

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/45534376>

Design of a Learning Environment for Embedded System

Article · June 2010

Source: DOAJ

CITATIONS

4

READS

54

4 authors, including:



António Espírito Santo
Universidade da Beira Interior

67 PUBLICATIONS 149 CITATIONS

[SEE PROFILE](#)



Pedro Dinis Gaspar
Universidade da Beira Interior

175 PUBLICATIONS 434 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Project of PhD research programme and thesis [View project](#)



EDGWISE [View project](#)

Design of a Learning Environment for Embedded System

Tiago Gonçalves, A. Espírito-Santo, B.J.F. Ribeiro, P. D. Gaspar

Abstract— Embedded systems have an everyday presence and direct impact in our lives. Therefore, Universities are continuously improving their courses in microprocessor and embedded programming. Although the diversity of curriculums, the availability of learning tools, where the student can practice and improve their skills, is a key factor to the success of the learning process. The platform developed and presented in this paper results from author's experience in teaching embedded systems. From the analysis of teaching/learning needs, a learning environment based on the MSP430 family was designed. This tool can be expanded by modules and adjusted, at a specific time, to student's real needs. All modules can be interconnected by an I2C bus, allowing to expand the capabilities of the platform. The developed modules allow the practice of subjects related with digital IO, analog interface, user interface, wireless communications, and energy management and conservation.

Index Terms—Embedded system, learning, laboratory, programming, microprocessors.

I. INTRODUCTION

EMBEDDED systems courses are today included in most of the graduate and undergraduate curriculums of electrical engineering and Computer Science from all over the world [1-5]. Increasing impact of microprocessors in our daily lives justifies this option. These devices can be found either in very simple products, like toasters or toys, or in highly technological products, as for examples, cellular phones, camcorders or cars.

The complexity of an embedded system can change from product to product, depending on the task that they must perform. Therefore, embedded system designers must have knowledge in different areas that, sometimes, are separately handled. Hardware project requires knowledge related with digital and/or analog electronics, and, at the same time, with electromagnetic compatibility issues that cannot be forgotten in high frequency operation or in products that must work in very restrictive environments as the ones found in hospitals. In turn, the designer must project the software, required by the hardware, allowing it to work as expected. Subjects like

operating systems, real time systems, fixed and floating-point arithmetic, digital signal processing, and programming languages as assembly, C/C++ or Java are of major relevance to the development of embedded systems.

As stated previously, the skills that an embedded systems designer must hold are, itself, highly complex and spread [6-7]. If beyond the previously knowledge, the student need to learn how to work with a complex development kit, then he probably will fail. Even if the student is adequately prepared in the required subjects, the time taken to obtain visible results is, sometime, responsible for student demotivation and consequent failure.

This paper presents a platform designed from scratch to support embedded systems teaching. With the improvement of student's skills, tool complexity can also be increased, adding more functionality to it. In its bare configuration, the student can practice only with the digital IO. Functionalities normally found in development kits are not present, avoiding this way student distraction. The complexity of the associated modules can be increased with student progress. Functionalities continuity between modules is granted by an I2C bus. The developed communication protocol allows data exchange between all modules.

This paper is organized as follows. Section II briefly overviews the proposed embedded system learning platform, identifying its main characteristics and the devices used to build it. In Section III, a detailed description of the platform is presented. Section IV describes the hardware interface used to implement the interface bus. The firmware that implements the communication protocol is presented in Section V.

II. HARDWARE PLATFORM DESIGN OVERVIEW

The development of an embedded system relies in a large and assorted set of technologies in constant and rapid evolution. The learning platform here described wish to offer the possibility of access to this set of technologies in an educational environment. This way, the usage of commercial evaluation kits is discouraged, since; they are mainly developed to check the potentialities of a specific product without educational concerns.

The MSP430 family selection is justified by the amount of configurations available, with a high count of peripherals, different memory sizes, and popularity. Another relevant attribute is the learning curve of these devices that, from author's experience, allows to rapidly obtain results [8].

Four modules were developed with increasing complexity. The learning platform can thus satisfy the needs from

Tiago Gonçalves was a MSc student at the Electromechanical Engineering Department, University of Beira Interior, Covilhã, Portugal. (e-mail: blueflash.pt@gmail.com).

A. Espírito-Santo is with the Electromechanical Engineering Department, University of Beira Interior, Covilhã, Portugal. (e-mail: aes@ubi.pt).

B.J.F. Ribeiro is with the Electromechanical Engineering Department, University of Beira Interior, Covilhã, Portugal. (e-mail: bruno@ubi.pt).

P. D. Gaspar is with the Electromechanical Engineering Department, University of Beira Interior, Covilhã, Portugal. (e-mail: dinis@ubi.pt).



Fig. 1. Example of one allowed configuration - Module 0 and Module 1 interconnected.

beginners to experienced students.

The learning platform here presented allows users to experiment with different kinds of interfaces, as for example: OLED display, seven-segment displays, and conventional LED. Students can interact with several other peripherals, as for example, among others: pressure switches, a touch-screen pad, a joystick, and an accelerometer.

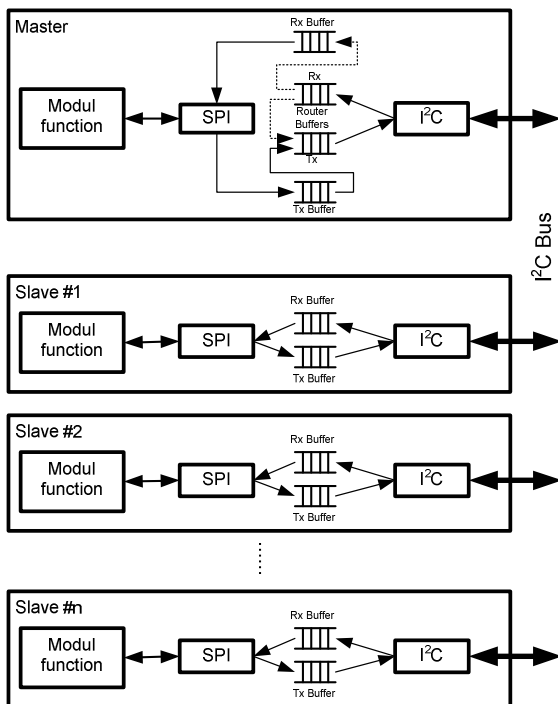


Fig. 2. Structure overview of the learning environment for embedded system.

The MSP430 family has good performance in situations where the energy consumption is a major concern [9]. The energy consumed by each module can be monitored and displayed in real time. Beyond these functionalities, the student can also improve their skills in real time data exchange. The developed modules overview structure can be observed in Fig. 2.

The strategy adopted to design the teaching platform took in consideration the following general characteristics:

- Modules can exchange data through an I2C bus;
- Each module has two microcontrollers, one manage the communications, while the other it is available for student's work.

- A SPI bus connects both processors in the same module.
- Energy consumed in each module can be measured and displayed in real time.
- All modules have standard dimensions.

The modules were constructed around three devices from the MSP430 family: MSP430F2112, MSP430FG4618 and MSP430F5419. The main characteristics of these microcontrollers are summarized next.

The MSP430F2112 microcontroller, that can operate with a maximum clock frequency of 16MHz, it has 32kB of flash memory, and 256B of RAM, one 10-bits ADC, two timers with respective compare/capture units, and a number of digital IO sufficient to satisfy the needs of the applications. The device also supports SPI, UART, LIN, IrDA, and I2C communication protocols. All the functionalities specified to the Module 0 – Basic Interface and Power– and for the Module 1 – Basic Interface – can easily be implemented with this device. Inter-module communications are managed by one of the devices present in each module.

The Module 2 – Analog and Digital Interface – developed to teach analog data acquisition and signal processing was constructed around the MSP430FG4618. This device can operate at maximum clock frequency of 8MHz, it has 116kB of flash memory and 8kB of RAM. The high count of digital IO is shared with the on-chip LCD controller. Other peripherals that are normally used in the analog/digital processing are also present: 12-bits ADC, two DAC, three operational amplifiers, DMA support, two timers with compare/capture units, a high number of digital IO, and hardware multiplier. As the previous device, this can also support SPI, UART, LIN, IrDA, and I2C communication protocols. On-chip LCD controller allow the connection of LCD with 160 segments.

The MSP430F5419 is a device from the MSP430 family that has a high processing power. It can operate with a maximum clock frequency of 18MHz, and it has 128kB of flash memory and 16kB of RAM. This microcontroller also has a hardware multiplier, a real time clock, direct memory access, and a 12-bits ADC. The communication protocols SPI, UART, LIN, IrDA, and I2C can be implemented in four independent peripherals. The device has three independent timers with compare/capture units.

III. MODULES DEVELOPMENT DETAILS

A. Module 0 – Basic interface and power

This module was thought to help the student first's steps in learning embedded systems. Usually, this kind of user does not have any experience with the development of embedded systems. This module will help him to take contact with the microcontroller architecture and, at the same time, with the software development tool. The internal structure of the module is illustrated in Fig. 3.

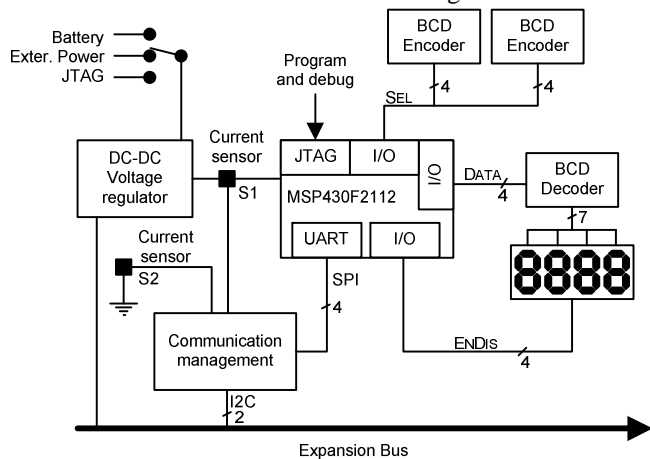


Fig. 3. Nodule 0 – Structure of the Basic interface and power module.

Beyond provide power to itself, the module can also power the modules connected to it through the Expansion Bus. Three different options are available to power the system: a battery, an external power source, or the JTAG programmer. The DC-DC converter allows an input range from 1.8V to 5.5V, providing a 3.3V regulated output. Powering from the JTAG is only available to support programming and debugging activities.

A numerical display was built with four seven-segments independent digits. The writing operation is controlled by four data lines (DATA) that, using a BCD decoder, which allows to write the desired value in the display. The selection of which digit will be written, at specific moment, is performed by four control lines (ENDIS). Because the BDC decoder do not latch the output, the value to exhibit in the display must be continually refreshed by the microcontroller, with a minimal frequency of 15Hz.

Beyond the MSP430F2112 microcontroller, available to the student's work, exist also another microcontroller to manage all the communications made through the I2C channel of the Expansion Bus. A SPI channel connects these two microcontrollers.

Despite the simplicity of this module, its predefined task is the visualization of the current and the energy consumed by each one of the modules connected by the Expansion Bus. Two BCD encoders, with ten positions each, are connected to the microcontroller through eight selection lines (SEL), and are used to select from which module the information will come. To execute this feature, the microcontroller must be loaded with a specific firmware.

B. Module 1 – Basic interface

This module is directed to the student that already has some basic knowledge in the embedded systems field. As the

previous module, this is also based in the MSP430F2112 microcontroller. Connected to this device, as can be seen in Fig. 4, can be found eight switches, eight LEDs, and a seven-segment display with two digits.

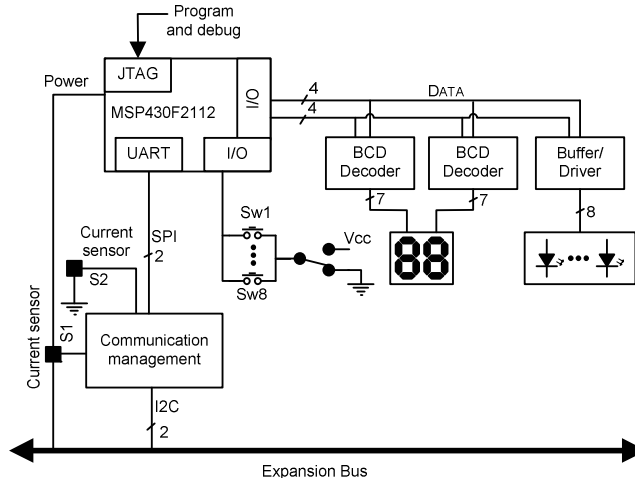


Fig. 4. Module 1 – Structure of the Basic interface module.

This module intends to develop student's competences related with synchronous and asynchronous interruptions. Simultaneously, the student can also explore programming techniques, as for example, the ones based in interrupts or port polling to check the status of the digital inputs or impose the status of the digital outputs. With this module, the student can have the first contact with the connection of the microprocessor with other devices, forcing him to respect accessing times.

Two BCD decoders, with the ability to latch the outputs, are used to write in the display. While the DATA lines are used to write the value in the display, the LE lines are used to select which digit will be written. The DATA data lines are also used, with a buffer/driver, to turn on or off each one of the eight LED. The inactive state of the switches can be configured by the user.

C. Module 2 – Analog and digital interface

The student with more advanced knowledge in the embedded systems field can use this module to improve his capabilities to develop applications where human-machine interface, analogue signal conversion, and digital processing are key aspects.

The design of this module was performed around the MSP430FG4618 microcontroller. On-board can be found a navigation joystick with four positions and a switch, a rotational encoder with 24 pulses/turn and a switch, a speaker output, a microphone input, generic IO, and a alphanumeric LCD. The internal structure of the module 2 – Analog and digital interface - is illustrated in Fig. 5.

Students can explore the operation of LCD devices, taking advantage from the on-chip LCD controller. The module can also be used to improve the knowledge related with the development of human-machine interfaces.

An example of a laboratory experience that students can perform with this module is the acquisition of an analogue signal, conditioned it with the on-chip op-amps, and digitally processes the result of the conversion with a software application. The result can be converted again to the analog

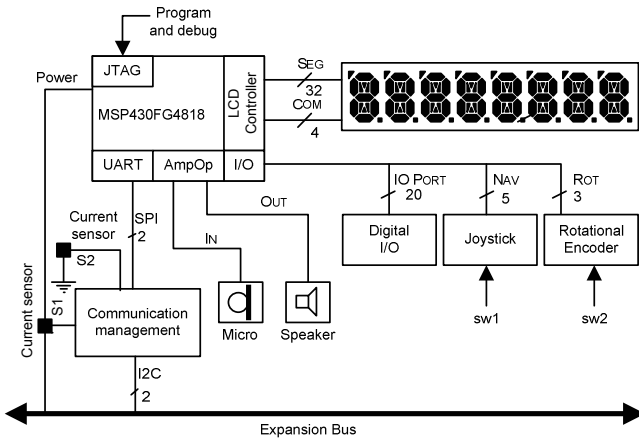


Fig. 5. Module 2 – Structure of the Analog and digital interface module.

world using the on-chip DAC.

Taking advantage from the on-chip op-amps its possible to verify the work of different topologies, as for example: buffer, comparator, inverter and non-inverter amplifier with programmable gain, and differential.

The digitalized signal can be processed using the multiply and accumulate hardware peripheral. Students can conclude about the relevance of this peripheral in the development of real time applications.

D. Module 3 – Communication interface

This is the most advanced module. With this module, the student has access to a set of sophisticated devices that are normally included in embedded systems. The student can explore how to work with: an OLED display with 160x128 pixels and 262 thousand colors, a three axis accelerometer, a SD card, a touch-screen, an USB port, two PS/2 interfaces. The module also owns connectors to support the radio frequency modules Chipcon-RF and RF-EZ430. Module 3 – Communications interface module - is illustrated in Fig. 6.

This microcontroller allows a good performance in the communication field, since it includes four peripherals that can support up to several different protocol buses, as for example, serial, infrared, SPI or I2C.

The accelerometer MMA7455L, with adjustable sensibility, uses an I2C interface to connect with the microcontroller in the port USCIB1. It has two outputs that can signal different conditions, like data available, free fall, or motion detection. This device can be enabled by the microcontroller through the VACCEN line.

The resistive touch-screen with four wires, and a 6x8cm area, needs a permanent management of their outputs. To free the microcontroller from this task, a touch-screen controller is used to connect it with the microcontroller through I2C bus (USCIB1). The line PENIRQ informs the microcontroller that the touch-screen is requesting attention. To save power, the microcontroller can enable or disable the touch-screen controller through the VTSCEN line.

The two PS/2 ports are connected to the microcontroller by a bidirectional level shifter to adapt the working voltage levels from 5V to 3.3V. The clock signal is provided by the Timer B.

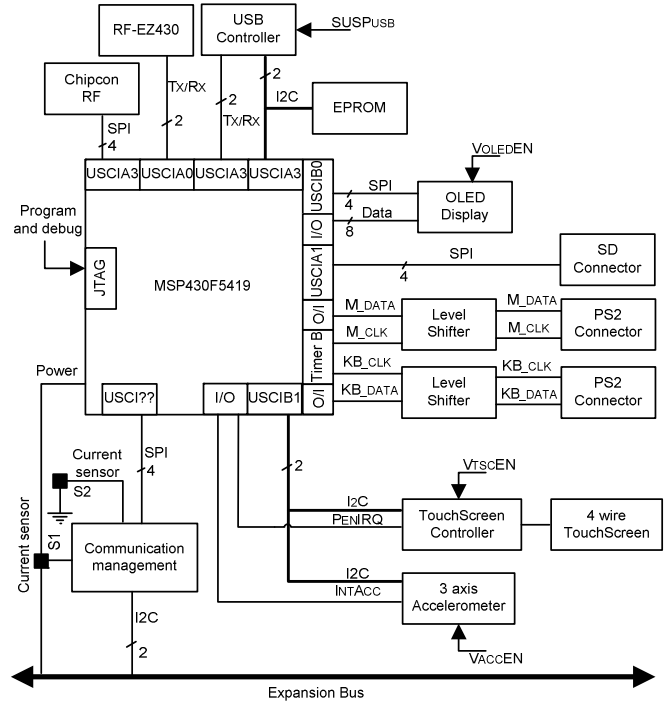


Fig. 6. Module 3 – Structure of the Communication interface module.

The communication port USB uses a dedicated controller that makes the interface with the UART (USCIA2). The USB controller firmware is saved in an EPROM that can communicate with the USB controller or with the microcontroller by an I2C bus (USCIB3). The USB can be suspended or reset by the microcontroller though the line *SUSPUSB*.

The SD socket is only a physical support to allow the connection of the card to the microcontroller through a SPI (USCIA1). Two additional lines enable detection and inhibition of writing operations.

The OLED display support two different interface methods: 8-bits parallel interface, or SPI (USCIO). The parallel interface requests a specific software driver. A simpler interface can be implemented with the SPI bus.

The radio frequency interfaces allow the connection of a RF module from the Chipcon, connected to the microcontroller by a SPI interface (USCIA3). The RF-EZ430 radio frequency module can be connected through an UART (USCIA0).

IV. COMMUNICATIONS MANAGEMENT HARDWARE

Each module has hardware dedicated to the management of the communications and supervision of module hardware working. Through its own functionalities, beyond allow the sharing of data between modules, it also allows to monitor in real time the voltage, the current, and the energy consumption by the module.

A. Hardware overview

The structure of the communication management hardware is illustrated in Fig. 7. The implementation of the protocol is described in Section V.

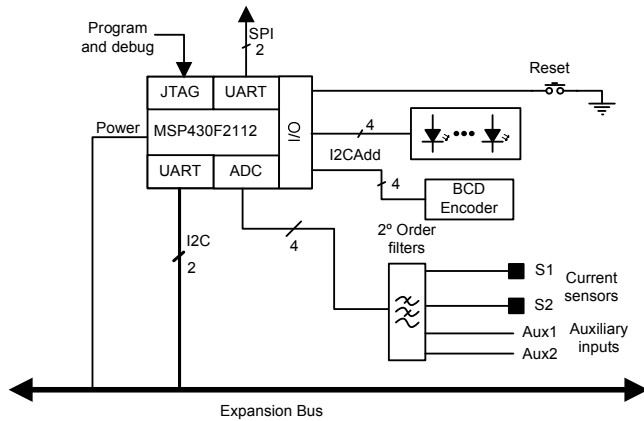


Fig. 7. Communication hardware interface.

The communication management hardware was built around the MSP430F2112. A BCD encoder is used to set the address of the module in the I2C Bus. Four LED are used as visual indicators of the communication status. The communication management hardware can be reset by the pressure switch *reset*. The connection between the user's microcontroller and the microcontroller used to manage the communications is made by a SPI channel. The communication management hardware is connected to the I2C bus through the UART2. Beyond the previously described hardware, it is also possible digitally to acquire four analog acquisition channels that are described next.

B. Energy consumption measurement

Current can be measured in two distinct points. At one measuring point (S_1) is possible to acquire the current of the user's microcontroller. The total current required by all module components can be observed at the second measuring point (S_2).

The current signals present high working frequencies. Therefore, the acquisition frequency must also be high. This problem was solved with a second order low-pass filter that provides, at the output, the mean value of the current.

Two other analog inputs are also available. The student can use them to acquire the working module voltage. With the current and voltage information, it is possible to compute the energy consumed.

The current of the user's microcontroller is acquired with the INA198 current shunt monitor. This device outputs an image of the sensed current. A precision resistance is used to sense the current required by the module.

V. COMMUNICATIONS MANAGEMENT SOFTWARE

Data exchange between modules is based upon two different communication methods: (1) the communication method between the module application function and the network function; and (2) the communication method supported by the bus, which has the main task of interconnect all modules, working as router among them. This characteristic gives to the system a high versatility, allowing to increase the set of applications supported.

The adopted solution was a serial bus, being the key criteria of this choice: the physical dimensions, the maximum length of the bus, the transmission data rate and, of course, the availability of the serial communications

interface.

The I2C communication protocol is oriented to connections of the type master/slave, i.e., the exchange of data will occur always through the master. This need led to the definition of two distinct functional units. Although, this technology supports multi-master operation, it was decided to use a single master, giving to the network a hierarchical structure with two levels.

The master role is performed by the module 0 – Basic interface and power. This choice is justified by the fact that this module has an obligatory presence in the network. The procedure of data exchange among slave modules is described in Fig. 8. If slave unit B wants to send a message to the slave unit A (path 3), first the message must be sent to the master (path 1) that will sent it to the destination unit (path 2).

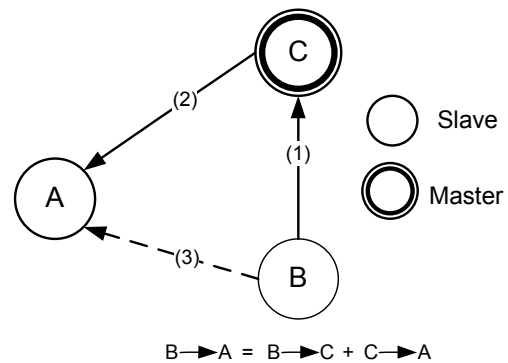


Fig. 8. Data exchange procedure among slave A and B.

To satisfy the specifications it was necessary to define a communication frame, as can be observed in Fig. 9.

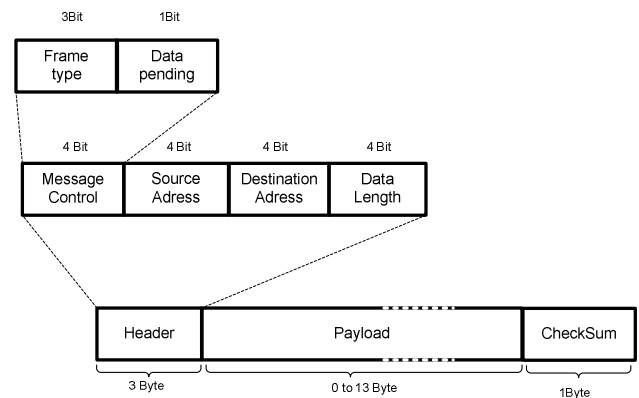


Fig. 9. General MAC frame format.

A. Protocol definition

In order to implement the communication service layer, a media access protocol was outlined with the mechanisms required to exchange information between modules. Therefore, the service frame and the communication protocols between the master unit and the slaves are defined as follows.

The service frame has three different fields. The header field, with two bytes length, has the information about the message type. The routing information includes sender and receiver addresses, and information about the length of the payload in bytes. The payload field is used to carry the

information from one nodule to another. Finally, the checksum field, with one-byte length, is used to control communications integrity.

The service layer protocol has three different types of frames. The data frame is used to transport the information between the applications running in the modules. The command frame establishes the service layer protocol. The acknowledge frame signalizes the data integrity and the end of the communication. The message type is specified in the sub-field frame type of the header field (see Table 1).

TABLE 1
VALUES OF THE FRAME TYPE SUBFIELD

Frame type value $b_2 b_1 b_0$	Description
000	Reserved
001	Data
010	Acknowledgment
011	MAC command
100-111	Reserved

The service layer uses a command set with three main goals: network management and setup; information transference tasks synchronization; and measurement. The command frame layout can be observed in Fig. 10. The command set is listed in Table 2.

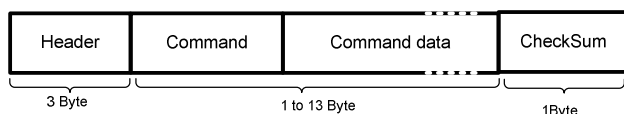


Fig. 10 – Service command frame

TABLE 2
MAC COMMAND FRAMES

Command frame identifier	Command Name	Direction
0x00	Heart_Beat Request	M ->S
0x01	Heart_Beat Response	S ->M
0x02	Get_ID Request	M ->S
0x03	Get_ID response	S ->M
0x04	Software Reset Request	M ->S
0x05	Get_Status Request	M ->S
0x06	Req_Data Request	M ->S
0x07	Req_Data Response	S ->M

VI. CONCLUSION

This paper presents the development of a platform to support the teaching of embedded systems. The modules here presented allow the implementation of different experimental laboratories, with increasing level of difficulty. At the same time, the student has access to key technologies related with the development of embedded systems.

The presence of an expansion bus gives a high versatility to the learning platform, since, it allows the development of new modules. This versatility is further enhanced by the existence of a communication bus that allows the sharing of data between modules.

REFERENCES

- [1] Diane T. Rover, *et al.*, “Reflections on Teaching and Learning in an Advanced Undergraduate Course in Embedded Systems,” in *IEEE Transactions on Education*, Vol. 51, No. 3, August 2008.
- [2] Kenneth G. Ricks, David Jeff Jackson, and William A. Stapleton, “An Embedded Systems Curriculum Based on the IEEE/ACM Model Curriculum,” in *IEEE Transactions on Education*, Vol. 51, No. 2, May 2008.
- [3] Hiroyasu Mitsui, Hidetoshi Kambe, and Hisao Koizumi, “Use of Student Experiments for Teaching Embedded Software Development Including HW/SW Co-Design,” in *IEEE Transactions on Education*, Vol. 52, No. 3, august 2009.
- [4] Saeid Nooshabadi, and Jim Garside, “Modernization of Teaching in Embedded Systems Design—An International Collaborative Project,” in *IEEE Transactions on Education*, Vol. 49, No. 2, May 2006.
- [5] Ken Ferens, Marcia Friesen, and Sandra Ingram, “Impact Assessment of a Microprocessor Animation on Student Learning and Motivation in Computer Engineering,” in *IEEE Transactions on Education*, Vol. 50, No. 2, May 2007.
- [6] Darko Hercog, *et al.*, “A DSP-Based Remote Control Laboratory,” in *IEEE Transactions on Industrial Electronics*, Vol. 54, No. 6, DECEMBER 2007.
- [7] Chien-Ying Chen, *et al.*, “EcoSpire: An Application Development Kit for an Ultra-Compact Wireless Sensing System,” in *IEEE Embedded Systems Letters*, Vol. 1, No. 3, October 2009.
- [8] Pedro Dinis, António Espírito-Santo, Bruno Ribeiro, Humberto Santo, “MSP430 Teaching ROM,” Texas Instruments, 2009.
- [9] MSP430™16-bit Ultra-Low Power MCUs, Texas Instruments, Available: <http://www.ti.com>.