ABSTRACT:

An embedded system is a computer system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts.

Modern embedded systems are often based on microcontrollers but ordinary microprocessors are also common, especially in more-complex systems. In either case, the processor(s) used may be types ranging from general purpose to those specialised in certain class of computations, or even custom designed for the application at hand. A common standard class of dedicated processors is the Digital signal processor (DSP).parts.

Since the embedded system is dedicated to specific tasks, design engineers can optimize it to reduce the size and cost of the product and increase the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale.

**HISTORY:**

One of the very first recognizably modern embedded systems was the Apollo Guidance Computer, developed by Charles Stark Draper at the MIT Instrumentation Laboratory. At the project's inception, the Apollo guidance computer was considered the riskiest item in the Apollo project as it

employed the then newly developed monolithic integrated circuits to reduce the size and weight. An early mass-produced embedded system was the Autonetics D-17 guidance computer for the Minuteman missile, released in 1961.

Since these early applications in the 1960s, embedded systems have come down in price and there has been a dramatic rise in processing power and functionality. An early microprocessor for example, the Intel 4004, was designed for calculators and other small systems but still required external memory and support chips. In 1978 National Engineering Manufacturers Association released a "standard" for programmable microcontrollers, including almost any computer-based controllers, such as single board computers, numerical, and event-based controllers.

**CHARACTERISTICS:**

**1.DEBUGGING**

Embedded debugging may be performed at different levels, depending on the facilities available. From simplest to most sophisticated they can be roughly grouped into the following areas:-

An in-circuit debugger (ICD), a hardware device that connects to the microprocessor via a JTAG or Nexus interface. This allows the operation of the microprocessor to be controlled externally, but is typically restricted to specific debugging capabilities in the processor.

• modified, and allowing debugging on a normal PC. The downsides are expense and slow operation, in some cases up to 100X slower than the final system.

• For SoC designs, the typical approach is to verify and debug the design on an FPGA prototype board. Tools such as Certus are used to insert probes in the FPGA RTL that make signals available for observation. This is used to debug hardware, firmware and software interactions across multiple FPGA with capabilities similar to a logic analizer.

TRACING:-

Real-time operating systems (RTOS) often supports tracing of operating system events. A graphical view is presented by a host PC tool, based on a recording of the system behaviour. The trace recording can be performed in software, by the RTOS, or by special tracing hardware. RTOS tracing allows developers to understand timing and performance issues of the software system and gives a good understanding of the high-level system behaviours. Commercial tools like RTXCQuadros or IAR Systems exists.

RELIABILITY:

Embeddd systems often reside in machines that are expected to run continuously for years without errors, and in some cases recover by themselves if an error occurs.Therefore,the software is usually developed and tested unreliable mechanical moving parts such as disc drives, switches or buttons are avoided.

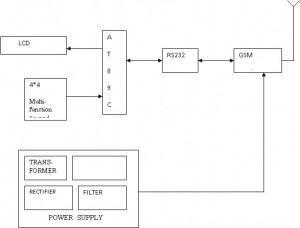
Specific reliability issues may include:

1.The system cannot safely be shutdown for repair or it is to inaccessible to repair.Examples include space systems,under sea cables, navigational beacons, bore-hole systems,and automobiles.

2.The system must be kept running for safety reasons.”Limp modes” are less tolerable. Often backups are selected by an operator.Example include Air craft navigation , Reactor control systems, safety- critical chemical factory controls.

**APPLICATIONS**

|  |  |
| --- | --- |
| *1.*[*SMS & Call Service Implementation for Mobile Phones Using 8051*](http://1000projects.org/sms-call-service-implementation-for-mobile-phones-using-8051.html)*:* |  |

             In this sms & **call service implementation for mobile phones using 8051 project** we are going to implement sms and call service for mobile phones using 8051 micro-controller. We are also going to implement the multi-functional keypad by which we can type the messages and send the messages and we can call to a mobile. These functions satisfy the basic things in a mobile phone using 8051 controller.   [](http://1000projects.org/wp-content/uploads/2011/12/SMS-CALL-SERVICE.jpg)

**Working Principle:**

|  |
| --- |
|  |
|  |
|  |

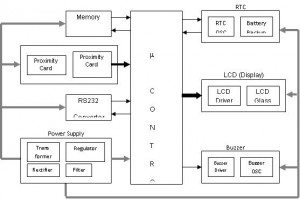
We are going to implement 4\*4 keypad to micro-controller and write the code for multi-functional keypad .GSM modem is interfaced to the micro –controller using RS232 .we interface LCD to the controller. When we want to send a message  just type message using keypad  and  the message is send to mobile using Gsmmodem.For call service we are going to type thenumber using keypad and press call button, we can interface a mic controller and a speaker for voice conversation.

# [*Tracking Police Man Using RF Proximity Card*](http://1000projects.org/tracking-police-man-using-rf-proximity-card.html)

|  |  |
| --- | --- |
|  |  |

**Tracking police man using RF proximity card embedded system project** explains about new method for bank officials for checking their security of the bank by implementing accurate method to control police officials. In present scenario banks are playing important role in providing security for people for saving their money, gold..Etc. So banks are taking serious steps to maintain security like hiring police officials at night time for patrolling banks, but there are chances in irregularity of police officials so this method will provide card system for every police official who should scratch his card. Data and time is stored inside system which can connect to computer and analyze details of his attendance.

Embedded systems range from portable devices such as [digital watches](https://en.wikipedia.org/wiki/Digital_watch) and [MP3 players](https://en.wikipedia.org/wiki/Digital_audio_player), to large stationary installations like [traffic lights](https://en.wikipedia.org/wiki/Traffic_light), [factory controllers](https://en.wikipedia.org/wiki/Programmable_logic_controller), and largely complex systems like [hybrid vehicles](https://en.wikipedia.org/wiki/Hybrid_vehicles), [MRI](https://en.wikipedia.org/wiki/MRI), and [avionics](https://en.wikipedia.org/wiki/Avionics). Complexity varies from low, with a single [microcontroller](https://en.wikipedia.org/wiki/Microcontroller) chip, to very high with multiple units, [peripherals](https://en.wikipedia.org/wiki/Peripheral) and networks mounted inside a large [chassis](https://en.wikipedia.org/wiki/Chassis).

[](http://1000projects.org/wp-content/uploads/2011/12/TRACKING-POLICE-MAN-USING-RF-PROXIMITY-CARD.jpg)

|  |  |
| --- | --- |
|  |  |

**ADVANTAGES:-**

 Enables real-time, deterministic scheduling and task prioritization

 Abstracts away the complexities of the processor

 Provides a solid infrastructure constructed of rules and policies

 Simplifies development and improves developer productivity

 Integrates and manages resources needed by communications stacks and middleware

 Optimizes use of system resources

 Improves product reliability, maintainability and quality

 Promotes product evolution and scaling

## The Advantages Of Embedded Android In Medical Devices:

So what advantages does Android have over other embedded systems? When it comes to medical devices, there are quite a few:

* Open-source, stable, and proven operating system
* Advanced networking and communication out-of-the-box
* Refined and modern User Interface and User Experience
* Specifically designed for touch screen systems, both large and small
* Open and extensible
* A complete computing platform

**DISADVANTAGES:-**

The disadvantages, is that the hardware is often just enough to do the work (requiring an upgrade to expand functions). When the product reaches EOL, the embedded system usually serves no further practical use, or is hard to re-purpose, includig getting around security.

**1. Difficult to change configurations and features**  
Once an embedded system is deployed (or finalized), it will be difficult to change its configuration - both its hardware and software. Remote update of software is possible provided the capability is included. Hence, proper requirement analysis is a must before deployment. Hardware configuration change will be much more trickier which may require existing boards be completely replaced. I have seen this happen and it is not pretty.  
  
**2. Issue of scalability**  
     Because it is difficult to change configuration, an embedded system cannot be easily scaled up as demand/scope changes. Said so, embedded systems can be designed to scale up for example using expansion ports or networking etc. This means it must be decided before hand during design phase for scale up provisions.    
  
**3. Limitation of hardware**  
     With a limited memory or computing capability in most embedded systems, there is always a limitation (or an upper limit) on our software design(upgrade). Be always aware of "Memory" and "Speed".   
  
**4. Applied for a specific purpose**  
By definition, embedded systems are constrained in their objectives. If it is decided to "rehash" an existing embedded system for a completely different purpose, it will normally result in significant change(s) in either or both its hardware or software.

**CONCLUSION:-**

The embedded system design growing complexity and the known design strategies deficiencies have motivated this work. We have designed two embedded systems following the AOSD methodology. We found advantages and disadvantages in this strategy, contributing to the evolution of this methodology. By using AOSD techniques, it was possible to infer specific hardware components from application code, using the artifactof hardware mediators. This reduces development time and helps the development of a system with the least number of hardware components possible. The evolution of the AOSD methodology, especially in the area of hardware generation from application code.