ISSN: 2582-8118 ***Volume 1, Issue 3; May 2021***

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYSIS IN SCIENCE AND ENGINEERING

Web: https://www.iarj.in/index.php/ijrase/index

11. Sensor Integration in Embedded Systems

Basheer P. I.
HOD, Dept of Electronics,
SSM Polytechnic College Tirur, Kerala.

Shajil Ameer V. V.
Lecturer, Electronics Engineering,
SSM Polytechnic College, Tirur, Kerala.

Subair P. H.
Rtd. HOD, Dept. of Electronics,
SSM Polytechnic, Tirur, Kerala.

ABSTRACT

Complex Embedded Systems rely heavily on sensors and actuators (CES). Catastrophic events can occur in these CES if the sensors or actuators malfunction. System performance can be severely impacted if a sensor or actuator malfunctions, which is difficult to detect. Integration testing of the 'Subsystem under test (SUT)' with sensors or actuators is necessary to ensure that the 'Embedded Processing Component (EPC)' meets specifications. In order to properly integrate sensors and actuators into the overall system, it is necessary to detect and correct any failure modes that may arise. Diagnosing a defective machine in a series of two CFSM requires conducting sensor/actuator integration testing within ES. The diagnostic methodology is utilised to find a solution to the problem at hand. Integrating sensors and actuators is just one part of the larger diagnostic challenge. It is becoming increasingly necessary in the age of IoT to have a solid foundation in areas such as signal processing, artificial intelligence, and multi-media communication in order to effectively use embedded systems based on mobile sensor technology. "Defective process " is used to describe a communication channel that has a faulty communication channel, which alters or possibly modifies events between the fault-free processes. The use of functional tests in integration testing ensures that the known and tested system requirements are appropriately identified. It is possible to determine whether an integration test is successful or unsuccessful by comparing the actual output with the expected response across a communication channel.

KEYWORDS

Sensor, Embedded, EPC, IOT, Real Time, RTE, Embedded System, Sensor Integration.

Introduction:

In the context of a larger system, an embedded system is a computer system that performs some of the functions of the bigger system. Automobile control systems, industrial process control systems, mobile phones, and compact sensor controllers are all examples of these systems. There is a wide variety of embedded systems, ranging from tiny computer-based devices to massive systems that monitor and control complicated processes in real time. More than ninety-nine percent of all computing units currently belong to the embedded system category[1]. A major issue in designing embedded systems is that they have so many requirements: they must be small, high-performance, low-cost and long-lasting, while also sensing and impacting their environment.

The real-time features of these embedded systems, such as reaction time, worst-case execution time, etc., are critical design considerations. To ensure the safety, reliability, availability, and other dependability characteristics of these systems. However, these systems have a restricted hardware capacity because they are compact in size and need to be mobile yet also have minimal production costs. To keep up with the increasing demands of embedded real-time systems, it is critical to manage life-cycle features like maintainability and portability in a way that is effective in detecting early errors and detecting problems before they become major problems.

Traditionally, embedded system software was written in assembly language, but today high-level languages like Java and C++ are utilised since embedded systems require compactness, which means that the amount of hardware required to run the software will be reduced, resulting in smaller devices.An energy efficient and high-speed gadget will be produced as a result of effective use of software in conjunction with a low-power operating system.

There is a new type of computer called RTEs that arose as an alternative that is surely helpful. RTEs are computers that focus on the application and are based on computer technology, but they can be customised based on the user's needs. "Real-time" normally refers to the 'time' specified by external sources, but it can also refer to a special-purpose computational system integrated inside another machine[2]. The overall system's requirements dictate the timeframes.

The Structure of Embedded System:

In order to enable a wide range of communication methods, a variety of architectures and platforms have been developed. Data acquisition, monitoring devices, network connections, and data transmission for control functions to higher DAS, DCS, and monitoring systems through Ethernet or ICB are all part of the platform's primary focus. Distinct components are found in each of these smaller systems. The hardware configuration refers to all of this as a whole. In order to store the programmes, some of these discrete pieces may be programmable, hence they will need memory[3]. The non-volatile memory on-chip or on-board of an RTE is where these programmes are stored. Both the operating system (RTOS) and application programmes are contained in these programmes. Figure 11.1 depicts the usual architecture of an embedded system.

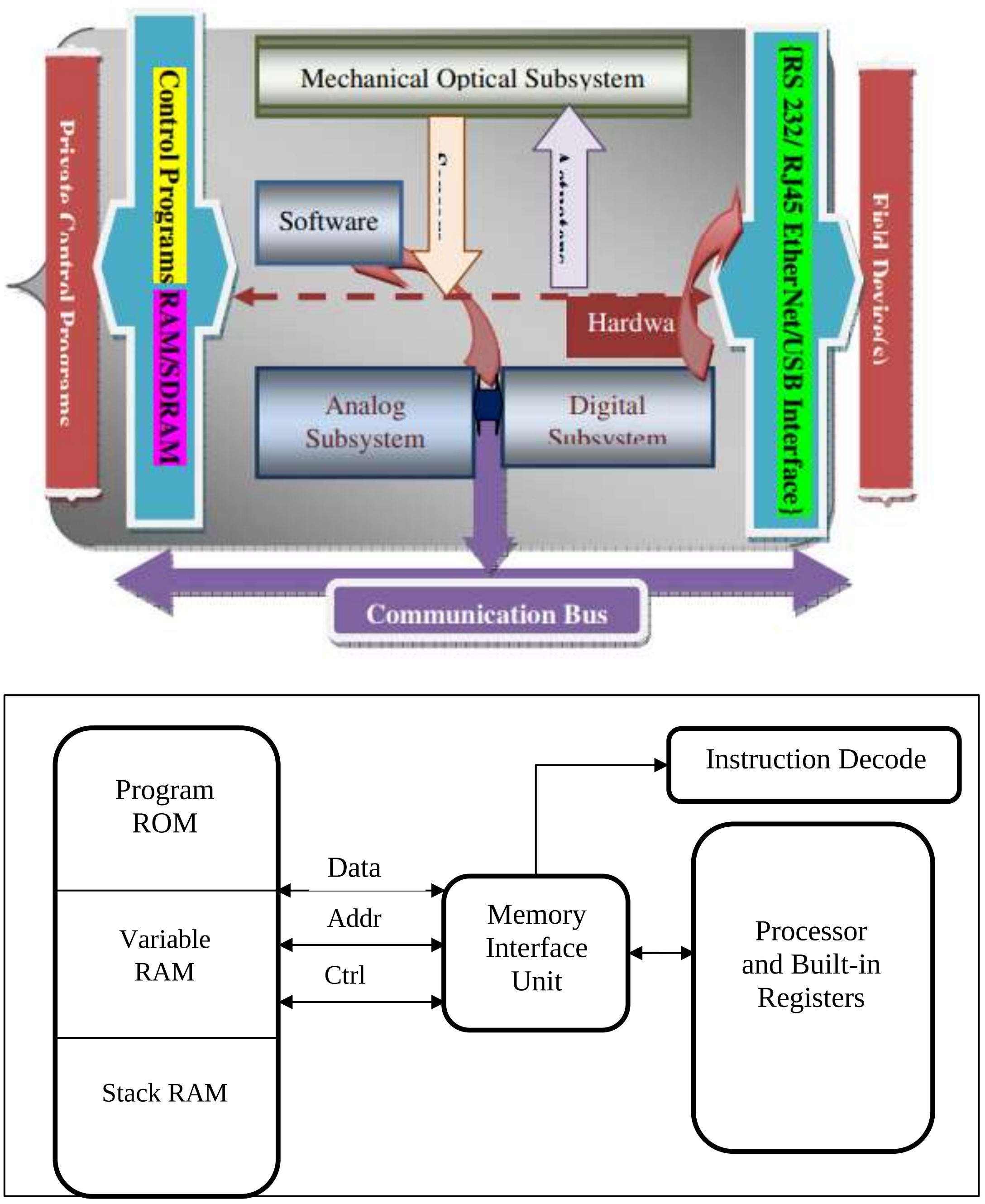

Figure 1: Embedded system architecture for Intelligent System

Embedded systems are found in abundance in today's automobiles. Among the embedded systems, one manages the anti-lock brakes, another keeps tabs on the vehicle's emissions, and a third provides dashboard information. Numerous embedded systems can be found even in the most general-purpose personal computer today. It's not just a keyboard and mouse; it includes a video card, modem, hard disc, floppy drive, and a sound card.

The advent of the microprocessor in 1971 signalled the beginning of the digital age[4]. Embedded systems were first used in unmanned space probes, computerised traffic lights, and aircraft flight control systems, among other things.

Embedded systems revolutionised our personal and professional life in the 1980s. As embedded system technology improves, new gadgets are being developed to make our lives easier and more comfortable. Mobile phones, PDAs, and digital cameras are just the tip of the iceberg in this rapidly expanding area.

Embedded System:

The electronics industry is built on embedded systems. These systems have a specific function in mind. Hardware and software come together to form an embedded system. The type of microprocessor or microcontroller required depends on the system. Embedded systems can be divided into two broad categories.

- Based on performance and functional Requirements.
- Based on performance of microcontroller

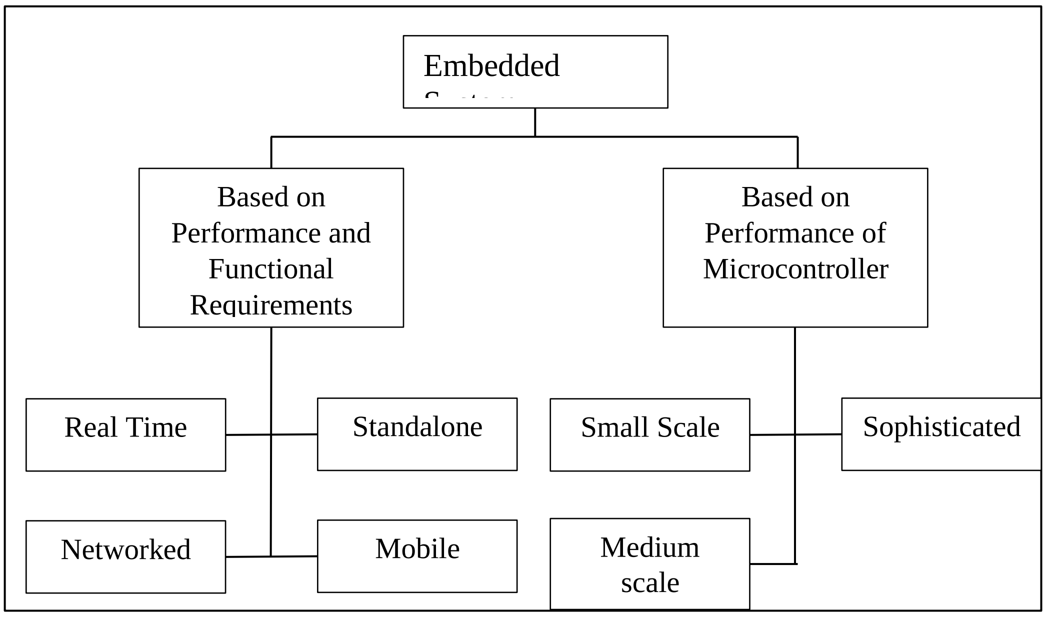
Embedded
Based on Performance and Functional Requirements
Based on Performance of Microcontroller
Real Time
Standalone
Small Scale
Sophisticated
Networked
Mobile
Medium scale

Figure 2: Embedded Systems Categaries

Review of Literature:

Cache and SPM in embedded systems have been shown to have advantages by Panda et al. Using this advancement, it is now possible to combine the advantages of both strategies and improve the performance of dynamic binary transformation in resource-constrained embedded systems by dislodging the address spaces occupied by SPM and main memory.

They found that altering the arrangement of programme items (code and data) in memory into non-cacheable, cacheable, and SPM sections had significant benefits for boosting cache efficiency.

The disadvantage of this strategy is that it places programme objects in a condensed position. Because the cache mapping rule evaluates whether two series items compete for the same cache block, this restriction is pointless.

Since cache conflicts can be alleviated by allowing unused memory space between programme objects in the main memory, allocation methods have adopted the locality optimization technique in order to accomplish the efficiency.

A vector/matrix abstraction technique is used by Tanja Van Achteren and Francky Catthoor to check at compile time for fresh data that needs to be positioned in cache and reused in the future in order to reduce the power consumption of the system.

Manish verma is Using a weighted CFG layout and an execution trace energy subsystem, we were able to acquire a trace of the programme blocks that were transported into memory during execution for varying cache sizes. A system employing one word SPM and two-word SPM generates an energy value that is then used to create a loop cache with one- and two-word energy values. Cache misses are eliminated, and the process becomes more energy efficient as a result.

Objectives:

a. Included are the upcoming ES technologies that are expected to be novel and promising in the next decade or so
b. Give a complete explanation of the many CES for Sensors and the state of the art.
c. Recognize the most potential future directions and developments in the field of CES for Sensors for those looking to participate.

Research Methodology:

It is our goal here to find out how electrical sensors may be used in the assessment of product quality using technical breakthroughs to integrate modelling, simulation, and analysis solutions.

Result and Discussion:

Embedded systems are the most common type of real-time system. There is no need to remove the processor or the software from the equipment they are managing. An example of a typical embedded application might be a cellular phone or a microwave oven or a laser printer or an electronic toy or video game [5-7].

It is expected that the operating system will be integrated into the system. Because the operating system can't be loaded from a disc or floppy, all of the code must be stored in the system's Read-Only Memory.

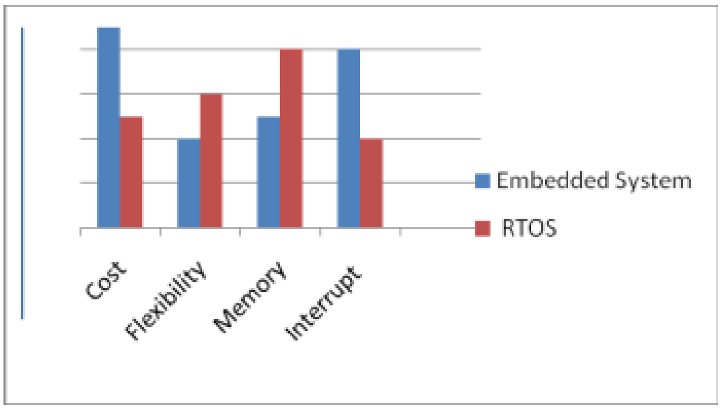

Figure 3: Comparison b/w Embedded System & RTOS

Sensors:

An electronic device, module, machine, or subsystem, known as a sensor, detects and communicates with other electronic devices, most typically a computer processor. Digital signals generated by sensors can be displayed, read, or processed in many ways. The diagram shows how a sensor works.[8]

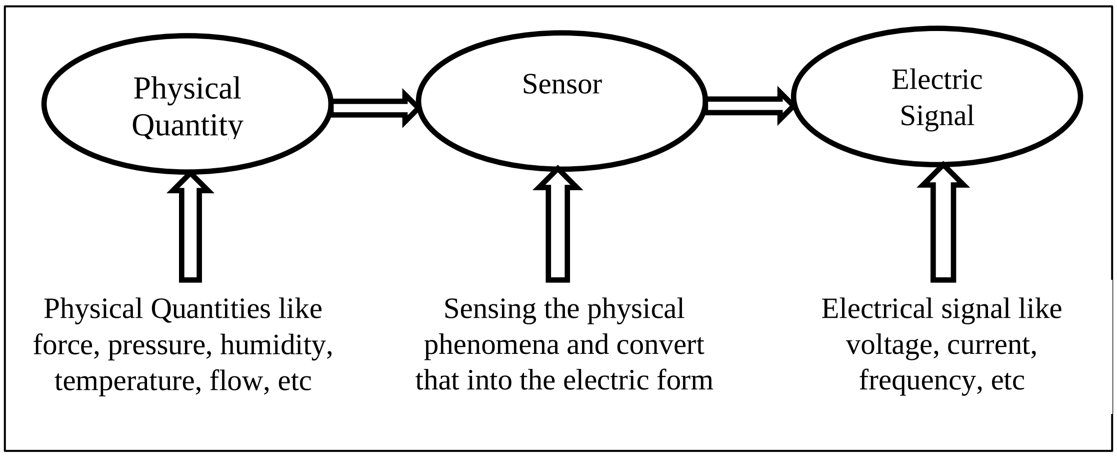
Physical Quantity
Sensor
Electric Signal
Physical Quantities like force, pressure, humidity, temperature, flow, etc
Sensing the physical phenomena and convert that into the electric form
Electrical signal like voltage, current, frequency, etc

Figure 4: working of sensor

Model of Sensor:

Finite state machines can be used to model sensors and actuators, which are both crucial components (DFSM). System performance can be severely impacted by the detection of sensor/actuator problems, which is difficult [10]. Various layers of the system hierarchy are affected by sensor and actuator problems. Low-level control processes sensor data, followed by mid- and high-level control, which decides how the system behaves. Low-level control is at the bottom of the stack. Failures of sensors or actuators have a direct impact on the system's hardware, causing everything from a minor issue to a catastrophic failure. For the "transduction process" (transfun) and the "related connection hardware" (conhd) (both electrical / electronic and mechanical), the sensor model depicted in Fig.5 officially captures faults and is based on an embedded system model presented in. For obvious reasons, all data exchanges between the environment and its transfun take place across the conhd and then on to the function (ip). An example of this is depicted in Figure 11.5(a) by allowing ip to carry inputs from the environment (a, b, C, D, and E) [11-14]. It is also necessary to send through the mechanical hardware the outputs (0; 1; 2; 3) that are generated (0; 1). A single input or output connection is assumed for each ip address. If we look at the model in Figure 11.5(b), we can see that input and output are either absent or have been diverted. Input variables of Controller (algorithm) that decide the (next) output state are these defects. Output variables of Controller are (combined action of controller and actuator).

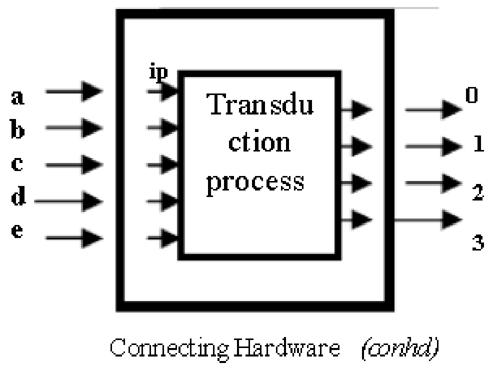

Figure 2: Sensor Model

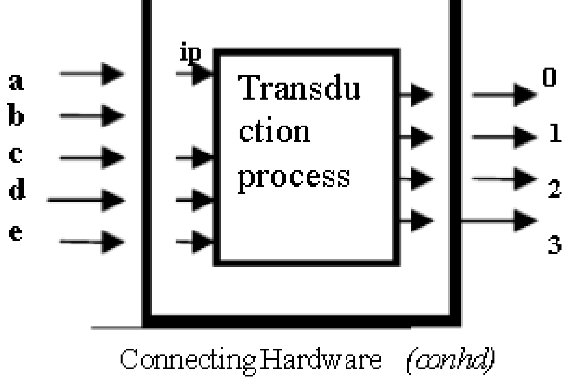

Figure 2: (b) Sensor Fault Model - missing input fault

Conclusion:

The rapid growth of digital data collecting and control systems is now necessitated by digital communication. Actuality Operating systems play a significant role in embedded systems, particularly for the development tools that are used to create the devices themselves. Thus, the first step in selecting an RTOS is to determine the processor, real-time performance, and budget constraints. The next step is to look at the many RTOSes on the market to see which one best fits our needs. Many modern kernels (RTOS) are utilised in real-time embedded systems; however they increase the cost and restrict system adaptability. Future real-time operating systems will necessitate new operating systems and job designs to ensure predictability and high adaptability. It is proposed to use an FSM-based sensor model, and faults are integrated via communication channel (link) architectures. An EPC sensor's integration with the SUT is considered as an indicator of a problematic communication route within the SUT. To ensure the integrity of the system, functional tests should be used rather than creating a new test sequence (NTS) for integration testing. It is possible to determine whether or not an integration test passed or failed by comparing observed output to what is expected to be received in the communication channel.

References:

1. P.R. Panda, N.D. Dutt, and A. Nicolau, “On-Chip vs. Off-Chip Memory: The Data Partitioning Problem in Embedded Processor- Based Systems,” ACM Trans. Design Automation of Electronic Systems, vol. 5, no. 3, pp. 682-704, 2000.
2. Hyungmin Cho Bernhard Egger Jaejin Lee Heonshik Shin “Dynamic Data Scratchpad Memory Management for a Memory Subsystem with an MMU”pp 541-547
3. Mahumut Kandemir,Associate Member ,IEEE,J Ramanujam,Member,IEEE,Mary Jane Irwin,Fellow,IEEE,N Vijaykrishna,Associate Member,IEEE,Ismail Kadayif, and Amisha Parikh“ Compiler directed Scratch pad Memory Hierarchy Design and Management” .
4. M kandemir,associate member, IEEE,J. Ramanujam, Member IEEE, Mary Jane Irwin ,Fellow ,IEEE “A Complier Based Approach from Dynamically Managing Scratch Pad Memories in Embedded System”
5. T.M. Overman, et al., “High-Assurance Smart Grid: A Three-Part Model for Smart Grid Control Systems”, Proceedings. of the IEEE, vol.99, no. 6, pp. 1046-1062, June 2011.
6. N. Kularatna and B. H. Sudantha, “An environmental air pollution monitoring system based on the IEEE 1451 standard for low-cost requirements,” IEEE Sensors J., vol. 8, pp. 415–422, Apr. 2008.
7. W. Chung and C. H. Yang, “Remote monitoring system with wireless sensors module for room environment,” Sens. Actuators B, vol. 113 no. 1, pp. 35–42, 2009
8. Brian Santo, “Embedded Battle Royale,” IEEE Spectrum, Dec. 2001, pp.36-41.
9. Dedicated Systems Experts, RTOS Evaluation Project. Brussels, Belgium: Dedicated Systems Experts, 2001.
10. John A. Stankovic, Krithi Ramamritham, “The Design of the Spring Kernel,” Proc. IEEE- Real-Time Systems Symposium, Dec.1987, pp.146-57
11. G.S. Yang, H. Zhang, Z.G Niu, etc, “Reasearch and realization of embedded linux system based on MPC8248,” Measuring & Control Technology, 2008,27(1): 51-54.

12. David S. Linthicum. Next Generation Application Integration: From Simple Information to Web Services. Addison-Wesley Professional, 2003
13. Tanja Van Achteren, Geert Deconinck, K.U.Leuven ESAT/ACCA, Belgium and F Catthoor ,Rudy Lauwereins "Data Reuse Exploration Technique for Loop dominated Application
14. Edward Balaban, Abhinav Saxena, Prasun Bansal, Kai F. Goebel, Paul Stoelting and Simon Curran, "A Diagnostic Approach for Electro-Mechanical Actuators in AerospaceSystems" IEEEAC paper #1345, Version 4, Updated January9, 2009
15. Bratthall, L.G., Runeson, P., Ädelsward, K., and Eriksson, W., A survey of lead-time challenges in the development and evolution of distributed real-time systems. Information and Software Technology, 42(13):947–958, September 2000.
16. Gerrit Muller, "Coping with System Integration Challenges in Large Complex Environments", Embedded Systems Institute.