3rd International Conference on Automation, Control, Engineering and Computer Science (ACECS'16)
Proceedings of Engineering & Technology (PET)
pp. 180-184

AES implementation on a low cost embedded system

S. BOUZAIANE, N. SAHLI

Unité de Recherche: Sécurité des Systèmes sensibles
Académie Navale, Route kankala, Menzel Bourguiba, 7050 Bizerte - Tunisie
samibouzaiane@yahoo.fr

***Abstract*— The objective of this work is the demonstration of the implementation of the commonly used AES encryption and decryption algorithm on a low cost embedded system based on PIC18F14K22 microcontroller. This device belongs to the PIC18 family and it has an excellent performance cost ratio. The simulations are performed on the PROTEUS environment and there are considered as the final step before hardware conception. The software was developed on MikroBasic pro for PIC with additional crypto library. The results obtained shows an interesting hardware and software solution suited for up to 32 kbps low cost embedded system applications like the wired or wireless low bandwidth analog or digital sensors networks, the active tags in RFID system or the compressed digital audio transmission.**

***Keywords*— AES encryption algorithm, Implementation, Microcontroller, Real time simulations, Low-cost embedded system.**

I. INTRODUCTION

Electronic devices surround us in almost all aspects of our everyday lives. Everything from entertainment to mobile phones to medical equipment etc. are electronic devices. In all this systems, a microcontroller is incorporated in order to implement some sort of functionality. A microcontroller could perform any task, from turning on and off the device when a button is pressed to more complex tasks like data processing, interfacing, communications etc.

Communication between electronic systems is largely used in many fields today. In several applications, it is necessary that this communication is carried out in a secure manner, in other words: The data transmitted should be inaccessible to anyone else than the data is intended for. An evident solution for this problem is the use of cryptographic algorithms. Numerous models of cryptographic algorithms have been developed for this purpose, and currently, The Advanced Encryption Standard, AES, become one of the most widely used.

In many applications cases such as wired or wireless, analog or digital sensors networks, active tags in RFID system or compressed digital audio transmission, we need a low-cost technical solution providing secure connections between sensors, digital data streams and transducers. This is what we will try to propose in this work.

The transmission and reception modules will be mainly composed by judiciously selected microcontrollers that will support all the tasks and functions necessary for this kind of application. The transmission module will support analog to digital conversion, data formatting, encryption and transmission through a wired or wireless interface. The receiving module will support the data acquisition via the interface, formatting data, decryption and digital to analog conversion.

Several designs have been proposed in the literature related to the implementation on specific microcontrollers. Patil et al proposed an implementation of AES algorithm on ARM processor for wireless network [1]. Taki El_Deen et al proposed an implementation of AES Algorithm in microcontroller using PIC18F452 [2]. The Zilog Company proposed in an application note AES 128-Bit Implementation with Z8 Encore! XP Microcontrollers [3]. Lee et al proposed an AES implementation and performance evaluation on 8-bit Microcontrollers [4].

Several works have been also proposed in the literature related to the optimization of the AES algorithm for the embedded systems like Yue et al on a low-cost round encryption method [5] and Babu et al on the optimization of memory for AES Rijndeal algorithm implementation [6].

The structure of the paper is organized as follows: Section 2 gives a brief presentation of AES. Section 3 present a description of the hardware system. Section 4 present a description of the related software. Section 5 details the simulations results and analysis. Section 6 concludes the paper.

II. BRIEF PRESENTATION OF AES

The Advanced Encryption Standard or AES (also named Rijndael) is a symmetric block cipher used initially by the United States to protect sensitive information and is implemented in software and hardware around the world to encrypt classified data. The origins of AES date back to 1997 when the National Institute of Standards and Technology NIST [7] announced the need of a successor to the old Data Encryption Standard (DES). This new encryption algorithm had to be robust against various attack techniques. It was to be easy to implement in hardware and software, as well as in restricted environments (smart cards, resource-constrained embedded systems etc.).

AES comprises three block ciphers, AES-128, AES-192 and AES-256. Each cipher encrypts and decrypts data in blocks of 128 bits using cryptographic keys of 128-bits, 192-bits and 256-bits, respectively. Symmetric or secret-key ciphers use the same key for encrypting and decrypting, so both the sender and the receiver must know and use the same secret key.

There are 10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys. A round consists of

ISSN: 2356-5608

several processing steps that include substitution, transposition and mixing of the input plaintext and transform it into the final output of cipher text. As a cipher, AES has proven its reliability and robustness against attacks.

Figure 1 shows a flow chart of the AES-128 encryption and decryption algorithm [1]. The encryption stages begins at the top left “128 Bit Plain Text Block” and finishes at the bottom left “128 Bit Cypher Text Block”. The decryption stages begins at the bottom right “128 Bit Cypher Text Block” and finishes at the top right “128 Bit Plain Text Block”.

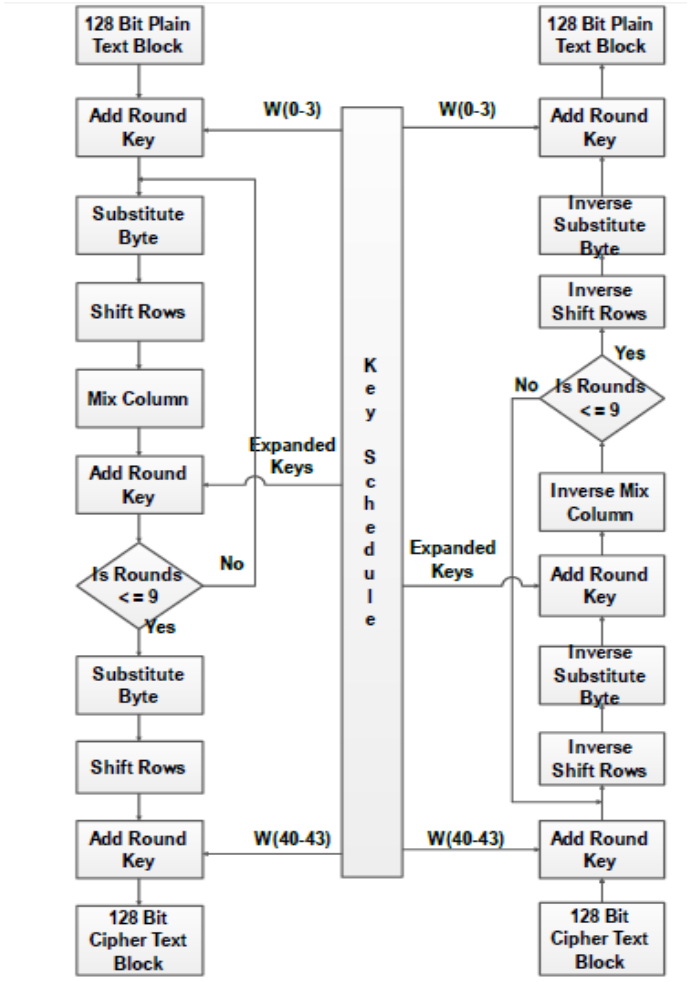

Fig. 1 Flow chart of the AES-128 encryption and decryption algorithm

The encryption phase begins with an initial round during which the first round key is XORed to the plain text (the “Add Round Key” step), nine equally-structured rounds follow. Each round consists of “Substitute Byte”, “Shift Rows”, “Mix Column” and “Add Round Key”. The last step consists of “Substitute Byte”, “Shift Rows” and “Add Round Key” and we obtain the “128 Bit Cypher Text Block”. The decryption phase consists of the same operations in a reverse order.

III. HARDWARE DESCRIPTION

Fig. 2 shows a general block diagram of an encryption and decryption hardware essentially based on microcontroller embedded systems.

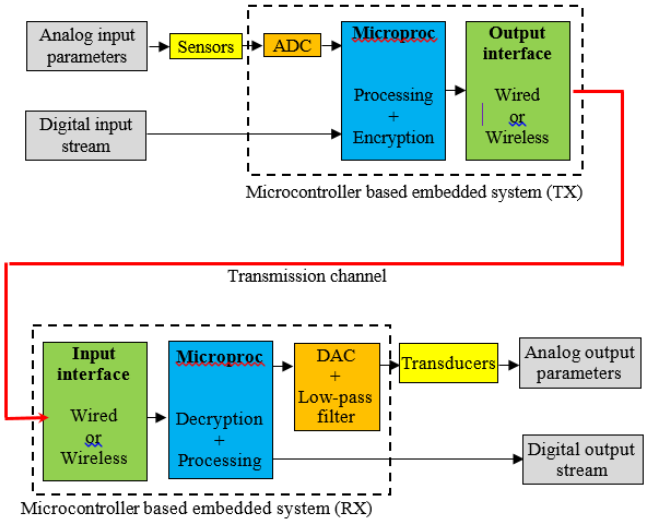

Fig. 2 Block diagram of encryption and decryption hardware

The physical analog input parameters like temperature, lighting, humidity, sound etc. are converted via the sensors to electrical signals. These analog signals are applied to the microcontroller integrated analog to digital converter. Moreover, the digital input stream is directly applied to its digital input. After processing and encryption, the digital output stream is transmitted via wired (UART, LAN, ONE-WIRE…) or wireless (RF, BLUETOOTH, WIFI…) output interface.

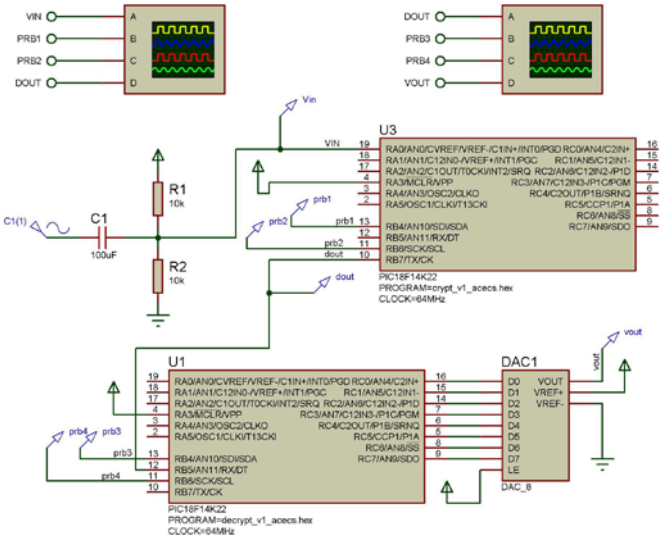

Fig. 3 Simulation diagram of encryption and decryption hardware

The digital crypt stream is received by the input interface. After decryption and processing, the analog data channels are applied to the digital to analog converters, the corresponding

low-pass filters and finally the transducers. The digital channels are directly applied to the output ports.

Fig. 3 shows the simulation diagram of the encryption and the decryption hardware carried on the PROTEUS software.

The encryption stage is formed by the PIC18F14K22 (U1) micro [8]. To demonstrate the real time performances of the used hardware and the associated developed software, a conventional 100 Hz sine wave is applied to the ANO analog input of U1 through a 2.5V bias circuit. The analog to digital converter of U1 is configured for 8-bit resolution and a sampling time of 0.5 ms. The 128-bit encryption key is stored on the internal data eeprom. The output of the encryption stage is the RB7/TX pin and it is configured as asynchronous digital transmitter at 115200 bps. RB4 for prb1 and RB6 for prb2 are configured as digital outputs and used as digital software probes for sequences timing measurements.

The connection between pin 10 of U1 (TX) and pin 12 of U2 (RX) represents the digital transmission channel between encryption and decryption modules. Depending on the application, this channel can be wired or wireless via specific hardware modules.

The decryption stage is formed by the PIC18F14K22 (U2) micro. The RB5/RX pin is configured as asynchronous digital receiver at 115200 bps. The same 128-bit encryption key is stored on the internal data eeprom of U2. The PORTC is configured to 8-bit digital output and is connected to the digital to analog converter DAC1. For synchronization requirements, an internal timer is configured at the same sampling time of the encryption stage.

IV. Software Description

The integrated development environment IDE used is MIKROBASIC PRO FOR PIC. A crypto library [9] is added to the IDE. It contains a collection of useful cryptographic algorithms. With the PIC18F14K22 at 64 Mhz, the duration of the AES128 crypt and decrypt routines is about 2.28 ms.

Fig. 4 shows the Flowchart of the AES encryption software.

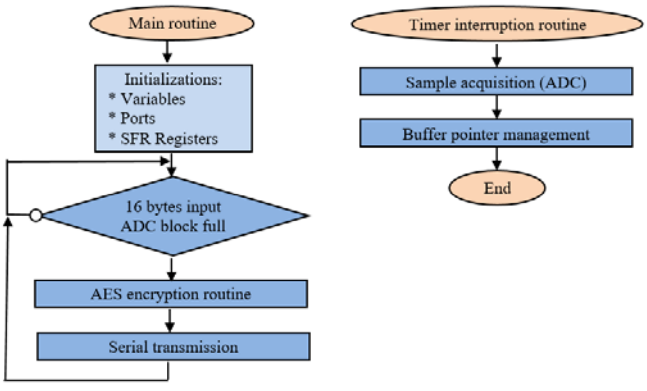

Fig. 4 Flowchart of the AES encryption software

The main routine starts with the initializations of the variables, ports and the special function registers. Then, the processor waits until the 16 bytes ADC input buffer is full. In this case, the AES encryption is called before the crypt 16 bytes are transmitted by the serial port.

The timer interruption routine set to 0.5 ms as mentioned above starts with a byte acquisition via the analog to digital converter then the storage in a 16 bytes block buffer before pointer management.

Fig. 5 shows the Flowchart of the AES decryption software.

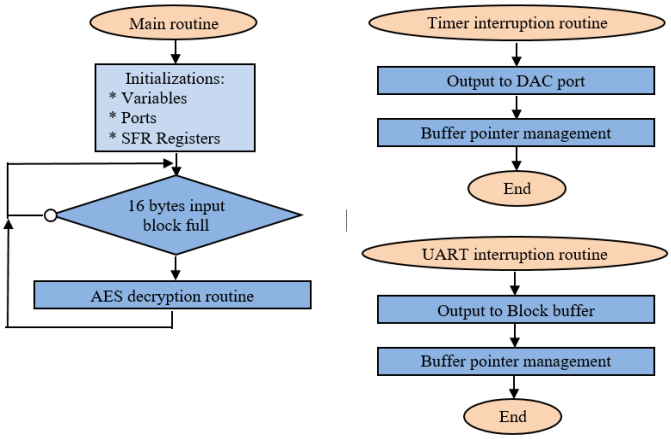

Fig. 5 Flowchart of the AES decryption software

The main routine starts with the initializations of the variables, ports and the special function registers. Then, the processor waits until the 16 bytes serial port input buffer is full. In this case, the AES decryption routine is called.

The timer interruption routine set to 0.5 ms also as mentioned above apply a pointed byte of the output buffer to PORTC before pointer management.

The UART interruption routine copy each received byte to the block buffer before pointer management.

V. Simulations Results and Analysis

Fig. 6 shows the simulations results of the AES encryption stage.

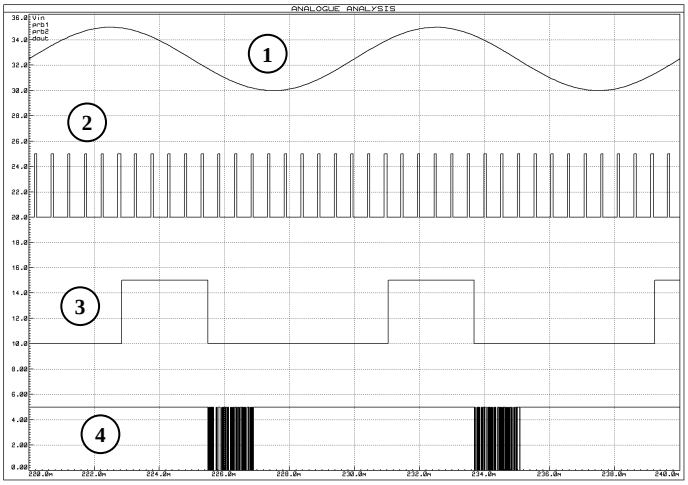

Fig. 6 Simulation results of the AES encryption stage

Signals 1, 2 and 3 are vertically shifted for clarity purposes.

Signal 1 is the conventional 100 Hz sine input. Signal 2 is issued from prb1 and shows the timer interruption cycles corresponding to sampling time. Signal 3 is issued from prb2 and shows when it is high the AES encryption routine activity. Signal 4 is the digital output from the serial port and shows the transmission phases.

The total duration time of 128-bit block processing is given by:

$$T_{tot} = (16 * Te) + Ta + Ts \quad (1)$$

Where Te is the duration of the timer interrupt, Ta the 128-bit block encryption time and Ts the duration of serial transmission. The simulations gives T_{tot}=4ms.

The performances in real time are determined by the data rate given by

$$D_r = \frac{N_{bits}}{T_{tot}} \quad (2)$$

Where N_{bits} is the number of bits of the treated block. We obtain here D_r=32 kbps.

Fig. 7 shows the simulations results of the AES decryption stage.

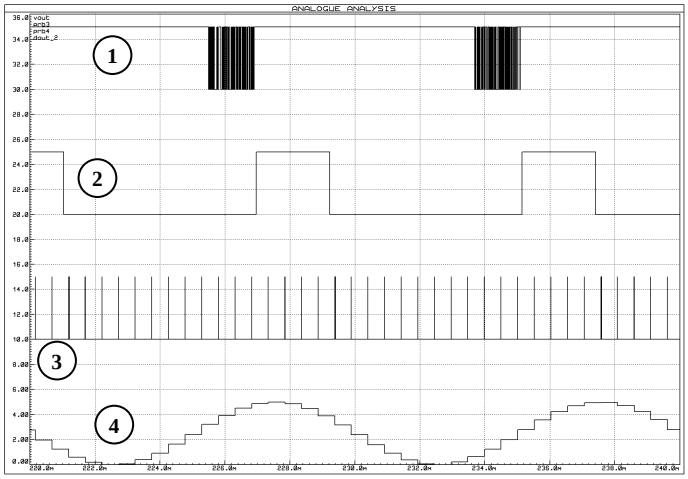

Fig. 7 Oscillogram of the AES decryption stage

Signal 1 is the digital input from transmission channel to the serial port and shows the reception phases. Signal 2 is issued from prb3 and shows when it is high the AES decryption routine activity. Signal 3 is issued from prb4 and shows the timer interruption cycles corresponding to output buffer to PORTC. Signal 4 is the output of the digital to analog converter and we obtain the discretized sine input signal. Evidently, a low pass filter (not presented here) is required in order to retrieve the same input sine signal.

Obviously, it's impossible to obtain identical input output signals if we have different encryption and decryption keys or if there are problems in synchronization or in buffers managements.

VI. Conclusions

This work has allowed us to validate, by using high performances simulation software, the complete hardware and software designing of a low cost microcontroller-based AES encryption and decryption embedded system. For higher data rate applications, it is recommended to use a more powerful microcontroller working at higher clock frequencies.

References

[1] V. B. Patil, U. L. Bombale, P. H. Dixit, "Implementation of AES algorithm on ARM processor for wireless network," *International Journal of Advanced Research in Computer and Communication Engineering.*, vol. 2, issue 8, pp. 3204–3209, Aug. 2013.

[2] A. E. Taki El_Deen, A. M. Fanni, "Implementation of AES Algorithm in MicroController Using PIC18F452," *IOSR Journal of Computer Engineering.*, vol. 15, pp. 35–38, Dec. 2013.

[3] Zilog. (2012) AES 128-Bit Implementation with Z8 Encore! XP Microcontrollers. [Online]. Available: http://www.zilog.com/docs/appnotes/AN0338.pdf

[4] H. Lee, K. Lee, Y. Shin, "AES Implementation and Performance Evaluation on 8-bit Microcontrollers," *International Journal of Computer Science and Information Security.*, vol. 6, No. 1, 2009.

[5] Q. Yue, L. Xinqiang, W. Yadong, "Low-Cost Round Encryption Method for Embedded System," *International Journal of Security and Its Applications.*, vol. 9, No. 4, pp. 117–124, 2015.

[6] M. R. Babu, A. R. Reddy, "Optimization Of Memory For AES Rijndeal Algorithm Implementation On Embedded System," *International Journal of Engineering Research & Technology (IJERT).*, vol. 1, Issue. 7, Sept 2012.

[7] National Institute of Standards and Technology. (2001) Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197. [Online]. Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[8] Microchip. (2011) PIC18(L)F1XK22 Data Sheet, 20-Pin Flash Microcontrollers with nanoWatt XLP Technology. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/41365E.pdf

[9] A. Vukelic. (2014) Crypto Library. [Online]. Available: http://www.libstock.com/projects/view/896/crypto-library