Implementation of the MAC Layer of a DSRC identification system

Daniel Fidalgo Morais, Ricardo Matos Abreu, João Nuno Matos

Abstract[1]— DSRC (Dedicated Short Range Communications) is a well known widespread technology. One of the main applications of DSRC is RTTT (Road Transport and Traffic Telematics), especially EFC (Electronic Fee Collection) on tolled roads, bridges or tunnels. Other vehicle-related applications had been envisaged, such as car parking or fuel refilling automatic payment. These applications were successfully developed, deployed and are in current use in Portugal. Especially for EFC, medium access control is of utmost importance. In this paper it is presented an implementation scheme for the MAC-layer of the RSE (road side equipment), suitable for low-cost microcontrollers, to operate between the two equipments (as in Figure 1), and also the MAC layer of the SPI connection, of the two Microcontrollers present within the RSE. The result is interface software that controls the referred communication, embracing hardware and software into a join solution. This hardware works with both transfer rates, the LDR (Low Data Rate), the current technology and the MDR (Medium Data Rate).

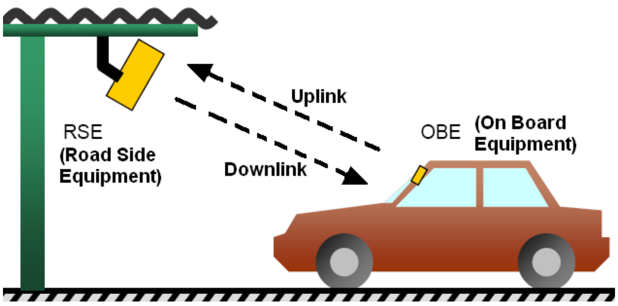

Figure 1 - Practical application of the project.

I. INTRODUCTION

Lately the world has witnessed the phenomenon of globalization, political social and economical changes have happening. Technology has ridden along with this tendency at many levels, providing a great number of products and services accessible to all of us.

In Europe, the European Union has the purpose of unify us into a united political, monetary and legislative system. There are also Committees in the field, managing all the technical issues, such as the Standardization European Committee.

Nowadays, Via Verde, which is an EFC (Electronic Fee Collection) system, uses is a DSRC (Dedicated Short Range Communication) system, in the RTTT (Road Traffic and Transport Telematics) environment, based on LDR (Low Data Rate) transmission, with a data rate of approximately 32Kbits/s. There are also other powerful data signaling rates such as MDR (Medium Data Rate), with data rates of approximately 256 and 512 Kbits/s.

This work is part of a larger project which aims the construction of two compliant units: an OBE (On Board Equipment) and a RSE (Road Side Equipment), using MDR data signaling rate, according to the European Standard EN 12795[1] (released by the Standardization European Committee) and using the know-how of the existing technologies already in use. The purpose of this work is to improve, and probably replace, the existing technology. This allows the technological unification of several countries under the same working pattern. The advantage is the usage of this service without the need of hardware changes, i.e., the user can travel in all the countries using the same tag for electronic toll payment. There are also other applications, such as the fuel and parking payment and also a new project the electronic license plate, also known as e-Plate.

Definitions:

- **Downlink:** is the communication channel on which fixed equipment transmits its information. Public and Private Downlink windows are distinguished by their information, Public if it is generic information and private if user specific information is present. Using MDR, the data signaling rate is 512 Kbits/s.

- **Uplink**: is the communication channel on which mobile equipment transmits its information. Public if it is generic information and private if user specific information is present. Using MDR, the data signaling rate is 256 Kbits/s.

II. COMMUNICATION ISSUES

This work is part of a larger project which aims the construction of two compliant units: an OBE and a RSE.

1 - The authors would like to thank Brisa - Auto-estradas de Portugal S. A. for the support of this work.

This DSRC system can be divided in two parts, please see Figure 2. The first part is the RSE and OBE wireless communication and the second part is the RSE internal microcontrollers' communication.

Regarding the first part, the main issues to study, in detail, are the standard EN12795, the features of the microcontroller and the existing software functions, of the physical layer. Standard EN 12795 imposes some time constrains, described in **Window management** section.

Concerning the second part, the studying issues are, regarding the hardware, the SPI (Serial Peripheral Interface) [2] of the microcontroller, and regarding the software, the Communication Protocol between the microcontrollers.

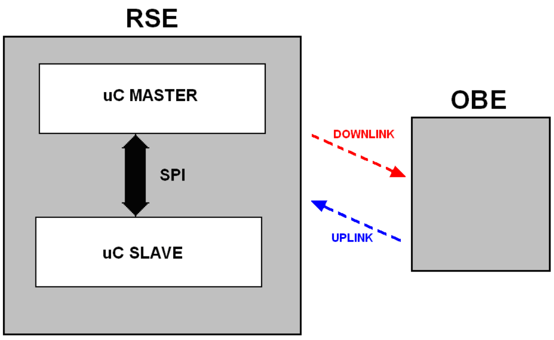

Figure 2 - Types of communication

1) Wireless Communication between the RSE and OBE

The Communication between OBE and RSE, both Uplink and Downlink, is based on frames exchange, according to constrains imposed by EN12795.

- **Standard EN 12795**

The Standard EN 12795 is one of a series of European Standards defining the framework of a DSRC (Dedicated Short Range Communication) in the RTTT (Road Traffic and Transport Telematics) environment.

This standard defines the Data Link Layer of the DSRC, it is subdivided in two sub layers, the MAC (Medium Access Control) and the LLC (Logic Link Control) Sublayer. The following paragraphs describe the main aspects of the standard.

- Frames Format:

The frames have the format shown in Figure 3 and are described in the following paragraphs. The size of the whole frame varies from 9 bytes up to 138 bytes, and this variation is associated with the LPDU (Link layer Protocol Data Unit) field.

The frame is delimited at the beginning and at the end, by a flag with the value 01111110 (base 2).

The Link Address Field has 5 bytes and contains the LID (Link Identifier), which is used to keep the communication private between different users.

Flag	Link Address Field	MAC Control Field	LPDU	Frame Check Sequence	Flag

Figure 3 - EN 12795 frames format

The MAC field is a single byte and it is used to indicate if the frame contains an LPDU, specify the transmission direction, allocate public/private windows and also request private windows.

The LPDU has from 10 up to 128 bytes and carries information regarding the OBE User and one byte for control.

The Frame Check Sequence field has two bytes and is used for error detection.

- Window management

There are different situations of message exchange each one with a specific time specifications, in which the RSE and OBE must be aware of and behave according to it. Here just the main situations are presented.

The first situation happens when the fixed equipment broadcasts a message in a Public Downlink Window and expects the response(s) in the following three consecutive Public Uplink Windows (please see Figure 4). T3 (160µs) is the downlink to uplink turn around time, T4b (32µs) is the maximum time to start of transmission and T5 (448µs) is the time duration of the uplink window

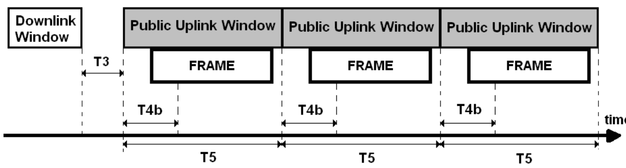

Figure 4 - Exchanged Windows after a Public Downlink

The second situation happens when the fixed equipment transmits in a Private Downlink Window (please see Figure 5) and the corresponding response arrives in a Private Uplink Window. T3 (160µs) is the downlink to uplink turn around time, T4a (320µs) is the maximum time to start of transmission.

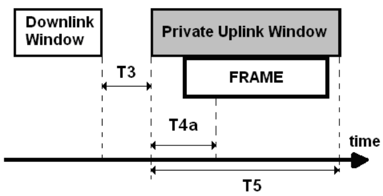

Figure 5 - Exchanged Windows after a Private Downlink

Public and private Downlink/Uplink windows are distinguished by their LID, whether a Broadcast LID or a Private LID is present.

- **Developed Software (wireless communication)**

Due to the differences of behavior between the public and private (number of consecutive windows and time constrains), there was the need of creating different functions for the Downlink. The following functions implement the desired solution.

- **void public_downlink (*array_out_public)**

This function executes a public downlink, by sending the information (parameter *array_out_public), from the RSE to the OBE. In this case data is generic for all vehicles. The time constrains are described in EN 12795. In public mode, there are three consecutive public windows. So, this function has to send the information and start the timer to control the turn around time in order to start the receiving mode,

- **void private_downlink (*array_out_private)**

This function executes a private downlink, by sending the information, of (parameter *array_out_private), from the RSE to the OBE. In this case data is specific for each vehicle. The time constrains are described in EN 12795. In private mode, there is a single window. So, this function has to send the information and start the timer, to control the turn around time in order to start the receiving mode.

- **void uplink ()**

This function executes an uplink, by receiving the information in data field, from the OBE to the RSE. This function can receive generic or specific OBE user data. The time constrains are described in EN 12795. The timer is set and if a timeout occurs, this function terminates.

2) Communication between the microcontrollers

The communication protocol between the two microcontrollers is an unbalanced Master/Slave, which means that all the exchanges are controlled by the Master. The master is a TX27 i.MX27 module and it is managed by our partner Brisa. The Slave is an Atmega 128 [3].

The hardware connection between microcontrollers, Master and Slave, was established using the SPI (Serial Peripheral Protocol), available in both microcontrollers, (Please see Figure 6)

- **Hardware connection SPI (Serial Peripheral Interface)**

The pins SCKL (SPI clock), SS\ (Slave Select), MOSI (Master Out Slave In) and MISO (Master In Slave Out), are standard SPI pins, while Busy and RTS (Request To Send) are stipulated for proper use of the protocol. Busy is high when the slave is not available for message exchange and RTS is high when the slave has information to transmit.

- **Communication Protocol between master/slave**

This chapter describes the main issues regarding the communication protocol. The frame format is displayed in the following Figure 7. The type of communication demands that, after the reception of a frame, the receiver must send an Ack (Acknowledge) if the frame is valid or a Nack (Not Acknowledge) if the frame is not valid, back to the emitter. Please see Figure 8.

- **Developed Software (master/slave communication)**

The software was developed in C language. Here only the main functions are presented, layer by layer, as in Figure 9 (the master functions were developed only for test purposes).

For the Lower MAC, the send and receive functions are:

- int slave_send_lower_mac (*str);
- int slave_receive_lower_mac();

These functions deal with the frame's fields *start* and *crc* (cyclic redundancy check), please see Figure 7.

In reception mode, the first step is to detect the *start* field (with value 0x02), then receive the rest of the frame (in accordance with size field). After the whole reception, the function calculates the crc of the received frame (not including the last two bytes corresponding to the crc field) and then compares it with those bytes. If they match the frame is valid, if not it means that the crc sent by the emitter is different than the one received by the receiver, so it is not a valid frame. If any of these two conditions do not occur the frame is discarded.

In sending mode, the sending frame has to be in accordance with the frame format, please see Figure 7.

For the Upper MAC, the send and receive functions are:

- int slave_send_upper_mac (*str);
- int slave_receive_upper_mac (*str);

These functions deal with the frame's fields *frame type* and *fsn* (frame sequence number), please see Figure 7.

In reception mode, it is in this stage that the frame type and the fsn are validated. The frame type has to be a data, an ack or a nack. The fsn is incremented whenever a valid frame exchange occurs. So the received fsn has to be in accordance with last increment. If any of these two conditions do not occur a nack is sent, if not an ack is sent.

For the Application layer, the function is:

- int slave_application ();

This function deals with the frame fields *size*, *message* and *data*. In this stage the validation of the content of these fields is done, according to a table of Application messages, with corresponding size and messages.

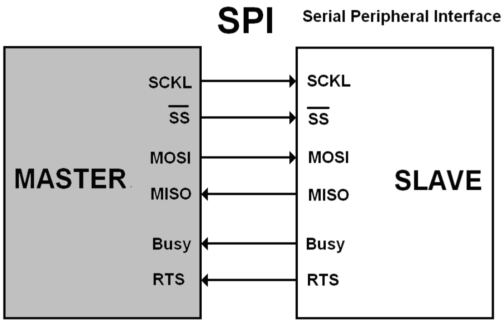

Figure 6 - SPI (Serial Peripheral Interface) connections

Start	Frame Type	FS N	Size	Message	Data	CRC

Figure 7 - Frame format of the Protocol

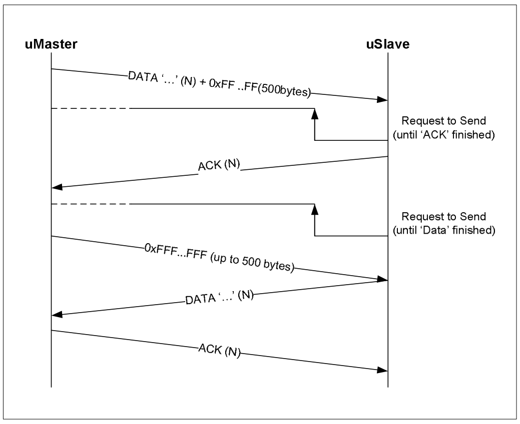

Figure 8 - Communication example between master and slave

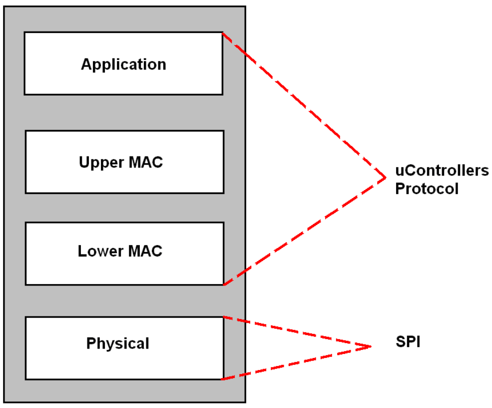

Figure 9 – Software layers model.

III. CONCLUSIONS AND FUTURE WORK

- MDR versus LDR

MDR technology, as expected, is better than LDR technology due to the higher data rate and the presence of information security protocols in MDR which are absent on LDR. The difference of baud rate, between both technologies, 32Kbits/s for the LDR and 256kbits/s or 512 kbits/s for the MDR, allows more versatility and robustness in the communication. The use of standard EN12795 as reference to establish the rules, transfer security is highly increased.

The developed hardware for this project is able to work, sending and receiving frames, with both technologies LDR and MDR. The RSE starts communication, by sending LDR and MDR alternatively and waits for a response of the corresponding sent frame, and completes the rest of the frames exchange in the respective mode.

In terms of software, the upper layers need to be able to reconfigure the operating mode constantly, according to the technology being used at that moment.

- Communication software, between RSE and OBE:

Regarding this part, the software complies with the demanded features and constraints required by EN 12795. Moreover, it is robust to communication errors and/or unexpected messages. Multiple error case scenarios, in communication, were considered and the software is robust enough to deal with those situations.

- Communication software between Microcontrollers:

Concerning this part, the software complies with the demanded features and constraints required by the existing protocol. Multiple error case scenarios, in communication, were considered and the software is robust enough to deal with those situations.

This approach is still simple enough to be implemented in a low-cost microcontroller with typical features. This is an ongoing work which will culminate with field tests in a short term.

REFERENCES

[1] [EN12795] EN 12795:2003 Dedicated Short-Range Communication (DSRC) - DSRC Data link layer: Medium Access and Logical Link Control (review);

[2] http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_ Bus (site visited in July 2009);

[3] http://www.atmel.com (site visited in May 2009)