MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

LECTURE HANDOUTS | **L01**

ECE | **III / VI**

Course Name with Code : **Embedded Systems and RTOS & 19ECC15**

Course Faculty : **Dr.P.Padmaloshani**

Unit : **I - INTRODUCTION** **Date of Lecture:**

Topic of Lecture: Introduction to Embedded Systems

Introduction :

- Embedded system is an Electronic/Electro mechanical system which is designed to perform a specific function and is a combination of both hardware and firmware (Software)

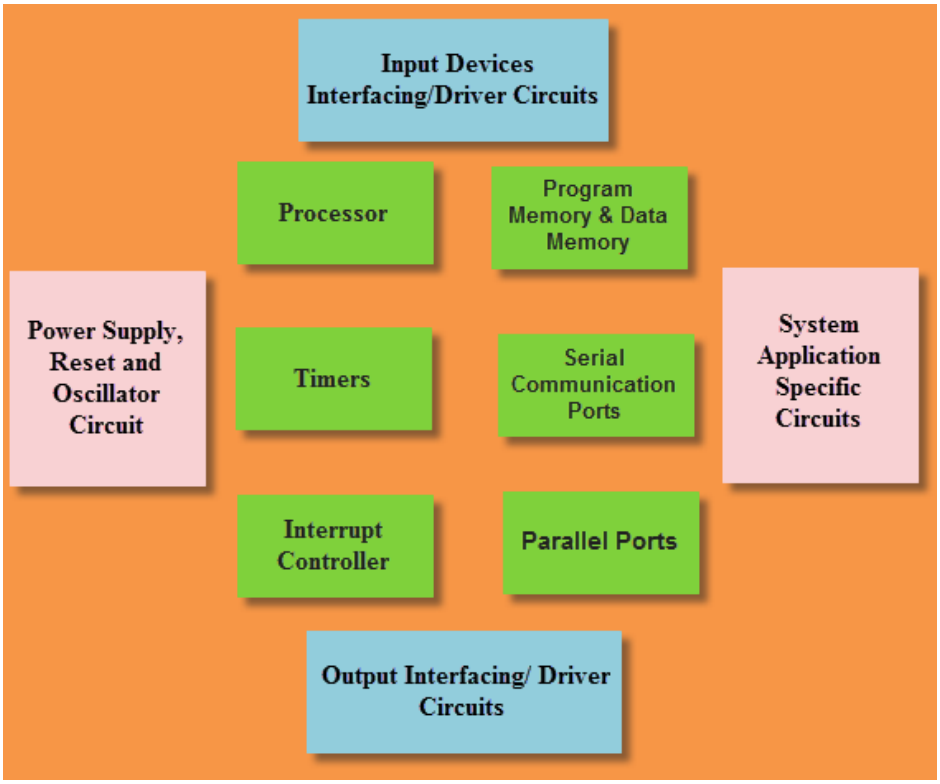
Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Introduction to Embedded Systems :

- An embedded system is an electronic system that has a software and is embedded in computer hardware.
- It is programmable or non- programmable depending on the application.
- An Embedded system is defined as a way of working, organizing, performing single or multiple tasks according to a set of rules
- In an embedded system, all the units assemble and work together according to the program.
- Examples of embedded systems include numerous products such as microwave ovens, washing machine, printers, automobiles, cameras, etc.
- These systems use microprocessors, microcontrollers as well as processors like DSPs.
- **Important terminologies used in embedded system**
- **Reliability:**
- This measure of the survival probability of the system when the function is critical during the run time.
- **Fault-Tolerance:**
- Fault-Tolerance is the capability of a computer system to survive in the presence of faults.
- **Real-Time:**
- Embedded system must meet various timing and other constraints. They are imposed on it by the real-time natural behavior of the external world.
- For example, an airforce department which keeps track of incoming missile attacks must precisely calculate and plan their counter-attack due to hard real-time deadline. Otherwise, it'll get destroyed.
- **Flexibility:**
- It's building systems with built-in debugging opportunities which allows remote maintenance.

- For example, you are building a spacecraft which will land on another planter to collect various types of data and send collected detail back to us.
- If this spacecraft went insane and lost the control, we should be able to make some important diagnostic.
- So, flexibility is vital while designing an embedded system.
- **Portability:**
- Portability is a measure of the ease of using the same embedded software in various environments.
- It requires generalized abstractions between the application program logic itself and the low-level system interfaces.
- **Embedded System Block Diagram**



- The embedded systems basics include the components of embedded system hardware, embedded system types and several characteristics.
- An embedded system has three main components: Embedded system hardware, Embedded system software and Operating system

Advantages of Embedded System

- It is able to cover a wide variety of environments
- Less likely to encore errors
- Embedded System simplified hardware which, which reduces costs overall.
- Offers an enhanced performance
- The embedded system is useful for mass production.
- The embedded system is highly reliable.
- It has very few interconnections.
- The embedded system is small in size.
- It has a fast operation.
- Offers improved product quality.
- It optimizes the use of system resources.
- It has a low power operation

Disadvantages of Embedded System

- To develop an embedded system needs high development effort.
- It needs a long time to market.
- Embedded systems do a very specific task, so it can't be programmed to do different things.
- Embedded systems offer very limited resources for memory.
- It doesn't offer any technological improvement.

- It is difficult to backup of embedded files.

Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=uFhDGagZzjs

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 4

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

LECTURE HANDOUTS | **L02**

ECE | **III / VI**

Course Name with Code : **Embedded Systems and RTOS & 19ECC15**

Course Faculty : **Dr.P.Padmaloshani**

Unit : **I - INTRODUCTION** **Date of Lecture:**

Topic of Lecture: Definition of Embedded System, Embedded Systems Vs General Computing Systems

Introduction :

- An Embedded system is a combination of computer hardware and software.
- As with any electronic system, this system requires a hardware platform and that is built with a microprocessor or microcontroller.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

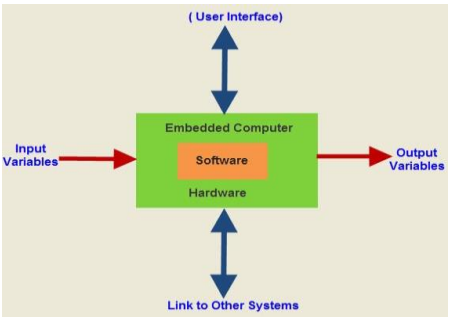
Definition of Embedded System, Embedded Systems Vs General Computing Systems:

- An Embedded system is a combination of computer hardware and software.
- As with any electronic system, this system requires a hardware platform and that is built with a microprocessor or microcontroller.
- The Embedded system hardware includes elements like user interface, Input/Output interfaces, display and memory, etc.
- Generally, an embedded system comprises power supply, processor, memory, timers, serial communication ports and system application specific circuits.

E.g. Electronic Toys, Mobile Handsets, Washing Machines, Air Conditioners, Automotive Control Units, Set Top Box, DVD Player etc…

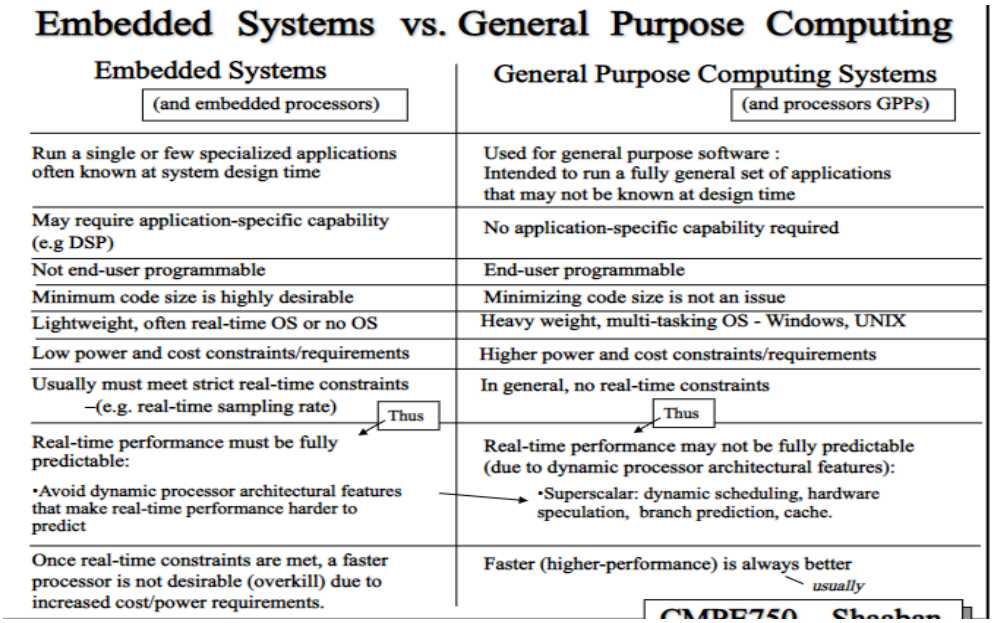
Embedded Systems are:

- ✈ Unique in character and behavior
- ✈ With specialized hardware and software



- Embedded system software is written in a high-level language, and then compiled to achieve a specific function within a non-volatile memory in the hardware.
- Embedded system software is designed to keep in view of three limits.
- They are availability of system memory and processor speed.

- When the system runs endlessly, there is a need to limit the power dissipation for events like run, stop and wake up.



Video Content / Details of website for further learning (if any):
https://www.techtarget.com/iotagenda/definition/embedded-system
https://www.youtube.com/watch?v=17VCuvKZNAA

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 4

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

LECTURE HANDOUTS | **L03**

ECE | **III / VI**

Course Name with Code : **Embedded Systems and RTOS & 19ECC15**

Course Faculty : **Dr.P.Padmaloshani**

Unit : **I - INTRODUCTION** **Date of Lecture:**

Topic of Lecture: History of Embedded Systems, Classification

Introduction :

- An Embedded system is a combination of computer hardware and software.
- As with any electronic system, this system requires a hardware platform and that is built with a microprocessor or microcontroller.

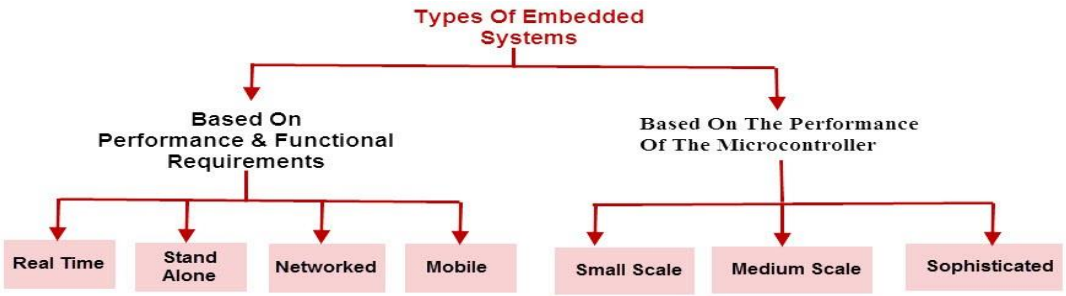
Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

History of Embedded Systems, Classification:

- Here, are important milestones from the history of embedded system:
- In 1960, embdded system was first used for developing Apollo Guidance System by Charles Stark Draper at MIT.
- In 1965, Autonetics, developed the D-17B, the computer used in the Minuteman missile guidance system.
- In 1968, the first embedded system for a vehicle was released.
- Texas Instruments developed the first microcontroller in 1971.
- In 1987, the first embedded OS, VxWorks, was released by Wind River.
- Microsoft's Windows embedded CE in 1996.
- By the late 1990s, the first embedded Linux system appeared.
- The embedded market reach $140 billion in 2013.
- Analysts are projecting an Embedded market larger than $40 billion by 2030.

Embedded systems can be classified into different types based on performance, functional requirements and performance of the microcontroller.



- Embedded systems are classified into four categories based on their performance and functional requirements:

Stand alone embedded systems
Real time embedded systems
Networked embedded systems
Mobile embedded systems

- Embedded Systems are classified into three types based on the performance of the microcontroller such as

Small scale embedded systems
Medium scale embedded systems
Sophisticated embedded systems

Stand Alone Embedded Systems

- Stand alone embedded systems do not require a host system like a computer, it works by itself.
- It takes the input from the input ports either analog or digital and processes, calculates and converts the data and gives the resulting data through the connected device-Which either controls, drives and displays the connected devices.
- Examples for the stand alone embedded systems are mp3 players, digital cameras, video game consoles, microwave ovens and temperature measurement systems.

Real Time Embedded Systems

- A real time embedded system is defined as, a system which gives a required o/p in a particular time.
- These types of embedded systems follow the time deadlines for completion of a task.
- Real time embedded systems are classified into two types such as soft and hard real time systems.

Networked Embedded Systems

- These types of embedded systems are related to a network to access the resources.
- The connected network can be LAN, WAN or the internet. The connection can be any wired or wireless.
- This type of embedded system is the fastest growing area in embedded system applications.
- The embedded web server is a type of system wherein all embedded devices are connected to a web server and accessed and controlled by a web browser.
- Example for the LAN networked embedded system is a home security system wherein all sensors are connected and run on the protocol TCP/IP

Mobile Embedded Systems

- Mobile embedded systems are used in portable embedded devices like cell phones, mobiles, digital cameras, mp3 players and personal digital assistants, etc.
- The basic limitation of these devices is the other resources and limitation of memory.

Small Scale Embedded Systems

- These types of embedded systems are designed with a single 8 or 16-bit microcontroller, that may even be activated by a battery.

For developing embedded software for small scale embedded systems, the main programming tools are an editor, assembler, cross assembler and integrated development environment (IDE)

Medium Scale Embedded Systems

- These types of embedded systems design with a single or 16 or 32 bit microcontroller, RISCs or DSPs.
- These types of embedded systems have both hardware and software complexities.
- For developing embedded software for medium scale embedded systems, the main programming tools are C, C++, JAVA, Visual C++, RTOS, debugger, source code engineering tool, simulator and IDE.

Sophisticated Embedded Systems

- These types of embedded systems have enormous hardware and software complexities, that may need ASIPs, IPs, PLAs, scalable or configurable processors.
- They are used for cutting-edge applications that need hardware and software Co-design and components which have to assemble in the final system.

Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=FUWnYHBGw1o

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 5, 6

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

LECTURE HANDOUTS | **L04**

ECE | **III / VI**

Course Name with Code : **Embedded Systems and RTOS & 19ECC15**

Course Faculty : **Dr.P.Padmaloshani**

Unit : **I - INTRODUCTION** **Date of Lecture:**

Topic of Lecture: Major Application Areas

Introduction :

- An Embedded system is a combination of computer hardware and software.
- As with any electronic system, this system requires a hardware platform and that is built with a microprocessor or microcontroller.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Major Application Areas:

- **Robotic science**
- **Ground Vehicles**
- **Drones**
- **Underwater Vehicles**
- **Industrial Robots**
- **Medical**
- **Dialysis Machine**
- **Infusion Pumps**
- **Cardiac Monitor**
- **Prosthetic Device**
- **Automotive**
- **Engine Control**
- **Ignition System**
- **Brake System**
- **Networking**
- **Router**
- **Hubs**
- **Gateways**
- **Electronics Instruments**
- **Home Devices**
- **TVs**
- **Digital Alarm**
- **Air Conditioner**
- **DVD Video Player**

- **Cameras**
- **Automobiles**
- **Fuel Injection**
- **Lighting System**
- **Door Locks**
- **Air Bags**
- **Windows**
- **Parking Assistant System**
- **Anti-stealing Alarms Whippers Motion**
- **Industrial Control**
- **Robotics**
- **Control System**
- **Missiles**
- **Nuclear Reactors**
- **Space Stations**
- **Shuttles**

Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=fE9li1T92vI

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 7

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

LECTURE HANDOUTS | **L05**

ECE | **III / VI**

Course Name with Code : **Embedded Systems and RTOS & 19ECC15**

Course Faculty : **Dr.P.Padmaloshani**

Unit : **I - INTRODUCTION** **Date of Lecture:**

Topic of Lecture: Major Application Areas

Introduction :

- An Embedded system is a combination of computer hardware and software.
- As with any electronic system, this system requires a hardware platform and that is built with a microprocessor or microcontroller.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Major Application Areas:

Telecom: Cellular Telephones, Telephone switches, Handset Multimedia Applications etc.

❑ **Computer Peripherals:** Printers, Scanners, Fax machines etc.

❑ **Computer Networking Systems:** Network Routers, Switches, Hubs, Firewalls etc.

❑ **Health Care:** Different Kinds of Scanners, EEG, ECG Machines etc.

❑ **Measurement & Instrumentation:** Digital multi meters, Digital CROs, Logic Analyzers PLC systems etc.

❑ **Banking & Retail:** Automatic Teller Machines (ATM) and Currency counters, Point of Sales (POS)

❑ **Card Readers:** Barcode, Smart Card Readers, Hand held Devices etc.

- **Consumer Electronics:** Camcorders, Cameras etc.

❑ **Household Appliances:** Television, DVD players, washing machine, Fridge, Microwave Oven etc.

❑ **Home Automation and Security Systems:** Air conditioners, sprinklers, Intruder detection alarms, Closed Circuit Television Cameras, Fire alarms etc.

❑ **Automotive Industry:** Anti-lock breaking systems (ABS), Engine Control, Ignition Systems, Automatic Navigation Systems etc.
Video Content / Details of website for further learning (if any): https://www.youtube.com/watch?v=fE9li1T92vI
Important Books/Journals for further learning including the page nos.: Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 7

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

LECTURE HANDOUTS | **L06**

ECE | **III / VI**

Course Name with Code : **Embedded Systems and RTOS & 19ECC15**

Course Faculty : **Dr.P.Padmaloshani**

Unit : **I - INTRODUCTION** **Date of Lecture:**

Topic of Lecture: Purpose of Embedded Systems

Introduction :

- An Embedded system is a combination of computer hardware and software.
- As with any electronic system, this system requires a hardware platform and that is built with a microprocessor or microcontroller.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Purpose of Embedded Systems:

Each Embedded Systems is designed to serve the purpose of any one or a combination of the following tasks.

o Data Collection/Storage/Representation

o Data Communication

o Data (Signal) Processing

o Monitoring

o Control

o Application Specific User Interface

1. **Data Collection/Storage/Representation:-**

❖ Performs acquisition of data from the external world.

❖ The collected data can be either analog or digital

❖ Data collection is usually done for storage, analysis, manipulation and transmission

❖ The collected data may be stored directly in the system or may be transmitted to some other systems or it may be processed by the system or it may be deleted instantly after giving a meaningful representation

2. Data Communication:-

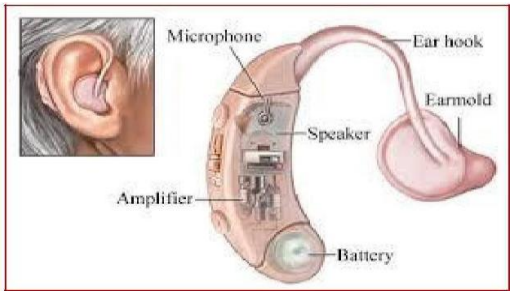
Embedded Data communication systems are deployed in applications ranging from complex satellite communication systems to simple home networking systems

Embedded Data communication systems are dedicated for data communication

The data communication can happen through a wired interface (like Ethernet, RS-232C/USB/IEEE1394 etc) or wireless interface (like Wi-Fi, GSM,/GPRS, Bluetooth, ZigBee etc)

Network hubs, Routers, switches, Modems etc are

typical examples for dedicated data transmission embedded systems



3. Data (Signal) Processing:-

Embedded systems with Signal processing functionalities are employed in applications demanding signal processing like Speech coding, synthesis, audio video codec, transmission applications etc

Computational intensive systems

Employs Digital Signal Processors (DSPs)

Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=vNedLKVoLJk

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 8

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

LECTURE HANDOUTS | **L07**

ECE | **III / VI**

Course Name with Code : **Embedded Systems and RTOS & 19ECC15**

Course Faculty : **Dr.P.Padmaloshani**

Unit : **I - INTRODUCTION** **Date of Lecture:**

Topic of Lecture: Purpose of Embedded Systems

Introduction :

- An Embedded system is a combination of computer hardware and software.
- As with any electronic system, this system requires a hardware platform and that is built with a microprocessor or microcontroller.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Purpose of Embedded Systems:

4. Monitoring:-

Embedded systems coming under this category are specifically designed for monitoring purpose
They are used for determining the state of some variables using input sensors
They cannot impose control over variables.
Electro Cardiogram (ECG) machine for monitoring the heart beat of a patient is a typical example for this
The sensors used in ECG are the different Electrodes connected to the patient‟s body

Measuring instruments like Digital CRO, Digital Multi meter, Logic Analyzer etc used in Control & Instrumentation applications are also examples of embedded systems for monitoring purpose

5. Control:-

Embedded systems with control functionalities are used for imposing control over some variables according to the changes in input variables
Embedded system with control functionality contains both sensors and actuators
Sensors are connected to the input port for capturing the changes in environmental variable or measuring variable

The actuators connected to the output port are controlled according to the changes in input variable to put an impact on the controlling variable to bring the controlled variable to the specified range
Air conditioner for controlling room temperature is a typical example for embedded system with „Control‟ functionality
Air conditioner contains a room temperature sensing element (sensor) which may be a thermistor and a handheld unit for setting up (feeding) the desired temperature

The air compressor unit acts as the actuator. The compressor is controlled according to the current room temperature and the desired temperature set by the end user

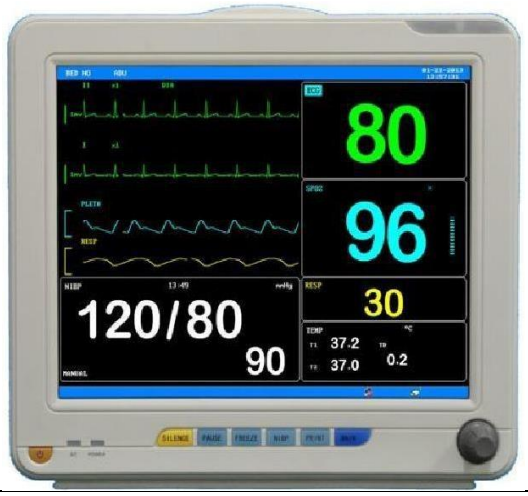
6. Application Specific User Interface:-

Embedded systems which are designed for a specific application

Contains Application Specific User interface (rather than general standard UI) like key board, Display units etc

Aimed at a specific target group of users

Mobile handsets, Control units in industrial applications etc are examples



Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=vNedLKVoLJk

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 8

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

LECTURE HANDOUTS | **L08**

ECE | **III / VI**

Course Name with Code : **Embedded Systems and RTOS & 19ECC15**

Course Faculty : **Dr.P.Padmaloshani**

Unit : **I - INTRODUCTION** **Date of Lecture:**

Topic of Lecture: Characteristics

Introduction :

- An Embedded system is a combination of computer hardware and software.
- As with any electronic system, this system requires a hardware platform and that is built with a microprocessor or microcontroller.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Characteristics:

Embedded systems possess certain specific characteristics and these are unique to each Embedded system.

1. Application and domain specific

2. Reactive and Real Time

3. Operates in harsh environments

4. Distributed

5. Small Size and weight

6. Power concerns

7. Single-functioned

8. Complex functionality

9. Tightly-constrained

10. Safety-critical

1. Application and Domain Specific:-

• Each E.S has certain functions to perform and they are developed in such a manner to do the intended functions only.

• They cannot be used for any other purpose.

• Ex – The embedded control units of the microwave oven cannot be replaced with AC‟S embedded control unit because the embedded control units of microwave oven and AC are specifically designed to perform certain specific tasks.

2. Reactive and Real Time:- • E.S are in constant interaction with the real world through sensors and user-defined input devices which are connected to the input port of the system.

• Any changes in the real world are captured by the sensors or input devices in real time and the control algorithm running inside the unit reacts in a designed manner to bring the controlled output variables to the desired level.

• E.S produce changes in output in response to the changes in the input, so they are referred as reactive systems.

• Real Time system operation means the timing behavior of the system should be deterministic ie the system should respond to requests in a known amount of time.

• Example – E.S which are mission critical like flight control systems, Antilock Brake Systems (**ABS**) etc are Real Time systems.

3. Operates in Harsh Environment :– • The design of E.S should take care of the operating conditions of the area where the system is going to implement.

• Ex – If the system needs to be deployed in a high temperature zone, then all the components used in the system should be of high temperature grade.

• Also proper shock absorption techniques should be provided to systems which are going to be commissioned in places subject to high shock.

4. Distributed: – • It means that embedded systems may be a part of a larger system.

• Many numbers of such distributed embedded systems form a single large embedded control unit.

• Ex – Automatic vending machine. It contains a card reader, a vending unit etc. Each of them are independent embedded units but they work together to perform the overall vending function.

5. Small Size and Weight:- • Product aesthetics (size, weight, shape, style, etc) is an important factor in choosing a product.

• It is convenient to handle a compact device than a bulky product.

6. Power Concerns:- • Power management is another important factor that needs to be considered in designing embedded systems.

• E.S should be designed in such a way as to minimize the heat dissipation by the system.

7. Single-functioned:- Dedicated to perform a single function

8. Complex functionality: - We have to run sophisticated algorithms or multiple algorithms in some applications.

9. Tightly-constrained:-
Low cost, low power, small, fast, etc
10. Safety-critical:-

- Must not endanger human life and the environment

Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=pddnCt3bUOE

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 72

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

LECTURE HANDOUTS | **L09**

ECE | **III / VI**

Course Name with Code : **Embedded Systems and RTOS & 19ECC15**

Course Faculty : **Dr.P.Padmaloshani**

Unit : **I - INTRODUCTION** **Date of Lecture:**

Topic of Lecture: Quality Attributes of Embedded Systems

Introduction :

- An Embedded system is a combination of computer hardware and software.
- As with any electronic system, this system requires a hardware platform and that is built with a microprocessor or microcontroller.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Quality Attributes of Embedded Systems:

Quality Attributes of Embedded System: Quality attributes are the non-functional requirements that need to be documented properly in any system design.

Quality attributes can be classified as

I. Operational quality attributes

II. Non-operational quality attributes.

I. Operational Quality Attributes: The operational quality attributes represent the relevant quality attributes related to the embedded system when it is in the operational mode or online mode.

Operational Quality Attributes are:

1. Response :-

It is the measure of quickness of the system.

It tells how fast the system is tracking the changes in input variables. Most of the E.S demands fast response which should be almost real time.

Ex – Flight control application.

2. Throughput :-

It deals with the efficiency of a system.

It can be defined as the rate of production or operation of a defined process over a stated period of time.

The rates can be expressed in terms of products, batches produced or any other meaningful measurements.

Ex – In case of card reader throughput means how many transactions the reader can perform in a minute or in an hour or in a day.

Throughput is generally measured in terms of "Benchmark".

A Benchmark is a reference point by which something can be measured

3. Reliability :-

• It is a measure of how much we can rely upon the proper functioning of the system.

• Mean Time Between Failure (MTBF) and Mean Time To Repair (MTTR) are the terms used in determining system reliability.
• MTBF gives the frequency of failures in hours/weeks/months.
• MTTR specifies how long the system is allowed to be out of order following a failure.
• For embedded system with critical application need, it should be of the order of minutes.

4. Maintainability:-

• It deals with support and maintenance to the end user or client in case of technical issues and product failure or on the basis of a routine system checkup.
• Reliability and maintainability are complementary to each other.
• A more reliable system means a system with less corrective maintainability requirements and vice versa.
• Maintainability can be broadly classified into two categories
1. Scheduled or Periodic maintenance (Preventive maintenance)
2. Corrective maintenance to unexpected failures

5. Security:-

• Confidentiality, Integrity and availability are the three major measures of information security.
• Confidentiality deals with protection of data and application from unauthorized disclosure.
• Integrity deals with the protection of data and application from unauthorized modification.
• Availability deals with protection of data and application from unauthorized users.

6. Safety :-

Safety deals with the possible damages that can happen to the operator, public and the environment due to the breakdown of an Embedded System.
The breakdown of an embedded system may occur due to a hardware failure or a firmware failure.
Safety analysis is a must in product engineering to evaluate the anticipated damages and determine the best course of action to bring down the consequences of damage to an acceptable level.

II. Non-Operational Quality Attributes: The quality attributes that needs to be addressed for the product not on the basis of operational aspects are grouped under this category. **1. Testability and Debug-ability:-**

I

• Testability deals with how easily one can test the design, application and by which means it can be done.
• For an E.S testability is applicable to both the embedded hardware and firmware.
• Embedded hardware testing ensures that the peripherals and total hardware functions in the desired manner, whereas firmware testing ensures that the firmware is functioning in the expected way**.**
• Debug-ability is a means of debugging the product from unexpected behavior in the system

• Debug-ability is two level process
• 1.Hardware level 2.software level
• **1. Hardware level:** It is used for finding the issues created by hardware problems.
• **2. Software level:** It is employed for finding the errors created by the flaws in the software.

2. Evolvability :-

• It is a term which is closely related to Biology.
• It is referred as the non-heritable variation.
• For an embedded system evolvability refers to the ease with which the embedded product can be modified to take advantage of new firmware or hardware technologies.

3. Portability:-

• It is the measure of system independence.
• An embedded product is said to be portable if the product is capable of functioning in various environments, target processors and embedded operating systems.
• „Porting“ represents the migration of embedded firmware written for one target processor to a different target processor.

4. Time-to-Prototype and Market:-

• It is the time elapsed between the conceptualization of a product and the time at which the product is ready for selling.

• The commercial embedded product market is highly competitive and time to market the product is critical factor in the success of commercial embedded product.

• There may be multiple players in embedded industry who develop products of the same category (like mobile phone).

5. Per Unit Cost and Revenue:-

• Cost is a factor which is closely monitored by both end user and product manufacturer.

• Cost is highly sensitive factor for commercial products

• Any failure to position the cost of a commercial product at a nominal rate may lead to the failure of the product in the market.

• Proper market study and cost benefit analysis should be carried out before taking a decision on the per-unit cost of the embedded product.

• The ultimate aim of the product is to generate marginal profit so the budget and total cost should be properly balanced to provide a marginal profit.

Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=pddnCt3bUOE

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 74

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L10

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : **Embedded Systems and RTOS & 19ECC15**

Course Faculty : **Dr.P.Padmaloshani**

Unit : **II - TYPICAL EMBEDDED SYSTEMS** **Date of Lecture:**

Topic of Lecture: Core of the Embedded System: General Purpose and Domain Specific Processors

Introduction :

- Embedded system is an Electronic/Electro mechanical system which is designed to perform a specific function and is a combination of both hardware and firmware (Software)

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Core of the Embedded System: General Purpose and Domain Specific Processors :

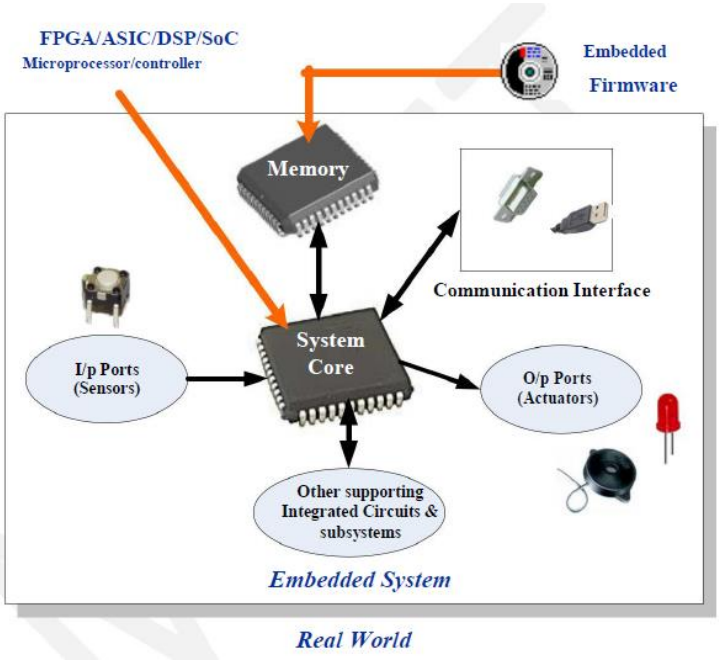
ELEMENTS OF EMBEDDED SYSTEMS:

An embedded system is a combination of 3 things, Hardware Software Mechanical Components and it is supposed to do one specific task only. A typical embedded system contains a single chip controller which acts as the master brain of the system. Diagrammatically an embedded system can be represented as follows:

Embedded systems are basically designed to regulate a physical variable (such Microwave Oven) or to manipulate the state of some devices by sending some signals to the actuators or devices connected to the output port system (such as temperature in Air Conditioner), in response to the input signal provided by the end users or sensors which are connected to the input ports. Hence the embedded systems can be viewed as a reactive system.

The control is achieved by processing the information coming from the sensors and user interfaces and controlling some actuators that regulate the physical variable.

Keyboards, push button, switches, etc. are Examples of common user interface input devices and LEDs, LCDs, Piezoelectric buzzers, etc examples for common user interface output devices for a typical embedded system.The requirement of type of user interface changes from application to application based on domain.



The Core of the Embedded Systems: The core of the embedded system falls into any one of the following categories.

General Purpose and Domain Specific Processors

- Microprocessors
- Microcontrollers
- Digital Signal Processors

Programmable Logic Devices (PLDs)

Application Specific Integrated Circuits (ASICs)

Commercial off the shelf Components (COTS)

General Purpose Processor (GPP) Vs Application Specific Instruction Set Processor (ASIP)

General Purpose Processor or GPP is a processor designed for general computational tasks

GPPs are produced in large volumes and targeting the general market. Due to the high volume production, the per unit cost for a chip is low compared to ASIC or other specific ICs

A typical general purpose processor contains an Arithmetic and Logic Unit (ALU) and Control Unit (CU)

Application Specific Instruction Set processors (ASIPs) are processors with architecture and instruction set optimized to specific domain/application requirements like Network processing, Automotive, Telecom, media applications, digital signal processing, control applications etc.

ASIPs fill the architectural spectrum between General Purpose Processors and Application Specific Integrated Circuits (ASICs)

The need for an ASIP arises when the traditional general purpose processor are unable to meet the increasing application needs

Some Microcontrollers (like Automotive AVR, USB AVR from Atmel), System on Chips, Digital Signal Processors etc are examples of Application Specific Instruction Set Processors (ASIPs)

ASIPs incorporate a processor and on-chip peripherals, demanded by the application requirement, program and data memory

Digital Signal Processors (DSPs):

Powerful special purpose 8/16/32 bit microprocessors designed specifically to meet the computational demands and power constraints of today's embedded audio, video, and communications applications
Digital Signal Processors are 2 to 3 times faster than the general purpose microprocessors in signal processing applications
DSPs implement algorithms in hardware which speeds up the execution whereas general purpose processors implement the algorithm in firmware and the speed of execution depends primarily on the clock for the processors

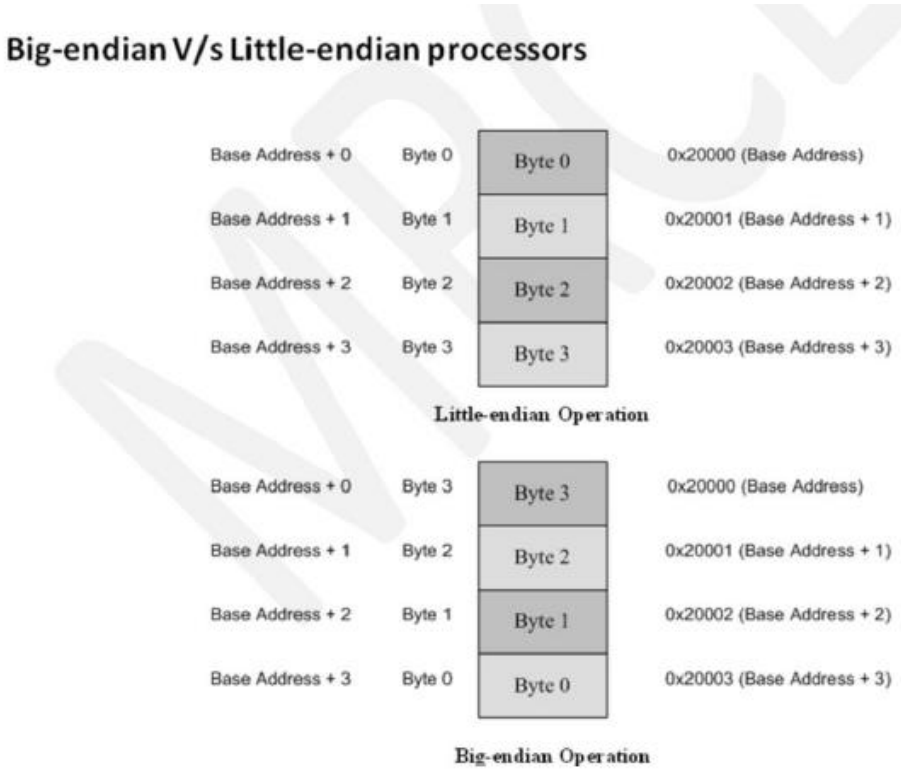
DSP can be viewed as a microchip designed for performing high speed computational operations for „addition", „subtraction", „multiplication" and „division"

Big-endian V/s Little-endian processors:

Endianness specifies the order in which the data is stored in the memory by processor operations in a multi byte system (Processors whose word size is greater than one byte). Suppose the word length is two byte then data can be stored in memory in two different ways
Higher order of data byte at the higher memory and lower order of data byte at location just below the higher memory
Lower order of data byte at the higher memory and higher order of data byte at location just below the higher memory

✓ *Little-endian* means the lower-order byte of the data is stored in memory at the lowest address, and the higher-order byte at the highest address. (The little end comes first)

✓ *Big-endian* means the higher-order byte of the data is stored in memory at the lowest address, and the lower-order byte at the highest address. (The big end comes first.)



Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=wAAPSgrpNYs

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 17

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L11

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : Embedded Systems and RTOS & 19ECC15

Course Faculty : Dr.P.Padmaloshani

Unit : II - TYPICAL EMBEDDED SYSTEMS Date of Lecture:

Topic of Lecture: ASICs

Introduction :

- Embedded system is an Electronic/Electro mechanical system which is designed to perform a specific function and is a combination of both hardware and firmware (Software)

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Application Specific Integrated Circuit (ASIC):

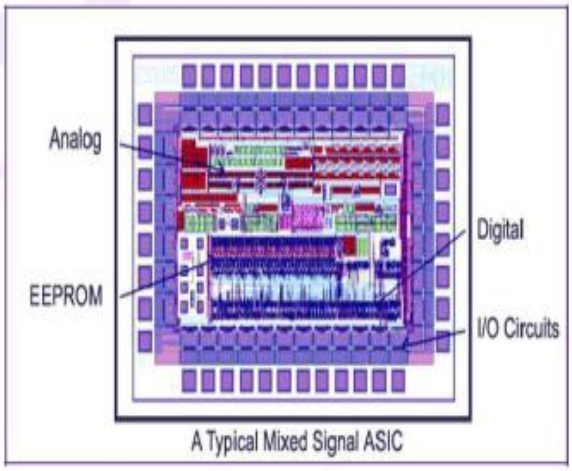
A microchip designed to perform a specific or unique application. It is used as replacement to conventional general purpose logic chips.
ASIC integrates several functions into a single chip and thereby reduces the system development cost
Most of the ASICs are proprietary products. As a single chip, ASIC consumes very small area in the total system and thereby helps in the design of smaller systems with high capabilities/functionalities.

ASICs can be pre-fabricated for a special application or it can be custom fabricated by using the components from a re-usable „*building block*" library of components for a particular customer application

Fabrication of ASICs requires a non-refundable initial investment (Non Recurring Engineering (NRE) charges) for the process technology and configuration expenses
If the Non-Recurring Engineering Charges (NRE) is born by a third party and the Application Specific Integrated Circuit (ASIC) is made openly available in the market, the ASIC is referred as Application Specific Standard Product (ASSP)

The ASSP is marketed to multiple customers just as a general-purpose product , but to a smaller number of customers since it is for a specific application.

Some ASICs are proprietary products , the developers are not interested in revealing the internal details.



Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=ccKj8kGFhUg

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 20

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L12

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : Embedded Systems and RTOS & 19ECC15

Course Faculty : Dr.P.Padmaloshani

Unit : II - TYPICAL EMBEDDED SYSTEMS Date of Lecture:

Topic of Lecture: PLDs

Introduction :

- Embedded system is an Electronic/Electro mechanical system which is designed to perform a specific function and is a combination of both hardware and firmware (Software)

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

PLDs :

Programmable Logic Devices (PLDs):

❖ Logic devices provide specific functions, including device-to-device interfacing, data communication, signal processing, data display, timing and control operations, and almost every other function a system must perform.

❖ Logic devices can be classified into two broad categories - Fixed and Programmable. The circuits in a fixed logic device are permanent, they perform one function or set of functions - once manufactured, they cannot be changed

❖ Programmable logic devices (PLDs) offer customers a wide range of logic capacity, features, speed, and voltage characteristics - and these devices can be re-configured to perform any number of functions at any time

❖ Designers can use inexpensive software tools to quickly develop, simulate, and test their logic designs in PLD based design. The design can be quickly programmed into a device, and immediately tested in a live circuit

❖ PLDs are based on re-writable memory technology and the device is reprogrammed to change the design

Field Programmable Gate Arrays (FPGAs) and Complex Programmable Logic Devices (CPLDs) are the two major types of programmable logic devices

FPGA:

FPGA is an IC designed to be configured by a designer after manufacturing.
FPGAs offer the highest amount of logic density, the most features, and the highest performance.

Logic gate is Medium to high density ranging from **1K to 500K** system gates

These advanced FPGA devices also offer features such as built-in hardwired processors (such as the IBM Power PC), substantial amounts of memory, clock management systems, and support for many of the latest, very fast device-to-device signaling technologies

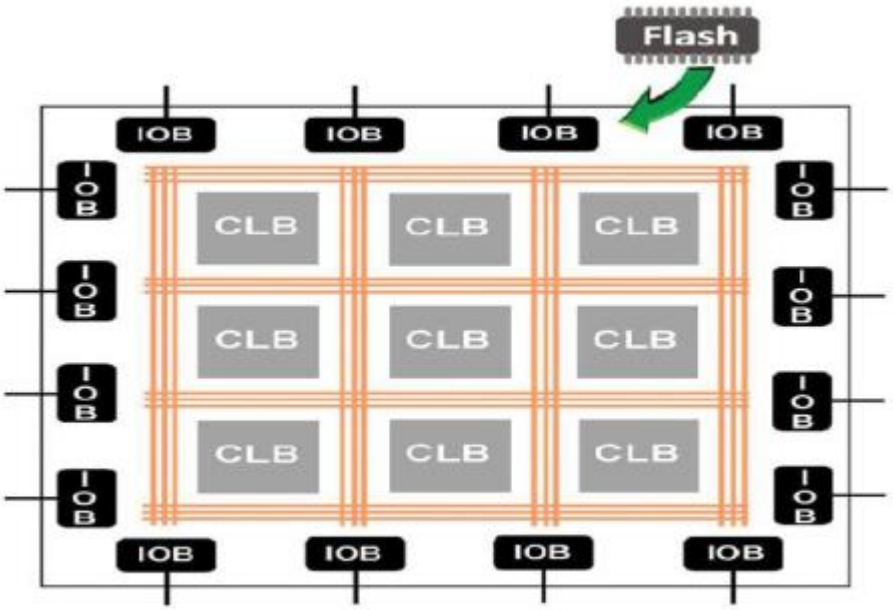


Figure: FPGA Architecture

CPLD:

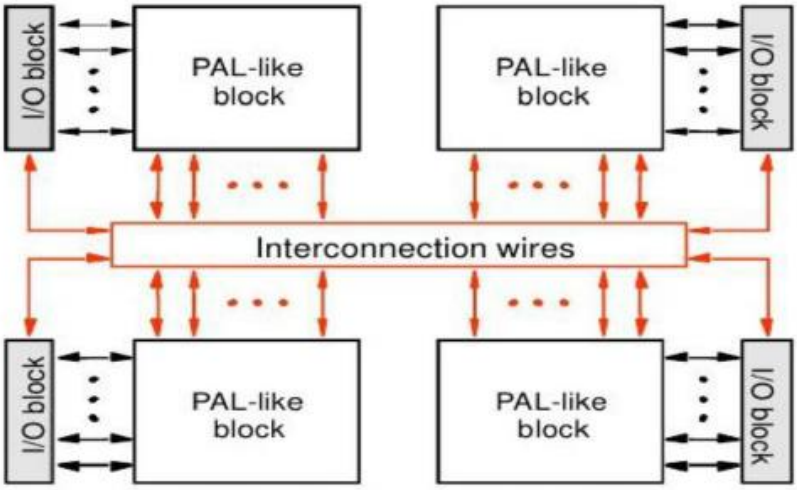
▪ A **complex programmable logic device** (**CPLD**) is a programmable logic device with complexity between that of PALs and FPGAs, and architectural features of both.

CPLDs, by contrast, offer much smaller amounts of logic - up to about 10,000 gates.

CPLDs offer very predictable timing characteristics and are therefore ideal for critical control applications

CPLDs such as the Xilinx **CoolRunner** series also require extremely low amounts of power and are very inexpensive, making them ideal for cost-sensitive, battery-operated, portable applications such as mobile phones and digital handheld assistants.

Structure of a C PLD



ADVANTAGES OF PLDs:

• PLDs offer customer much more flexibility during design cycle

• PLDSs do not require long lead times for prototype or production-the PLDs are already on a distributor"s self and ready for shipment

• PLDs do not require customers to pay for large NRE costs and purchase expensive mask sets

• PLDs allow customers to order just the number of parts required when they need them. allowing them to control inventory.

• PLDs are reprogrammable even after a piece of equipment is shipped to a customer.

• The manufacturers able to add new features or upgrade the PLD based products that are in the field by uploading new programming file

Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=z_L1VlAxZl8

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 23

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L13

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : Embedded Systems and RTOS & 19ECC15

Course Faculty : Dr.P.Padmaloshani

Unit : II - TYPICAL EMBEDDED SYSTEMS Date of Lecture:

Topic of Lecture: Commercial Off-The-Shelf Components (COTS)

Introduction :

- Embedded system is an Electronic/Electro mechanical system which is designed to perform a specific function and is a combination of both hardware and firmware (Software)

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Commercial off the Shelf Component (COTS):

A Commercial off-the-shelf (COTS) product is one which is used „*as-is*"

COTS products are designed in such a way to provide easy integration and interoperability with existing system components

Typical examples for the COTS hardware unit are Remote Controlled Toy Car control unit including the RF Circuitry part, High performance, high frequency microwave electronics (2 to 200 GHz), High bandwidth analog-to-digital converters, Devices and components for operation at very high temperatures, Electro-optic IR imaging arrays, UV/IR Detectors etc

A COTS component in turn contains a General Purpose Processor (GPP) or Application Specific Instruction Set Processor (ASIP) or Application Specific Integrated Chip (ASIC)/Application Specific Standard Product (ASSP) or Programmable Logic Device (PLD)

The major advantage of using COTS is that they are readily available in the market, cheap and a developer can cut down his/her development time to a great extend.
There is no need to design the module yourself and write the firmware .

Everything will be readily supplied by the COTs manufacturer.

The major problem faced by the end-user is that there are no operational and manufacturing standards.
The major drawback of using COTs component in embedded design is that the manufacturer may withdraw the product or discontinue the production of the COTs at any time if rapid change in technology

This problem adversely affect a commercial manufacturer of the embedded system which makes use of the specific COTs

Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=CdlnhpdsDqY

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 26

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L14

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : **Embedded Systems and RTOS & 19ECC15**

Course Faculty : **Dr.P.Padmaloshani**

Unit : **II - TYPICAL EMBEDDED SYSTEMS** **Date of Lecture:**

Topic of Lecture: Memory: ROM

Introduction :

- Embedded system is an Electronic/Electro mechanical system which is designed to perform a specific function and is a combination of both hardware and firmware (Software)

Prerequisite knowledge for Complete understanding and learning of Topic:

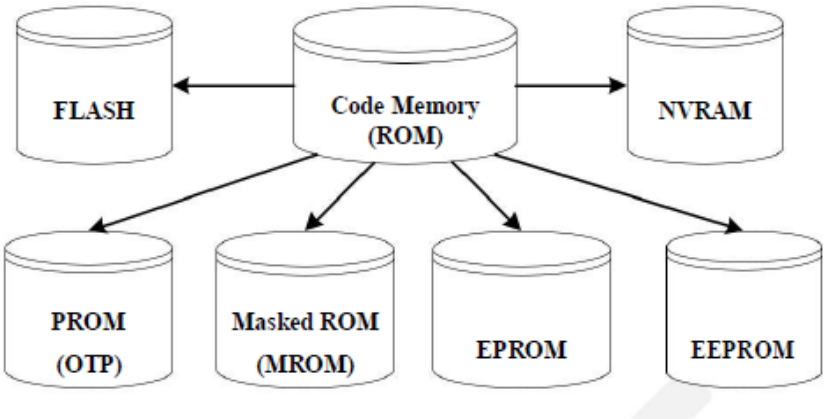
- Basic knowledge in microprocessors and microcontrollers-based system design

ROM :

Memory is an important part of an embedded system. The memory used in embedded system can be either Program Storage Memory (ROM) or Data memory (RAM)

Certain Embedded processors/controllers contain built in program memory and data memory and this memory is known as on-chip memory

Certain Embedded processors/controllers do not contain sufficient memory inside the chip and requires external memory called **off-chip memory or external memory.**



Masked ROM (MROM):
One-time programmable memory.
Uses hardwired technology for storing data.
The device is factory programmed by masking and metallization process according to the data provided by the end user.
The primary advantage of MROM is low cost for high volume production.
MROM is the least expensive type of solid state memory.
The limitation with MROM based firmware storage is the inability to modify the device firmware against firmware upgrades.
The MROM is permanent in bit storage, it is not possible to alter the bit information

Programmable Read Only Memory (PROM) / (OTP) :
It is not pre-programmed by the manufacturer
The end user is responsible for Programming these devices.
PROM/OTP has *nichrome* or *polysilicon* wires arranged in a matrix, these wires can be functionally viewed as fuses.
It is programmed by a PROM programmer which selectively burns the fuses according to the bit pattern to be stored.
Fuses which are not blown/burned represents a logic "1" where as fuses which are blown/burned represents a logic "0".The default state is logic "1".
OTP is widely used for commercial production of embedded systems whose proto-typed versions are proven and the code is finalized.
It is a low cost solution for commercial production.
OTPs cannot be reprogrammed.

3. Erasable Programmable Read Only Memory (EPROM):
Erasable Programmable Read Only (EPROM) memory gives the flexibility to re-program the same chip.
During development phase , code is subject to continuous changes and using an OTP is not economical.
EPROM stores the bit information by charging the floating gate of an FET
Bit information is stored by using an EPROM Programmer, which applies high voltage to charge the floating gate
EPROM contains a quartz crystal window for erasing the stored information. If the window is exposed to Ultra violet rays for a fixed duration, the entire memory will be erased
Even though the EPROM chip is flexible in terms of re-programmability, it needs to be taken out of the circuit board and needs to be put in a UV eraser device for 20 to 30 minutes

4. Electrically Erasable Programmable Read Only Memory (EEPROM):
Erasable Programmable Read Only (EPROM) memory gives the flexibility to re-program the same chip using electrical signals
The information contained in the EEPROM memory can be altered by using electrical signals at the register/Byte level
They can be erased and reprogrammed within the circuit
These chips include a chip erase mode and in this mode they can be erased in a few milliseconds
It provides greater flexibility for system design
The only limitation is their capacity is limited when compared with the standard ROM (A few kilobytes).

5. Program Storage Memory – FLASH
FLASH memory is a variation of EEPROM technology.
FALSH is the latest ROM technology and is the most popular ROM technology used in today"s embedded designs
It combines the re-programmability of EEPROM and the high capacity of standard ROMs
FLASH memory is organized as sectors (blocks) or pages
FLASH memory stores information in an array of floating gate MOSFET transistors
The erasing of memory can be done at sector level or page level without affecting the other sectors or pages
Each sector/page should be erased before re-programming

The typical erasable capacity of FLASH is of the order of a few 1000 cycles.
Video Content / Details of website for further learning (if any): https://www.youtube.com/watch?v=1z0Gh2bFxeE
Important Books/Journals for further learning including the page nos.: Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 28

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L15

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : Embedded Systems and RTOS & 19ECC15

Course Faculty : Dr.P.Padmaloshani

Unit : II - TYPICAL EMBEDDED SYSTEMS Date of Lecture:

Topic of Lecture: RAM

Introduction :

- Embedded system is an Electronic/Electro mechanical system which is designed to perform a specific function and is a combination of both hardware and firmware (Software)

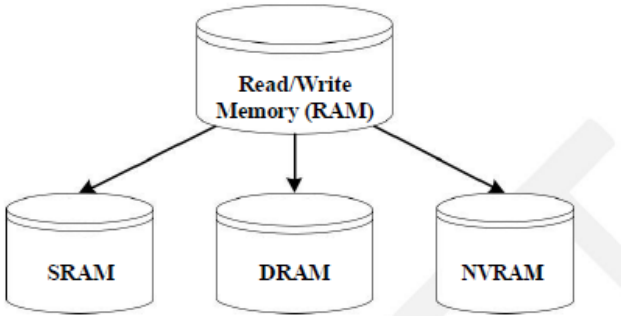
Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

RAM **:**

RAM is the data memory or working memory of the controller/processor

RAM is volatile, meaning when the power is turned off, all the contents are destroyed
RAM is a direct access memory, meaning we can access the desired memory location directly without the need for traversing through the entire memory locations to reach the desired memory position (i.e. Random Access of memory location)



1. Static RAM (SRAM):

Static RAM stores data in the form of Voltage

They are made up of flip-flops

In typical implementation, an SRAM cell (bit) is realized using 6 transistors (or 6 MOSFETs
Four of the transistors are used for building the latch (flip-flop) part of the memory cell and 2 for controlling the access.

Static RAM is the fastest form of RAM available

SRAM is fast in operation due to its resistive networking and switching capabilities

2. Dynamic RAM (DRAM)

Dynamic RAM stores data in the form of charge. They are made up of MOS transistor gates

The disadvantage is that since the information is stored as charge it gets leaked off with time and to prevent this they need to be refreshed periodically

Special circuits called DRAM controllers are used for the refreshing operation. The refresh operation is done periodically in milliseconds interval

SRAM Vs DRAM:

SRAM Cell	DRAM Cell
Made up of 6 CMOS transistors (MOSFET)	Made up of a MOSFET and a capacitor
Doesn"t Require refreshing	Requires refreshing
Low capacity (Less dense)	High Capacity (Highly dense)
More expensive	Less Expensive
Fast in operation. Typical access time is 10ns	Slow in operation due to refresh requirements. Typical access time is 60ns. Write operation is faster than read operation.

Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=1z0Gh2bFxeE

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 30

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L16

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : Embedded Systems and RTOS & 19ECC15

Course Faculty : Dr.P.Padmaloshani

Unit : II - TYPICAL EMBEDDED SYSTEMS Date of Lecture:

Topic of Lecture: Memory according to the type of Interface

Introduction :

- Embedded system is an Electronic/Electro mechanical system which is designed to perform a specific function and is a combination of both hardware and firmware (Software)

Prerequisite knowledge for Complete understanding and learning of Topic:

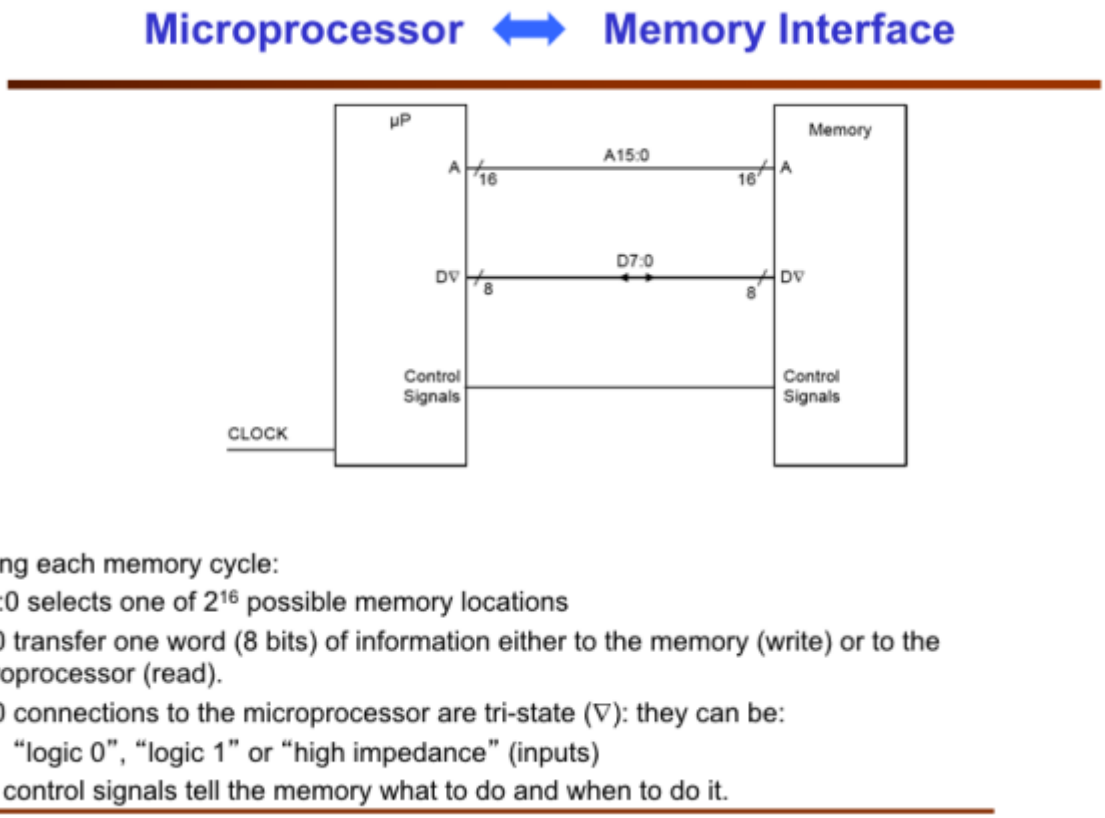
- Basic knowledge in microprocessors and microcontrollers-based system design

Memory according to the type of Interface **:**

Memory interfacing is an essential topic for digital system design. In fact the among silicon area devoted to memory in a typical digital embedded system or a computer system is substantial. For example, in a mobile phone, the number of transistors devoted to memory is many times more than those used for computation. For the second year course, I will only focus on interfacing to static memory, known as RAM (Random Access Memory) or ROM (Read-Only Memory). There are other types of memory such as dynamic memory (DRAM), Synchronous DRAM (SDRAM) and flash memory (Flash RAM) which will not be covered on this course.

For a 8k x 8 RAM, there are 8 data bits, and therefore 8 separate 1-bit arrays. Let us assume that each data bit array is organised as a 256 rows x 32 column (=8192) of memory cells. Eight such array are placed next to each other to form the 8 data bits required. This makes the memory chip roughly square (which is generally desirable). You can think of the row decoder and the column selector driven by the 13-bit address as a 8192 way multiplexer, selecting one of 8192 cells organised as 256 x 32, to be accessed. The simplified circuit of each memory cell shown here consists of two inverters and two switches is a schematic of the read-write circuit. When reading from the cell, A12:0 select one of 8192 cells to route its signal via the right inverter to Dn. Now Dn is an output pin. This only happens if CE*OE* !WR = 1 (i.e. asserting CE and OE, but not asserting WR). When writing to the memory cell, the right switch is open, Dn is an input pin driving the left hand inverter and the output switch from that

inverter is closed because both CE and WR are asserted. Some memory chips have separate Din and Dout pins, but that's expensive on pins and is not particularly common nowadays.



Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=-S911PHuQV0

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 33

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L17

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : Embedded Systems and RTOS & 19ECC15

Course Faculty : Dr.P.Padmaloshani

Unit : II - TYPICAL EMBEDDED SYSTEMS Date of Lecture:

Topic of Lecture: Memory Shadowing, Memory selection for Embedded Systems

Introduction :

- Embedded system is an Electronic/Electro mechanical system which is designed to perform a specific function and is a combination of both hardware and firmware (Software)

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Memory Shadowing, Memory selection for Embedded Systems **:**

Memory Shadowing

Shadow memory is **a method used to keep track of the memory utilized during the execution of a program in your computer**. Shadow memory consists of shadow bytes that help you guide through singular bits in the fundamental/main memory.

This is a technique that is probably best known from its implementation with the BIOS ROMs used in a PC. The idea behind shadowing is to copy the contents of the slow ROM into faster RAM and execute the code from the RAM. As a result the time taken to execute the code is greatly reduced. The shadowing refers to the fact that the RAM contains a copy of the original ROM contents.

This mechanism can be implemented either with hardware assist or entirely in software. The basic principles behind the shadowing mechanism are as follows:

Typically the ROM contains the start up code as well as the system software. When the CPU is reset it will start executing this start-up code. As part of the initialisation, the contents of the ROM is copied into the RAM area where it can be executed. This part is common to both implementations.

Memory Selection

Selection of suitable memory is very much essential step in high performance applications, because the challenges and limitations of the system performance are often decided upon the type of memory architecture.

Systems memory requirement depend primarily on the nature of the application that is planned to run on the system. Memory performance and capacity requirement for low cost systems are small, whereas memory throughput can be the most critical requirement in a complex, high performance system.

Following are the factors that are to be considered while selecting the memory devices,

- Speed
- Data storage size and capacity
- Bus width
- Latency
- Power consumption
- Cost

SRAM's have lower data storage and capacity hence they are suitable for lower end systems where as SDRAM for higher end systems with complex requirements.

Among the high speed types of SDRAM, DDR2 memory modules can have memory capacities from 256MB to 4GB capacities. Most of the DDR2 memory chips come in FBGA (Fine Ball Grid Array) package. The package allows higher memory densities in smaller space with better electrical properties. DDR2 memory uses 1.8V for power, resulting in lower power and cooler operation, whereas the DDR uses 2.5V.

Another factor, when going for non-volatile programmable storage, is deciding the programming model. For example, it could be ISP (In-System Programming) that allows programming the flash but needs the application to be stopped at that time. Or it could be IAP (In-Application Programming) that will allow re-programming of the memory even when the application firmware is running. This is determined by the memory architecture. Nowadays many microcontrollers support both the options and ISP is used for manufacturing and IAP is appropriate for field updates.

Video Content / Details of website for further learning (if any):
https://jillian-greenberg.com/qa/why-memory-shadowing-technique-is-adopted.html

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 34

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L18

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : Embedded Systems and RTOS & 19ECC15

Course Faculty : Dr.P.Padmaloshani

Unit : II - TYPICAL EMBEDDED SYSTEMS Date of Lecture:

Topic of Lecture: Sensors and Actuators

Introduction :

- Embedded system is an Electronic/Electro mechanical system which is designed to perform a specific function and is a combination of both hardware and firmware (Software)

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Sensors & Actuators:

• Embedded system is in constant interaction with the real world

• Controlling/monitoring functions executed by the embedded system is achieved in accordance with the changes happening to the Real World.

• The changes in the system environment or variables are detected by the sensors connected to the input port of the embedded system.

• If the embedded system is designed for any controlling purpose, the system will produce some changes in controlling variable to bring the controlled variable to the desired value.

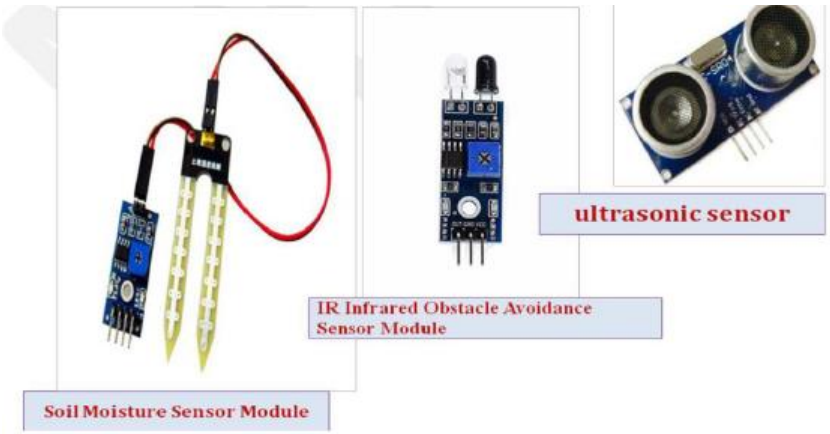
• It is achieved through an actuator connected to the out port of the embedded system.

Sensor:

A transducer device which converts energy from one form to another for any measurement or control purpose. Sensors acts as input device

Eg. Hall Effect Sensor which measures the distance between the cushion and magnet in the Smart Running shoes from adidas

Example: IR, humidity , PIR(passive infra red) , ultrasonic , piezoelectric , smoke sensors



Actuator:
A form of transducer device (mechanical or electrical) which converts signals to corresponding physical action (motion). Actuator acts as an output device

Eg. Micro motor actuator which adjusts the position of the cushioning element in the Smart Running shoes from adidas

Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=F0cNJjfVe-U

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 35

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L19

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : **Embedded Systems and RTOS & 19ECC15**

Course Faculty : **Dr.P.Padmaloshani**

Unit : **III - EMBEDDED FIRMWARE** **Date of Lecture:**

Topic of Lecture: Reset Circuit

Introduction :

- Embedded system is an Electronic/Electro mechanical system which is designed to perform a specific function and is a combination of both hardware and firmware (Software)

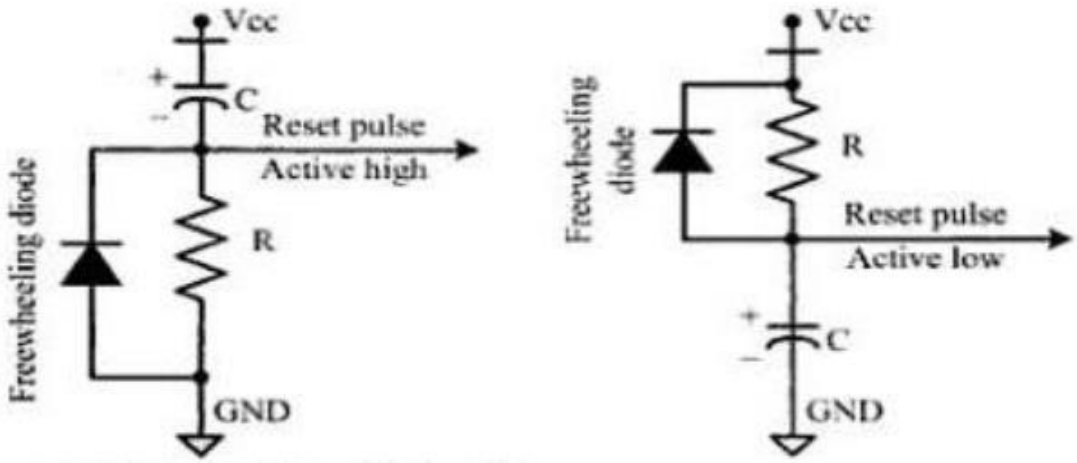
Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Reset Circuit **:**

The reset circuit is essential to ensure that the device is not operating at a voltage level where the device is not guaranteed to operate, during system power ON. The reset signal brings the internal registers and the different hardware systems of the processor/controller to a known state and starts the firmware execution from the reset vector (Normally from vector address 0x0000 for conventional processors/controllers. The reset vector can be relocated to an address for processors/controllers supporting bootloader). The reset signal can be either active high (The processor undergoes reset when the reset pin of the processor is at logic high) or active low (The processor undergoes reset when the reset pin of the processor is at logic low). Since the processor operation is synchronised to a clock signal, the reset pulse should be wide enough to give time for the clock oscillator to stabilise before the internal reset state starts. The reset signal to the processor can be applied at power ON through an external passive reset circuit comprising a Capacitor and Resistor or through a standard Reset IC like MAX810 from Maxim Dallas (www.maxim-ic.com). Select the reset IC based on the type of reset signal and logic level (CMOS/TTL) supported by the processor/controller in use. Some microprocessors/controllers contain built-in internal

reset circuitry and they don't require external reset circuitry. Figure 2.35 illustrates a resistor capacitor based passive reset circuit for active high and low configurations. The reset pulse width can be adjusted by changing the resistance value *R* and capacitance value *C*.



Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=BVwvZAk5nlg

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 60

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L20

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : **Embedded Systems and RTOS & 19ECC15**

Course Faculty : **Dr.P.Padmaloshani**

Unit : **III - EMBEDDED FIRMWARE** **Date of Lecture:**

Topic of Lecture: Brown-out Protection Circuit

Introduction :

- Embedded system is an Electronic/Electro mechanical system which is designed to perform a specific function and is a combination of both hardware and firmware (Software)

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Brown-out Protection Circuit **:**

Brown-out protection circuit prevents the processor/controller from unexpected program execution behaviour when the supply voltage to the processor/controller falls below a specified voltage. It is essential for battery powered devices since there are greater chances for the battery voltage to drop below the required threshold. The processor behaviour may not be predictable if the supply voltage falls below the recommended operating voltage. It may lead to situations like data corruption. A brown-out protection circuit holds the processor/controller in reset state, when the operating voltage falls below the threshold, until it rises above the threshold voltage. Certain processors/controllers support built in brown-out protection circuit which monitors the supply voltage internally. If the processor/controller doesn't integrate a built-in brown-out protection circuit, the same can be implemented using external passive circuits or supervisor ICs. Figure 2.36 illustrates a brown-out circuit implementation using Zener diode and transistor for processor/controller with active low Reset logic.

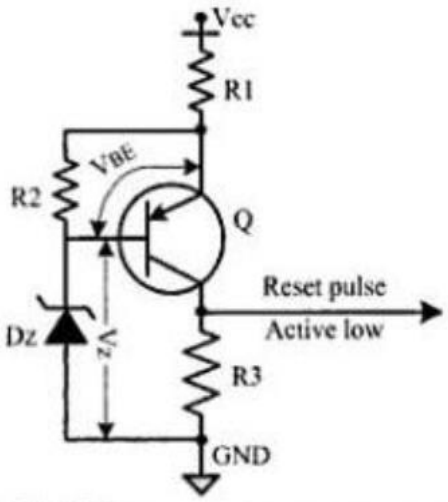


Fig. 2.36 **Brown-out protection circuit with Active low output**

The Zener diode Dz and transistor Q forms the heart of this circuit. The transistor conducts always when the supply voltage V_{cc} is greater than that of the sum of V_{BE} and V_z (Zener voltage). The transistor stops conducting when the supply voltage falls below the sum of V_{BE} and V_Z. Select the Zener diode with required voltage for setting the low threshold value for Vcc. The values of R1, R2, and R3 can be selected based on the electrical characteristics (Absolute maximum current and voltage ratings) of the transistor in use. Microprocessor Supervisor ICs like DS1232 from Maxim Dallas (www.maxim-ic.com) also provides Brown-out protection.

Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=2JzyjMxI__I

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 64

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L21

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : **Embedded Systems and RTOS & 19ECC15**

Course Faculty : **Dr.P.Padmaloshani**

Unit : **III - EMBEDDED FIRMWARE** **Date of Lecture:**

Topic of Lecture: Oscillator Unit

Introduction :

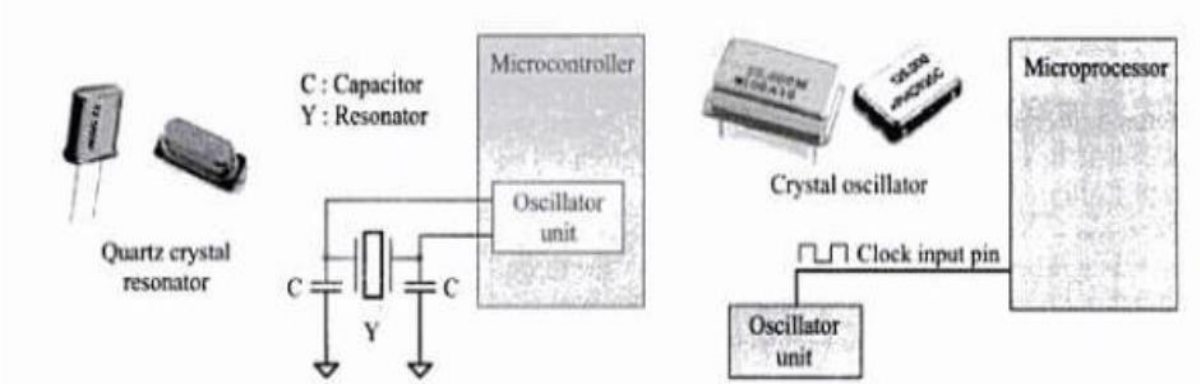
- Embedded system is an Electronic/Electro mechanical system which is designed to perform a specific function and is a combination of both hardware and firmware (Software)

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Oscillator Unit **:**

A microprocessor/microcontroller is a digital device made up of digital combinational and sequential circuits. The instruction execution of a microprocessor/controller occurs in sync with a clock signal. It is analogous to the heartbeat of a living being which synchronises the execution of life. For a living being, the heart is responsible for the generation of the beat whereas the oscillator unit of the embedded system is responsible for generating the precise clock for the processor. Certain processors/controllers integrate a built-in oscillator unit and simply require an external ceramic resonator/quartz crystal for producing the necessary clock signals. Quartz crystals and ceramic resonators are equivalent in operation, however they possess physical difference. A quartz crystal is normally mounted in a hermetically sealed metal case with two leads protruding out of the case. Certain devices may not contain a built-in oscillator unit and require the clock pulses to be generated and supplied externally. Quartz crystal Oscillators are available in the form chips and they can be used for generating the clock pulses in such a cases. The speed of operation of a processor is primarily dependent on the clock frequency. However we cannot increase the clock frequency blindly for increasing the speed of execution. The logical circuits lying inside the processor always have an upper threshold value for the maximum clock at which the system can run, beyond which the system becomes unstable and non functional. The total system power consumption is directly proportional to the clock frequency. The power consumption increases with increase in clock frequency. The accuracy of program execution depends on the accuracy of the clock signal. The accuracy of the crystal oscillator or ceramic resonator is normally expressed in terms of +/-ppm (Parts per million). Figure 2.37 illustrates the usage of quartz crystal/ceramic resonator and external oscillator chip for clock generation.



Oscillator circuitry using quartz crystal and quartz crystal oscillator

Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=dVxssvcIPrY

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 94

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L22

LECTURE HANDOUTS

ECE **III / VI**

Course Name with Code : **Embedded Systems and RTOS & 19ECC15**

Course Faculty : **Dr.P.Padmaloshani**

Unit : **III - EMBEDDED FIRMWARE** **Date of Lecture:**

Topic of Lecture: Real Time Clock

Introduction :

- Real-time clocks (RTC) are timers dedicated to maintaining a one-second timebase. In addition, an RTC is often used to keep track of clock time and calendar date either in software or hardware. Many of the features of an RTC are very specialized and required for maintaining high accuracy and very reliable operation. There are RTC devices external to a microcontroller which interface with an I²C or SPI bus.

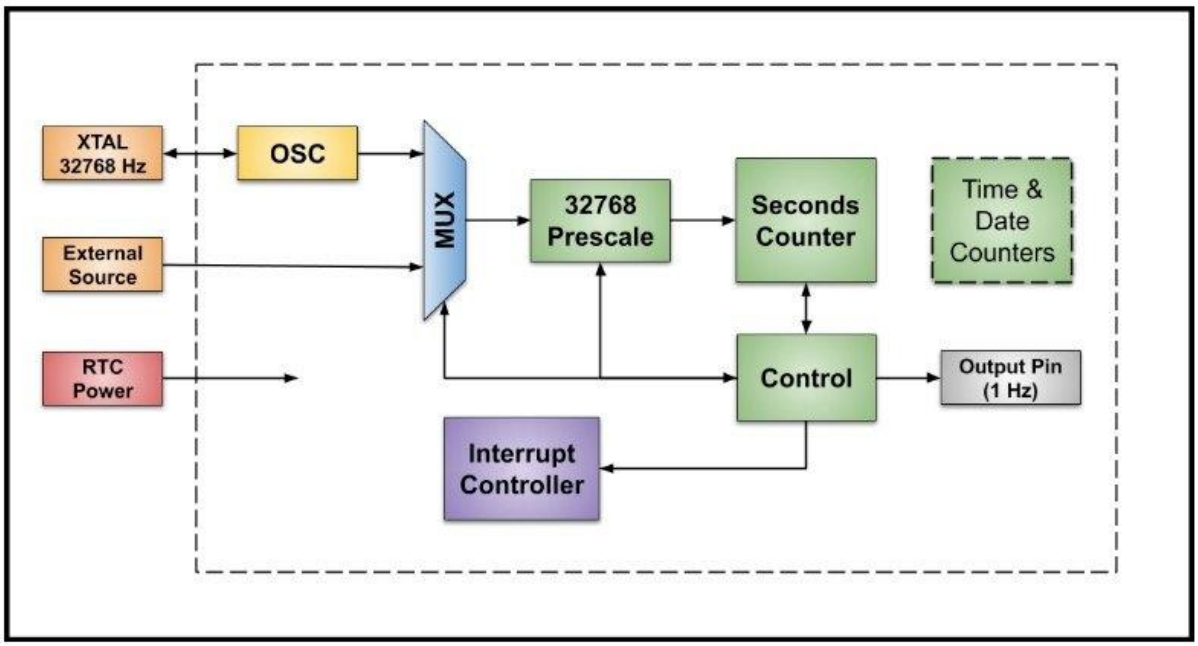
Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Real Time Clock **:**

Real-Time Clock (RTC) is a system component responsible for keeping track of time. RTC holds information like current time (In hours, minutes and seconds) in 12 hour/24 hour format, date, month, year, day of the week, etc. and supplies timing reference to the system. RTC is intended to function even in the absence of power. RTCs are available in the form of Integrated Circuits from different semiconductor manufacturers like Maxim/Dallas, ST Microelectronics etc. The RTC chip contains a microchip for holding the time and date related information and backup battery cell for functioning in the absence of power, in a single IC package. The RTC chip is interfaced to the processor or controller of the embedded system. For Operating System based embedded devices, a timing reference is essential for synchronising the operations of the OS kernel. The RTC can interrupt the OS kernel by asserting the interrupt line of the processor/controller to which the RTC interrupt line is connected. The OS kernel identifies the interrupt in terms of the Interrupt Request (IRQ) number generated by an interrupt controller. One IRQ can be assigned to the RTC interrupt and the kernel can perform necessary operations like system date

time updation, managing software timers etc when an RTC timer tick interrupt occurs. The RTC can be configured to interrupt the processor at predefined intervals or to interrupt the processor when the RTC register reaches a specified value (used as alarm interrupt).



Real time clock Hardware features

Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=IO0HZ6IRKrA

Important Books/Journals for further learning including the page nos.:
Rajkamal, Embedded Systems: Architecture, Programming and Design, Tata McGraw Hill, 2015, pp. 158

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L23

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : Embedded Systems and RTOS & 19ECC15

Course Faculty : Dr.P.Padmaloshani

Unit : III - EMBEDDED FIRMWARE Date of Lecture:

Topic of Lecture: Watchdog Timer

Introduction :

- **Watchdog timer is a piece of hardware in micro-controller**. **Watchdog timer is used to generates system reset if system gets stuck somewhere** i.e. if system goes into endless loop of execution watchdog timer will reset the system to come out of endless loop. Watchdog is safety mechanism in embedded system which makes your system reliable, but **it depends on how you make use of watchdog timer.**

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

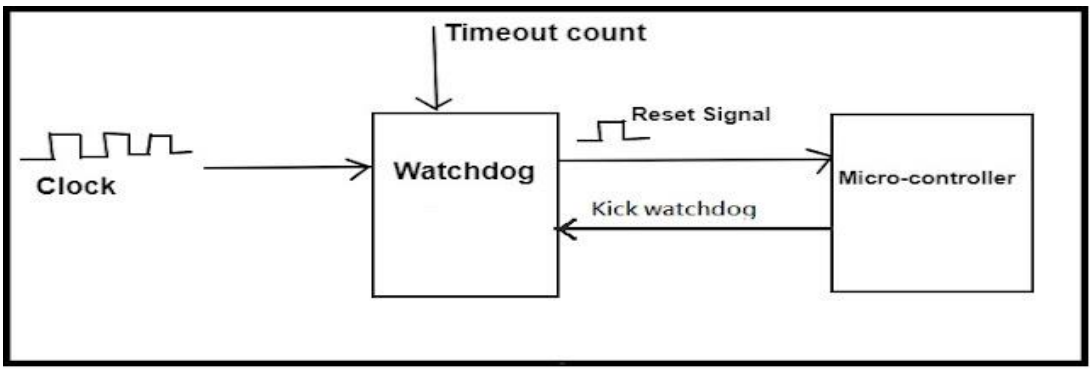
Watchdog Timer :

Watchdog timer is a piece of hardware in micro-controller. Watchdog timer is used to generates system reset if system gets stuck somewhere i.e. if system goes into endless loop of execution watchdog timer will reset the system to come out of endless loop. Watchdog is safety mechanism in embedded system which makes your system reliable, but it depends on how you make use of watchdog timer. How does watchdog works :

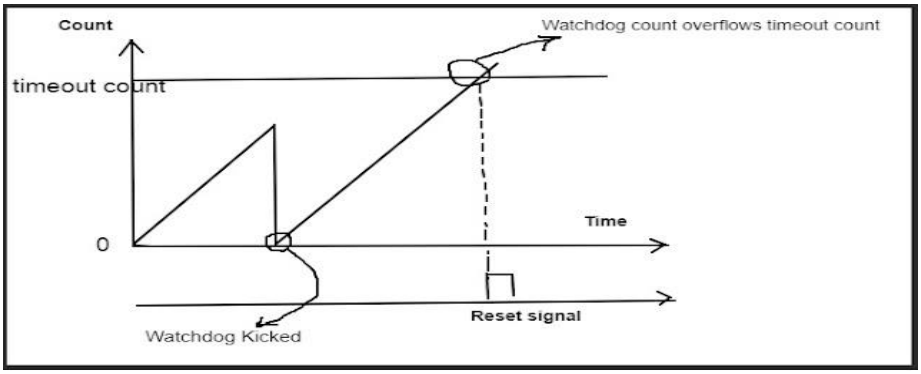
Watchdog is basically a counter, which starts from counting zero and reaches to a certain value. If counter reaches to certain value then watchdog hardware will generates a watchdog reset. To avoid system reset, software needs to kick the watchdog i.e. need to reset the counter to zero. In case software stuck into endless loop it system will not able to kick the watchdog hence counter reaches to certain value and resets the system.

Watchdog is initially loaded with certain value. This value is calculated based on timeout time of watchdog (Further section it is been shown how to calculate counter value based on timeout value). Before timeout time, system should reset the counter.

e.g. If your system if performing 3 tasks periodically and to perform 3 tasks is takes 500 ms. Then timeout time is considered as 600 ms (considering worst case scenario), counter value is calculated with respect to 600 ms and loaded into watchdog.
Following figures shows watchdog hardware. Input to watchdog hardware is clock. Based on every clock tick watchdog internal counter increments. Then there is comparator which compares count value with loaded count value (timeout value) and if count matches watchdog hardware generates and reset signal.



Watchdog timer



Watchdog timer working

Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=jeWLei8C0S0

Important Books/Journals for further learning including the page nos.:
Rajkamal, Embedded Systems: Architecture, Programming and Design, Tata McGraw Hill, 2015, pp. 157

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L24

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : Embedded Systems and RTOS & 19ECC15

Course Faculty : Dr.P.Padmaloshani

Unit : III - EMBEDDED FIRMWARE Date of Lecture:

Topic of Lecture: Embedded Firmware Design Approaches

Introduction :

The embedded firmware can be developed in various methods like

o Write the program in high level languages like Embedded C/C++ using an Integrated Development Environment (The IDE will contain an editor, compiler, linker, debugger, simulator etc. IDEs are different for different family of processors/controllers.

- o Write the program in Assembly Language using the Instructions Supported by your application's target processor/controller

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Embedded Firmware Design Approaches **:**

The embedded firmware is responsible for controlling the various peripherals of the embedded hardware and generating response in accordance with the functional requirements of the product.

The embedded firmware is the master brain of the embedded system. The embedded firmware imparts intelligence to an Embedded system. It is a onetime process and it can happen at any stage.

The product starts functioning properly once the intelligence imparted to the product by embedding the firmware in the hardware.

The firmware design approaches for embedded product is purely dependent on the complexity of the functions to be performed and speed of operation required.

There exist two basic approaches for the design and implementation of embedded firmware, namely;

- **The Super loop based approach**
- **The Embedded Operating System based approach**

1. Embedded firmware Design Approaches – The Super loop:

The Super loop based firmware development approach is Suitable for applications that are not time critical and where the response time is not so important (Embedded systems where missing deadlines are acceptable).

It is very similar to a conventional procedural programming where the code is executed task by task

The tasks are executed in a never ending loop.
The task listed on top on the program code is executed first and the tasks just below the top are executed after completing the first task
A typical super loop implementation will look like:
1. Configure the common parameters and perform initialization for various hardware components memory, registers etc.
2. Start the first task and execute it
3. Execute the second task
4. Execute the next task 5. :

6. :
7. Execute the last defined task
8. Jump back to the first task and follow the same flow.
Pros:
Cons:
Doesn't require an Operating System for task scheduling and monitoring and free from OS related overheads
Simple and straight forward design Reduced memory footprint
Cons:
Non Real time in execution behavior (As the number of tasks increases the frequency at which a task gets CPU time for execution also increases)
Any issues in any task execution may affect the functioning of the product (This can be effectively tackled by using Watch Dog Timers for task execution monitoring)
Enhancements:
 Combine Super loop based technique with interrupts
 Execute the tasks (like keyboard handling) which require Real time attention as Interrupt Service routines

Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=-HL-VnLnmIE

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 303

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L25

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : Embedded Systems and RTOS & 19ECC15

Course Faculty : Dr.P.Padmaloshani

Unit : III - EMBEDDED FIRMWARE Date of Lecture:

Topic of Lecture: Embedded Firmware Design Approaches

Introduction :

The embedded firmware can be developed in various methods like

o Write the program in high level languages like Embedded C/C++ using an Integrated Development Environment (The IDE will contain an editor, compiler, linker, debugger, simulator etc. IDEs are different for different family of processors/controllers.

o Write the program in Assembly Language using the Instructions Supported by your application's target processor/controller

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Embedded Firmware Design Approaches:

Embedded firmware Design Approaches – Embedded OS based Approach:

The embedded device contains an Embedded Operating System which can be one of:

- **A Real Time Operating System (RTOS)**
- **A Customized General Purpose Operating System (GPOS)**

The Embedded OS is responsible for scheduling the execution of user tasks and the allocation of system resources among multiple tasks

It Involves lot of OS related overheads apart from managing and executing user defined tasks

Microsoft® Windows XP Embedded is an example of GPOS for embedded devices

Point of Sale (PoS) terminals, Gaming Stations, Tablet PCs etc are examples of embedded devices running on embedded GPOSs

'*Windows CE*', '*Windows Mobile*','*QNX*', '*VxWorks*', '*ThreadX*', '*MicroC/OS-II*', '*Embedded Linux*', '*Symbian*' etc are examples of RTOSs employed in Embedded Product development

Mobile Phones, PDAs, Flight Control Systems etc are examples of embedded devices that runs on RTOSs The embedded device contains an Embedded Operating System which can be one of: ¬A Real Time Operating System (RTOS) ¬A Customized General Purpose Operating System (GPOS) ¬The Embedded OS is responsible for scheduling the execution of user tasks and the allocation of system resources among multiple tasks ¬It Involves lot of OS related overheads apart from managing and executing user defined tasks ¬Microsoft® Windows XP Embedded is an example of GPOS for embedded device
Video Content / Details of website for further learning (if any): https://www.youtube.com/watch?v=-HL-VnLnmIE
Important Books/Journals for further learning including the page nos.: Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 303

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L26

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : Embedded Systems and RTOS & 19ECC15

Course Faculty : Dr.P.Padmaloshani

Unit : III - EMBEDDED FIRMWARE Date of Lecture:

Topic of Lecture: Development Languages

Introduction :

- Embedded system is an Electronic/Electro mechanical system which is designed to perform a specific function and is a combination of both hardware and firmware (Software)

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Development Languages **:**

Embedded firmware Development Languages/Options

Assembly Language High Level Language

- Subset of C (Embedded C)
- Subset of C++ (Embedded C++)
- Any other high level language with supported Cross-compiler

Mix of Assembly & High level Language

o Mixing High Level Language (Like C) with Assembly Code

o Mixing Assembly code with High Level Language (Like C)

o Inline Assembly

Embedded firmware Development Languages/Options – Assembly Language

'***Assembly Language***' is the human readable notation of '***machine language***'

'M***achine language***' is a processor understandable language

Machine language is a binary representation and it consists of 1s and 0s Assembly language and machine languages are processor/controller dependent
An Assembly language program written for one processor/controller family will not work with others
Assembly language programming is the process of writing processor specific machine code in mnemonic form, converting the mnemonics into actual processor instructions (machine language) and associated data using an assembler
The general format of an assembly language instruction is an Opcode followed by Operands
The Opcode tells the processor/controller what to do and the Operands provide the data and information required to perform the action specified by the opcode

It is not necessary that all opcode should have Operands following them. Some of the Opcode implicitly contains the operand and in such situation no operand is required. The operand may be a single operand, dual operand or more

Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=JHbSVK8divM

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 306

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L27

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : Embedded Systems and RTOS & 19ECC15

Course Faculty : Dr.P.Padmaloshani

Unit : III - EMBEDDED FIRMWARE Date of Lecture:

Topic of Lecture: Development Languages

Introduction :

- Embedded system is an Electronic/Electro mechanical system which is designed to perform a specific function and is a combination of both hardware and firmware (Software)

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Development Languages **:**

2. Assembly Language – Source File to Hex File Translation:

The Assembly language program written in assembly code is saved as *.asm* (Assembly file) file or a *.src* (source) file or a format supported by the assembler Similar to 'C' and other high level language programming, it is possible to have multiple source files called modules in assembly language programming. Each module is represented by a '***.asm***' or '***.src***' file or the assembler supported file format similar to the '***.c***' files in C programming

The software utility called 'Assembler' performs the translation of assembly code to machine code

The assemblers for different family of target machines are different. A51 Macro Assembler from Keil software is a popular assembler for the 8051 family micro controller

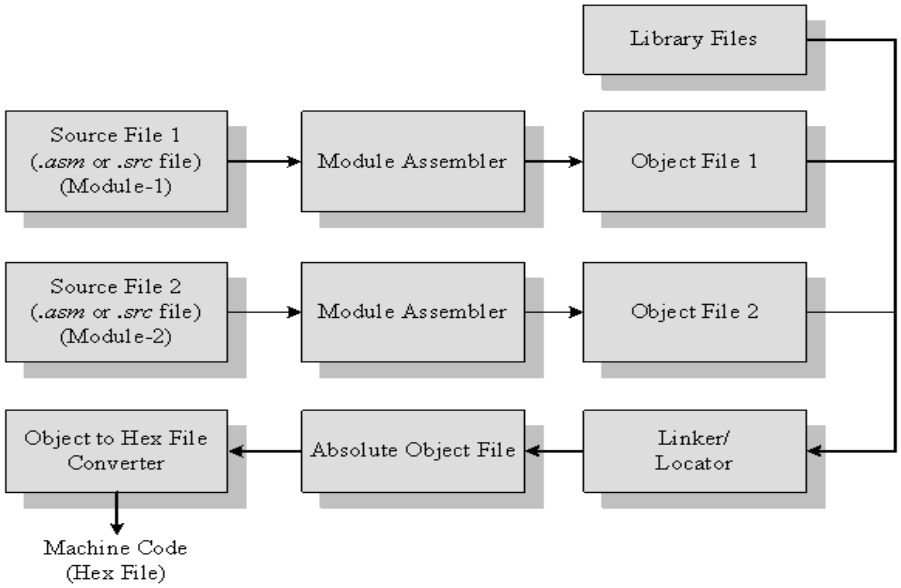


Figure 5: Assembly Language to machine language conversion process

Advantages:
1.Efficient Code Memory & Data Memory Usage (Memory Optimization
2.High Performance
3.Low level Hardware Access:
4.Code Reverse Engineering
Drawbacks:

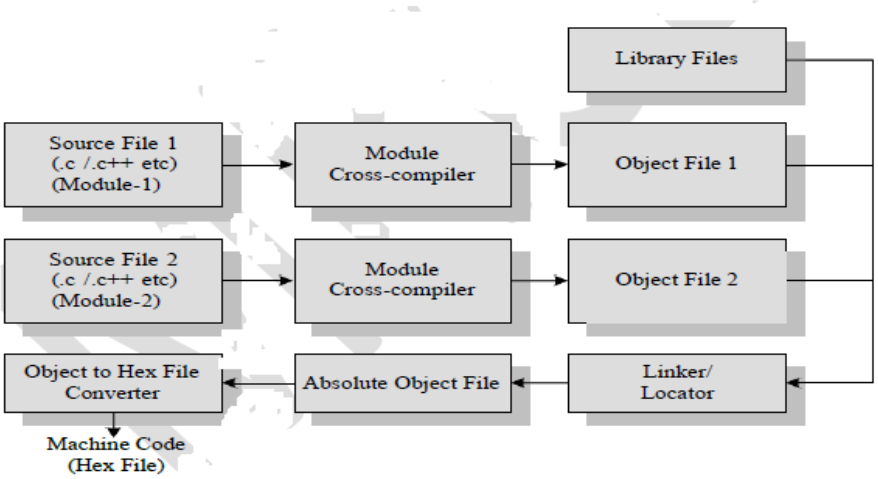
1.High Development time
2.Developer dependency
3.Non portable

2. **Embedded firmware Development Languages/Options – High Level Language**
The embedded firmware is written in any high level language like C, C++
A software utility called 'cross-compiler' converts the high level language to target processor specific machine code
Advantages:
Reduced Development time
Developer independency
Portability

Drawbacks:
The cross compilers may not be efficient in generating the optimized target processor specific instructions.

• Target images created by such compilers may be messy and non- optimized in terms of performance as well as code size.

• The investment required for high level language based development tools (IDE) is high compared to Assembly Language based firmware development tools.



High level language to machine language conversion process

Embedded firmware Development Languages/Options – Mixing of Assembly Language with High Level Language

Embedded firmware development may require the mixing of Assembly Language with high level language or vice versa.

Interrupt handling, Source code is already available in high level language\Assembly Language etc are examples

High Level language and low level language can be mixed in three different ways

- ✓ Mixing Assembly Language with High level language like 'C'
- ✓ Mixing High level language like 'C' with Assembly Language
- ✓ In line Assembly

The passing of parameters and return values between the high level and low level language is cross-compiler specific

Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=JHbSVK8divM

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 306

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L28

LECTURE HANDOUTS

ECE **III / VI**

Course Name with Code : Embedded Systems and RTOS & 19ECC15

Course Faculty : Dr.P.Padmaloshani

Unit : IV - REAL TIME OPERATING SYSTEMS Date of Lecture:

Topic of Lecture: Defining RTOS, The Scheduler

Introduction :

- **A real-time operating system (RTOS)** is a special-purpose operating system used in computers that has strict time constraints for any job to be performed. It is employed mostly in those systems in which the results of the computations are used to influence a process while it is executing. Whenever an event external to the computer occurs, it is communicated to the computer with the help of some sensor used to monitor the event. The sensor produces the signal that is interpreted by the operating system as an interrupt. On receiving an interrupt, the operating system invokes a specific process or a set of processes to serve the interrupt.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Defining RTOS, The Scheduler **:**

The Operating System acts as a bridge between the user applications/tasks and the underlying system resources through a set of system functionalities and services

The primary functions of an Operating system is

➢ Make the system convenient to use

➢ Organize and manage the system resources efficiently and correctly

Real Time Purpose Operating System (RTOS):

Operating Systems, which are deployed in embedded systems demanding real-time response

• Deterministic in execution behavior. Consumes only known amount of time for kernel applications

• Implements scheduling policies for executing the highest priority task/application always

• Implements policies and rules concerning time-critical allocation of a system's resources

• Windows CE, QNX, VxWorks , MicroC/OS-II etc are examples of Real Time Operating Systems (RTOS)

The Real Time Kernel: The kernel of a Real Time Operating System is referred as Real Time kernel. In complement to the conventional OS kernel, the Real Time kernel is highly specialized and it contains only the minimal set of services required for running the user applications/tasks. The basic functions of a Real Time kernel are

a) Task/Process management
b) Task/Process scheduling
c) Task/Process synchronization
d) Error/Exception handling
e) Memory Management
f) Interrupt handling
g) Time management

Task/Process Scheduling: Deals with sharing the CPU among various tasks/processes. A kernel application called '*Scheduler*' handles the task scheduling. Scheduler is nothing but an algorithm implementation, which performs the efficient and optimal scheduling of tasks to provide a deterministic behavior.

Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=95yUbClyf3E

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 382

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L29

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : **Embedded Systems and RTOS & 19ECC15**

Course Faculty : **Dr.P.Padmaloshani**

Unit : **IV - REAL TIME OPERATING SYSTEMS** **Date of Lecture:**

Topic of Lecture: Objects, Services

Introduction :

- **A real-time operating system (RTOS)** is a special-purpose operating system used in computers that has strict time constraints for any job to be performed. It is employed mostly in those systems in which the results of the computations are used to influence a process while it is executing. Whenever an event external to the computer occurs, it is communicated to the computer with the help of some sensor used to monitor the event. The sensor produces the signal that is interpreted by the operating system as an interrupt. On receiving an interrupt, the operating system invokes a specific process or a set of processes to serve the interrupt.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Objects, Services **:**

OBJECTS

Kernel objects are special constructs that form the building blocks for application development for real time embedded systems

The most common RTOS kernel objects are:

Tasks – are concurrent and independent threads of execution that can compete for CPU execution time

○ Semaphores – are token-like objects that can be incremented or decremented by tasks for synchronization or mutual exclusion

○ Message Queues – are buffer-like data structures that can be used for synchronization, mutual exclusion, and data exchange

SERVICES

Along with objects most kernels provide services that help developers create applications for real-time embedded systems.

• The most common services found comprise sets of API (Application Program Interface) calls that can be used to perform operations on kernel objects or can be used in general to facilitate timer management, interrupt handling, device I/O, and memory management

Real Time Kernel Task/Process Management: Deals with setting up the memory space for the tasks, loading the task's code into the memory space, allocating system resources, setting up a Task Control Block (TCB) for the task and task/process termination/deletion. A Task Control Block (TCB) is used for holding the information corresponding to a task. TCB usually contains the following set of information

❖ *Task ID:* Task Identification Number

❖ *Task State:* The current state of the task. (E.g. State= 'Ready' for a task which is ready to execute)

❖ *Task Type*: Task type. Indicates what is the type for this task. The task can be a hard real time or soft real time or background task.

❖ *Task Priority*: Task priority (E.g. Task priority =1 for task with priority = 1)

❖ *Task Context Pointer*: Context pointer. Pointer for context saving

Task Memory Pointers: Pointers to the code memory, data memory and stack memory for the task

❖ *Task System Resource Pointers*: Pointers to system resources (semaphores, mutex etc) used by the task

❖ *Task Pointers:* Pointers to other TCBs (TCBs for preceding, next and waiting tasks)

❖ *Other Parameters* Other relevant task parameters

Task/Process Scheduling:

Task/Process Synchronization

Error/Exception handling

Memory Management

Interrupt Handling

Time Management

The scheduler is the heart of every kernel and it is contained within each kernel.

It follows a set of algorithms that determines which task executes when. Some common examples of scheduling algorithms include round-robin and preemptive scheduling.

SCHEDULABLE ENTITIES

A schedulable entity is a kernel object that can compete for execution time on a system, based on a predefined scheduling algorithm: Ex: tasks and processes

Processes are similar to tasks

Processes provide better memory protection features, at the expense of performance and memory Overhead. Note: message queues and semaphores are not schedulable entities

Video Content / Details of website for further learning (if any):
https://electricalfundablog.com/rtos-real-time-operating-system/

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 390

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L30

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : **Embedded Systems and RTOS & 19ECC15**

Course Faculty : **Dr.P.Padmaloshani**

Unit : **IV - REAL TIME OPERATING SYSTEMS** **Date of Lecture:**

Topic of Lecture: Characteristics of RTOS

Introduction :

- **A real-time operating system (RTOS)** is a special-purpose operating system used in computers that has strict time constraints for any job to be performed. It is employed mostly in those systems in which the results of the computations are used to influence a process while it is executing. Whenever an event external to the computer occurs, it is communicated to the computer with the help of some sensor used to monitor the event. The sensor produces the signal that is interpreted by the operating system as an interrupt. On receiving an interrupt, the operating system invokes a specific process or a set of processes to serve the interrupt.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Characteristics of RTOS **:**

Applications define the requirements of its underlying RTOS. Some of the more common ones are:
Reliability -- depending on the application, the system might need to operate for long periods without human intervention.
Predictability -- meeting time requirements is key to real-time embedded systems -- the term deterministic describes RTOSs with predictable behavior, in which the completion of OS calls occurs within known timeframes.
Performance -- this requirement dictates that the embedded system must perform fast enough to fulfill its timing requirements.
Compactness -- application design constraints and cost constraints help determine how compact and embedded system needs to be.
Scalability -- because RTOSs can be used in a wide variety of applications, they must be able to scale up or down to meet application-specific requirements.

MEMORY REQUIREMENTS FOR REAL-TIME SYSTEMS

The size of a kernel is from 1K to 100K bytes

A minimal kernel for an 8-bit CPU that provides onlyscheduling, context switching, semaphore

management, delays, and timeouts is 1K to 3K bytes.

Total code space = application space + kernel space

Total RAM if the kernel does not support a separate interrupt stack is:
application code requirements + data space needed by the kernel itself + SUM (task staks + MAX(ISR nesting)).

If the kernel supports a separate stack for interrupts is:

application + data space needed by the kernel + SUM(task stacks) + MAX(ISR nesting).

Video Content / Details of website for further learning (if any):
https://frameboxxindore.com/other/quick-answer-what-are-the-characteristics-of-real-time-operating-systems.html

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 394

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L31

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : Embedded Systems and RTOS & 19ECC15

Course Faculty : Dr.P.Padmaloshani

Unit : IV - REAL TIME OPERATING SYSTEMS Date of Lecture:

Topic of Lecture: Defining a Task, Tasks States and Scheduling

Introduction :

- In the Operating System context, a task is defined as the program in execution and the related information maintained by the Operating system for the program
- Task is also known as '*Job*' in the operating system context

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Defining a Task, Tasks States and Scheduling **:**

- In the Operating System context, a task is defined as the program in execution and the related information maintained by the Operating system for the program
- Task is also known as '*Job*' in the operating system context
- A program or part of it in execution is also called a '*Process*'
- The terms '*Task*', '*job*' and '*Process*' refer to the same entity in the Operating System context and most often they are used interchangeably
- A process requires various system resources like CPU for executing the process, memory for storing the code corresponding to the process and associated variables, I/O devices for information exchange etc

The structure of a Processes

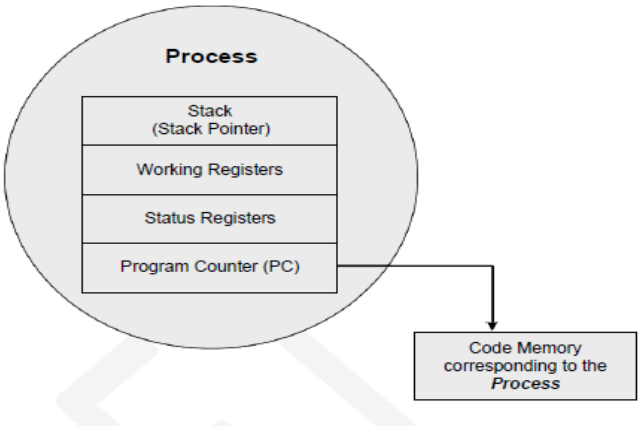
▪ The concept of '*Process*' leads to concurrent execution (pseudo parallelism) of tasks and thereby the efficient utilization of the CPU and other system resources

▪ Concurrent execution is achieved through the sharing of CPU among the processes.

▪ A process mimics a processor in properties and holds a set of registers, process status, a Program Counter (PC) to point to the next executable instruction of the process, a stack for holding the local variables associated with the process and the code corresponding to the process

A process, which inherits all the properties of the CPU, can be considered as a virtual processor, awaiting its turn to have its properties switched into the physical processor

- When the process gets its turn, its registers and Program counter register becomes mapped to the physical registers of the CPU



Structure of a process

Process States & State Transition

The creation of a process to its termination is not a single step operation

▪ The process traverses through a series of states during its transition from the newly created state to the terminated state

▪ The cycle through which a process changes its state from '*newly created*' to '*execution completed*' is known as '*Process Life Cycle*'. The various states through which a process traverses through during a Process Life Cycle indicates the current status of the process with respect to time and also provides information on what it is allowed to do next

Process States & State Transition:

- **Created State:** The state at which a process is being created is referred as 'Created State'. The Operating System recognizes a process in the '*Created State*' but no resources are allocated to the process
- **Ready State:** The state, where a process is incepted into the memory and awaiting the processor time for execution, is known as '*Ready State*'. At this stage, the process is placed in the '*Ready list*' queue maintained by the OS
- **Running State:** The state where in the source code instructions corresponding to the process is being executed is called '*Running State*'. Running state is the state at which the process execution happens

Blocked State/Wait State: Refers to a state where a running process is temporarily suspended from execution and does not have immediate access to resources. The blocked state might have invoked by various conditions like- the process enters a wait state for an event to occur (E.g. Waiting for user inputs such as keyboard input) or waiting for getting access to a shared resource like semaphore, mutex etc

Completed State: A state where the process completes its execution

 The transition of a process from one state to another is known as '*Statetransition*'

 When a process changes its state from Ready to running or from running toblocked or terminated or from blocked to running, the CPU allocation for the process may also change

Task Scheduling:

In a multitasking system, there should be some mechanism in place to share the CPU among the different tasks and to decide which process/task is to be executed at a given point of time

- Determining which task/process is to be executed at a given point of time is known as task/process scheduling

- Task scheduling forms the basis of multitasking
- Scheduling policies forms the guidelines for determining which task is to be executed when
- The scheduling policies are implemented in an algorithm and it is run by the kernel as a service
- The kernel service/application, which implements the scheduling algorithm, is known as '*Scheduler*'
- The task scheduling policy can be *pre-emptive*, *non-preemptive* or *co- operative*
- Depending on the scheduling policy the process scheduling decision may take place when a process switches its state to ➢ '*Ready*' state from '*Running*' state

➢ '*Blocked/Wait*' state from '*Running*' state

➢ '*Ready*' state from '*Blocked/Wait*' state

➢ '*Completed*' state

Task Scheduling - Scheduler Selection:

The selection of a scheduling criteria/algorithm should consider

• **CPU Utilization:** The scheduling algorithm should always make the CPU utilization high. CPU utilization is a direct measure of how much percentage of the CPU is being utilized.

• **Throughput:** This gives an indication of the number of processes executed per unit of time. The throughput for a good scheduler should always be higher.

• **Turnaround Time:** It is the amount of time taken by a process for completing its execution. It includes the time spent by the process for waiting for the main memory, time spent in the ready queue, time spent on completing the I/O operations, and the time spent in execution. The turnaround time should be a minimum for a good scheduling algorithm

Waiting Time: It is the amount of time spent by a process in the '*Ready*' queue waiting to get the CPU time for execution. The waiting time should be minimal for a good scheduling algorithm.

• **Response Time:** It is the time elapsed between the submission of a process and the first response. For a good scheduling algorithm, the response time should be as least as possible.

Video Content / Details of website for further learning (if any):
https://electricalfundablog.com/rtos-real-time-operating-system/

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 404

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L32

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : Embedded Systems and RTOS & 19ECC15

Course Faculty : Dr.P.Padmaloshani

Unit : IV - REAL TIME OPERATING SYSTEMS Date of Lecture:

Topic of Lecture: Task Operations, Structure

Introduction :

- **A real-time operating system (RTOS)** is a special-purpose operating system used in computers that has strict time constraints for any job to be performed. It is employed mostly in those systems in which the results of the computations are used to influence a process while it is executing. Whenever an event external to the computer occurs, it is communicated to the computer with the help of some sensor used to monitor the event. The sensor produces the signal that is interpreted by the operating system as an interrupt. On receiving an interrupt, the operating system invokes a specific process or a set of processes to serve the interrupt.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Task Operations, Structure **:**

Every RTOS has some specific steps for system initialization and starting. Here we present the initialization steps for the uC/OS-II.

• OSInit( ) initializes all uC/OS variables and data structures (see OS_CORE.C).

○ OSInit( ) creates the idle task OS_TaskIdle( ), which is always ready to run. The priority of OS_TaskIdle( ) is always set to OS_LOWEST_PRIO.

○ If OS_TASK_STAT_EN and OS_TASK_CREATE_EXT_EN (see OS_CFG.H) are both set to 1, OSInit( ) also creates the statistic task OS_TaskStat( ) and makes it ready to run.

○The priority of OS_TaskStat( ) is always set to OS_LOWEST_PRIO-1.

A requirement of uC/OS is that you call OSInit( ) before you call any of uC/OS's other services

(4) Because both tasks (Idle and Stat) are ready to run, their corresponding bits in OSRdyTbl[ ] are set to 1.

(5) Also, because the bits of both tasks are on the same row in OSRdyTbl[ ] only one bit in OSRdyGrp is

(1) The task control blocks of the Idle and Stat tasks are chained together in a doubly linked list.

(2) OSTCBList points to the beginning of this chain --when a task is created, it is always placed at the beginning of the list.
(3) both ends of the doubly linked list point to NULL (i.e., 0).
Because both tasks are ready to run their corresponding bits in OSRdyTb[] are set to 1; also only one bit in OSRdyGrp is set to 1.
uC/OS-II also initializes five pools of free data structure
Each of these pools is a singly linked list and allows uC/OS-II to obtain and return an element from and to a pool quickly.
After OSInit() has been called:
OS_TCB pool contains OS_MAX_TASKS entries
OS_EVENT pool contains OS_MAX_EVENTS entries
OS_Q pool contains OS_MAX_QS entries
OS_FLAG_GRP pool contains OS_MAX_FLAGS entries; and
OS_MEM pool contains OS_MAX_MEM_PART entries
• After OSInit() has been called:
Each of the free pools are NULL-pointer terminated -- their size is defined in OS_CFG.H.
You start multitasking by calling OSStart( ).
The start-up task is unique because it is meant to run only one time. As we have mentioned most tasks are endless loops. To make sure that the start-up task is executed only one time, it ends with a trap that calls the OSTaskSuspend() service. OSTaskSuspend() does stop the task, but just in case another task accidentally starts it up again with OSTaskResume(), it is contained in an endless trap.
TYPICAL TASK STRUCTURES
When writing code for tasks, the code is structuredin one of two ways:

- run to completion, or
- endless loop

```
RunToCompletionTask() {
  Initialize application
  Create 'endless loop tasks'
  Create kernel objects
  Delete or suspend this task
}
```

```
EndlessLoopTask() {
   Initialization code
   Loop forever
   {
        Body of the loop
        Make one or more blocking calls
    }
}
```

Video Content / Details of website for further learning (if any):
https://electricalfundablog.com/rtos-real-time-operating-system/

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 422

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L33

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : Embedded Systems and RTOS & 19ECC15

Course Faculty : Dr.P.Padmaloshani

Unit : IV - REAL TIME OPERATING SYSTEMS Date of Lecture:

Topic of Lecture: Synchronization

Introduction :

- **A real-time operating system (RTOS)** is a special-purpose operating system used in computers that has strict time constraints for any job to be performed. It is employed mostly in those systems in which the results of the computations are used to influence a process while it is executing. Whenever an event external to the computer occurs, it is communicated to the computer with the help of some sensor used to monitor the event. The sensor produces the signal that is interpreted by the operating system as an interrupt. On receiving an interrupt, the operating system invokes a specific process or a set of processes to serve the interrupt.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Synchronization **:**

Real time system applications use multiple concurrent tasks to maximize efficiency -- coordinating these activities requires inter-task synchronization and communication.

Synchronization is classified in two categories:

resource synchronization -- determines whether access to a shared resource is safe, and, if not, when it will be safe.

activity synchronization -- determines whether the execution of a multithreaded program has reached a certain state, and, if it hasn't, how to wait for and be notified when this state is reached.

RESOURCE SYNCHRONIZATION

Access by multiple tasks must be synchronized to maintain the integrity of a shared resource.

Shared Resource: is a resource that can be used by more than one task. Each task should gain exclusive access to the shared resource to prevent data corruption. This process is called mutual exclusion.

Resource synchronization is closely associated with critical sections and mutual exclusions.

•Mutual exclusion is a provision by which only one task at a time can access a shared resource.

•**Critical section**: a critical section of code, also called a critical region, is code that needs to be treated indivisibly. After the section of code starts executing, it must not be interrupted. To ensure that execution is not interrupted, interrupts are typically disabled before the critical code is executed and enabled when the critical code is finished.
○A critical section is the section of code from which the shared resource is accessed.
MEMORY
• Problem arises if access to the shared memory is not exclusive, and multiple tasks can simultaneously access it -- As an example, consider two tasks trying to access shared memory.
One task (the sensor task) periodically receives data from a sensor and writes the data to shared memory. • Meanwhile, a second task (display task) reads the data from the memory and displays it.
•Erroneous data interpretation can take place
ACTIVITY SYNCHRONIZATION
Activity synchronization, also called condition synchronization or sequence control, ensures that the correct execution order among cooperating tasks is used.
Methods of activity synchronization:
1) rendezvous synchronization;
2) barrier synchronization.
RENDEZVOUS SYNCHRONIZATION
A simple rendezvous method can use kernel primitives such as semaphores or message queues, instead of entry calls. Both binary semaphores are initialized to 0.
When task 1 reaches the rendezvous point, it gives semaphore #2, and then it gets on semaphore #1.
• When task 2 reaches the rendezvous point, it gives semaphore #1, and it gets on semaphore #2.
• Task 1 has to wait on semaphore #1 before task 2 arrives, and vice versa, thus achieving rendezvous synchronization
BARRIER SYNCHRONIZATION
A barrier is a point where some tasks need to present some results that must be analyzed before the decision on the next execution path can be made
Barrier synchronization comprises three actions:
1. A task posts its arrival at the barrier,
2. The task waits for other participants to reach the barrier,
3. The task receives notification to proceed beyond the barrier.
EXAMPLE of barrier: In an embedded control system, a complex computation can be divided and distributed among multiple tasks. Some parts of this complex computations are I/O bound, other parts are CPU intensive, and still others are mainly floating point operations that rely heavily on specialized floating-point coprocessor hardware. These partial results must be collected from the various tasks for the final calculation. The result determines what other partial computations each task is to perform next.

Video Content / Details of website for further learning (if any):
https://www.coursera.org/lecture/iot-architecture/synchronisation-and-communication-CRERd

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 442

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L34

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : Embedded Systems and RTOS & 19ECC15

Course Faculty : Dr.P.Padmaloshani

Unit : IV - REAL TIME OPERATING SYSTEMS Date of Lecture:

Topic of Lecture: Communication and Concurrency

Introduction :

- **A real-time operating system (RTOS)** is a special-purpose operating system used in computers that has strict time constraints for any job to be performed. It is employed mostly in those systems in which the results of the computations are used to influence a process while it is executing. Whenever an event external to the computer occurs, it is communicated to the computer with the help of some sensor used to monitor the event. The sensor produces the signal that is interpreted by the operating system as an interrupt. On receiving an interrupt, the operating system invokes a specific process or a set of processes to serve the interrupt.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Communication and Concurrency **:**

COMMUNICATION

Tasks in a multitasking environment communicate with one another so they can pass information to each other and coordinate their activity.

Types of communications are:

signal-centric -- all necessary information is conveyed within the event signal itself.

data-centric -- the information is carried within the transferred data.

or both -- data transfer accompanies event notification.

COMMUNICATION PURPOSES

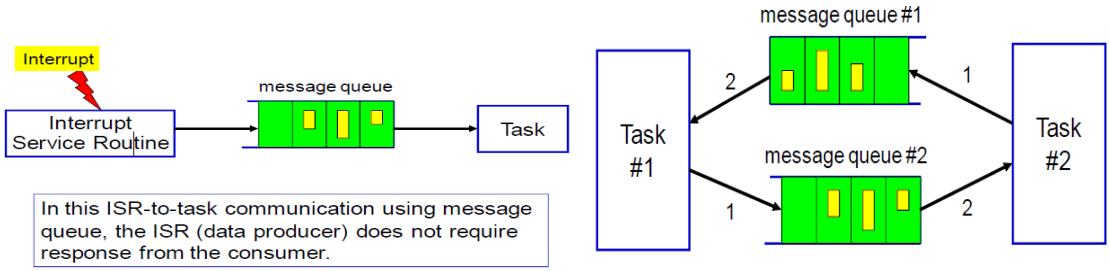
- Transferring data from one task to another,
- signaling the occurrences of events between tasks
- allowing one task to control the execution of other tasks,

• synchronizing activities, and implementing custom synchronization protocols for resource sharing.

The main kernel objects used for task communications are **message mailboxes and message queues**

Loosely coupled communication – when communication involves unidirectional data flow

Tightly coupled communication – the data movement is bidirectional.



The data producer synchronously waits for a response to its data transfer before resuming execution, or the response is returned asynchronously while the data producer continue its function.

RESOURCE SYNCHRONIZATION METHODS IN RTOS

Resource synchronization or mutual exclusion methods are:

1) interrupt locking (disabling system interrupts),
2) preemptive locking (disabling the kernel scheduler);
3) performing test-and-set operations;
4) using semaphores and event flags

DISABLING SYSTEM INTERRUPTS

Disable interrupts;

Access the resource (read/write from/to variable);

Reenable interrupts;

Interrupt locking (disabling system interrupts) is the method used to synchronize exclusive access to shared resources between tasks and ISRs.

This process guarantees that the current task continues to execute until it voluntarily relinquishes control. As such, interrupt locking can also be used to synchronize access to shared resources between tasks. Interrupt locks, although the most powerful and the most effective synchronization method, can introduce indeterminism into the system when used indiscriminately. Therefore, the duration of interrupt locks should be short, and interrupt locks should be used only when necessary to guard a task-level critical region from interrupt activities.

DISABLING AND ENABLING THE SCHEDULER

Can be used when a task is not sharing variables or data structures with an ISR. NOTE that while the scheduler is locked, interrupts are enabled, but the behavior of the scheduler is very similar with that of a non-preemptive kernel.

Video Content / Details of website for further learning (if any):
https://www.coursera.org/lecture/iot-architecture/synchronisation-and-communication-CRERd

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 426

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L35

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : Embedded Systems and RTOS & 19ECC15

Course Faculty : Dr.P.Padmaloshani

Unit : IV - REAL TIME OPERATING SYSTEMS Date of Lecture:

Topic of Lecture: Defining Semaphores, Operations and Use

Introduction :

- **A real-time operating system (RTOS)** is a special-purpose operating system used in computers that has strict time constraints for any job to be performed. It is employed mostly in those systems in which the results of the computations are used to influence a process while it is executing. Whenever an event external to the computer occurs, it is communicated to the computer with the help of some sensor used to monitor the event. The sensor produces the signal that is interpreted by the operating system as an interrupt. On receiving an interrupt, the operating system invokes a specific process or a set of processes to serve the interrupt.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Defining Semaphores, Operations and Use **:**

DEFINING A SEMAPHORE

A semaphore (sometimes called a semaphore token) is a kernel object that one or more tasks can acquire or release for the purpose of synchronization or mutual exclusion.

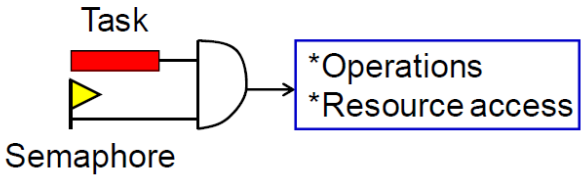
Types of semaphores:

a) binary semaphores,

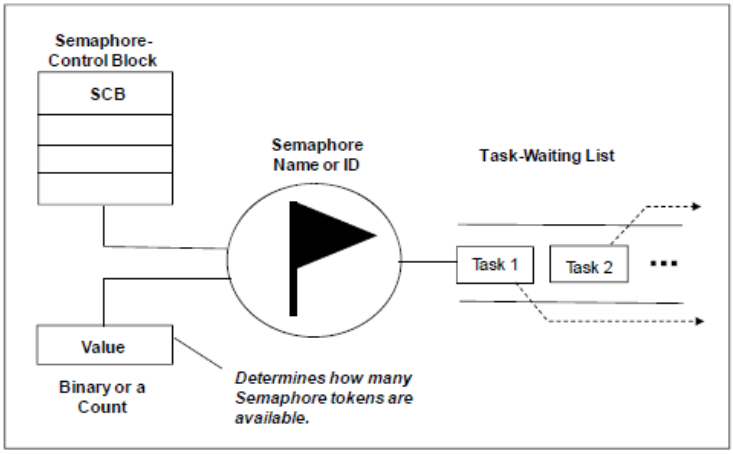
b) counting semaphores,

c)mutual exclusion (mutex) semaphores.

A semaphore is like a key when used for mutual exclusion and like a flag when used for synchronization -- if a task can acquire the semaphore, it can carry out the intended operation or access the resource.



When a semaphore is first created, the kernel assigns to it an associated semaphore control block (SCB), a unique ID, a value (binary or count), and a task-waiting list.



A single semaphore can be acquired a finite number of times. The kernel tracks the number of times a semaphore has been acquired or released by maintaining a token count, which is initialized to a value when the semaphore is created.

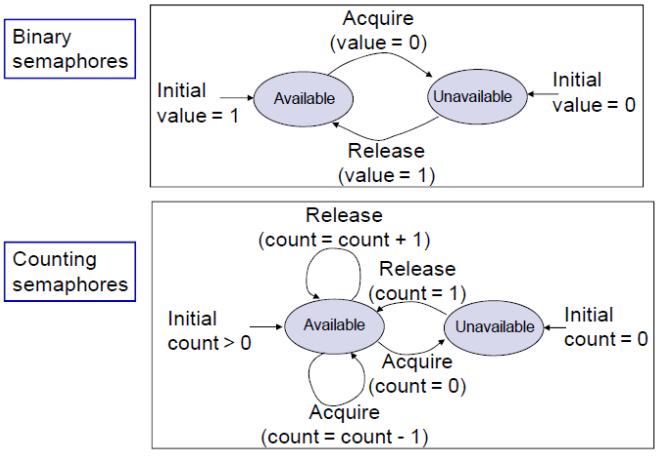
• As a task acquires the semaphore, the token is decremented; as a task releases the semaphore, the count is incremented.

• If the token count reaches 0, the semaphore has no tokens left. A requesting task, therefore, cannot acquire the semaphore, and the task blocks if it chooses to wait for the semaphore to become available.

•The task waiting list tracks all tasks blocked. These blocked tasks are kept in the task waiting list in either first in/first out (FIFO) order or highest priority first order.

• When an unavailable semaphore becomes available, the kernel allows the first task in the task-waiting list to acquire it. The kernel moves this unblocked task either to the running state, if it is highest priority task, or to the ready state, until it becomes the highest priority task and is able to run.

• Note that the exact implementation of a task-waiting list can vary from one kernel to another.



OPERATIONS PERFORMED ON A SEMAPHORE

The operations performed on a semaphore are:

The initial value of the semaphore must be provided when the semaphore is initialized.
• The waiting list of tasks is always initially empty.
ACQUIRING A SEMAPHORE
if the semaphore is available (the semaphore value is greater than 0) the semaphore value is decremented and the task continue execution.
○ if semaphore's value is 0, the task performing WAIT is placed in a waiting list.
○ A task desiring a semaphore performs a WAIT operation.
• most kernels allow to specify a timeout for the wait
RELEASING A SEMAPHORE
If no task is waiting for the semaphore, the semaphore value is simply incremented
If any task is waiting, one of the tasks is made ready to run and the semaphore value is not incremented - depending on the kernel.
A task releases a semaphorethe task that receives the semaphore is the highest priority task waiting or, the first task that requested
the semaphore -- uC/OS supports the first method
Semaphores are useful when task share I/O devices. Note: a timeout value of 0 indicates that the task is willing to wait forever.
Example: • Two tasks need to access the same printer.
Solution: The resource (device) has a semaphore associated to it.
Initialize the semaphore to 1.
Rules to access the printer: each task must first obtain the semaphore.

Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=firiu8_3DZA

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 450

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L36

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : **Embedded Systems and RTOS & 19ECC15**

Course Faculty : **Dr.P.Padmaloshani**

Unit : **IV - REAL TIME OPERATING SYSTEMS** **Date of Lecture:**

Topic of Lecture: Defining Message Queue

Introduction :

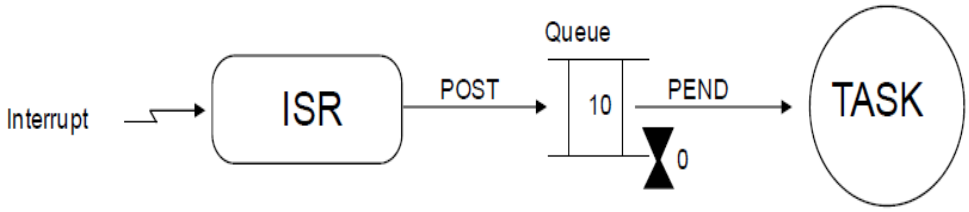
- **A real-time operating system (RTOS)** is a special-purpose operating system used in computers that has strict time constraints for any job to be performed. It is employed mostly in those systems in which the results of the computations are used to influence a process while it is executing. Whenever an event external to the computer occurs, it is communicated to the computer with the help of some sensor used to monitor the event. The sensor produces the signal that is interpreted by the operating system as an interrupt. On receiving an interrupt, the operating system invokes a specific process or a set of processes to serve the interrupt.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Defining Message Queue **:**

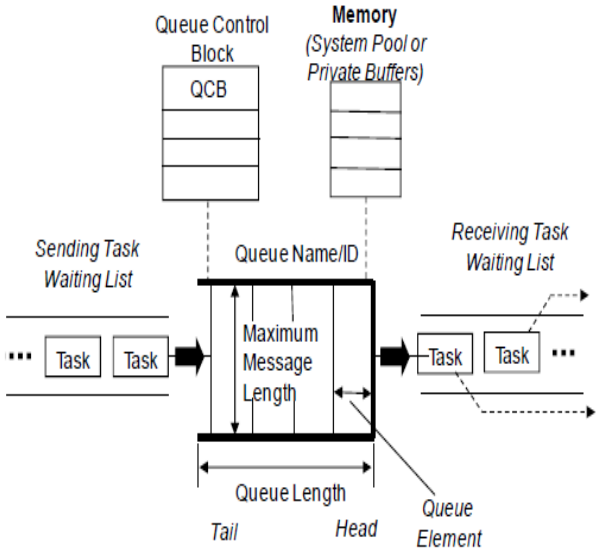
A message queue is used to send one or more messages to a task – is an array of mailboxes.
The messages are extracted in FIFO fashion or LIFO fashion.
A waiting list is associated with the queue -- highest priority task waiting or FIFO will get the message.



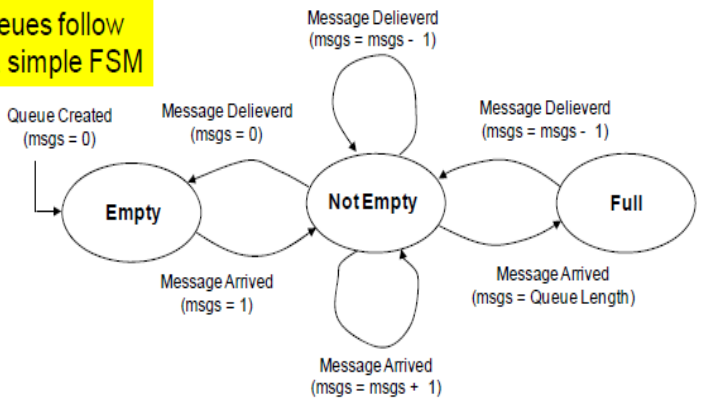
Kernels typically provide these message queue services:
initialize the queue (queue empty after initialization).
deposit a message into the queue (POST).
wait for a message to be deposited into the queue (PEND).
get a message from a queue, if one is present, but do not suspend the caller if the mailbox is empty (ACCEPT).

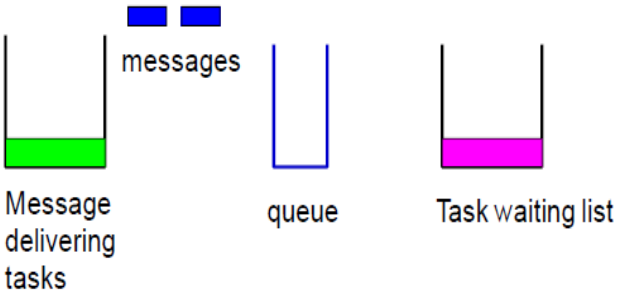
Defining Message Queues

Message queue created: an associated queue control block (QCB), a message queue name, a unique ID, memory buffers, a queue length, a maximum message length, and one or more task-waiting lists



Message queues follow the logic of a simple FSM





A message queue in any kernel is a buffer-like object through which a task and ISRs send and receive messages to communicate and synchronize with data.

A message queue temporarily holds messages from sender until the intended receiver is ready to read them -- this temporary buffering decouples a sending and receiving task; that it frees the tasks from having to send and receive messages simultaneously. The message queue itself consists of a number of elements, each of which can hold a single message.
Video Content / Details of website for further learning (if any): https://www.youtube.com/watch?v=ESL4NmhxXQY
Important Books/Journals for further learning including the page nos.: Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 476

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L37

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : **Embedded Systems and RTOS & 19ECC15**

Course Faculty : **Dr.P.Padmaloshani**

Unit : **V - RTOS BASED EMBEDDED SYSTEM DESIGN**

Date of Lecture:

Topic of Lecture: Defining Case Studies of RTOS RT Linux

Introduction :

- RT LINUX : Linux is a free general purpose OS. Several real-time implementations of Linux (RTLinux) are available.
- RTLinux is a self-host operating system that runs along with Linux system.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Case Studies of RTOS RT Linux **:**

RT LINUX

Linux is a free general purpose OS. Several real-time implementations of Linux (RTLinux) are available.
RTLinux is a self-host operating system that runs along with Linux system.
The real-time kernel sits between the hardware and the Linux system; to the standard Linux kernel, the RT Linux layer appears to be the actual hardware.
The RT Linux kernel intercepts all interrupts generated by hardware; interrupts not related to real-time activity are passed to the Linux kernel as software interrupts.
○ If an interrupt is to cause a real-time task to run, the real time kernel preempts Linux; thus Linux runs as a low priority background task of RT Linux.
○Real-time applications are written as loadable kernel modules and run in the kernel space.
This approach is known as the dual kernel approach.
The following are some of the important shortcomings of the dual-kernel approach:
Duplicate coding efforts - new drivers and system services must be created specifically for the real-time kernel;

○Fragile execution environment - real-time tasks that contain a coding error such as a corrupt C pointer can easily cause a fatal kernel fault.
○ Limited portability - real-time programs written using one vendor's RT-Linux version may not run on another.
○ Programming difficulty - supports only a limited subset of POSIX APIs.
LYNX
Linx is a self-host real-time OS and is available from www.lynuxworks.com;
Lynx 3.0 is a microkernel based real-time OS
Lynx is fully compatible with Linux. A Linux program's binary image can be run directly on
Lynx - on the other hand for other Linux compatible OS such as QNX, Linux applications need to be recompiled in order to run on them.
Lynx is 28 KB in size and provides essential services for task scheduling, interrupt dispatch, and synchronization

Other services are provided as Kernel Plug-Ins (KPIs) By adding KPIs to the microkernel the system can be configured to support I/O, file systems, sockets, and so on.

○With full configuration, it can even function as a multipurpose Unix machine on which both hard and soft real-time tasks can run.
• Unlike many embedded RTOSs Lynx supports memory protection

Video Content / Details of website for further learning (if any):
https://www.coursera.org/lecture/real-time-embedded-systems-concepts-practices/comparison-of-cyclic-executive-rtos-and-linux-rt-service-implementation-jCQDV

Important Books/Journals for further learning including the page nos.:
Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 499

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L38

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : **Embedded Systems and RTOS & 19ECC15**

Course Faculty : **Dr.P.Padmaloshani**

Unit : **V - RTOS BASED EMBEDDED SYSTEM DESIGN**

Date of Lecture:

Topic of Lecture: MicroC/OS-II, Vx Works

Introduction :

- VxWorks is a product from Wind River Systems. It is a host-target type RTOS and the host can be either Windows or Unix machines.
- VxWorks confirms to POSIX-RT and comes with an Integrated Development Environment (IDE) called Tornado.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

MicroC/OS-II, Vx Works **:**

uC/OS-II

uC/OS-II is available from Micrium Corporation. This RTOS is written in ANSI C and contains a small portion of assembly code. The assembly language portion has been kept to minimum to allow for easy portability.

uC/OS-II has been ported to over 100 different processor architectures ranging from 8-bit to 64-bit microprocessors, microcontrollers, and DSPs Some of the important features of uC/OS-II are:

Programmers have the option of using just a few of the offered services or select the entire range

○ Has a fully preemptive kernel

○ Allows up to 64 tasks to be created

Uses a partitioned memory management scheme.

Each memory partition consists of several fixed sized blocks. A task obtains memory blocks from the memory partition and the task must create a memory partition before it can be used. Allocation and deallocation of fixed-sized memory blocks is done in constant time and is deterministic.

○ Has been certified by the Federal Aviation Administration (FAA) for use in commercial aircraft and meets the demanding requirements of its standard to be used in avionics. VxWorks is a product from Wind River Systems. It is a host-target type RTOS and the host can be either Windows or Unix machines. •VxWorks confirms to POSIX-RT and comes with an Integrated Development Environment (IDE) called Tornado. Tornado comes with standard support for program development tools such as editor, cross-compiler, cross-debugger, etc. plus it contains VxSim and WindView VxSim simulates a VxWorks target for use as a prototyping and testing environment in the absence of the actual target board ○ WindView provides debugging tools for the simulator environment ○VxMP is the multiprocessor version of VxWorks VxWorks was deployed in the Mars Pathfinder which was sent to Mars in 1997 The system proved to have a bug due to unbounded priority inversion that caused real time tasks to miss their deadlines ○The problem was fixed by remote debugging
Video Content / Details of website for further learning (if any): https://www.wazeesupperclub.com/what-are-the-features-of-ucos-ii/
Important Books/Journals for further learning including the page nos.: Shibu K.V, Introduction to Embedded Systems, Tata McGraw Hill, 2009, pp. 514

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L39

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : **Embedded Systems and RTOS & 19ECC15**

Course Faculty : **Dr.P.Padmaloshani**

Unit : **V - RTOS BASED EMBEDDED SYSTEM DESIGN**

Date of Lecture:

Topic of Lecture: Digital camera, washing machine

Introduction :

- The digital still camera bears some resemblance to the film camera but is fundamentally different in many respects. The digital still camera not only captures images, it also performs a substantial amount of image processing that formerly was done by photofinishers.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Digital camera, washing machine **:**

The digital still camera bears some resemblance to the film camera but is fundamentally different in many respects. The digital still camera not only captures images, it also performs a substantial amount of image processing that formerly was done by photofinishers.

Digital image processing allows us to fundamentally rethink the camera. A simple example is digital zoom, which is used to extend or replace optical zoom. Many cell phones include digital cameras, creating a hybrid imaging/communication device.

Digital still cameras must perform many functions:

It must determine the proper exposure for the photo.

It must display a preview of the picture for framing.

It must capture the image from the image sensor.

It must transform the image into usable form.

It must convert the image into a usable format, such as JPEG, and store the image in a file system.

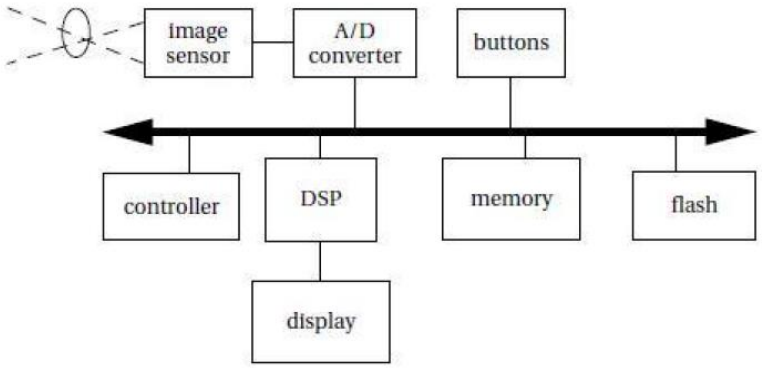
System Architecture:

Typical hardware architecture for a digital still camera is shown in Figure 5.19. Most cameras use two processors. The controller sequences operations on the camera and performs operations like file system

management. The DSP concentrates on image processing. The DSP may be either a programmable processor or a set of hardwired accelerators. Accelerators are often used to minimize power consumption.

Requirements:

Name	Digital still camera
Purpose	Digital still camera with JPEG compression
Inputs	Image sensor, shutter button
Outputs	Display, flash memory
Functions	Determine exposure and focus. capture image, Perform Bayer pattern interpolation, JPEG compression, store in flash file system.
Performance	Take one picture in 2 sec.
Manufacturing cost	Approximately $75
Power	Two AA batteries
Physical size and weight	Less than 4 ounces



Architecture of digital camera

Washing Machine:

Embedded System for Automatic Washing Machine using Microchip PIC18F Series Microcontroller

The design uses the PIC18F series microcontroller. All the control functionalities of the system are built around this. Upgradeability is the unique feature of this system. Control card hardware and software allows the manufacturer to add or remove the features as per customer requirement and model. Thus once the whole system is designed it is very economic in large quantity production. Single-phase motor is considered for the design. Front panel consists of a keypad and LCD display. Keypad provides automatic and manual wash options to the user. LCD display is convenient to convey machine information to user. One more design possibility is to use brushless DC motors or three phase induction motor. These types of motors are very efficient but requires complex control algorithm.

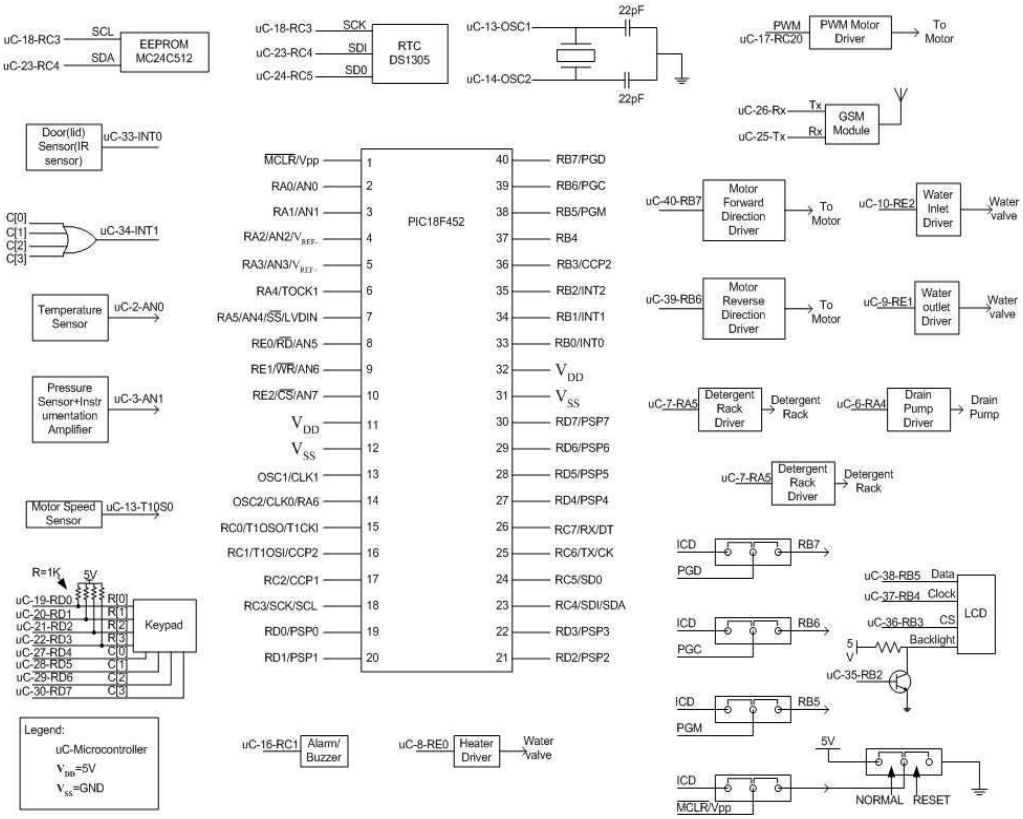
To implement such a complex and real time algorithm dedicated controller and software is required which a master controller controls. Even though cost is important criteria modern washing machines are designed with BLDC motors owing to efficiency and energy conservation. But in this assignment single phase universal motor has been used to design prototype due to its simplicity.

Design Specifications

This include both hardware and software specifications.

1. The system should provide fully automatic mode, semi-automatic mode and manual mode. Modes should be selectable by a keypad.

2. Under fully automatic mode user intervention requirement should be zero. Once the system is started in this mode it should perform its work independently and after the completion of work it should notify the user about the completion of work. This mode instantaneously should sense cloth quality and requirement of water, water temperature, detergent, load, wash cycle time and perform operation accordingly.
3. In semi-automatic mode also user requirement should be nil. But user has to choose any one of the semi-automatic mode in which washing conditions are predefined. Once the predefined mode is started the system should perform its job and after completion it should inform the user.
4. In manual mode continuous intervention of user is required. User has to specify which operation he wants to do and has to provide related information to the control system. For example, if user wants to wash only, he has to choose 'wash' option in manual mode. Then the system should ask the user to enter the wash time, amount of water and the load. After these data are entered, the user should start the machine. When the specified operation is completed system should inform the user.



Block Schematic of Washing Machine Controller

5. When the lid is open system should not work. If door is accidentally opened in between wash operation, then the system should stop working in minimum possible time (<10s)>
6. The system should provide all basic features of a washing machine like washing, rinsing, spinning, drying, cold wash, hot wash etc.
7. The system should provide easy options for upgradeability of new features. The hardware and the software should be compatible to both machines, which have fewer features, or more features. Removal of any feature should not affect the working of any other features or overall working of the system.
8. The system should work on single phase AC from 190VAC to 250VAC. The system should protect itself from power supply voltage variations.
9. In the event of power failure, the washing machine should automatically start its cycle from the point of interruption when power is resumed.

Hardware Design

Heart of this system is PIC18F452. Most of the peripheral features have been utilized to implement the design. Controlling the motor is very crucial part of the design. The PWM feature of the microcontroller controls motor speed. PWM output is fed to driver circuit and then to motor. To rotate the motor in two different directions 'forward' and 'reverse' direction control blocks are used. Motor speed sensor is interfaced to microcontroller. Microcontroller reads the speed of the motor and appropriately controls the speed of the motor in different phases of washing using PWM output. Door sensor, pressure sensor, keypad are also interfaced to microcontroller. Serial port in connected to GSM module. EEPROM and RTC are interfaced to MSSP module of controller. In circuit serial programming facility is provided for quick and easy programming and debugging.

Washing machine default parameters and user settable parameters are stored in external EEPROM. Internal EEPROM of the PIC is used to store status of the washing machine. The status is regularly logged to the internal EEPROM. In the event of power failure or whenever program resets, status flags are read from the internal EEPROM and thus status of the machine is determined and operation is continued from the point of interruption. Accessing of internal EEPROM is faster compared to external EEPROM. PIC18F452 provide 256 bytes of internal EEPROM that is not sufficient for storing parameters of machine. For this purpose, external EEPROM is used. Depending on the mode flag and status flag conditions corresponding machine parameters are read from the external EEPROM and temporarily stored in RAM and operations are performed.

RTC DS1305 is interfaced to SPI port of the microcontroller. This RTC is used as timing reference for all timing calculation of machine. Whenever a particular mode starts RTC is initialized to zero and there onwards RTC is read and compared with the set timings; with the battery backup provision actual RTC can also be implemented. Since PIC allows either I2C or SPI mode at a time, whenever we need to access EEPROM or RTC, MSSP port of the PIC has to be configured to respective protocol.

Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=1tsDqDqufXw

Important Books/Journals for further learning including the page nos.:
Rajkamal, Embedded Systems: Architecture, Programming and Design, Tata McGraw Hill, 2015, pp. 531

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L40

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : **Embedded Systems and RTOS & 19ECC15**

Course Faculty : **Dr.P.Padmaloshani**

Unit : **V - RTOS BASED EMBEDDED SYSTEM DESIGN**

Date of Lecture:

Topic of Lecture: Cell phones

Introduction :

- Embedded systems are **special-purpose computers built into devices not generally considered to be computers**. For example, the computers in vehicles, wireless sensors, medical devices, wearable fitness devices, and smartphones are embedded systems.
- Cellular phones are largely being used today. The cellular phones consist of a transmitter and receiver.it consist of microprocessor, Rom, Ram. The LCD display and keyboard is also available

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Cell phones **:**

Cellular phones have come a long way from analog communication devices to digital mobile computers. Today, a cellular phone is a paradigm of an embedded system having highly optimized cost, size, efficiency and performance. Challenges in RF circuits, implementation architecture, memory, and power consumption are still affecting the development and growth of mobile devices. New technologies such as decentralized architectures, reconfigurable circuits, advanced memories, and low power designs will help overcome challenges

Cellular phones are largely being used today. The cellular phones consists of a transmitter and receiver.it consist of microprocessor,Rom,Ram,. The LCD display and keyboard is also available

Digital convergence

Convergence enables people to create, share and consume digital content

Cellular phones evolve from traditional cost-optimized handhelds to multifunctional terminals

Cellular phones: key platform for mobile convergence applications (web browsing, video streaming, etc.) Digital Convergence

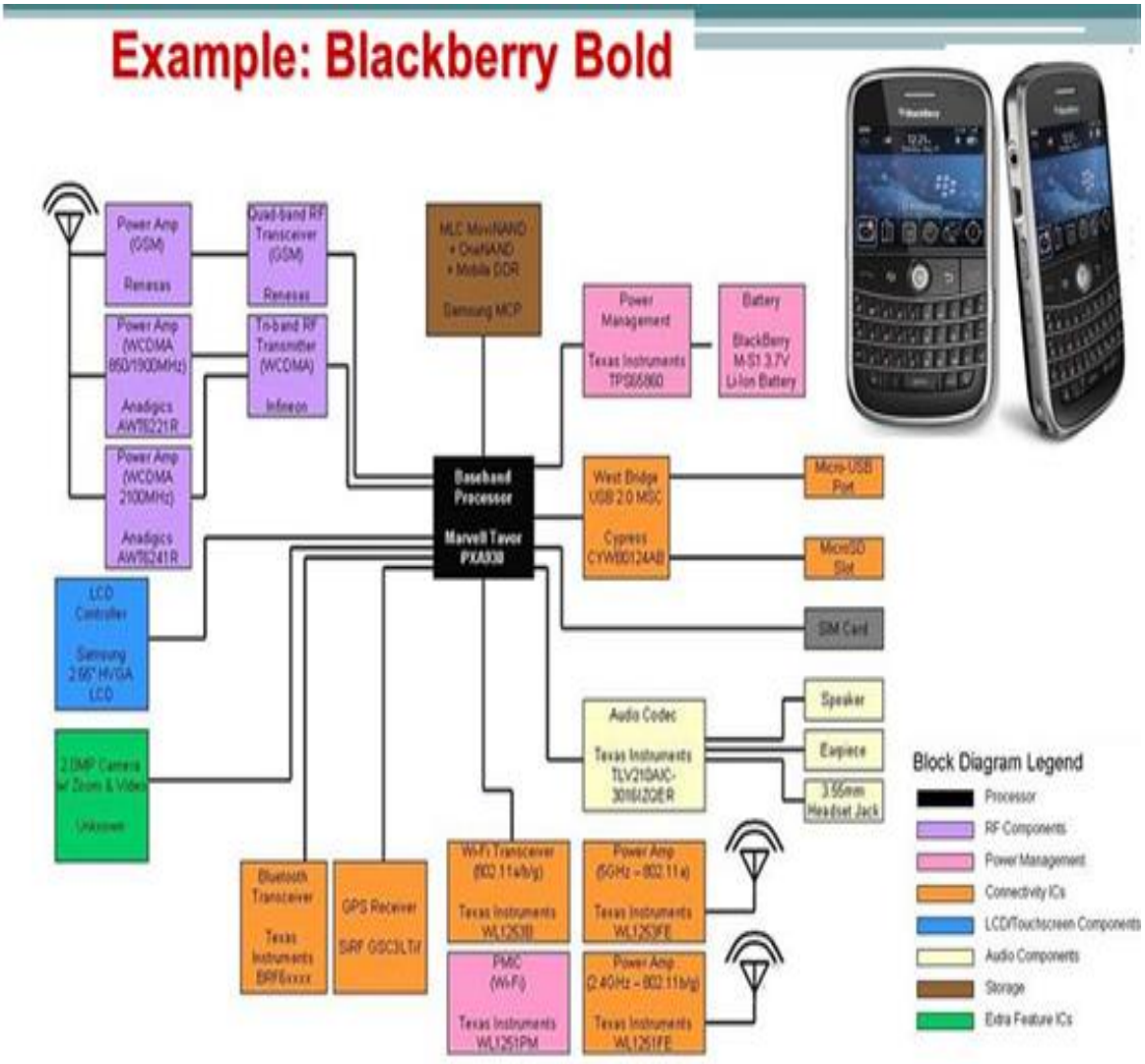
Todays' cell phone

Extremely complex embedded system

Functional blocks are custom-made for mobility

Chips are either proprietary designs or based on available chips.

sophisticated and better functionality Today's Cellular Phone



Example: Blackberry Bold

Technical Specifications

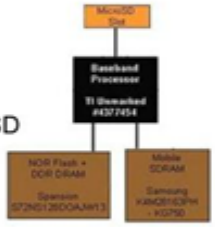
Operating System	BlackBerry OS 4.6
Network Support	GSM/GPRS/EDGE/HSDPA, UMTS: 850/1900/2100MHz, GSM: 850/900/1800/1900MHz
Screen	2.65" HVGA (320 x 480px), 65k Color TFT
Input Method	QWERTY Keyboard, Trackball
Wi-Fi	802.11a/b/g
GPS	Internal GPS equipped with BB Maps
Bluetooth	Bluetooth v2.0, A2DP and AVCRP support
Camera	2MP Camera with LED Flash
Storage	Internal 1GB Storage, External MicroSDHC Slot
Connection Ports	Mini-USB 2.0 Hi-Speed, 3.5mm stereo audio jack
Physical Attributes	Dimensions: 114mm x 66mm x 12.7mm, Weight: 133g

Media Support

Audio Support	Codecs: MP3/WMA/AAC+
Video Support	Codecs: DivX/XviD/WMV/3GP
Browser	Full HTML Browser, streaming video, RSS feed support

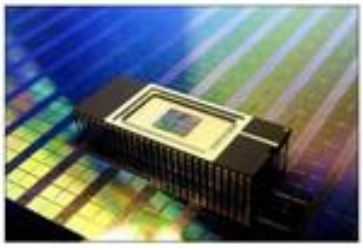
Memory Challenge

- Total memory requirement is increasing rapidly
- Mass Memories – interactive games, high quality video
  - Large memories are required to support data downloading and local storage
  - Supported by external memory cards: MMC or SD
- Small Memories – processing and small applications
  - Memory chips and their interconnections consume large areas on PCBs and are accessed frequently
  - New types of NVRAM may challenge memory chips to provide smaller and more cost effective memory solutions



NVRAM – Alternatives to Flash Memory

- Ferroelectric RAM (FeRAM)
  - DRAM cell with ferroelectric dielectric in the storage capacitor
  - Advantages: low power, faster reads and writes (single word vs. entire block erase), greater number of write-erase cycles (10^{16} vs. 10^{6})
  - Disadvantages: lower storage density, higher cost
- Phase Change Memory
  - Glass cells that become crystalline or amorphous by cooling
  - Advantages: faster reads and writes, greater number of write-erase cycles (10^{8} vs. 10^{6}), longer hold times
  - Disadvantages: temp sensitivity, no pre-programming



Application Platform

- Mobile internet – web browsing, video calls and high bit rate streaming
- Java ME – provides flexible user interfaces, built in network protocols, multimedia support
  - Fact: 2.1 Billion mobile phones use Java platforms
- 3rd Party Mobile Applications – if developed in Java, are portable enough to run on almost all cell phones

Power Challenge

- Recent evolution of communication and application functions have substantially increased power consumption
- Constant annual growth of 10% in battery capacity has enabled battery volume shrinkage while having mAh level constant
- However, when 3G or WLAN communication is run simultaneously with multimedia applications, power consumption must be reduced

Solutions to Power Gap

- 10% increase in battery capacity will continue forever
- Reduce power hungry components:
  - Antennas – Bluetooth, Wi-Fi, RF
  - Digital displays
- Dynamic voltage and frequency scaling (DVFS)
- Reconfigurable RF components to reduce the number of ICs.

Video Content / Details of website for further learning (if any):
https://www.eejournal.com/article/20120227-mobile/

Important Books/Journals for further learning including the page nos.:
Rajkamal, Embedded Systems: Architecture, Programming and Design, Tata McGraw Hill, 2015, pp. 604

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L41

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : **Embedded Systems and RTOS & 19ECC15**

Course Faculty : **Dr.P.Padmaloshani**

Unit : **V - RTOS BASED EMBEDDED SYSTEM DESIGN**

Date of Lecture:

Topic of Lecture: Home security systems

Introduction :

- The home security system can be achieved by accepting important controllers to switch home appliances that notice different variables using an appropriate sensor. This system involves the security hardware such as property, doors, locks, lighting, security cameras and alarm systems.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Home security systems **:**

The home security system can be achieved by accepting important controllers to switch home appliances that notice different variables using an appropriate sensor. This system involves the security hardware such as property, doors, locks, lighting, security cameras and alarm systems.

The main feature of this **home security system is a sensory system** that collects the information of the parameter such as temperature, gas, fire, human presence, etc., and sends the information to the Microcontroller.

Security is a major concern in our daily life. Everybody desires to be as much safe as possible. The progressive home security system has improved to the present security system in our day-to-day life. Security systems for homes protect from the thief. Security systems are protected in the home keeps the things safety from the thief, hazardous events in our home surrounds. The security in the home has to start the owner of the home and to look after the home. The finest way to guard the home is by connecting the security system in the home These systems are classified into two types, namely wired security system, and wireless security system.

Microcontroller Based Home Security System

Home Security System

Nowadays, most of them are using a wireless security system. It is simple to use and threatened by the unknown person. The **wireless security systems are motion sensor detector**, camera, etc. The home security systems are normally used by the motion sensor and monitor detector, that are placed in unknown places.

What is a Home Security System?

The home security system can be achieved by accepting important controllers to switch home appliances that notice different variables using an appropriate sensor. This system involves the security hardware such as property, doors, locks, lighting, security cameras and alarm systems. The main feature of this **home security system is a sensory system** that collects the information of the parameter such as temperature, gas, fire, human presence, etc., and sends the information to the Microcontroller.

Home Security System

This security system uses a microcontroller which is pre-programmed such that whenever the parameter crosses their specified limits, then it sends the control signals to various controlling devices such as relays, buzzer devices, and motors. The security system can be implemented by using the below function blocks.

Microcontroller Based Home Security System

The home security system project comprises of 3-basic modules, namely IR transmitter, IR receiver and IR sensor. Where IR transmitter and receiver modules work for the safety of doors from the thief in case we are out of the home. When the IR sensors are disturbed, a buzzer if switched to representing somebody has entered into the house. Another module is an **LPG gas sensor**. It is provided to notice the LPG Gas outflow. A buzzer will generate the sound when Gas is noticed by the sensor. The third part of the project helps the functionality of a door-latch opening using a password entered through a keypad. This module also makes the buzzer on if the three wrong passwords are entered consequently. The information from all the modules is sent to a computer through serial port. Microcontroller based home security system built with the following

IR (Infrared) Transmitter

The IR transmitter is used in the theft recognition module with the help of the one transmitter and one receiver. The IR transmitters are mainly used to detect the theft because an unauthorized person cannot noticeable to the human eyes.

IR (Infrared) Receiver

The IR (infrared) receiver is an active low device, therefore it gives the low o/p when the IR rays receive

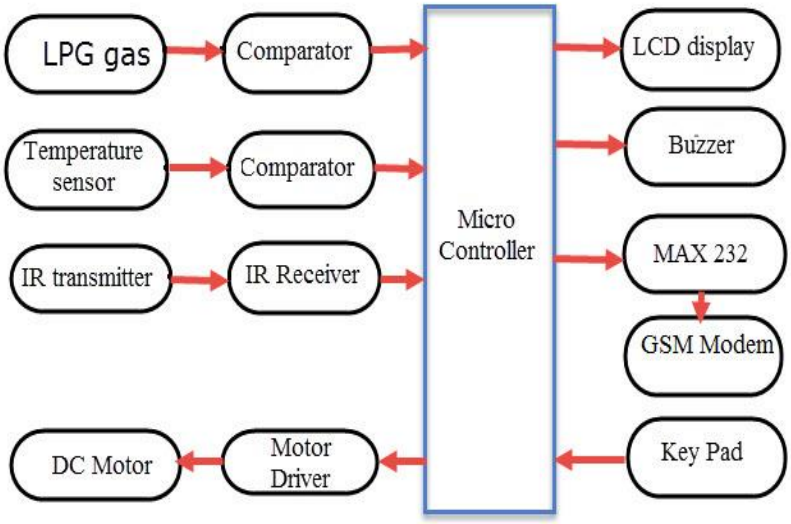
LPG Gas Sensor

The LPG gas sensor is one type of sensor used to notice the leakage of the gas in the home & also in kitchen rooms.

IR (Infrared) Transmitter

IR (Infrared) Receiver

LPG Gas Sensor



Microcontroller Based Home Security System

Microcontroller

The 8051 Microcontroller is nothing but the CPU. **The 8051 Microcontroller** includes the following functions which are described below.

- Reading in the digital i/p from the IR receiver.
- Reading the o/p of an **LPG gas sensor to generate the buzzer sound**.
- Detecting the password by keyword to check whether it is a correct password or not.
- Sending the information to the LCD display and to the serial port computer.

LCD

The proposed system uses a 16 * 2 alphanumeric LCD which contains 16 characteristics with two lines. Hence it can display the numbers and alphabets.

PC Interfacing

In the PC interfacing, we are **using the IC MAX 232** to interface a number of persons arrived in the room and also the state of entered password right or word is sent to the PC.

Keypad

By using the keypad the consumer can enter the password. There are different kinds of keypads are available like 0-9 keypad, Enter keypad, and Escape Keypad.

EEPROM

It is an external memory and used to store the passwords. For the communicative purpose of EEPROM with 8051 Microcontroller, the I2C bus protocol is used.

GSM Modem

In the proposed security system a is used as a GSM modem. It has to discover the number of persons inside the hall, a number of incorrect passwords entered on the keypad and also an outflow of LPG gas. The GSM modem can connect to the calls, SMS, or use MMS. In the wireless home security system, cameras play an important role. These are very easy to use and observed from the phone or from the personal computer. The camera can capture the tape from inside the signal and the footage can be kept on the computer hard drive.

Another feature of the security system is by using an alarm system. When an alarm generates the sound, then the authorized person will identify that another person has entered into the home or room. There are several reasons to generate the **security alarm sound** from restriction if doors are unlocked, and when the fire exits and a broken window. The **applications of home security system** mainly include houses for security purpose, industries, and educational institutes.
Video Content / Details of website for further learning (if any): https://slideplayer.com/slide/5911321/
Important Books/Journals for further learning including the page nos.: Rajkamal, Embedded Systems: Architecture, Programming and Design, Tata McGraw Hill, 2015, pp. 610

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L42

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : **Embedded Systems and RTOS & 19ECC15**

Course Faculty : **Dr.P.Padmaloshani**

Unit : **V - RTOS BASED EMBEDDED SYSTEM DESIGN**

Date of Lecture:

Topic of Lecture: Finger print identifiers

Introduction :

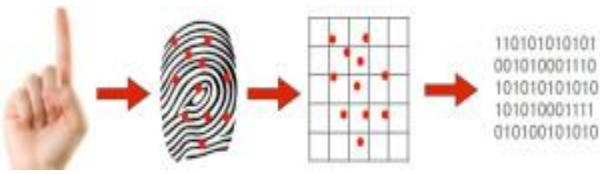
- **Embedded modules are most commonly used with fingerprint scanners firmly installed within the composition of the device**, for instance, in the case of laptops with inbuilt fingerprint verification; whereas USB fingerprint readers are more commonly used due to their portability and durability capabilities.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

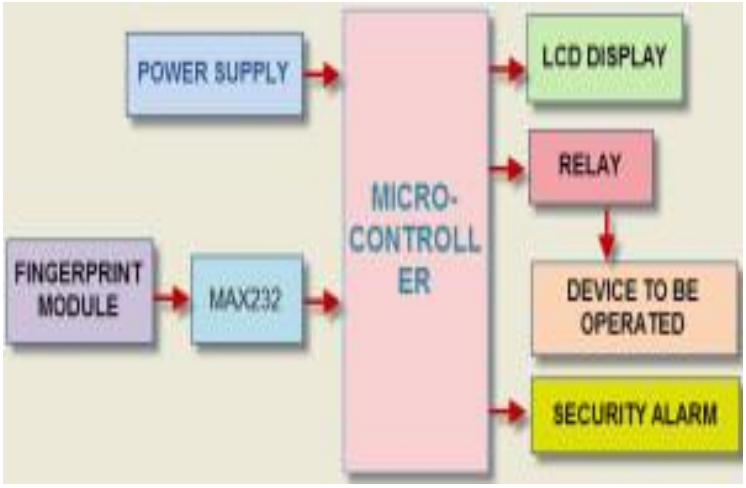
Finger print identifiers **:**

Fingerprints form a unique identification pattern for humans, which consist of a pattern of ridges on fingers that helps to grip things by hand.Fingerprint scanner is the heart of this automated authentication system which is responsible for the acquisition of images based on the patterns of ridges and valleys of human fingers, and then matching them with the pre-stored patterns. It consists of sensors that are optical, ultrasonic, thermal, capacitive, etc.,but mostly optical and capacitance scanning methods are used.



Fingerprint Devices

These sensors generate top quality images of finger ridges and valleys by overriding inconsistent and irregular designs of the scanned image.These scanners consist of Analog to digital converters which process the analog electric signals to produce digital representation of the image. When we press the finger on a fingerprint scanner, it collects the signals, processes the image and extracts minutiae information of the finger. Subsequently a processing unit or host acquires this ID information and store it in a fingerprint database.



Block Diagram of Fingerprint Authentication and Controlling System

Hardware Requirements

- Fingerprint sensor
- 8051 Microcontroller
- Buzzer
- Keyboard
- LCD display
- Capacitors
- Resistors
- Transistors

Software Requirements

- Keil Complier
- Embedded C or assembly language program
- A fingerprint module is interfaced to a microcontroller with a serial interfacing, and this project uses a relay, a Buzzer, an LCD, and switches.
- 4-pin fingerprint scanners consist of VCC, RX, TX and Ground pins and are connected to 10, 11, 16 pins of the microcontroller, as shown in the figure.
- The LCD is interfaced to thePORT0 of the microcontroller to display the information or data.
- A transistor driven buzzer is connected to the 24-pin of the microcontroller to give alarm for authentication.
- The pushbutton switches are connected to 1, 2, and 3input pins of the microcontroller for informing about the type of operation to themicrocontroller including scanning, adding and deleting fingerprint.
- The relay is connected to the25th pin of the microcontroller through a transistorto operate the loads or devices.

- The Microcontroller is programmed in embedded C language in Keil software and this hexa code is dumped to the microcontroller using a hardware circuitry.

This type of authentication for controlling devices and appliances is highly secured and more reliable at a lower cost due to inexpensive microcontroller. Thus, you might have got an idea of implementing a fingerprint authentication system by using a simple controller. This can also be implemented as an attendance system in schools and offices by adding external memory.

Video Content / Details of website for further learning (if any):
https://how2electronics.com/fingerprint-biometric-attendance-system-arduino/

Important Books/Journals for further learning including the page nos.:
Rajkamal, Embedded Systems: Architecture, Programming and Design, Tata McGraw Hill, 2015, pp. 620

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L43

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : **Embedded Systems and RTOS & 19ECC15**

Course Faculty : **Dr.P.Padmaloshani**

Unit : **V - RTOS BASED EMBEDDED SYSTEM DESIGN**

Date of Lecture:

Topic of Lecture: Printers

Introduction :

- Printer can be defined as one of the many peripheral devices that are specifically used for replicating the contents seen on the computer system onto a physical material, which is typically a paper of some kind.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Printers **:**

Printer can be defined as one of the many peripheral devices that are specifically used for replicating the contents seen on the computer system onto a physical material, which is typically a paper of some kind. There are many types of Printers available which include the classic toner-based printers, thermal printers, impact printers, daisy wheel printers, dot matrix printers, laser printers, inkjet printers, 3D printer, etc. The types of printers can also be classified based on its connectivity provisions such as the wired printers and wireless printer that can be connected with the wireless networks like Bluetooth, VLAN, wifi, etc.

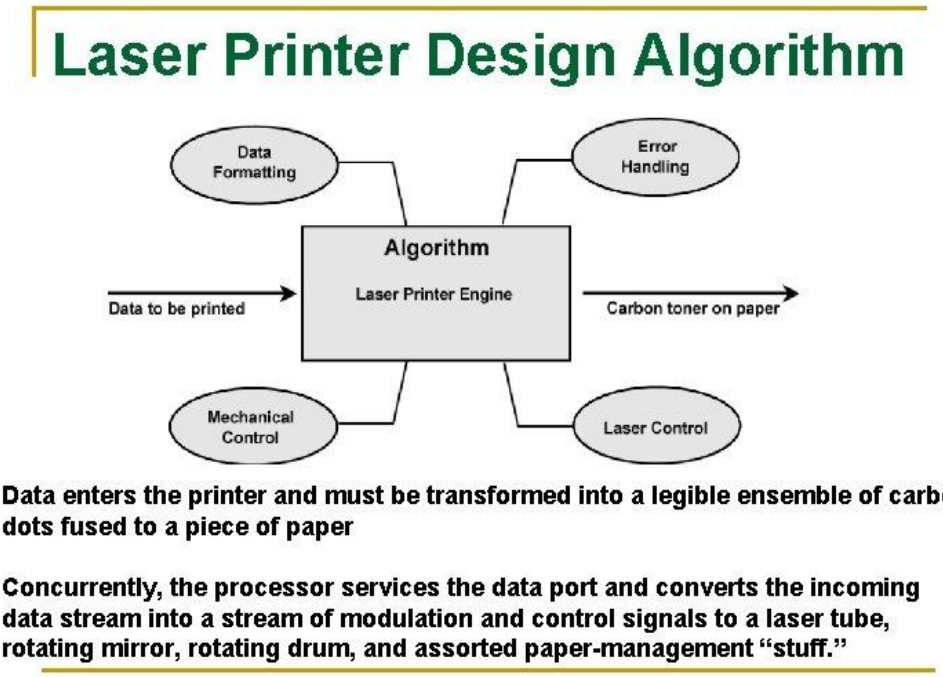
A typical printer should have all the basic parts in order to carry out the printing process successfully, and they are:

- Paper support
- Sheet feeder
- Printer cover
- Output tray
- Output tray extension
- connectors
- Edge guides
- Control Buttons

- Cartridge cover
- Print head

Laser Printer Design Algorithm

Data enters the printer and must be transformed into a legible ensemble of carbon dots fused to a piece of paper Concurrently, the processor services the data port and converts the incoming data stream into a stream of modulation and control signals to a laser tube, rotating mirror, rotating drum, and assorted paper-management "stuff. "



Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=aBX0V-XsxhQ

Important Books/Journals for further learning including the page nos.:
Rajkamal, Embedded Systems: Architecture, Programming and Design, Tata McGraw Hill, 2015, pp. 640

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L44

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : Embedded Systems and RTOS & 19ECC15

Course Faculty : Dr.P.Padmaloshani

Unit : V - RTOS BASED EMBEDDED SYSTEM DESIGN

Date of Lecture:

Topic of Lecture: Automated teller machine

Introduction :

- An automated teller machine (ATM) is a computerized machine used in banking that communicates with a host bank computer over a network. The bank computer verifies all the data entered by the users and stores all transactions, while the embedded system in the ATM displays the transaction data and processes inputs from the ATM keyboard.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

Automated teller machine **:**

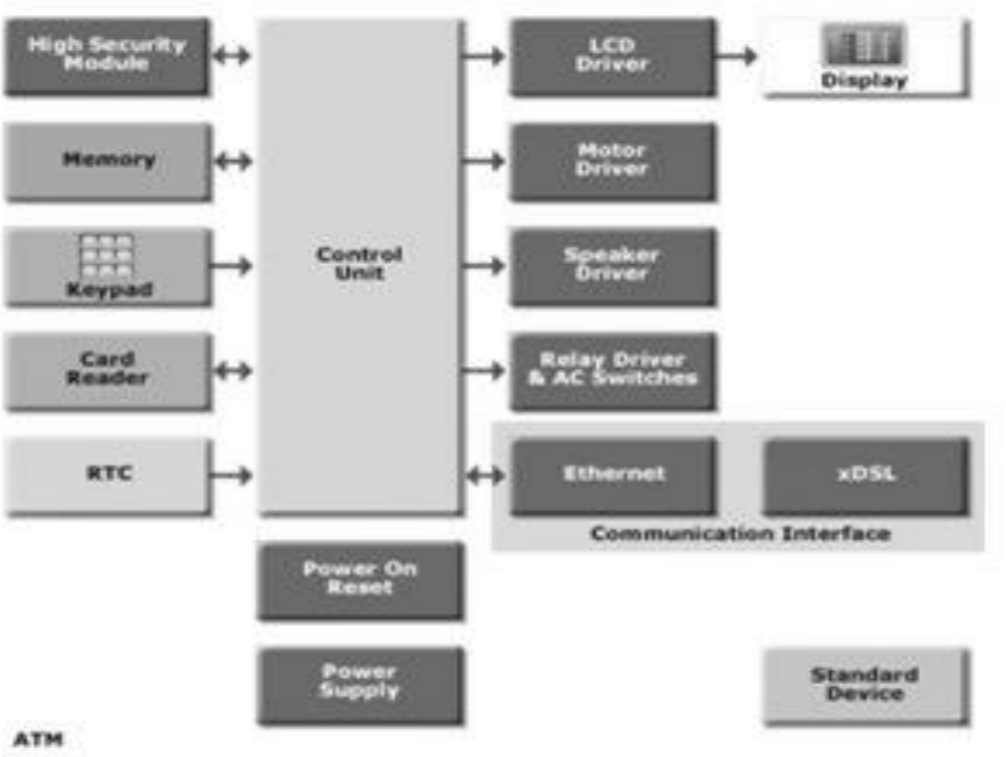
The term ATM stands for automated teller machine. It is an electronic device that is used by only bank customers to process account transactions. The users access their accounts through a special type of plastic card that is encoded with user information on a magnetic strip. The strip contains an identification code that is transmitted to the bank's central computer by modem. The users insert the card into ATMs to access the account and process their account transactions. The automated teller machine was invented by John Shepherd-Barron in the year 1960.

By using an automated teller machine or ATM we can perform different financial transactions such as cash deposits, withdrawals, transfer funds, information of account, ATM PIN change, and also linking the Aadhaar number to the bank account so that the interaction between the bank staff and the customer can be reduced.

The automated teller machine (ATM) is an automatic banking machine (ABM) that allows the customer to complete basic transactions without any help from bank representatives. There are two types of automated teller machines (ATMs). The basic one allows the customer to only draw cash and receive a report of the account balance. Another one is a more complex machine that accepts the deposit, provides credit card payment facilities and reports account information

History of Automated Teller Machines

In India, the development of ATMs was very slow because they were launched in the early 1990s to the Indians and assisted through foreign banks.
When most of the banks were facing severe problems because of the lack of a strong branch system, then automated teller machines appeared like the best solution to these problems.
The ATM machine was implemented to reduce the obstructions of the branch networking by reaching out to the customers by offering comfortable services with fewer transaction charges. After that, many improvements in this technology have arrived & also customer accessibility has also improved through boundaries.
The **block diagram of the automated teller machine** consists of mainly two input devices and four output devices. The input devices like card reader and keypad whereas output devices are speaker, display screen, receipt printer, and cash depositor.



Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=IhLBcrAIjBA

Important Books/Journals for further learning including the page nos.:
Rajkamal, Embedded Systems: Architecture, Programming and Design, Tata McGraw Hill, 2015, pp. 650

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L45

LECTURE HANDOUTS

ECE

III / VI

Course Name with Code : Embedded Systems and RTOS & 19ECC15

Course Faculty : Dr.P.Padmaloshani

Unit : V - RTOS BASED EMBEDDED SYSTEM DESIGN

Date of Lecture:

Topic of Lecture: Software modem, audio player

Introduction :

- Embedded modems **contain a data pump, DAA (data access arrangement) and modem code**. They may or may not include a controller, Internet protocol stack, and SDRAM. The number of chips depends on the application requirements, including the need for any additional functions
- Audio player (software), a piece of computer software for playing audio files.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic knowledge in microprocessors and microcontrollers-based system design

SOFTWARE MODEM :

Low-cost modems generally use specialized chips, but some PCs implement the modem functions in software.

Theory of Operation and Requirements:

The modem will use ***frequency-shift keying (FSK)***,a technique used in 1200-baud modems. As shown in Figure 5.14, the FSK scheme transmits sinusoidal tones, with 0 and 1 assigned to different frequencies. Sinusoidal tones are much better suited to transmission over analog phone lines than are the traditional high and low voltages of digital circuits.

The analog input is sampled and the resulting stream is sent to two digital filters (such as an FIR filter). One filter passes frequencies in the range that represents a 0 and rejects the 1- band frequencies, and the other filter does the converse.

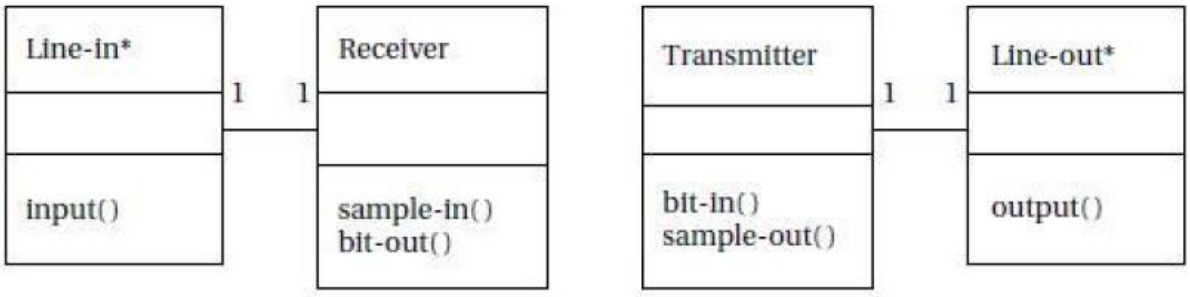
The outputs of the filters are sent to detectors, which compute the average value of the signal over the past *n* samples. When the energy goes above a threshold value, the appropriate bit is detected.

The receiver will detect the start of a byte by looking for a start bit, which is always 0. By measuring the length of the start bit, the receiver knows where to look for the start of the first bit. However, since the receiver may have slightly misjudged the start of the bit, it does not immediately try to detect the bit.

Instead, it runs the detection algorithm at the predicted middle of the bit.

Requirements:

Name	Modem
Purpose	A fixed baud rate frequency-shift keyed modem.
Inputs	Analog sound input, reset button.
Outputs	Analog sound output, LED bit display.
Functions	**Transmitter:** Sends data stored in microprocessor memory in 8-bit bytes. Sends start bit for each byte equal in length to one bit. **Receiver:** Automatically detects bytes and stores results in main memory. Displays currently received bit on LED.
Performance	1200 baud
Manufacturing cost	Dominated by microprocessor and analog I/O.
Power	Powered by AC through a standard power supply.
Physical size and weight	Small and light enough to fit on a desktop



Class diagram for the modem

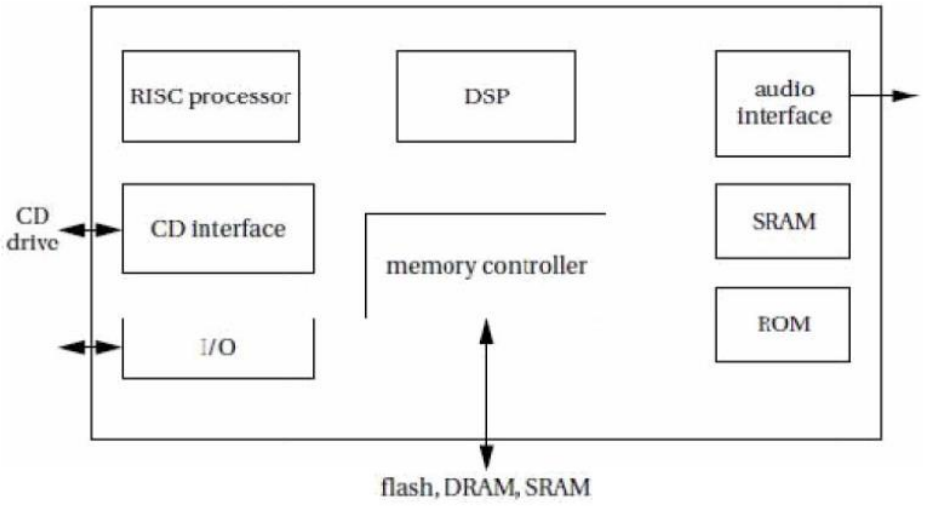
Audio Player:

Audio players are often called ***MP3 players*** after the popular audio data format. The earliest portable MP3 playerswere based on compact disc mechanisms. Modern MP3 players use either flash memory or disk drives to store music.

An MP3 player performs three basic functions: audio storage, audio decompression, and user interface. Although audio compression is computationally intensive, audio decompression is relatively lightweight. The incoming bit stream has been encoded using a Huffman-style code, which must be decoded. The audio data itself is applied to a reconstruction filter, along with a few other parameters. MP3 decoding can, for example, be executed using only 10% of an ARM7 CPU.

The user interface of an MP3 player is usually kept simple to minimize both the physical size and power consumption of the device. Many players provide only a simple display and a few buttons.

The file system of the player generally must be compatible with PCs. CD/MP3 players used compact discs that had been created on PCs. Today's players can be plugged into USB ports and treated as disk drives on the host processor.



Architecture of Cirrcus audio processor for CD/MP3 players

The Cirrus CS7410 is an audio controller designed for CD/MP3 players. The audio controller includes two processors.
The 32-bit RISC processor is used to perform system control and audio decoding.
The 16-bit DSP is used to perform audio effects such as equalization. The memory controller can be interfaced to several different types of memory: flash memory can be used for data or code storage; DRAM can be used as a buffer to handle temporary disruptions of the CD data stream.
The audio interface unit puts out audio in formats that can be used by A/D converters.
General-purpose I/O pins can be used to decode buttons, run displays, etc. Cirrus provides a reference design for a CD/MP3 player.

Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=aBX0V-XsxhQ

Important Books/Journals for further learning including the page nos.:
Rajkamal, Embedded Systems: Architecture, Programming and Design, Tata McGraw Hill, 2015, pp.670

Course Faculty

Verified by HOD