CONTROL ACCESS SYSTEM USING RFID TECHNOLOGY

Wagner Regis de Araujo – wraraujo@gmail.com
Michael Taynnan A de O Barros – michael.taob@gmail.com
Daniella Dias C da Silva – danielladias@cefetpb.edu.br
Israel Cavalcante Portela –rael.portela@gmail.com
Jeosafá de Melo Freitas Júnior – jotafreitasjr@gmail.com

IFPB – Federal Institute of Education, Science and Technology of Paraiba

ABSTRACT

Due to increased urban violence, comes the need for an efficient access control. In this context, RFID technology has become a good solution. The main advantage in using this solution is that it is possible to get identification tag without a direct contact with the RFID reader. So, it is possible to get information about persons and vehicles while they are moving. In this paper, a control access system using RFID technology is described. The system was implemented in a microcontroller since this architecture is more appropriate for places as access gates and watches. It was also developed a data encoding in order to save space in storage memory. To ensure the correct functioning of system, it was developed a prototype using a RFID reader and tags.

1. INTRODUCTION

The RFID technology (Radio-Frequency IDentification) has it roots in the radar system in the Second World War. The Germans, Japanese, Americans and English used the radars to warn the people that the plane was very near. With the success of this technology, studies and advances in the radar and RF (Radio Frequency) communications area continued through the decades [1]. So, until nowadays many applications emerged, such as: health care [2], postal package tracking [3], aviation industry [4], defense and military [5], baggage and passenger tracing in airports [6,7], robotics [8] and control access [9].

A RFID system follows a simple architecture: a tag, a reader, a processing unit and an output system. For identification a tag is necessary. This tag can be of two types: a passive and an active one. The active tag requires an internal energy. Unlike the passive one, the energy is provided by the RFID reader.

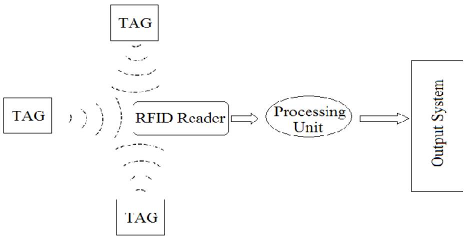

Figure 1- RFID System Diagram.

Basically the RFID has several methods of identification, but in this project and the most common one is to store a serial number that identifies a person or object [10]. After the captured data have been processed, it will be sent to an output system.

Depending on where the security system will be implanted, for example, in a watch, the use of a personal computer becomes inadequate. In this case, it is necessary another computational architecture for the project processing unit. In this context, the use of microcontroller becomes the most interesting solution and with a low cost. In this project, a RISC microcontroller PIC 16f877A from Microchip Technology Inc. [11,12] was used. This chip includes:

- A CPU (Central Processor Unit) to interpret the instruction program;
- A PROM memory (Programmable Read Only Memory) will record in a permanent way all the instruction program wrote in a high or a low language.
- A RAM memory (Random Access Memory) to record the variables utilized in the program;
- I/O lines (in/out) to control the external devices as sensors, keys, displays and others.
- Auxiliary devices like: clock generator, bus, counter, watchdog timer, UARTS, etc.

The PIC microcontrollers always have their use in: electronics consumption automation, robotics, instrumentation, embedded electronic and computer peripherals.

In this work, it was developed a control access system for students and teachers in an educational institute. Each user has one RFID tag that identifies only the user in the system. When the user approaches the watch, the place where a RFID reader is installed, he will be identified, facilitating the security job.

This paper is organized as follows. In Section 2 the developed system architecture is presented. In Section 3 results of this project are showed and the benefits of its use. Finally, Section 4 discusses about future implementations and the project extension.

2. DEVELOPED SYSTEM

The Figure 2 presents the developed system diagram.

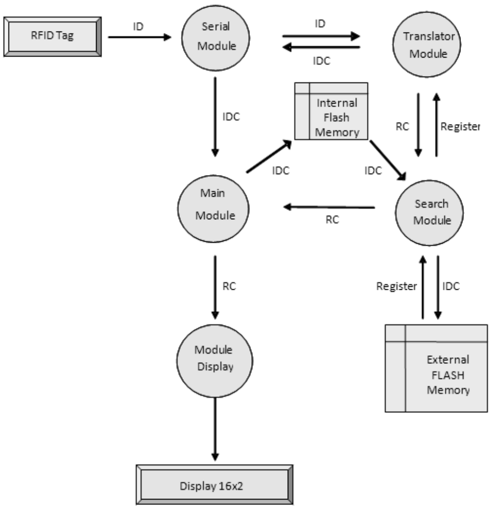

Figure 2- Developed System Diagram.

The reader sends an ID tag in four bytes which are in ASCII format. These digits are received by Serial Module (SD). SD receives these data and sends to the Translator Module (TM). TM returns the coded ID (IDC) in three bytes which are codificated in four nibbles. The IDC is sent to the Main Module (MM) which stores it in a FIFO (First In, First Out). The FIFO is stored in the PIC's internal memory and an interruption is started. When this interruption occurs the Search Module gets the first IDC in the FIFO. Based on IDC the user information is recovered from the external memory. Calculation from address is explained after that.

The user data are recorded in an external flash memory in a codified form. This codification intends to save space in memory and will be explained afterwards. The read data for the Search Module are passed to TM and it decodes the data to the ASCII format, forming a register (RC) that includes the user's name and function (student/teacher/server). The register is passed to Display Module (DM) which shows user's information in a LCD Display.

The operation of each module is presented in the next sections.

2.1. Serial Module

This module gets the tag identification through an internal RFID reader working on 125 KHz and RS232C is used as the data communication interface. This reading is performed with a physical connection of pin 3 from reader to microcontroller pin RC7. The reader sends serially the RFID tag identification using a baud rate of 9600 bauds. When the RFID tag is about the reader, it generates an interruption that triggers all the processing for this tag. The interruption is generated by CP pin (pin 1 from the reader) determining that an identification is ready for transmission through the serial interface. Pin 6 from reader is set in a high level state to use the ASCII characters. A library that makes the reading of data from reader is written using the C programming language

2.2. Translator Module

In encoding process the Translator Module intends to represent ASCII characters in a nibble with six bits. Once in decoding process each nibble is represented in ASCII characters.

To encode the characters:

1. Between 20H and 3FH the two least significant bits (MSB) are zero so they are discarded.
2. Between 40H and 5FH it is subtracted 40H and afterwards the two MSB are discarded.

To decode the characters:

1. Between 20H and 3FH a left shift two is performed and two MSB are made zero.
2. Between 0H and 1FH it is added 40H to nibble representation.

Figure 3 shows the coding and decoding process.

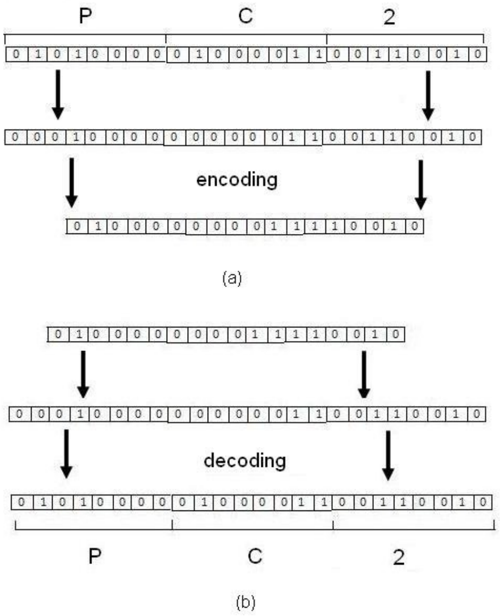

Figure 3 – (a) Encoding and (b) Decoding processes.

In the (a) part, the user’s information is represented in ASCII pure characters, which are the fist array of bits. Afterwards, the algorithm, explained in the beginning of this module, will be run with the characters, that is the second array. And finally, the two MSD bits are discarded forming the encoded register, which is the third array. After this process, the data will be recorded in the flash memory by the search module.

In the (b) part, an encoded register is rescued in the flash memory by the search module separating the array and completing the data with 0 bit in MSB. After that, the algorithm, showed in the beginning of this module, is run. So, the final data is pure ASCII characters, ready to be used in the display module.

2.3. Search Module

In this project, it was used an external flash memory since PIC internal memory size is not enough to store all users registers. The goal of this module is to perform the external memory search. I²C protocol [13] was used for serial communication with flash memory. To search the user's information an index search is made. The address is calculated based on the length of the register with the user's ID.

Each register has 33 ASCII characters, where 4 characters represent the ID, 28 characters the name and 1 character the codified information data (function). The register will be stored in the flash memory in 6 bits format (nibble). Calculating total of bits for 33 nibbles that will be stored in flash: 33 nibbles * 6 bits = 198bits that corresponds to 24.75 bytes. Hence, it was used 25 bytes to store the 33 nibbles (24bytes + 1nibble). Note that the last byte has 6 useful bits, so the two MSB (More Significant Bit) bits are defined as 0.

In storing it is used a union 3byte_4nibbles. This union receives 4 nibbles and converts it to 3 bytes format. The process was made in the fist nibble to 32th, 4 nibbles a time, totaling 192bits (24 bytes). The last nibble (33th) is treated separately, on account of the two 0 in MSB bits.

Byte format 25: [X X X X X X 0 0] (X= nibble.)

In reading a byte array is used to read information in flash memory.

After that, a union of 3bytes_4nibbles is used. This union gets 3 bytes a time and converts it to 4 nibbles format. The process was made in the first byte to the 24th, in 3 a group of 3 bytes, totaling 192 bits. The last byte (25th) is treated separately, on account of a needed right shift two (>>2) to a byte to nibble convertion.

3.4. Display Module

LCD (Liquid Crystal Display) was used to show user information. Basically the alphanumeric LCDs follow a default interface specification, which are 14 access pins (for LCD without back-light illumination) or 16 pins (for LCD with back-light illumination).

In this project it was used an LCD with 14 pins. Data bus consists of pins 7 to 14. This bus receives data and commands. Pins 4 and 5 signal if data bus information is command or data.

First step consists of configuring PIC ports that will be used. PortD (4 bits mode) was used for data and portE (E2,E1,E0) was used for control commands. Control pins correspond respectively to RS (Register Select), RW (Read/Write) and E (Enable).

After LCD configuration process a list of commands is performed for the LCD is really able to be used. In addition to this, the LCD is reset not to interfere with the data that will be displayed. So, Display Module is able to show the user's information.

3. RESULTS

The whole system has been simulated and the results were grateful. Coding and decoding process demonstrate a real saving space in memory.

In simulation process ID tag input was replaced by random data which was transmitted by serial interface.

To prove the correct functioning of the system it was developed a prototype using an internal RFID reader and MicroGenios PIC KIT 16f Family v3.0 [14] (Figure 4). Grateful results were obtained.

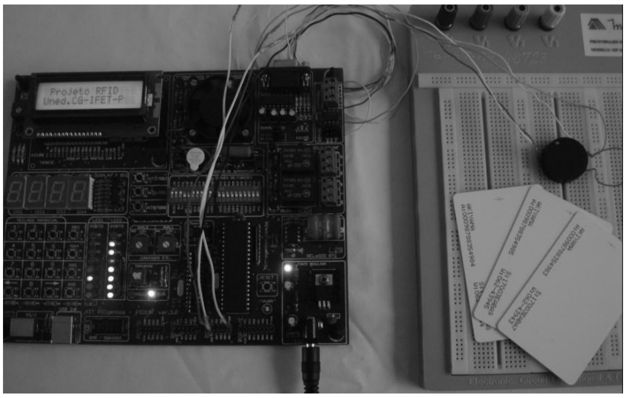

Figure 4 - System Prototype.

For implementation of the whole system, 77% of the PIC's Program Memory was used. PI02-(R2/W)/P RFID reader [15] was the choice for the prototype. It works in 125MHz frequency, and it reads tags up to 7 cm of distance. The reader sends the data in two ways: codified with the Wiegand26 protocol or in a serial mode. In this project, serial mode was used. When reader sends data in a serial mode, pin 6 must be connected to VCC and the data output is transmitted through pin 3. It is also necessary to connect pin 1 of reader to PIC's pin responsible by interruption. So, when a new tag is read by RFID reader, an interruption (through pin 1) is started and recognized by PIC which begins all process relating to this tag.

4. FINAL CONSIDERATIONS

After prototype development other improvements can be done. The next step is the system implantation in the Federal Institute. For this a RFID reader with a greater range will be necessary. Gate control through the system is also a desired improvement.

Studies have been made about the interference of the RFID reader in the microcontroller PIC. This interference is due to the magnetic field sent by the reader which can generate a lot of noise in the PIC's circuit. Possible solution for this problem is a bigger distance between PIC and RFID reader. Another characteristic is when a vehicle enters the reader's magnetic field all the data sent by the reader returns to it. It is because the metallic body of the vehicle returns all the signals, acting as a mirror. Solutions must be studied to solve these problems.

5. REFERENCES

[1] “The History of RFID Technology”.RFID Journal 2009

[2]C. Swedberg. “Hospital Uses RFID for Surgical Patients”. RFID Journal, 2005.

[3]P. Harrop. “RFID in the Postal Service. MoreRFID”, 2005.

[4]J. Collins. “Boeing Outlines Tagging Timetable”. RFID Journal, 2005.

[5]J. Collins. “DOD Tries Tags That Phone Home”. RFID Journal, 2006.

[6]R. B. Fergunson. “Logan Airport to Demonstrate Baggage, Passenger RFID Tracking”. eWeek, May 2006.

[7]Meingast, M., King, J., Mulligan, D. K. “Embedded RFID and Everyday Things: A Case Study of the Security and Privacy Risks of the U.S. e-Passport.” IEEE International Conference on RFID, 2007.

[8]ZHOU, Y., LIU, W. “Preliminary Research on Indoor Mobile Robot Localization using Laser-activated RFID”. IEEE International Conference on RFID, 2007.

[9]Blythe, P. “RFID for road tolling, road-use prcing and vehicle access control”.IEEE RFID Technology, 1999.

[10]Derakhshan, R., Orlowska, M. E., Li, X. “RFID Data Management: Challenges and Opportunities”. IEEE International Conference on RFID, 2007.

[11] Microchip 2009 http://www.microchip.com/

[12]Bates, M. Interfacing PIC Microcontrollers. Newnes, 2006.

[13]De Sousa, D. J. e Lavinia, N. C. PIC 16F877A Conectando o PIC. Editora Érica, 2003.

[14]MicroGenios 2009, http://www.microgenios.com.br/

[15]Akiyama Tecnologia 2009, http://www.akiyama.com.br/