Embedded Security For High Risk Areas With Multiple Fault Tolerant

P. Moorthy[1], R. Gauthami[2]
[1]Associate Professor, [2]PG Student
Vivekananda College of Engineering for Women
Tiruchengode, Namakkal Dt, Tamil nadu, India

Abstract- Embedded systems have been almost invisibly pervading our daily lives for several decades. These Embedded electronic devices plays vital role in almost the areas of human life, Decreased costing, increased production tends to use these devices with higher reliability areas. They facilitate smooth operation in sphere of human life, avionics, automotive electronics, or telecommunication. New problems arise by increasing employment, interconnections, EMI, RFI, harmonics, sag, and end of course thermal problems. These problems are not quick common but expected always due to the fast growing multi utilize service like induction machine, radio frequency devices, rotating machines and much more. The challenges unique to embedded systems require new approaches to security covering all aspects of embedded system design from architecture to implementation. This paper presents a design of fault – tolerant embedded system by piggyback method of two embedded processor. Both processors are involved in adaptive sharing with software module. The customized fault tolerant embedded system is designed by combining selective protection on both hardware and software. Security processing, which refers to the computations that must be performed in a system for the purpose of security, can easily overwhelm the computational capabilities of processors in both low- and high-end embedded systems.

Index Terms—Embedded system security, processing monitor, security enforcement.

1 INTRODUCTION

EMBEDDED systems are widely deployed and used in application domains ranging from cellular phones to smart cards, sensors, network infrastructure components, and a variety of control systems. Two key characteristics make these systems particularly vulnerable to attacks. First, the embedded nature of the processing system limits the complexity of the device in terms of processing capabilities and power resources. It also exposes the device to a operate. In dealing with security, the embedded systems can be self-sufficient and be able to deal with cut electrical and communication systems.

number of potential physical attacks. Second, as a direct result of the limited processing capabilities, embedded systems are limited in their capabilities to run software to identify and mitigate attacks. Unlike workstation computers that can afford to run virus scanners and intrusion detection software, embedded systems typically only run the target application. Thus, embedded systems are inherently more vulnerable to attacks than conventional systems. Attacks on embedded systems can be motivated by several different goals.

1. Extraction of secret information (e.g., reading of cryptographic key material from a smart card);

2. Modification of stored or sensed data (e.g., tampering with utility meter readings);

3. Denial of service attack (e.g., reducing the functionality of a sensor network);

4. Hijacking of hardware platform (e.g., reprogramming of TV set-top box).In each of these cases, the attack relies on the ability to get access to the embedded system and change its behaviour (i.e., change in instruction memory) or its data (i.e., change in data memory). In most attack scenarios, a modification of behaviour is necessary even when modification of or access to data is the ultimate goal of the attack. Therefore, we focus on the security of processing in this paper.

1.2 ATTACKS ON EMBEDDED DEVICES:

Attacks on embedded systems can be motivated by several different goals. The following list illustrates this point (but is not meant as a complete enumeration of all possible scenarios):

1. Extraction of secret information (e.g., reading of cryptographic key material from a smart card);
2. Modification of stored or sensed data (e.g., tampering with utility meter readings);
3. Denial of service attack (e.g., reducing the functionality of a sensor network);
4. Hijacking of hardware platform (e.g., reprogramming of TV set-top box).

Embedded systems are especially suited for use in transportation, fire safety, safety and security, medical applications and life critical systems as these systems can be isolated from hacking and thus be more reliable.

For fire safety, the systems can be designed to have greater ability to handle higher temperatures and continue to

In addition to commonly described embedded systems based on small computers, a new class of miniature wireless devices called motes are quickly gaining popularity as the field of wireless sensor networking rises. Wireless sensor networking, WSN, makes use of

miniaturization made possible by advanced IC design to couple full wireless subsystems to sophisticated sensors, enabling people and companies to measure a myriad of things in the physical world and act on this information through IT monitoring and control systems. These motes are completely self-contained, and will typically run off a battery source for many years before the batteries need to be changed or charged.

1.3SECURE EMBEDDED:

Embedded system finds varieties of application in almost all the sphere human life in many application areas starting from cooking utilities to high end technology like space craft's. Generally embedded system are very riskless devices because of its own unique features but often beyond the feature some unknown attacks may happened due to various reason like EMI,RFI,harmonics,sag,and swell of course thermal problems. These problems are not quit common, but expected always due to the fast growing multi utility service like induction machine radio frequency devices rotating machines and much more. When embedded system is utilized for simple work can be reset cd, during the attack but this is not possible in all the cases like places where the human cannot go and rectify. Attacks are of various kinds and all the attacks doesn"t destroy the whole CPU, but disturb flow of activities which in turn commits a malfunction. A malfunction can be a subroutine; there by abnormal proceeding will happen in the execution in sequential lines. In this situation for a seamless electronic device which is equal kind of embedded and can perform much more than original device is required to overcome the above said.

The device utilize a seamless operation will have execution program of its own as well as a program for redundancy operation. Similarly the main device can also have a program of second. In this case both are in networking and can perform in dually as well as mutually on requirement basis. The embedded processor reports on the progress of application processing by sending a stream of information to the monitoring system. Proposed method of seamless operation of desired output of any real-time application can be done using to different processor like RISC architecture based PIC Microcontroller.

1.4 NEED FOR HARDWARE SUPPORT:

The design of secure hardware is often overlooked in the product development lifecycle, leaving many devices vulnerable to hacker attacks resulting in theft of service, loss of revenue, or a damaged reputation. Many times, products must be redesigned after a harmful incident, which raises overall development costs and increases time-to-market. This paper focuses on general concepts for secure hardware design coupled with practical examples. Topics in this paper include recommendations on incorporating security into the product development cycle, attack and threat models, and design solutions for enclosure, circuit board, and firmware layers.

As designers, the best we can do is understand the potential attacks against our system, design methods to prevent such attacks, with the understanding that nothing is ever 100% secure. "Secure" can simply be defined as when the time and money required to break the product is greater than the benefits to be derived from the effort. Given enough determination, time, and resources, an attacker can break any system. Security is a process, not a product. Security must be designed into the product during the conceptual design phase and must be considered for every portion of the design. It must be continually monitored and updated in order to have the maximum effect against attacks. Security cannot be simply added to a product and forgotten about, assuming that the product will forever remain secure.

The embedded processing system should be monitored to verify the required original performance. Because, the embedded device should perform continuously in some cases without interruption. If it not happens the whole embedded system will get collapsed due to any of the attacks.

In each of these cases, the attack relies on the ability to get access to the embedded system and change its behaviour (i.e., change in instruction memory) or its data (i.e., change in data memory). In most attack scenarios, a modification of behaviour is necessary even when modification of or access to data is the ultimate goal of the attack. Therefore, we focus on the security of processing in this paper.

2. RELATED WORK

The term "embedded system" covers a broad range of possible system designs, processor architectures, and performance and functionality characteristics. In our work, we focus on embedded systems that can be broadly characterized as middle to lower end in the performance spectrum. Their main characteristics are :
(1) Medium to low-performance embedded processor core (e.g., single RISC processor);
2) Targeted use for one or only a handful of applications;
3) typically used in a networked setting.

Examples for practical embedded systems that fit these characteristics are : cell phones, networked sensors, smart cards (typically not networked though), low-end network routers (e.g., home/small office gateway), networked printers, etc. Attacks on embedded system can have a wide range of approaches.

Ravi et al. describe mechanisms to achieve physical security by employing tamper-resistant designs [1]. Wood and Stankovic consider a networked scenario where systems are exposed to additional remote attacks [2]. Embedded systems are also susceptible to side-channel attacks (e.g., differential power analysis [3]). Solutions to this problem have been proposed [4], and we do not consider this aspect in our work. In terms of developing a general, hardware-based architecture to protect embedded systems against a range of attacks.

Gogniat et al.have proposed one such in [5]. This work does not give details on what the proposed monitors would look like. Our work can be seen as one example of how to monitor processing to ensure secure execution of applications. In the context of monitoring processing on embedded systems,the system by Arora et al. [6] and the IMPRES system [7] are conceptually similar to our work. The main difference is that their finest granularity of monitoring is the basic block level due to the use of per-block hash values (in [6]) or per-block encrypted checksum (in [7]), and deviations in the program execution are detected when the hash value or checksum does not match at the end of a basic block.

In our work, deviations from the binary can be determined within a single (or a few) instructions. In addition, Arora et al. use control flow information to track program execution. As we discuss in Section 4, our proposed hash-based monitoring performs significantly better (i.e., faster detection) than control-flow-based monitoring. The SAFE-OPS system by Zambreno et al. [8] uses information that is collected across multiple executed instructions to determine valid operation. This system can detect errors and attacks at the end of such a sequence, whereas our system may immediately detect the first instruction that deviates. Abadi et al. [9] also use a control flow graph for monitoring program execution. Nakka et al. [10] introduce integrity checks into the micro architecture and use special check instructions. The main

difference to our work is that these approaches require changes in the machine code to implement the necessary checks, while, in our work, binaries do not need to be modified. We also believe it is important to separate the processor from the monitor by using separate system resources to reduce vulnerability. Suhet al. use the concept of "information flow" to track if data is considered authentic or spurious (i.e., potentially malicious) [12].This system requires a much more complex design that needs to be integrated with the processor.

A completely different approach to ensuring secure execution of programs is to identify non instruction memory pages with an NX (No eXecute) or XD (eXecute Disable) bit. The idea is to avoid a change of control flow to a piece of code that belongs to data memory. This mechanism is useful to avoid, for example, buffer overflow attacks. It does not consider a scenario where an attacker overwrites instruction memory. Another approach to defending against buffer overflow attacks is described by Shao et al. in [13], where bound checks are used and function pointers are protected by XORing them with a secret key.Anomaly and intrusion detection by comparing behaviour against a model is also used in other domains (e.g., mobile ad hoc networks [14]). In our case, we have a simpler problem since our model is derived from the actual binary of the application. Thus, there is no guesswork on how accurate the model is—it is exactly the same as the application.

3. BLOCK DIAGRAM:

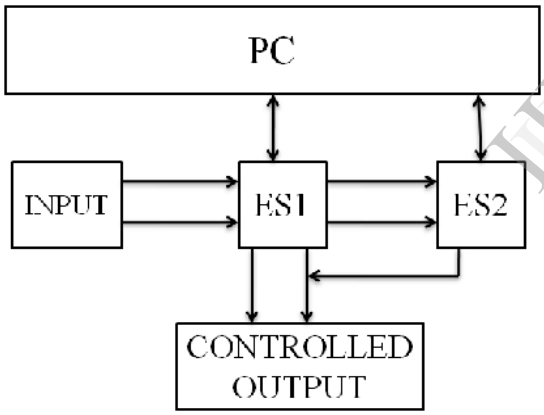


Fig3:Block Diagram

BLOK DIAGRAM EXPLANATION:

The proposed block diagram can be used for medical application, military application and where sophisticated ensured critical outputs. According to the proposed block input coming from the sensor to both of the Embedded controller and performance simulation to obtain consisting result at any causes like attacks of various kinds.

In processor 1 and processor 2 are inter connected and coupled with computer through RS 232 to see the output on the screen. We can execute any sot of application can be developed as a coding using VB front end and back end can be data base. To have a particular application we would like to represent medical application this time which can be will evaluated and presentable on the computer screen. Biomedical application is very numeral in nature and we would like to opt pacemaker as real time application. Simulate a pacemaker using our project.

3.1 OBJECTIVE:

Proposed method of seamless operation of desired output of any real-time application can be done using to different processor like RISC architecture based PIC Microcontroller and advanced micro controller like ARM 7.0.The PIC microcontroller has its own unique features and ARM7.0 has its own muti application hardware will jointly executes a similar program on both and executes different program individually. The following hardware feature is the specialty of microchip based PIC microcontroller.PIC means peripheral interface controller which can perform various operation at nominal speed with higher reliability and good repeatability. The only concentration we need to provide is two different power sources to be offered for analog and digital operations.

3.2 CIRCUIT OPERATION:

From the circuit it can be seen that the reference analog supply after being regulated by the 9v regulator enters the zener diode through the resistance R4 where it is again regulated to 5v since the zener diode used here has a cut off of 5v. Thus we have a double regulated completely filtered analog reference source. R6 is a potential divider used for setting the dynamic response range of the reference supply. This means that the reference 5v can be used as it is or it can be made into a fraction of the 5v for example 1v so that readings in this range can be read with more precision. This is because the ADC has 10 bit resolution which can be totally used for representing the 1v rather than 5v.

The pins 2-5, 7-10, 35 and 36 are used as the 10 channels of the ADC. To these pins the analog inputs to be processed by the ADC are given. Y1 is the crystal oscillator used. It is of 10 MHz and gives a baud rate of 9600 bits/s. The capacitors C2 and C3 are used as decoupling capacitors to remove the high frequency noise signals.

The capacitor C1 is in the off condition when power is switched off. When the power is switched on or reset then this capacitor gets charged through the resistor R2 and then through R1 this appears at the MCLR pin of the PIC. This is the memory clear pin and thus the memory is cleared and is ready for use as soon as power is switched on. S1 is the synchronous switch which is also used for the same operation and for PC and PIC synchronous operation.

The advantage of this architecture is to continuously monitor the operation of the system to detect abnormal behaviourand to use reconfigurable hardware to provide various levels of protection and performance. we propose to use for monitoring is the hashed pattern.

In this case, several pieces of information (in our case, instruction address and instruction word) can be compacted to a smaller hash value. This is particularly useful since opcode, operands, etc., can consume a lot of memory space. This pattern can be used with different lengths of hash functions. The key idea is to use a monitoring subsystem that operates in parallel with the embedded processor. The monitor verifies that two processing steps are performed that match up with the originally installed application.Any attack would disturb the pattern of execution steps, and thus,alert the monitor.

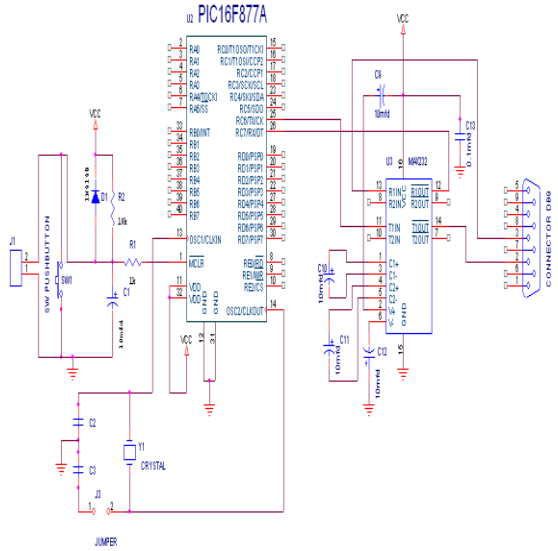
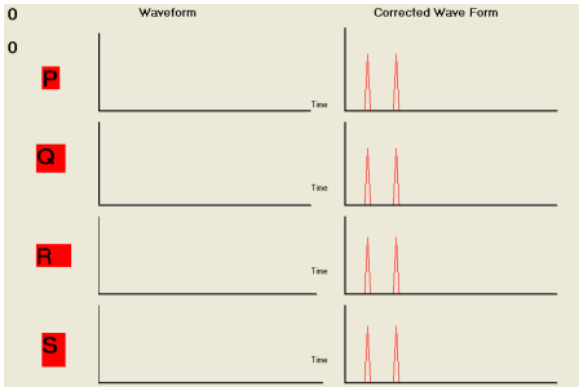
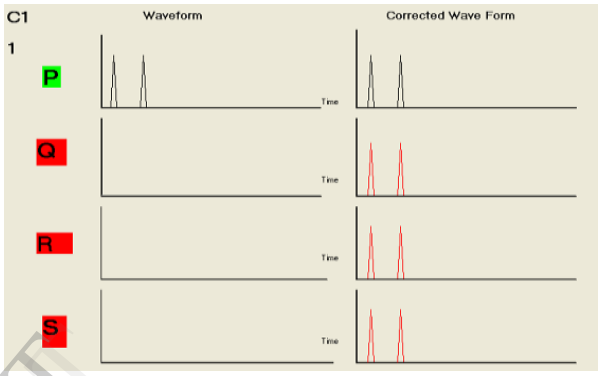


Fig. 2: PIC circuit operation

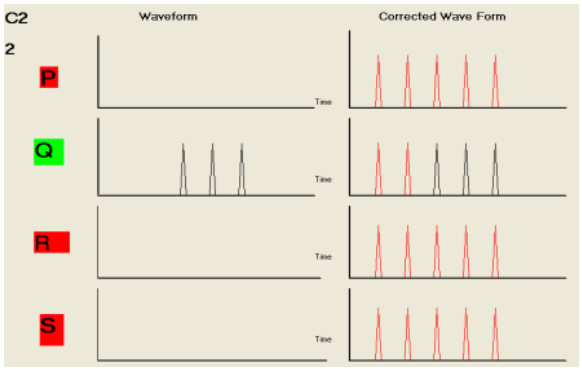


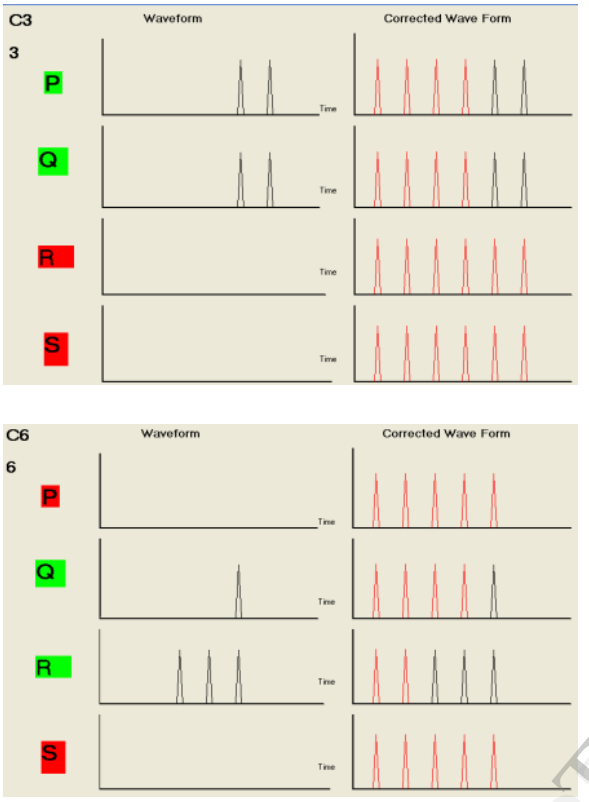


4. RESULTS AND DISCUSSIONS:

The proposed project and assembled hardware been evaluated and results of furnished has a VB coding and ASM coding. To evaluated and obtain a result we have used pacemaker is an application tool, where the inputs are given through push switches instead of and ECG waveform.

According to our project the system has to accept linear input compacts analyzer and must produce appropriate output as per the logic developed in the program is perfectly matched and graphical representation at the present result is attached with this thesis.





5. CONCLUSION

Thel conclusion of the research is submitted here with, assembling, testing, evaluation of application orientation is done for PIC embedded microcontroller. Embedded „C" coding enables us to interact with computer by serial communication.PIC microcontroller and computer connectivity is made in this project, instead of an ARM 7.0 processor, VB software is designed and the result is completely evaluated. Now to conclude PIC microcontroller and PC are connected secured for seamless project execution.

REFERENCES

[1]. T.wolf, and shufu Mao, "Hardware support for secure processing in embedded systems," IEEE Trans.vol 59,No.6, june 2010.

[2]. G. Gogniat, T. Wolf, W. Burleson, J.-P. Diguet, L. Bossuet, and R. Vaslin, "Reconfigurable Hardware for High-Security/High-
Performance Embedded Systems: The SAFES Perspective," IEEE Trans. Very Large Scale Integration (VLSI) Systems, vol. 16, no. 2, pp. 144-155, Feb. 2008

[3]. R.G. Ragel and S. Parameswaran, "IMPRES: Integrated Monitoring for Processor Reliability and Security," Proc. 43rd Ann. Conf. Design Automation (DAC), pp. 502-505, July 2006.

[4]. G.F. Cretu, J.J. Parekh, K. Wang, and S.J. Stolfo, "Intrusion and Anomaly Detection Model Exchange for Mobile Ad-Hoc Networks," Proc. Third IEEE Conf. Consumer Comm. and Networking (CCNC ″06), pp. 635-639, Jan. 2006.

[5]. D. Arora, S. Ravi, A. Raghunathan, and N.K. Jha, "Secure Embedded Processing through Hardware-Assisted Run-Time Monitoring," Proc. Design, Automation, and Test in Europe onference and Exhibition (DATE ″05), Mar.2005.

[6]. J. Zambreno, A. Choudhary, R. Simha, B. Narahari, and N. Memon, "SAFEOPS: An Approach to Embedded Software Security," ACM Trans. Embedded Computing Systems, vol. 4, no. 1, pp. 189-210, Feb. 2005.

[7]. M. Abadi, M. Budiu, _ U. Erlingsson, and J. Ligatti, "Control-Flow Integrity Principles, Implementations, and Applications," Proc. ACM Conf. Computer and Comm. Security (CCS), , Nov. 2005.

[8]. R.G. Ragel, S. Parameswaran, and S.M. Kia, "Micro Embedded Monitoring for Security in Application Specific Instruction-Set Processors," Proc. 2005 Int″l Conf. Compilers, Architectures, and Synthesis for Embedded Systems (CASES), pp. 304-314, Sept. 2005.

[9]. G.E. Suh, J.W. Lee, D. Zhang, and S. Devadas, "Secure Program Execution via Dynamic Information Flow Tracking," Proc. 11th Int″l Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS-XI), pp. 85-96, Oct. 2004.

[10]. .N. Nakka, Z. Kalbarczyk, R.K. Iyer, and J. Xu, "An Architectural Framework for Providing Reliability and Security Support," Proc. 2004 Int″l Conf. Dependable Systems and Networks (DSN, June 2004.

[11]. .S. Ravi, A. Raghunathan, and S. Chakradhar, "Tamper Resistance Mechanisms for Secure, Embedded Systems," Proc. 17th Int″l Conf. Very Large Scale Integration Design (VLSI Design ″04), Jan. 2004.

[12]. .Z. Shao, Q. Zhuge, Y. He, and E.H.-M. Sha, "Defending Embedded Systems Against Buffer Overflow via Hardware/Software," Proc. 19th Ann. Computer Security Applications Conf. (ACSAC), pp. 352-363, Dec. 2003.

[13]. .K.-S. Lhee and S.J. Chapin, "Buffer Overflow and Format String Overflow Vulnerabilities," Software: Practice and Experience, vol. 33, no. 5, pp. 423-460,Apr. 2003

[14]..X. Zhang, L. Doorn, T. Jaeger, R. Perez, and R. Sailer, "Secure coprocessor-based intrusion detection," in Proc. ACM SIGOPS European Wkshp., Sept. 2002.

[15.]A.Wood and J.A. Stankovic, "Denial of Service in Sensor Networks,"Oct. 2002.

[16].M.R. Guthaus, J.S. Ringenberg, D. Ernst, T.M. Austin, T. Mudge, and R.B. Brown, "MiBench: A Free, Commercially Representative Embedded Benchmark Suite," Proc. IEEE Fourth Ann. Workshop Workload Characterization,Dec. 2001.

[17].D. Burger and T.M. Austin, "The SimpleScalar Tool Set, Version 2.0," Dept.of Computer Science, Univ. of Wisconsin in Madison, Technical Report 1342, June 1997.

[18].C. Ko, M. Ruschitzka, and K. Levitt, "Execution monitoring of security critical programs in distributed systems: A specification-based approach," in Proc. IEEE Symp. on Security & Privacy, May 1997.

[19].P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," Proc. 19th Ann.Int″l Cryptology Conf,1999.

[20].S. Chari, C.S. Jutla, J.R. Rao, and P. Rohatgi, "Towards Sound Approaches to Counteract Power-Analysis Attacks," Proc. 19th Ann. Int″l Cryptology Conf. Advances in Cryptology (CRYPTO ″99), pp. 398-412,1999.