Buletinul Ştiinţific al Universităţii Politehnica Timişoara

TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Volume 59(73), Issue 1, 2014

# Telemetric Applications for the Automotive Industry using IQRF devices

Victor Serban, Dan Andreiciuc, Aurel Filip

***Abstract*— The proposed system implies exploiting a technology previously utilized in smart home automation, namely IQRF boards, in the automotive industry. Employing these devices, one can obtain a more versatile control of his/her vehicle, as well as various aspects of environmental conditions, like temperature or the intensity of light outside the vehicle. This system constitutes of a modular hardware part, and multiple software parts, each accomplishing some of the previously enumerated notions.** (*Abstract*)

***Keywords*—IQRF; Automotive; IQMESH; Telemetry (key words)**

## I. INTRODUCTION

IQRF is a sub GHz wireless communications technology. It is intended where wireless general use connectivity is needed, be it point to point, or in complex networks. Its functionality depends only on a dedicated application written in the C programming language.

The object of this application that is presented in this paper is the integration of smart house applications in the automotive industry.

The elementary IQRF communication is the transceiver module including a microcontroller with an onboard operating system, implementing the physical level, as well as the data link level upholding the MESH network which utilizes the open IQMESH protocol. No other superior level, like the transport level is part of this technology.

## II. IQRF MODULAR HARDWARE UTILIZED

*IQRF Technology Characteristics*

Among the main properties of the IQRF technology are: low speeds, low power consumption and low data rate, packet based oriented data radio frequency up to 128 Bits/packet, RF software selectable parameters, sub-GHz radio frequency bands using multi canal systems and FSK modulation, bit rate between 1,2 – 86,2kb/s, output power of maximum 20mW, output range of over 1 Km per hop, up to 240 hops per packet, up to 65000 devices in a single network, low power consumption (380nA in idle mode and 25 uA when receiving), the code can be uploaded wirelessly to all nodes in the same time and last but not least, there is no license acquisition fee.

## III. HARDWARE CHARACTERISTICS

### A. *DK-EVAL-04*

The DK-EVAL-04 is a universal development kit for wireless IQRF transceivers. Its reduced size, LiPol accumulator and low cost make this kit ideal for intelligent network application usage.

Applications for this device include: the development of wireless applications, host for TR IQRF modules, testing and debugging of IQMESH networks and portable battery powered wireless systems.

The simplified diagram of the board is presented in Fig.1.

### B. *CK-USB-04*

The CK-USB-04 is a development kit oriented towards the programming and debugging of user applications with IQRF transceivers. It can also serve as a IQRF USB gateway (USB – SPI converter) or just a simple host for the transceiver module.

Among the key applications are the following: programmer board for the IQRF transceiver, IQRF debug kit, final IQRF application host, USB host, USB – SPI converter and finally PC connectivity.

The simplified circuit diagram is presented in the Fig.2.

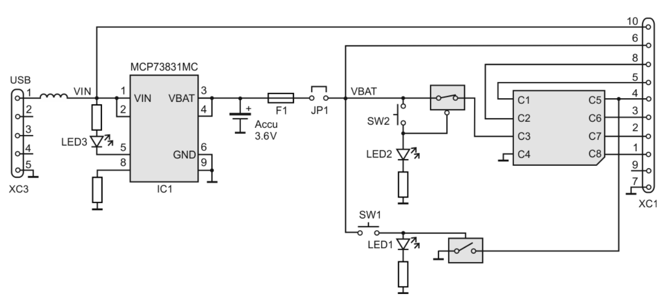



Fig.1 – Simplified diagram of the DK-EVAL-04 board

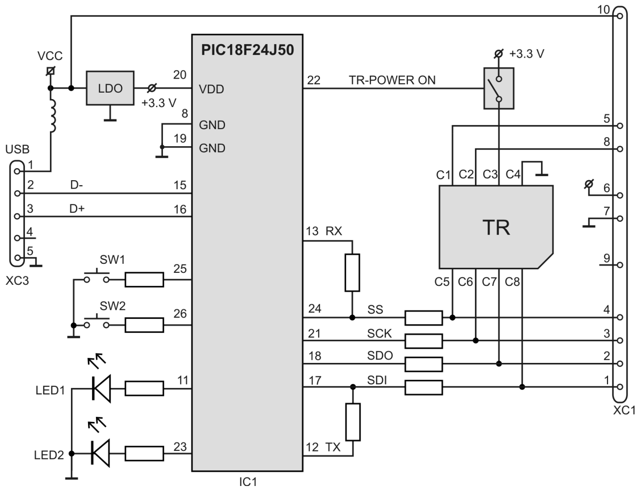



Fig.2 – Simplified diagram for the CK-USB-04 board

The network can be controlled using three words:

1. Node address
2. Peripheral number – indicates with which peripheral I can communicate with
3. Each peripheral number has different commands

Each packet can have 58 bits that can be used.

### *C. SHD-SE-01*

The SHD-SE-01 is a multifunctional wireless sensor that gives temperature readings, illumination, acceleration measurement, has a real time clock as well as an EEPROM memory.
Its low power usage design allows its battery to last for years on end.
The sensor can be adapted to user specific functionality through application software for the microcontroller in the operating system included transceiver module.
The key characteristics of the sensor are the following: smart Transceiver station with built-in antenna, Selectable band of FR 868 / 916 MHz with multiple channels, internal microcontroller equipped with an operating system, compatible with TR-54D, 3 axis accelerometer, temperature sensor, light sensor, real time clock, serial EEPROM, tactile button, LED indicator, ultra reduced power consumption, integrated primary battery with a multi-year lifetime and a programmable application in the internal transceiver module.
Fig.3 represents a picture of the SHD-SE-01 module.
Fig.4 will represent the simplified block diagram of the SHD-SE-01 module.

Fig.3 Image of the SHD-SE-01 module

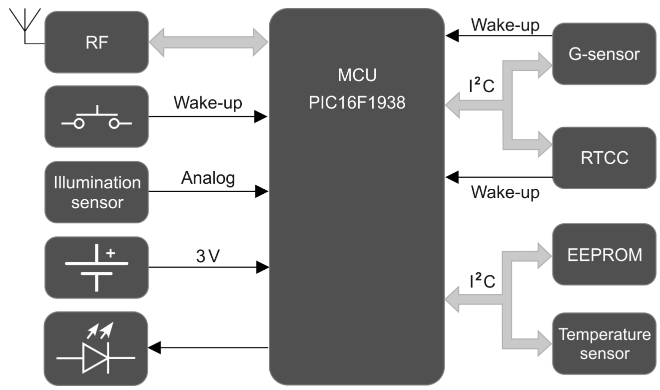



Fig.4 Simplified block diagram of the SHD-SE-01 module

## IV. Software Architecture

Software architecture is defined by IEEE as: basic organization of a system embodied in the system components, the relationships between them and between system components and the environment, and the principles governing the design and evolution of the system. [ANSI/IEEE Std. 1471-2000, Recommended Practice for Architectural Description of Software-Intensive Systems].
The definition proposed by IEEE says that architecture captures system structure in terms of its components and how these components interact. The system architecture defines the rules by which the system is designed as well as defining how it can be changed.
An embedded enterprise architecture is organized in four levels:

- drivers working with hardware (its abstraction).
- Basic Software - eg AdcHandler (mode like reading series features several ADC channels, applying transformations on the results).
- OS - basic skeleton of an operating system based on tasks (model time-slice).
- Application - Module generic state machines, controllers, error checking, etc.

This type of architecture is commonly used in automotive industry. The proposed variant satisfies the portability and future expansion, given the requirements and the complexity of the application.
To demonstrate the feasibility of using IQRF devices, but also to have a starting point for the concept car telemetry integration in smart home, they realized some projects in the IQRF IDE 4.7. The first project aims to familiarize with the programming environment, the operating system and the attached framework and programming devices and using IQRF.
This first project carried out at switching supply or after reset, the LED flashing three times. Upload file consists Start.c (v. Annex) in one of microcontrollers (Fig.5).
The second project consists of a device programmed with Rx.c file that will ide receive a package through LED lights, a device that sends a message by pressing and sensor standard program that sends all the data to a message by pressing. In Fig.6's played such a system.

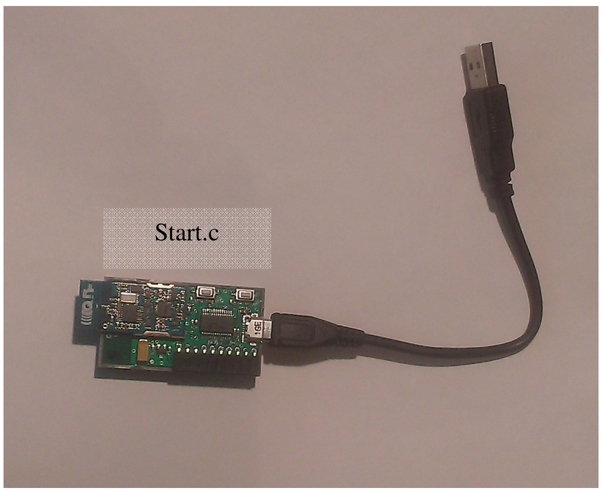



Fig.5 Start.c project

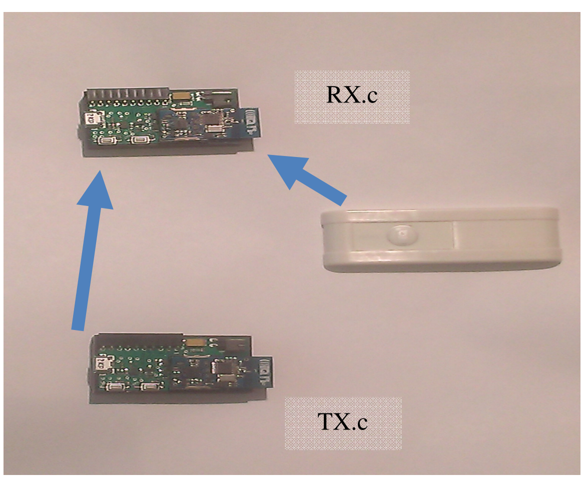



Fig.6 Second Project

The third project is a temperature sensor remote that receives every 5 seconds data and sends to the USB port to be displayed on the terminal that monitors communication between your device and computer programming, feature offered by IQRF IDE development environment. The devices involved in the project are shown in Fig.7 and a screen capture after the terminal window is shown in Fig.8. One can notice the message wrong (and detected as erroneous) when the device transmitter button is pressed and interfere with the communication of the sensor.
The fourth project contains two devices. In the first is entered a program that executes two commands: alters the duty cycle of the PWM signal generated and lights five times the LED. With this program you can control another device ON / OFF or can act an actuator (motor, etc.). The second device is a transceiver. Topology can be seen in Fig.9. In Fig.10 is shown connected to the terminal window with TR.c, control messages are sent manually with the author. Also, here is presented the information in a message from SHD device containing data that can provide it (from all sensors mounted).

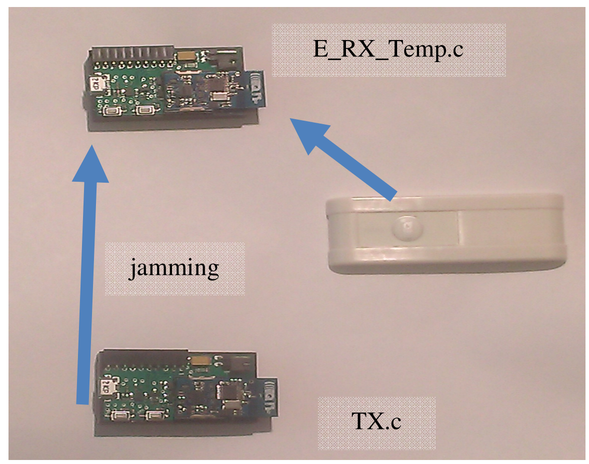
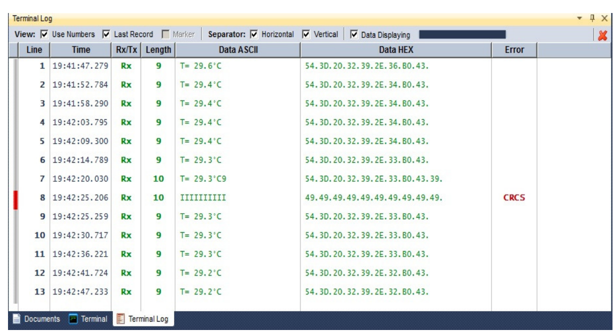



Fig.7 Third Project


Terminal Log
View: Use Numbers Last Record Marker Separator: Horizontal Vertical Data Displaying

| Line | Time | Rx/Tx | Length | Data ASCII | Data HEX | Error |
|---|---|---|---|---|---|---|
| 1 | 19:41:47.279 | Rx | 9 | T= 29.6°C | 54.3D.20.32.39.2E.36.B0.43. | |
| 2 | 19:41:52.784 | Rx | 9 | T= 29.4°C | 54.3D.20.32.39.2E.34.B0.43. | |
| 3 | 19:41:58.290 | Rx | 9 | T= 29.4°C | 54.3D.20.32.39.2E.34.B0.43. | |
| 4 | 19:42:03.795 | Rx | 9 | T= 29.4°C | 54.3D.20.32.39.2E.34.B0.43. | |
| 5 | 19:42:09.300 | Rx | 9 | T= 29.4°C | 54.3D.20.32.39.2E.34.B0.43. | |
| 6 | 19:42:14.789 | Rx | 9 | T= 29.3°C | 54.3D.20.32.39.2E.33.B0.43. | |
| 7 | 19:42:20.030 | Rx | 10 | T= 29.3°C9 | 54.3D.20.32.39.2E.33.B0.43.39. | |
| 8 | 19:42:25.206 | Rx | 10 | IIIIIIIIII | 49.49.49.49.49.49.49.49.49.49. | CRCS |
| 9 | 19:42:25.259 | Rx | 9 | T= 29.3°C | 54.3D.20.32.39.2E.33.B0.43. | |
| 10 | 19:42:30.717 | Rx | 9 | T= 29.3°C | 54.3D.20.32.39.2E.33.B0.43. | |
| 11 | 19:42:36.221 | Rx | 9 | T= 29.3°C | 54.3D.20.32.39.2E.33.B0.43. | |
| 12 | 19:42:41.724 | Rx | 9 | T= 29.2°C | 54.3D.20.32.39.2E.32.B0.43. | |
| 13 | 19:42:47.233 | Rx | 9 | T= 29.2°C | 54.3D.20.32.39.2E.32.B0.43. | |

Documents Terminal Terminal Log

Fig.8 Temperature readings as seen on the screen

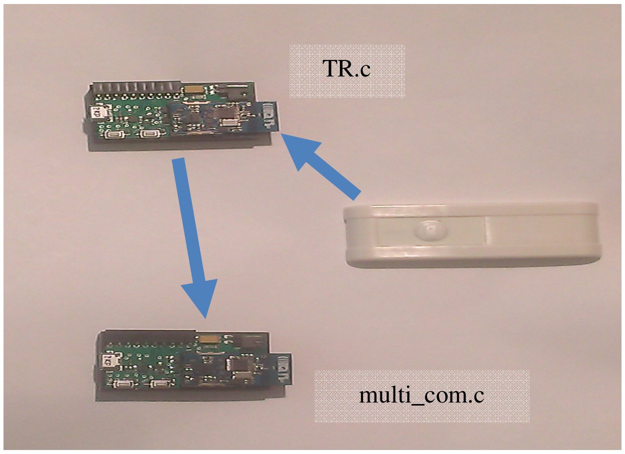



Fig.9 The fourth project

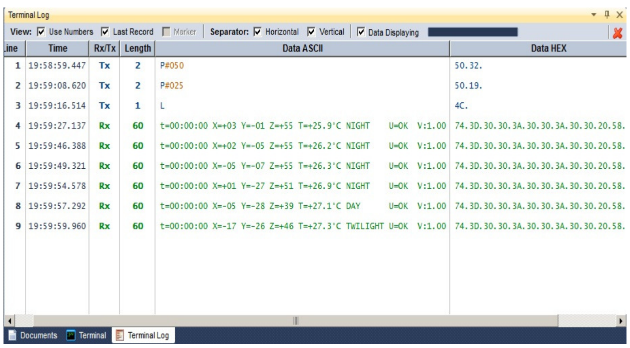



Fig.10 Terminal window, message transmission and reception

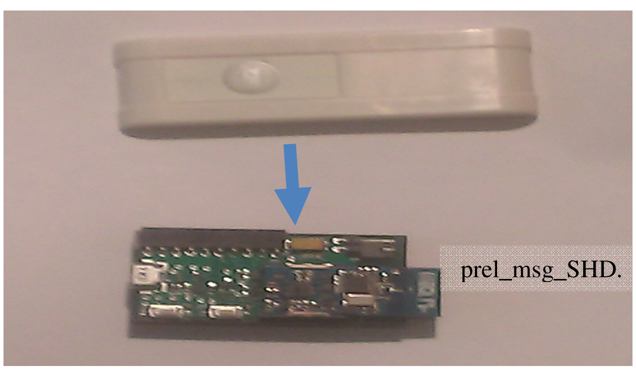



Fig.11 – The fifth Project

The last project contains an interpretation program of the messages coming from the SHD. The implementation is seen in Fig.11. This program can be directly used to monitor the automobile. Thus, alarms can be set if the vehicle is moved, started, what temperature is inside, etc. Commands can be sent, as seen in the previous project to lock the doors, stop the engine, start the air conditioning, etc. The car will be as such in a permanent state of monitoring and control.

One utility example can be seen when a person wants to leave home in an hour. This is signaled to the smart house intelligent system, which verifies the temperature in the car and in case it is too high, turns on the air conditioning. In such a way the vehicle will be ready for the voyage.

## V. CONCLUSIONS

In the proposed paper is presented the exploitation of a technology that until recently was used only in the smart home. The idea to gather telemetry from inside a car is also an innovative one not implemented by the time of the current study.

The programs used in the C programming language are simple, with no great complexity and can be easily carried, debugged and tested.

The sensors have a battery life of several years, providing a long and independent functionality comparable to the average user to overhaul a car.

We can advance the application on IQRF devices in cars by manipulating a visual control panel touchscreen display that can receive both data and sensor nodes and monitor activity programs running in them, besides work and monitor the integrated smart home.

## ACKNOWLEDGMENT

This work was partially supported by the Polithenica University of Timisoara, Department of Applied Electronics with the assistance of Assoc. Prof. Dr. Eng. Dan Andreiciuc.

## REFERENCES

[1] APLICAȚII TELEMETRICE PENTRU INDUSTRIA AUTOMOTIVE UTILIZÂND DISPOZITIVE IQRF, Serban Victor

[2] ***, http://en.wikipedia.org/wiki/Telemetry

[3] ***, http://en.wikipedia.org/wiki/Wireless_sensor_network

[4] ***, IQRF Quick Start Guide, http://www.iqrf.org/weben/index.php?sekce=support&id=download

[5] ***, IQRF Technical Guide, http://www.iqrf.org/weben/index.php?sekce=support&id=download

[6] ***, DK-EVAL-04 User's Guide, http://www.iqrf.org/weben/index.php?sekce=support&id=download

[7] ***, CK-USB-04 User's Guide, http://www.iqrf.org/weben/index.php?sekce=support&id=download

[8] ***, SHD-SE-01 User's Guide, http://www.iqrf.org/weben/index.php?sekce=support&id=download

[9] ***, Wikipedia, http://en.wikipedia.org/wiki/IQRF

[10] ***, SOS Electronic Webinar, http://www.soselectronic.com/?str=1426

[11]. Brian Kernighan and Dennis Ritchie, The C Programming Language, 1978 (1st edition), Englewood Cliffs, NJ: Prentice Hall. ISBN 0-13-110163-3.