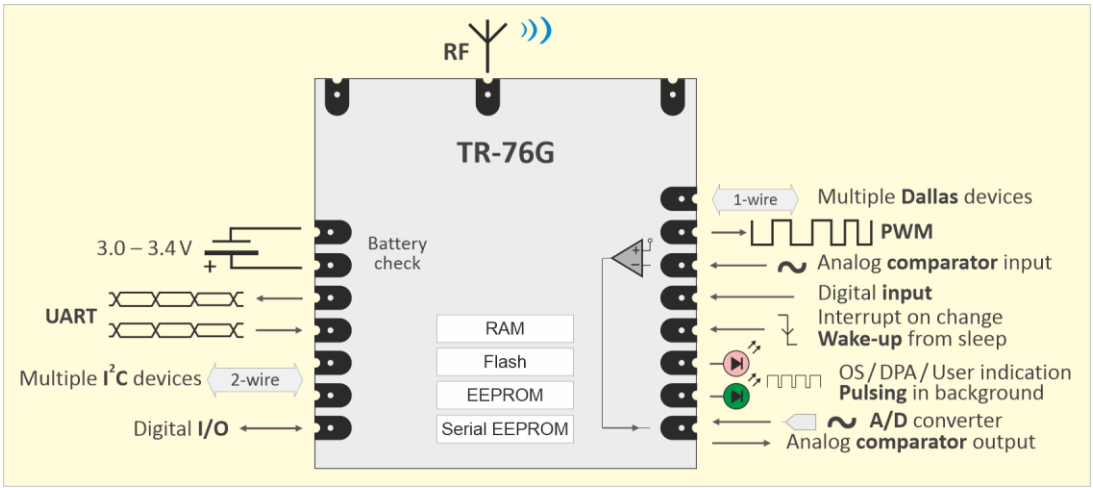
  Several HW peripherals are supported by MCU **HW** modules running in **MCU background**: **PWM**, **A/D** and **D/A** converters, **analog comparator**, **HW timer**, etc. Digital pins of all MCU peripherals are **remappable**.

For **DPA** approach:

- The IQMESH Coordinator is managed by local **Control interface** via **SPI** or **UART**.
- **Other** (common) **TR resources** are accessible as **embedded** Peripherals with DPA framework support.
- **User-specific peripherals** and non-standard behavior of embedded peripherals can be handled by **Custom DPA handler**.
- **IQRF Interoperability Standard** specifies the rules for unified handling with peripherals for all IQRF Alliance **interoperable** products.




*Example: one of interfacing possibilities of standalone TR-72GAT plugged in SIM connector*




*Example: one of a lot of interfacing possibilities of standalone TR-76G*

**Complex** standards (Ethernet, WiFi, LTE, …) can be accessed via IQRF gateways. Additionally, they are easily connectable to any cloud.

# Routers and repeaters

Routing means sending packets to addresses out of the sender's range using "**hops**" via Nodes that are in the range of each other. Network **topology** stands for the placement of devices with respect to mutual RF accessibility. For effective IQMESH, the topology should be designed in a **redundant** way. Every Node should have a sufficient number of routing Nodes in range. Routing allows **higher range**, more ways to deliver packets, higher noise immunity, resistance against failures and dropouts (self-healing) and flexibility with respect to dynamic changes in range between devices (movement of persons, obstacles or devices themselves) resulting in **higher throughput** and **reliability**.

## TR as a background router

In addition to normal operation, **any Node** device can **concurrently route** packets for other Nodes. This routing **in background** is transparent from the user's point of view, just by enabling routing in the TR configuration.


E.g., TRs controlling lights, wireless switches, and so on can route packets for all devices (other lights and switches, radiators, blinders, etc.) in the whole building. Dedicated repeaters are not necessary (but possible).

## Dedicated repeater

Dedicated repeater is a specialized ready-to-use device for **routing only** purpose, often with various **additional fetures**.

## Aggregating repeater

An aggregating repeater is a repeater providing not only a common routing but additionally concurrently serves as an interface to collect data from Beaming sensors and forward it to the Coordinator. It can be either **dedicated** (e.g. IQD-REP04 repeater) or implemented as an additional feature in an end device (e.g. IQD-LCZH02 lighting controller) routing in the **background**.

### IQD-REP04

Dedicated aggregating repeater IQD-REP04.

- **Aggregating** data from Beaming sensors
- TR-72G compatible
- Configurable, no programming needed
- **NFC** for contactless bonding via IQuip or a mobile phone
- Internal antenna
- Supplied from mains
- **Battery** backup (optional)
- **Wall plug** and **ceiling mount** versions

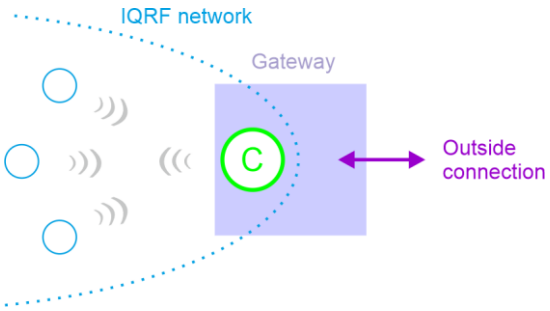
Wall plug
IQD-REP04-01A or IQD-REP04-02B

Ceiling mount
IQD-REP04-03A or IQD-REP04-04B

# Gateways

Typically, the IQMESH Coordinator is hosted inside a gateway (**GW**). It is a central wireless network device designed as an **interface** between the IQRF network and the rest of the world. It manages all devices in the entire IQMESH and provides communication from/to outside. It supports various connectivity standards, either **local** (USB, …) but especially **remote**, wired (Ethernet, …), or wireless (WiFi, LTE, …), enabling **Internet** connectivity and access to any existing **cloud**. A **datalogger** can also be implemented in GW. Such features are ideal for **IoT** applications.

GW is mostly based on a small single-board computer and **Linux** open source **IQRF GW Daemon**.

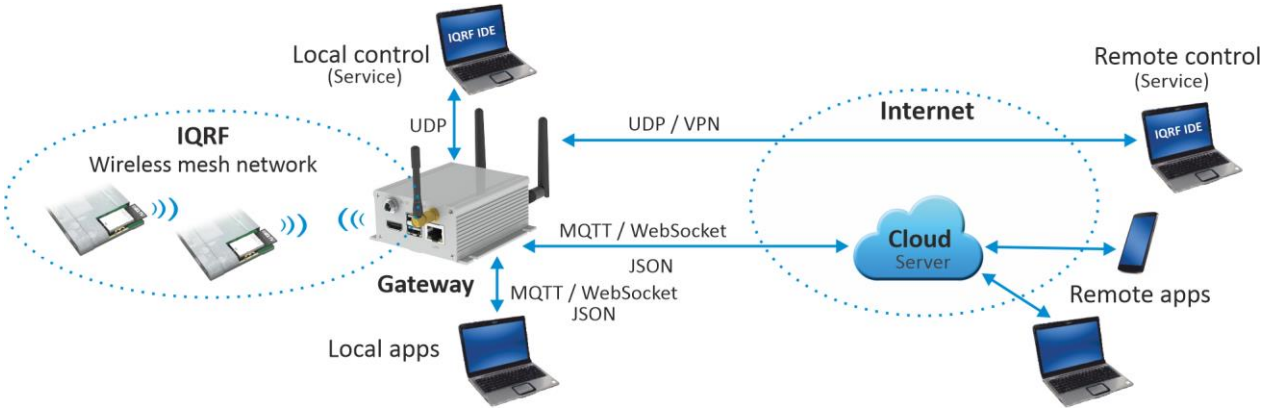



Typical IQRF **Gateway** = **TR** with **DPA** + **Linux machine** + IQRF GW **Daemon**

## Gateway types

- Standard Daemon-based **ready-made** gateways (e.g. IQD-GW04) are available. This is the easiest and the most common way to get a gateway.
- Alternatively, **GW Daemon** enables to construct a **user-specific gateway** from a **Linux machine** by a user's application engineer.
- And even more alternatively, construction of a **non-Daemon** IQRF GW is also possible. IQRF **SDK** **library package** enables to develop a user-specific gateway from Linux or even from a non-Linux machine (with Windows or without an operating system at all) by a user's application engineer.

## GW application example




## IQUBE (IQD-GW-02)

- Economical gateway IQD-GW-02, especially for low cost development, evaluation and samples
- Based on the Orange Pi Zero single-board computer with Linux (**Armbian** or **Yocto**), IQRF **Daemon**, and TR-76D transceiver with internal or external antenna
- **Ethernet** connectivity, connectable to **Internet** and optionally to any existing **cloud** servers
- Multiple communication channels **UDP**, **MQTT**, and **WebSocket**
- HW **watchdog** to recover the gateway from unexpected events (optional)
- Aggregating data from Beaming sensors
- Management through IQRF Gateway Webapp

IQD-GW-02W

IQD-GW-02EW

### IQD-GW04

- Industrial GW
- Based on the Raspberry Pi single-board computer with Linux (**Armbian** or **Yocto**), IQRF **Daemon**, and TR-76G transceiver
- Pre-configured **eMMC** flash memory
- **Ethernet**, **WiFi**, and (optional) **LTE** connectivity, connectable to **Internet** and optionally to any existing **cloud** servers
- Multiple communication channels **UDP**, **MQTT**, and **WebSocket**
- HW **watchdog** to recover the gateway from unexpected events (optional)
- **RTCC** with **power backup** for years (optional)
- Aggregating data from Beaming sensors
- Management through IQRF Gateway Webapp

### Gateways by IQRF Alliance members

Ready compact devices suitable to serve as gateways to IQRF are available from some 3rd parties (IQRF Alliance members). E.g., **Unipi Iris IQRF** by UniPi technology is a fully customizable modular solution for automation, control, regulation, and monitoring.

## Support for user's gateways

## Shields

Adapters, sometimes with additional supportive features are available to connect the TR to a single board computer (Raspberry Pi or many others with a compatible interface connector). In this way, you can easily create your own gateway, especially based on Linux open source IQRF GW Daemon.

### KON-RASP-01

Simple shield KON-RASP-01 for Raspberry Pi and lots of compatibles:

KON-RASP-01

Raspberry Pi with KON-RASP-01 and TR

### KONA-RASP04-10B

Shield KONA-RASP04-10B for Raspberry Pi and lots of compatibles, with various connectivity options and capability to aggregate data from Beaming sensors.

## Breakout boards

The simplest adapters are also available to allow universal connection of the TR to any 3rd party device regardless of connector type. For solderless development and prototyping.

### IQRF-BB-02

IQRF-BB-02 is a development adapter for connecting an IQRF TR transceiver to any device with 3 V or 3.3 V logic. It enables easy development with platforms like chipKIT, Arduino, Gemalto, Raspberry Pi and many others.

IQRF-BB-02

Arduino with IQRF-BB-02 and TR

# Low power

## IQRF wireless is extremely low power

IQRF is outstandingly suitable for **battery operated** devices. To optimize the consumption in the most crucial case (**RX**) for various needs, three RX/TX modes are available: **STD** (Standard), **LP** (Low Power) and **XLP** (Extra Low Power, not for IQMESH). For IQMESH, various advanced SW methods for further power consumption lowering are available: **FRC and sleep**, **LP** combined with **sub-µA precisely timed sleeping**, etc. The most power-saving operational mode for sensors is Beaming. A Beaming sensor is always sleeping, except for brief moments when it periodically or upon some event measures and asynchronously unidirectionally sends data.

For many IQRF applications, **battery life** is determined primarily by its **self-discharge**.

| Mode | Operation | Time spent | Requirements | Importance |
|---|---|---|---|---|
| **Transmit (TX)** | TR is transmitting an RF packet | Very short periods | As low as possible, but given mostly by RF output power | Important |
| **Receive (RX)** | TR is receiving or attempting to detect incoming RF packet | **Majority of the non-sleeping time** | **As low as possible** | **Crucial** |
| **Run** | TR is operating (either with or without RF circuitry sleeping) but no RX or TX is in progress | Negligible | Not critical | Negligible |
| **Sleep** | No operation (including incoming RF signal detection) is possible until TR wakes up | **Majority of time** | **1 µA** order of magnitude for battery life of years | **Necessary** |
| **Deep sleep** | No operation (including incoming RF signal detection) is possible. Complete initialization of RF circuitry must be performed after TR awakening. | Very long periods (if applicable) | **< 300 nA**. Extremely low power in very long periods of inactivity. **In special applications only.** | Either not critical or crucial |
| **Beaming** | Data sent from sensors periodically or upon some events | Permanently | **As low as possible** | **Crucial** |

### TR-76G power consumption example

- Deep sleep mode: **< 300 nA**
- Run mode
  - RF sleep: 1.8 mA
  - RF ready: 3.3 mA
- TX mode: **8 – 25 mA**
- Sleep mode: **< 1 µA**
- RX mode
  - STD: 12.5 mA (Standard)
  - Power saving:
    - LP: **190 µA**
    - XLP: **13 µA**

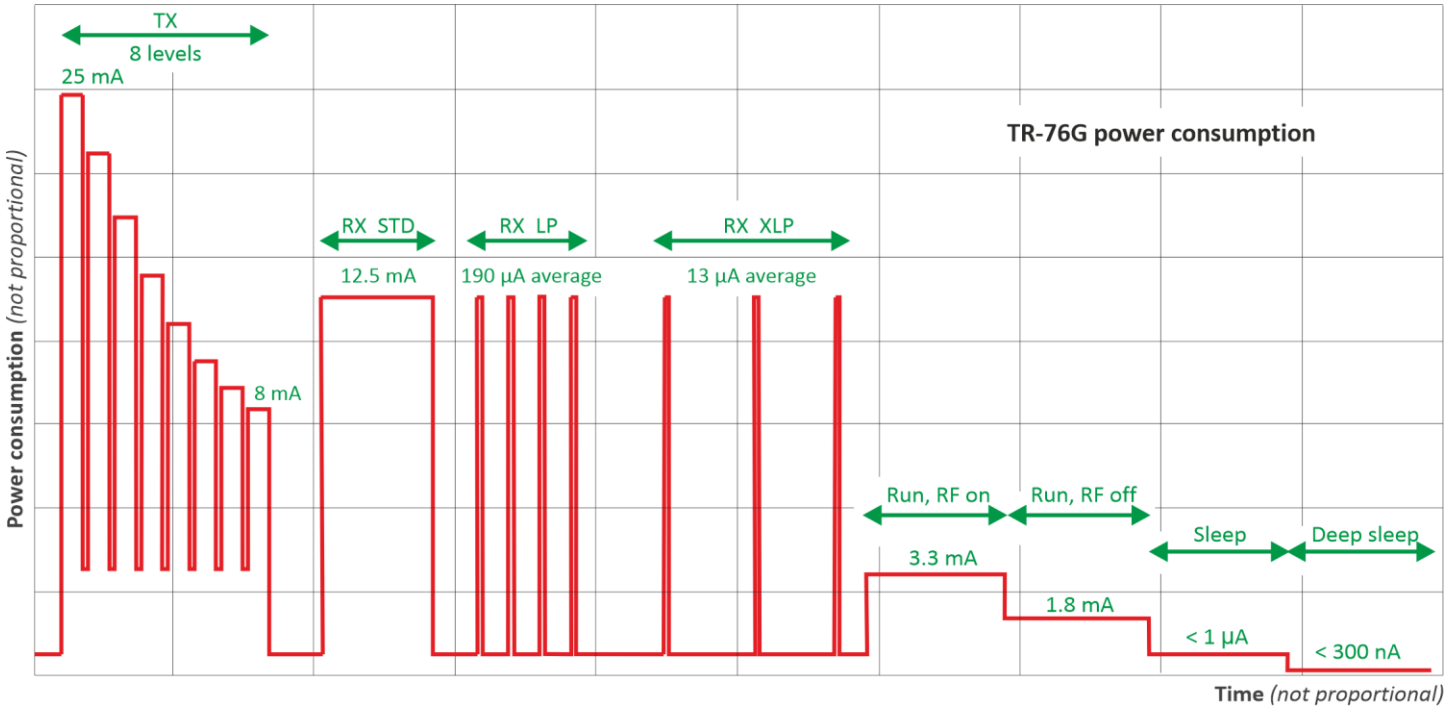
IQRF **lifetime** * for 1 Ah, 3.6 V, ½ AA sized **battery**:

- **1000 years** in Deep **sleep**
- **7 years** continually **receiving** in XLP **
- **300 MB** data **received**
- **200 MB** data **transmitted** (at highest RF output power)

* *Theoretical values*

** *Without an incoming RF signal including a noise*

Look at other RF platforms boasting ultra/extra low power.



# Security

Every wireless system is exposed to potential over-the-air (OTA) attacks. Unlike wired systems, there is no physical protection and any eavesdropper can listen to RF communication and try to fake RF packets. Thus, the security concept should consider more threats. Firstly, the communication must be protected when the network is created. During this bonding procedure, devices exchange sensitive information (e.g. network password and addresses). However, the bonding is typically launched only once within the installation process, whereas the subsequent communication runs continuously, so it is more vulnerable and must always be secured properly. IQRF brings ultimate security based on industrial standards ensuring authorized access to OTA-flowing data. IQRF utilizes **AES-128 encryption**, an **industrial standard** for wireless communication. Moreover, some supplemental extension layers are added. Besides hiding sensitive data, the encryption brings another contribution to IQRF multilevel consistency protection and prevents packets forging.

Three different protections based on **AES-128** are applied:

## Networking encryption

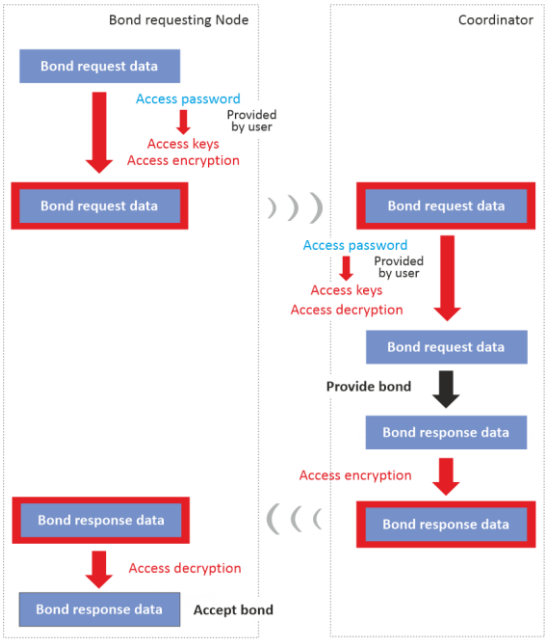
**All** networking communication is **encrypted**.

## Access encryption

Another independent encryption is always applied while **bonding** and all other network **maintenance**.

## User encryption

Moreover, payload data (for networking as well as non-networking packets) can optionally be encrypted and/or decrypted **end-to-end** by a user-specific key to hide the information content.

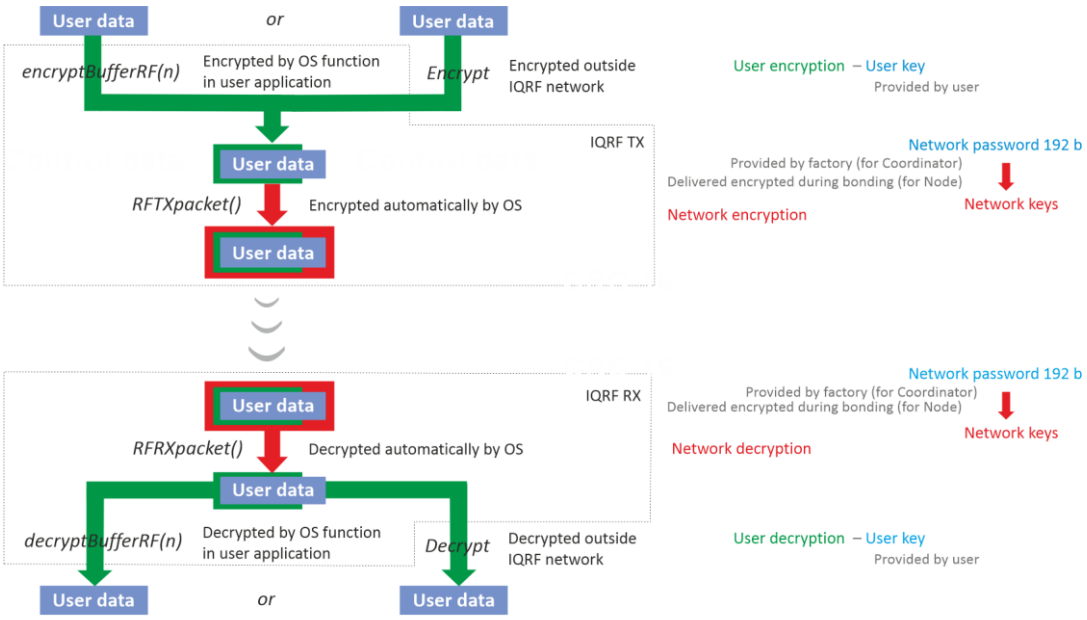



*Encryption during bonding*

## Features

IQRF minimizes and protects also manipulation with **keys**. Network and access keys are not known during physical manipulation as they are generated from respective **passwords**. Network password is generated randomly with high entropy and delivered encrypted to devices that are joining the network during bonding. This approach offers the following significant advantages:

- The user should take care of the passwords only but never handle with keys.
- The management and distribution of keys are completely handled by OS.
- The keys depend not on the passwords only but are modified by embedded **hash** functions.
- These **two separated layers** (with no simple direct relationship between passwords and keys) additionally increase the security.
- The keys are generated **dynamically**, varying in time, which is significantly more immune against attacks.
- The relationship between passwords and keys are **different in different networks**.
- Network encryption is **forced** (not optional), the encryption is done automatically by OS, fully transparently from the user's point of view. This prevents an incorrect implementation.
- **Breaking** the keys in one network has **no impact on other networks** at all.
- In the future, new vulnerabilities (even e.g. in AES) may be found and existing systems should be patched.
  All IQRF software is fully upgradeable (even wirelessly), which (among others) allows to completely **upgrade** the security functionality.




*User and networking encryption*

# Development

*This chapter applies to **generic** products (designed to implement a **user-specific** functionality), as opposed to off-the-shelf systems like IQAROS that require only simple installation (Place & Play) but no user development.*

TR transceivers are delivered with the following **SW from the factory**:

- **IQRF OS**. It is fixed and can only be replaced (**upgraded**) by another existing OS version released for the given TR type.
- **RF link** demo application example, peer-to-peer (directly above the OS). It is primarily intended for RF connection **demonstration** just after unpacking two TRs and connecting them to a power supply.

To implement a specific functionality, the user's application engineer must replace the demo with the required SW in three steps:

- **Programming:** writing the application code in C language. Lots of source code examples solving typical tasks are available.
  - **Peer-to-peer** application can be written in **C** directly above the OS.
  - If **DPA** is to be used (i.e. for IQMESH), the appropriate **DPA plug-in** (which is a special case of the application) must be uploaded first. Then, the specific functionality is achieved by a **Custom DPA handler**, also written in **C** similarly as above the OS. There are lots of examples of source code handlers that solve typical tasks. Particularly simple applications can therefore be implemented using some of these ready-made handlers even **without programming**.
- **Compiling** the SW from C to executable (`HEX`) format.
- **Uploading** the `HEX` code into TR.

So in any case it is necessary to compile the application from C and upload it into TR. All of the above operations can be performed very efficiently in IQRF IDE.

## What is needed to start?

### Hardware

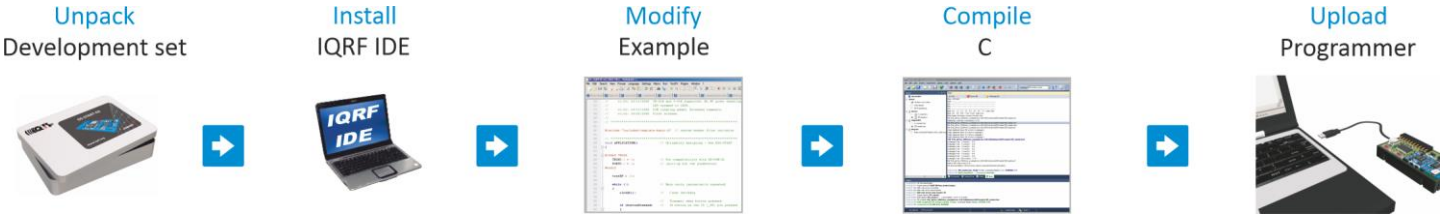
- **All hardware** needed to start IQRF application designs is available as the DS-START-05 or DS-DPA-03 **development set**.

### Software

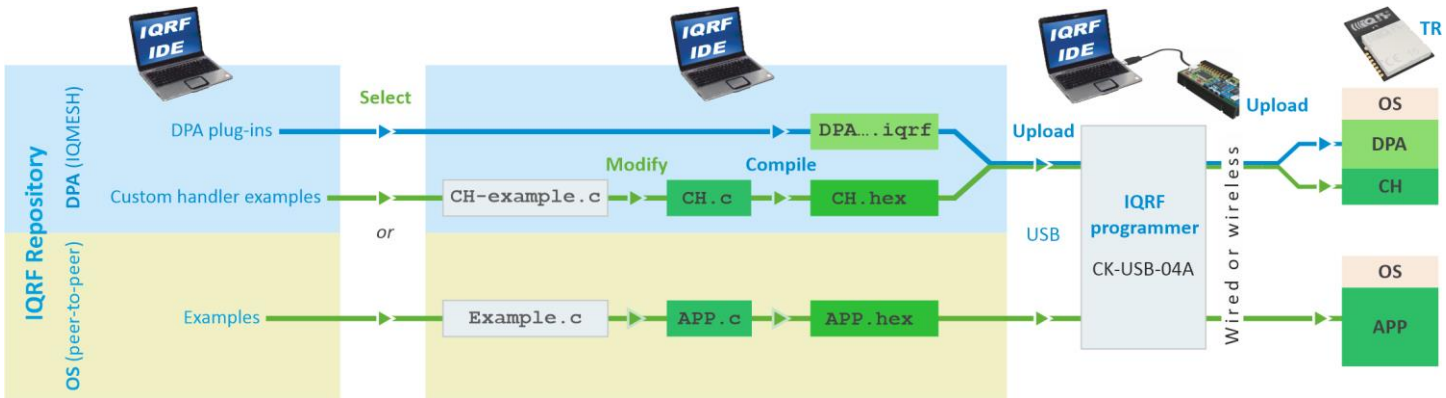
**All software** needed to effectively start IQRF application designs is **free**.

- IQRF IDE is a comprehensive specialized software that provides all the necessary **design functions** (including the free version of C compiler) with **many additional gadgets** integrated in a single **development environment**.
- IQRF Repository is a public online **storage** and **database** containing all files and relevant information for IQRF design.

## How to start?




## Programming and uploading process



# Programmers / debuggers

## CK-USB-04A

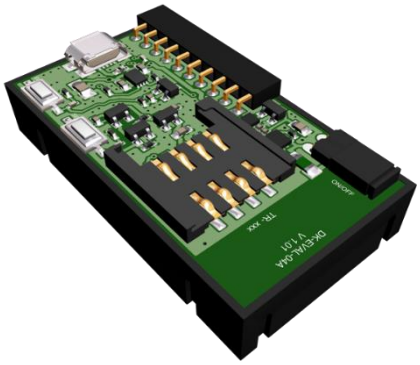
- One of the two main IQRF development tools. Optimized to work with IQRF IDE.
- **Programmer** to upload codes into TR transceivers
- **Debug** kit to **test** user applications
- **Host** for **TR**
- Works also as a **USB converter** (**USB** ↔ **SPI** and **USB** ↔ **UART**)
- USB modes: **Custom device**, **CDC IQRF**, **CDC SPI** and **CDC UART**
- It can be used to create other development tools, e.g. CATS including a programmer for wireless upload.
- **Interface connector** to access all TR pins
- Powered from USB (micro-USB connector)
- 2 LEDs, 2 pushbuttons
- Plastic SIM connector holder supports also SMT transceivers

## GW-USB-06

- USB transceiver suitable also for CATS
- See chapter USB transceiver GW-USB-06.

# End Node kit

## DK-EVAL-04A



- **Universal evaluation kit** for IQRF
- One of the two main IQRF development tools
- **Host for TR module**, plugged in **SIM** connector
- Ideal for network applications development
- LiPol accumulator, charged via microUSB connector
- Portable, space saving (48 x 27 x 11 mm)
- **Connector** for TR interfacing
- Compatible with other IQRF DDC kits, see below

The simplest usage of two DK-EVAL-04A kits:

# Development sets

- Optimized combinations of IQRF devices
- Ready-to-use SW examples fitting to typical tasks
- Documentation, demo SW, and support
- Affordable prices and favorable training

## DS-START-05

Development set for immediate start with IQRF
To operate 3 wireless devices, peer-to-peer or in a small network

| | |
|---|---|
| TR-72GAT | 3 pcs |
| CK-USB-04A | 1 pc |
| DK-EVAL-04A | 2 pcs |
| Micro USB cable | 1 pc |

## DS-DPA-03

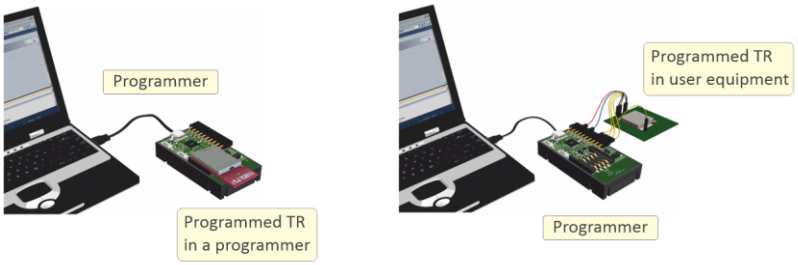
The best choice for starting with IQRF **DPA**
To operate a network with 6 wireless devices

This set consists of two identical sets DS-START-05 with affordable price.

# Upload

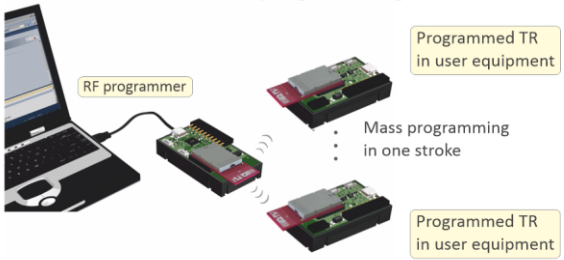
Uploading = programming (writing) the application program into the TR memories.

Wired upload



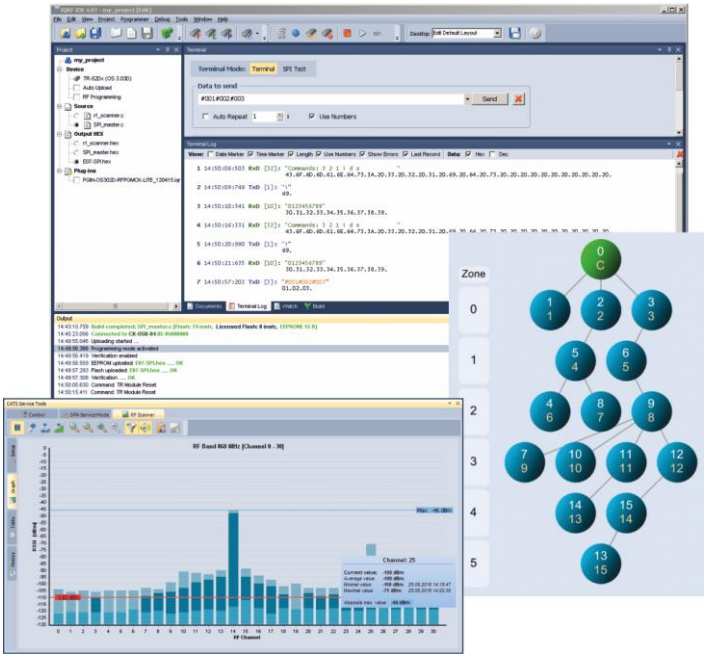

TR is plugged in a programmer (via SIM connector) or populated in an end equipment and connected with the programmer by wires.

Wireless upload
RFPGM – RF programming®




TR is populated in an end equipment. For **mass upload** (into all nodes in one stroke) it is a must.

# IQRF IDE



Free integrated development environment for **creation**, **service** and **maintenance** of IQRF applications. It is **project-oriented** and drawing data from IQRF Repository. IDE is a fundamental SW tool for general work with IQRF. Most of the IDE functions are also implemented in IQRF Daemon.

IQRF IDE is available also in limited (TR uploader only) **command line** version. It is primarily intended to **speed up** and **secure** the **production** (even for less skilled workers).

## Programming

**Edit** and **compile** the application and **upload** the code in TR.

## Debug

**Break**, **watch** and **continue**.
Internal variables can be inspected after every break.

## Communication test tool

- Rich terminal offering hints for packet construction, supporting macros
- Terminal log: Raw/DPA packets, bookmarks, notes
- Packet inspector decoding and interpreting packets

## IQMESH network manager

Tools for IQMESH network management, visualization and testing.

## CATS – IQRF service tools

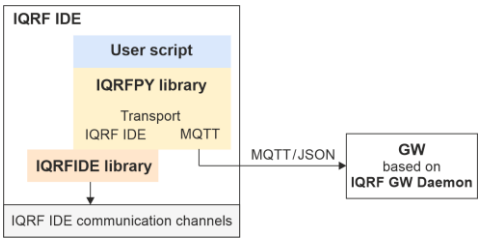
**RF programmer**, **DPA service tool** and **RF scanner**

## Project wizard

Easy guided way to create a project.

## Python scripting

A tool for Python script creation (typically for testing purposes) using the IQRFPY library to communicate with the IQMESH Coordinator via USB or MQTT.




## Upgrades

Upgrade (even over-the-air) OS and DPA in TR transceivers or firmware in auxiliary MCU inside IQRF devices.

## Many other features

TR configuration, Autonetwork, access to **IQRF Repository**, …

# IQRF Repository

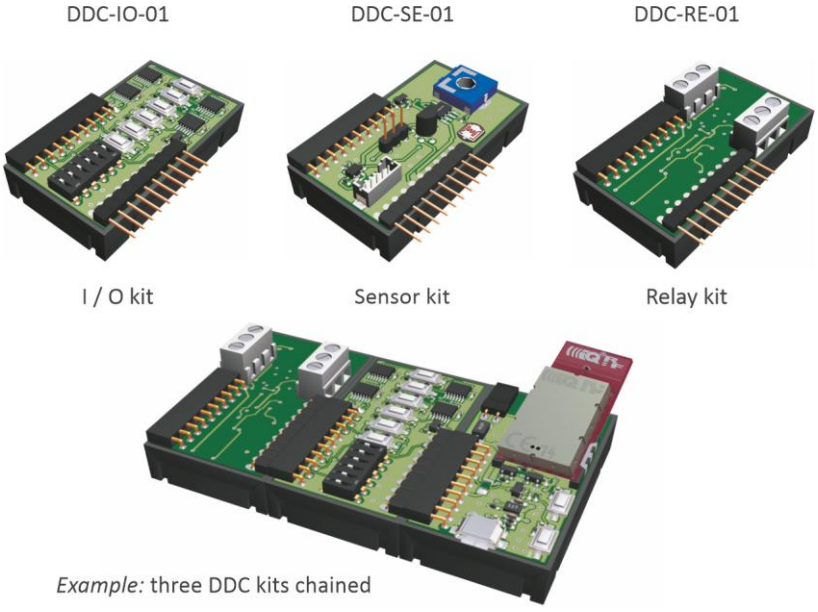
IQRF Repository is a public documented online **storage** and **database** containing the following:

- All **files** (DPA plug-ins, source code headers, application examples, and documentation) needed for creation of IQRF IDE **Project**.
- Information regarding all **IQRF interoperable products**. It provides information about and relations between IQRF ecosystem objects such as **companies**, **products**, **DPA handlers**, **standards**, **OS**, **DPA**, etc.

The Repository is developed in **REST API** with **JSON**. It is utilized by **gateways**, **clouds**, **IQRF IDE**, the **mobile application**, and so on, helpful not only for network creation and maintenance.

# DDC – Development Daisy Chain kits

- Modular **cascadable** kits
- To expand interfaces and peripherals
- For a **solderless development** and **prototyping**
- No wiring
- Compact, portable, small size 48 × 27 mm




*Example:* three DDC kits chained

## Base kit

- DK-EVAL-04A with TR transceiver
- Internal **accumulator** supplies the entire chain

## Expansion kits

With additional peripherals

- DDC-IO-01 **I/O** kit to control inputs/outputs
- DDC-SE-01 **Sensor** kit to measure temperature, voltage and illumination
- DDC-RE-01 **Dual relay** kit

# Application support and services

## Design support

**Free** application support and a lot of design services are available. Examples, video tutorials, consultancy, development support, troubleshooting, …

## Training

To speed up the startup, a one-day training is recommended. It can be **arranged** with respect to your **specific needs**.

## Consultancy

Before starting your commercial project, you can contact our technical support to consult how to proceed in the most effective way.

## FAE

IQRF FAE (Field Application Engineering) provides massive customer-facing help with applications (from an **assistance** with system-level design to complete **turn-key** solutions).

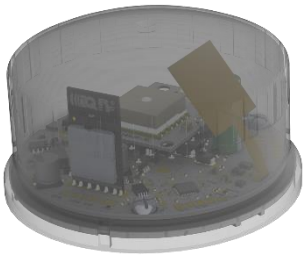
**Speed up your first** commercial **design** with our FAE. Then you can follow-up to **your next designs yourself**.

# End devices – lighting

*In the End devices chapters, only a small part of the available products is listed so far. Follow the IQRF product pages for complete portfolio.*

Controllers and bridges for wireless lighting control.

## IQD-LCZH02



Wireless controller IQD-LCZH02 for **street LED lights** with **Zhaga** socket. Outdoor, to be mounted on the cover of the lamp

It works as a bidirectional data tunnel (bridge) to transfer data frames between the IQRF wireless network and the standard **digitally controlled lighting bus IEC 62386**. Alternatively, **analog** control (**0 - 10 V**) is possible.

Additionally, the unit serves also as an aggregating repeater.

- **NFC** is equipped for easy **contactless** installation and maintenance using a mobile phone or IQuip.
- **Interoperable**, observing IQRF Standard
- RF range 450 m in free space
- **24 V** DC powered from the lamp
- Optional:
  - **RTCC** (real-time clock/calendar) with power backup
  - **GPS**
  - Thermometer

## IQD-LCNE02

Wireless control unit IQD-LCNE02 for **street LED lights** with **NEMA** socket. Outdoor, to be mounted on the cover of the lamp.

It works as a bidirectional data tunnel (bridge) to transfer data frames between the IQRF wireless network and the standard **digitally controlled lighting bus IEC 62386**. **Analog** control (**0 - 10 V**) is also possible.

Additionally, the unit serves also as an aggregating repeater.

- Bistable **relay** switching of LED driver power source
- Relay load up to 200 VA, switching on zero crossing
- **NFC** is equipped for easy **contactless** installation and maintenance using a mobile phone or IQuip.
- **Interoperable**, observing IQRF Standard
- Powered from mains
- RF range 400 m in free space
- Optional:
  - Measurement of **current** drawn by the **appliance** (approximate)
  - **RTCC** (real-time clock/calendar) with power backup
  - **GPS**
  - Thermometer

## IQD-DB1M-01A


Wireless control unit IQD-DB1M-01A for **LED luminaires**. Indoor, for **ceiling mounting**. Connectable via wires.

It works as a bidirectional data tunnel (bridge) to transfer data frames between the IQRF wireless network and the standard **digitally controlled lighting bus IEC 62386**.

- Connection to a luminaire by **wires** and **clamp connectors**
- **NFC** is equipped for easy **contactless** installation and maintenance using a mobile phone or IQuip.
- **Interoperable**, observing IQRF Standard
- Powered from mains
- RF range 300 m in free space
- Small dimensions, 22 mm height

# IQAROS

IQAROS is a **wireless** indoor environmental **monitoring system**, especially for **automated collection** of **temperature** and relative **humidity**. It is intended to measure conditions in the interior of buildings, e.g. warehouses, shops, factories, offices, archives, and other indoor areas, even at different heights in the room.

In addition to the normal meters, particularly robust sensors are available for outdoor or other harsh environments, either with a sensor in a stainless steel housing connected by a cable, e.g. for refrigerators, swimming pools, etc., or a sensor in a stainless steel rod (1.5 m) for measuring temperature in hay, straw, grain, etc.

IQAROS is available in various pre-prepared variants, or in a customized set. It is fully ready to use, pre-configured for immediate start-up and requires only basic parameter settings.

Sensors beam the measured values every minute.

» More info on IQAROS product page.

IQAROS sets are combined from the following parts:

Sensor Industrial
Based on IQD-SE03-01

Sensor Industrial Pro
Based on IQD-SE03-02

Sensor Industrial X
Based on IQD-SE03-03

Sensor Industrial SX
Based on IQD-SE03-07

IQRF Sensor SE02-T/H
Based on IQD-SE02-04A

Gateway IQUBE
Based on IQD-GW-02

Aggregation repeater
Based on IQD-REP04

IQRF®

Reliable by definition

IQRF Technology

www.iqrf.org

IQRF Alliance

www.iqrfalliance.org

IQRF Open Standard

standard.iqrf.org

IQRF E-shop

eshop.iqrf.org

IN MESH WE TRUST

SINCE 2004