IEEE 802.15.4 MAC Layer Adaptation for UWB

Jérôme Vernez*, Stephan Robert*, Alexandre Pollini**, Andréas Hutter**

* University of Applied Sciences of Western Switzerland (EIVD), Rue de Galilée 15, CH-1400 Yverdon-les-Bains
Email : {stephan.robert, jerome.vernez}@eivd.ch

** Centre Suisse d'Electronique et de Microtechnique (CSEM), Jaquet-Droz 1, CH-2007 Neuchâtel (Switzerland)
Email : {alexandre.pollini,andreas.hutter}@csem.ch

ABSTRACT

This paper presents an adaptation of the IEEE 802.15.4 MAC layer to an Ultra Wide Band (UWB) radiofrequency physical layer developed at the CSEM in the framework of a European project called URSafe (stands for "**U**niversal **R**emote **S**ignal **A**cquisition **F**or H**e**alth"). A hardware demonstrator based on a low power technology has been realized for supporting different test applications, the final objective being to realize a body-worn Wireless Personal Area Networks (WPAN) for telemedicine applications.

I. INTRODUCTION

The objective of this project is to adapt and implement the IEEE 802.15.4 MAC layer to an existing UWB physical layer developed at the CSEM [1]. Other proposals are available to adapt the MAC layer to UWB [2,3,4]. The baseband implementation is realized with a very low power microprocessor, the MSP430 from Texas Instrument. This approach extends the functionalities of the system used in the URSafe [5] project which aims to develop a telemedicine concept for medical care of elderly people who could, with such a platform, be treated and monitored at home. The basic idea of URSafe is to acquire physiological data from a person remotely. The medical team can monitor and react promptly in case of need. To support this application two wireless networks are required, as shown in Figure 1. The first one is short range and worn by the monitored person (Wireless Personal Area Network: WPAN) whereas the second one is longer range (WLAN, GPRS, UMTS and/or satellite). The gateway in between the networks is done through a unit called 'the Portable Base Station" (PBS) which is also worn by the monitored person. The short range wireless connection is done using UWB and allows the PBS to receive and pre-process measured medical data, which have been sent from the sensor-devices. These data can be forwarded to external devices via the longer range network. Control command can be received with the same support. Other features, like communicating with simple sentences or establishing a direct hand-free communication with the patient are considered. Regarding medical data acquisition, four kinds of sensors have been chosen [2]: 1) a miniaturized ECG sensor, 2) a portable SpO2 sensor, 3) a fall detector and 4) a speech sensor.

In this paper, section 2 shows briefly how the system works, some implementation choices and general concepts of the IEEE 802.15.4 MAC layer. Section 3 describes the actual software realization in terms of primitives, timers and interruptions. Finally, in section 4, details about applications which have been implemented to test specific features of the MAC layer (synchronization, association processes, indirect transmission of data,…) are provided.

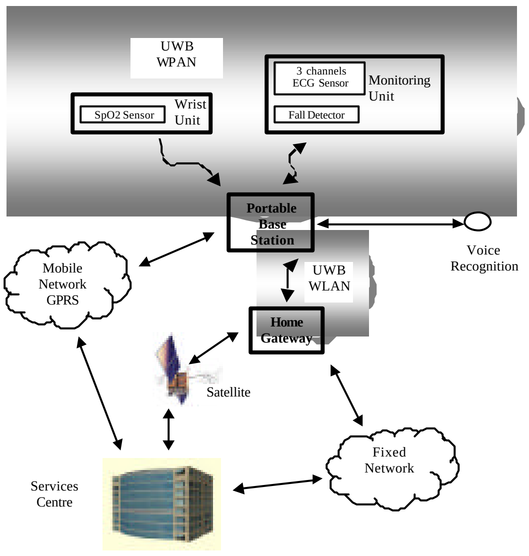


Figure 1: General overview of the European project URSafe which constitutes the framework of the developments exposed in the paper.

II. IEEE 802.15.4 OPERATIONS AND SYNTHESIS

In this section, we want to describe the programming choices, which are driven by applications and by the available resources. At the time of the project only two hardware modules were available, which have restricted the applications to use only one Personal Area Network (PAN). Security aspects have not been treated in this first phase. They are left out of the scope of this project. Due to the low power microprocessor restricted capacity, sequential programming has been realized in general. The base time frame is imposed by the UWB physical layer at 9.6 kb/s. Coding is based on Manchester, which means that one symbol (2 bits) is transmitted every 208 µs.

A. Communications

The two implementations of CSMA/CA for IEEE 802.15.4 are based on energy measurement [6]. The physical layer decides either if the channel is busy or not on the base of a quite complex energy measurement of the channel. In the framework of URSafe, such a measurement is not possible. However, a check on some pins of the radio-interface gives the information on the availability of a given channel. To minimize the energy consumption, an indirect transmission for data is realized. A coordinator has a transmission queue of up to seven MAC frames (according to the standard [6]) and waits for solicitations from the different devices to transmit data they ask for. If memory space is missing in the queue, an alert message is sent to upper layers if new transmission attempts are done. When either a frame or an acknowledgment is lost, it is sent again after a timeout, but no more than 3 consecutive times.. In the no beacon enabled mode, when the receiver is down, it is not possible for the MAC layer to be waked up by the physical layer. Therefore, it is the responsibility to the application layer to switch off or to switch on the RF front-end in accordance with the application. Data are rejected if they are not compliant to the IEEE 802.15.4 standard frame, after a CRC check. In beacon enabled mode each device can determine if data with its personal address are pending in the coordinator memory when it receives beacons. The coordinator regularly sends beacons to everybody, with addresses (corresponding to the different devices) associated to memorized data. When a device recognizes its address, it sends a message (data request) to the coordinator, which has to be acknowledged. Afterwards, the coordinator sends data to the device. If transmission errors happen in this data exchange, recovery processes (e.g. retransmissions) are provided by the IEEE 802.15.4 MAC layer. In accordance with IEEE 802.15.4, all devices, the Reduced Functionality Device (RFD) and the Full Functionality Device (FFD), are able to realize a passive scan and an orphan scanning. When the full functionality is implemented (FFD), active scan and energy detection scanning are also possible. Passive scanning allows every device to locate a coordinator transmitting beacons.. With passive scanning, no beacon request command is sent by the device which makes the scanning. The device does only polling. Active scanning allows a device to send beacon request command on each channel. If on one of the channels a PAN coordinator is already present , it does not answer this command frame. On the other hand, if the coordinator is in a non-beacon enabled state, it answers with a beacon in a CSMA/CA unslotted mode. The device can then establish a list of PANs reachable, from where beacons have been received. When a device looses the connection, it can make an orphan scanning which consists to send an orphan notification to all known channels. That way it tries to find the coordinator it was associated with before losing the synchronization. The procedure ends if a coordinator realignment frame (with the PAN ID and the old address of the device) is received or when all channels have been scanned without answer. In our devices, only passive and orphan scanning have been implemented.

B. PAN creation, association and dissociation

When all functionalities of the MAC layer are implemented, the creation of multiple PANs can be realized. In this case, an identity for the PAN and a free channel have to be chosen. Here, only one PAN is envisaged per channel and identified uniquely. Beacons are thought to indicate to devices that data are waiting for them, but also for synchronization purposes. They have the highest priority in all communications. To be able to join a PAN, a device must know on which channel it has to communicate. Normally, this is done with a channel scanning. The device has to initiate an association procedure which consists first to send an “association request” to the coordinator. After checking that resources are sufficient, the coordinator will send an association answer (with a new address valid inside the PAN) to the device. To quit the PAN, there are several possibilities: 1) The coordinator wants a device to quit. In this case, it sends a notification to the device which has to acknowledge that it understood. If the device does not acknowledge, the coordinator considers it as dissociated anyway. 2) The device wants to quit the PAN. It has to send a dissociation notification to the coordinator which will acknowledge the request. As before, if the device does not receive the acknowledgement, it considers itself as dissociated anyway. When application layers notice that data can not be sent because of successive failures, the device concludes that it is orphan. It can either realign to the coordinator with a “coordinator realignment” procedure or associate itself to another PAN using the standard procedure.

C. Synchronization

In a beacon-enabled network, every device has the possibility to synchronize with beacons received from the coordinator. It is necessary to verify the validity of the beacon and to be sure that the PAN identifier is the one to which the device is associated. When that is done, the device has to listen the channel regularly When the receiver has not received a beacon at the time envisaged, a procedure for listening a longer time is engaged. When many beacons are missed, the MAC layer notifies it to the upper layers indicating that the beacon is lost.

III. IMPLEMENTATION

Realization has been done in ANSI C, based on the standard [6]. The IEEE 802.15.4 is also described in SDL but this approach has not been followed. URSafe hardware has limited functionalities by comparison with a ‘true’ IEEE 802.15.4 RF front-end and does not provide all the requested functionalities. Furthermore, the standard has been interpreted and partly implemented in C language on a MSP430 low power microprocessor. Compilation, debugging and code transfer on the hardware have been performed in the IAR Embedded Workbench™ development environment [7].

A. Description

MAC primitives have been translated in C functions. The boards developed for URSafe are the platform on which the applications have been tested. A Linux-based PC has been used for sending and receiving data via a serial port RS-232 and hence, to explore the performances of the overall wireless link, as shown in Figure 2. At first, only basic functionalities have been implemented and the complexity of the implemented applications has been increased step by step.

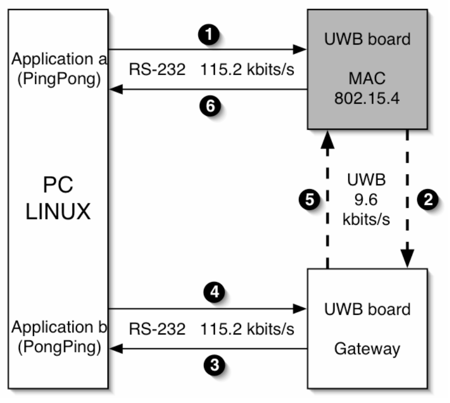


Figure 2: Assembly scheme of the testapplications which consisted to transmit and receive data at the MAC level. The construction, segmenting and reassembly of frames as well as acknowledgements have been checked with such a scheme of the application.

The implementation is based mainly on sequential programming at the application level. All primitives are executed sequentially in the application layer, including the primitives received from the MAC layer. At the time of the project, the only functionality that was not realized in a sequential manner is the emission and reception of beacon frames. We use interruptions to guarantee that synchronization frames are sent and received at the right times. In sequential programming, one has to consider time periods to use specific functionalities at the application layer. The use in this project of a microprocessor with strongly limited resources (e.g. 2 kbyte of RAM) requires a particular care of the memory management. A reservation of the memory space is preferable, because MAC and PHY frames are directly built in the RAM. Three "buffer" variables are generated at the start and are of fixed size. The first and second buffers are used to memorize frames from the MAC and the physical layers. The third one is used to accommodate data from upper layers. In a optimized final implementation only one buffer will be used but for debugging purposes, three buffers have been chosen (called Buffer UP, Buffer MAC and Buffer PHY). Another memory space is necessary for the transmission queue of which may contain up to seven frames.

B. CSMA/CA slotted and MLME-RX-ENABLE

These two functionalities are important. They can guarantee emission and reception of frames in a precise time frame. Implementation of CSMA/CA is slightly modified compared to the standard. The objective is to determine if data can be sent during a super-frame (shown in Figure 3).

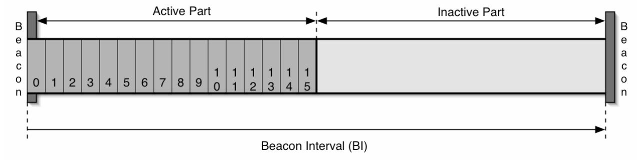


Figure 3: Structure of the IEEE 802.15.4 superframe, with an active part (16 slots) and an inactive part. During the active part, 8 slots are dedicated to contention access and 8 slots are dedicated to contention free access. The inactive part permits the coordinator to enter into a low power mode.

Transmission can be done only if the current local time is lower than a defined constant. An additional functionality has been added to the slotted CSMA/CA to avoid failures for channel access during inactive periods of the super-frame. During this time interval, transmission are forbidden and a station must wait until the next active period. MLME-RX-ENABLE [6] is also a bit modified, because many cases for reception are possible (control and data frames). In our project, this primitive allows us to calculate how much time the receiver can be on, based on the actual time. In other words, it enables upper layers to be either active or inactive during a certain time.

C. Cyclic Redundancy Check (CRC)

Two approaches have been selected. The first one corresponds to an implementation which computes the CRC based on a polynomial generator and the preliminary initialization of the polynomial content. This implementation uses few memory space but utilizes expensively the microprocessor processing resources to compute the CRC. The second implementation is based on a pre-calculated table, which corresponds to the polynomial generator (which also corresponds to the required CRC of the IEEE 802.15.4 standard). The initial value of the CRC can be chosen, but it is zero according to the standard. This way of doing requires more memory space, but it requests less processing resources of the microprocessor. The results of the two implementations are identical. For error control in certain applications, it can be advantageous to use the second method, especially with low power microprocessor with strongly limited processing capabilities.

D. Use of the timers

The standard defines 12 timing variables for the overall timing control. In the MSP 430, only three timers are available. Hence a good management of the timers is capital. In a "beacon enabled" network, many variables depend on the measurement of time. At the microprocessor level, it is not possible to guarantee real-time management, because time measurements are made in the software. A first idea was to use one timer per layer but it has been found that this method presents disadvantages. Time is not measured anymore at the appearance time of the beacon which is managed by the watchdog timer. The adopted idea is to use a global variable (Time) allowing the permanent control of the actual time in number of symbols. Timer A is used to drive variables indicating various time intervals. Three variables can be incremented at each exception: *Time*, which is the time basis for the implementation, a time variable of 32 bits (*macBeaconTxTime*) used by the PAN coordinator when the network is “beacon enabled”, a variable of the PAN descriptor of 32 bits (*TimeStamp*) used by the devices when the network is “beacon enabled”.

Measurements of the time proceed in the following way: A global variable (*StartChrono*) is set to the actual time. When the measurements stops, another global variable (*StopChrono*) is set to the actual time. The difference between these two variables gives the time past in number of symbols. Timer B is used at the physical layer and causes an exception when sending data or waiting for data takes too much time. The timing is not very precise and it is used as a guard time if something goes wrong in the physical layer. The watchdog timer manages the sending time frame of beacons in a coordinator or the beacon reception time frame in a device. when the network is beacon enabled. It indicates when a beacon has to be received from the coordinator. This timer is the basis to check that variables *macBeaconTxTime* (coordinator) and *TimeStamp* (device) correspond to the interval between beacons (defined in the PAN as a global parameter). These checks have to be performed quickly (in the order of one symbol duration). The incremental step is therefore 125 µs while one symbol duration is 208 µs. The microprocessor is clocked at 4 MHz when it is not power save mode. When the network is beacon enabled, the beacon has to be in the timeframe of a super-frame. Therefore a timing correction has to be done. Time to send and receive a beacon has to be taken into account. Otherwise, global variables have to be saved before the interruption to be able to be restored thereafter. In a non-beacon enabled network, the watchdog timer is not activated.

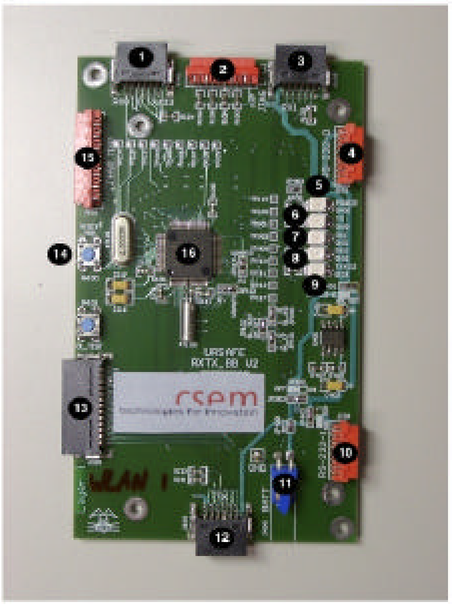
E. Interruptions

The implementation requires a careful management of the interruptions resources. For example, during the watchdog timer interrupt routine (beacon management), the timer A interrupt must always be enabled in order to not loose real time synchronization. By default in the microprocessor [8] - during an exception - all interruptions are prohibited. To know the current state of interruption authorizations, one has to look first in the state register of the microprocessor one specific bit (Generic Interrupt Enable) which should be set to 1. Secondly, the mask of each interrupt source must be set or reset in accordance with the application. We have to activate exceptions by setting the Generic Interrupt Enable bit during the execution of the interruption routine of the watchdog to allow the timer A to interrupt the process. This interruption is critical. For certain applications, the microprocessor blocked when it had to execute other jobs than treating only the interruptions of the timer A and the watchdog. The cause was that exceptions appeared too often. An optimization of the interruption management was necessary to release the resources of the microprocessor. The number of interruptions raised by the timer A has been decreased. Instead of raising an interruption at each symbol, it is preferable to interrupt every a given number of symbols but we loose a bit of the real time accuracy. A limit has to be determined. Now, the timer A raises an interruption four times less often than previously envisaged. To save even more microprocessor resources when the network is beacon enabled, the rate of the interruptions of the watchdog timer has been decreased to 2 ms instead of 125 µs. Sending of beacons is then realized in a less precise way. In the final implementation, the choice was to privilege a better operation rather than a good precision. However, it is not a drawback it is possible to take into consideration this kind of small imprecision by allowing , for example, the switching on of a receiver before the real arrival of a beacon.

IV. APPLICATIONS

Applications have been basically created to check the functionalities of our implementation. Four applications have been created for the project. Hardware modules are CSEM boards designed in the framework of the URSafe project (shown in Figure 4). The software is programmed on the microprocessor and debugged through the JTAG port. Data transmission between two UWB modules can be done at a baseband level for debugging purpose. A serial RS-232 port on the boards allows their connection to a PC computer. Leds allow to determine the directions of the transmissions and to know the actual mode of transmission.

The goal of the first application is to check transmission and reception of data at the MAC layer. Construction, reading , segmentation and reassembly of frames can be tested. The used scheme is shown in Figure 2. This application proceeds in two phases. First, only transmission at the MAC layer level is tested and secondly programs have been modified to be able to realize a segmented transmission and a reassembly at the receiver side. The computer, equipped with a Linux operating system, simulates the upper layers in generating a succession of bits sent to the boards via the RS-232 interfaces. Bits can be received via the same way. In the framework of URSafe, two application-programs have been created, *PingPong* and *PongPing*. *PingPong* is used to transmit 256 bytes and waits for an answer through the serial interface (RS232) several time. *PongPing* waits for data on the serial interface and when it receives the data, sends them back through the interface as a confirmation.



1 : UART 0	9 : TX433 LED
2 : CPLD JTAG Port	10 : RS-232-1 port
3 : UART 1	11 : Power connector (Batt)
4 : RS-232-0 port	12 : TX port
5 : Power Supply LED	13 : Reserved port (debugging)
6 : Mode TX LED	14 : Reset button
7 : Mode RX1 LED	15 : Microprocessor JTAG port
8 : Mode RX0 LED	16 : Microprocessor MSP 430

Figure 4 : CSEM card called RXTX_BB V2, on which the microprocessor has been programmed. The card was used in the framework of URSafe. The card has two serial ports which allow reception and emission of data from a personal computer. LEDs are present essentially for debugging purposes (direction of communications).

Figure 2 shows the course of operations. (1) the program *PingPong* sends a first testing sequence of 256 bytes which is transmitted through the interface RS-232 to the UWB module (RXTX_BB V2) where the MAC layer has been implemented. Data are segmented into 802.15.4 data frames and transmitted to the MAC layer. They are then forwarded to the second module (2). The second module recovers one frame after the other and transmits them through the second RS-232 interface to the *PongPing* program which is first in a reception mode and then in a transmission mode (3). If the received data are correct the frames header is modified the data are sent back (4, 5, 6).This platform allow to send, to receive and control data through the UWB physical layer and the 802.15.4 MAC layer from one application to another. Other applications have been tested. The second applications aimed at simulating an information exchange with acknowledgements in an unslotted CSMA/CA mode. The goal of the third application was to set up a network with a coordinator. Two important functionalities have been integrated for this application: the synchronization between devices and the coordinator, association to a PAN. The objective of the fourth application was to realize an indirect transmission of ASCII data.

V. CONCLUSION

The standard used to implement the MAC layer has been found accurate. Moreover, the use of the 802.15.4 MAC layer extends satisfactorily the functionalities of the UWB front-ends that were previously driven by a non-standardized, in-house MAC layer. it's the MAC adaptation to an UWB-based physical layer has been well carried out although few routines have not been implemented yet (security, multi-PAN). The applications, whose goal were to check the different functionalities, performed as required. Data transfer with acknowledgments, synchronization between devices and the coordinator, association/dissociation processes have been tested. Differences found between the standard and our current implementation are relative basically to the differences between the UWB front-end and the true IEEE 802.15.4 functionalities.

REFERENCES

[1] J. Gerrits, A. Pollini, P. Müller, B. Gros, *A UWB Wireless PAN for Telemedecine Applications*. Research report, CSEM, Neuchâtel, 2002.

[2] J.-Y. Le Boudec, R. Merz, B. Radunivic, J. Widmer, "A MAC protocol for UWB Very Low Power Mobile Ad-hoc Networks based on Dynamic Channel Coding with Interference Mitigation", EPFL, Technical report No. IC/2004/02, January 2004.

[3] A. Hicham, Y. Souilmi and C. Bonnet, *Self-balanced receiver-oriented MAC for ultra-wide band mobile ad hoc networks*. In International Workshop on Ultra Wideband Systems (IWUWBS'03), june 2003.

[4] M.Z. Win and R.A. Scholtz. *Ultra-wide bandwidth time-hopping spread-spectrum impulse radio for wireless multiple-access communications*. IEEE Transactions on Communications, 48(4):679-691, April 2000.

[5]F. Castinié et al., *The U-R-Safe project: A multidisciplinary approach for a fully "nomad" care for patients*. Invited paper for SETIT 2003, Tunisia, March 17-21, 2003.

[6] IEEE Standards: Draft IEEE 802.15.4/D18, February 2003. http://www.ieee.org.

[7] http://www.iar.com

[8] Texas User's guide MSP430.