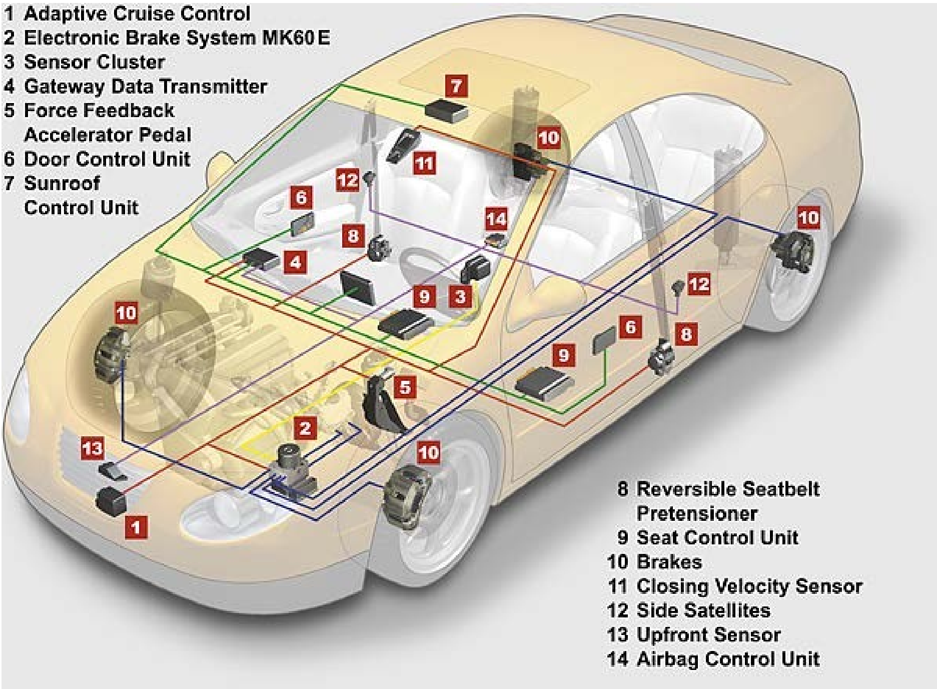
MODULE 4
EMBEDDED SYSTEMS


1 Adaptive Cruise Control
2 Electronic Brake System MK60E
3 Sensor Cluster
4 Gateway Data Transmitter
5 Force Feedback Accelerator Pedal
6 Door Control Unit
7 Sunroof Control Unit
8 Reversible Seatbelt Pretensioner
9 Seat Control Unit
10 Brakes
11 Closing Velocity Sensor
12 Side Satellites
13 Upfront Sensor
14 Airbag Control Unit

An embedded system can be defined as a computing device that does a specific focussed job. Both the hardware and software in an embedded system are opimized for that specific job. Appliances such as the air conditioner,VCD player, DVD player,Printer, Fax machine, Mobile phones etc. are examples of embedded systems.

Each of these appliances will have a processor or controller and special hardware to meet the specific requirement of the application along with the embedded software that is executed by the processor for meeting that specific requirement.

The embedded software is called firmware

CHARACTERISTIC FEATURES OF EMBEDED SYSTEMS

1. Embedded systems do a very specific task,they cannot be programmed to do different things.

2. Embedded systems have very limited resources, particularly the memory.
   They do not have secondary storage devices such as the CDROM or the floppy disk.

3. Embedded systems have to work against some deadlines. A specific job has to be completed within a specific time. In some Embedded systems , called real-time systems, the deadlines are stringent. Missing a deadline may cause a catastrophe – loss of life or damage property.
4. Embedded systems are constrained for power. As many embedded systems operate through a battery, the power consumption has to be very low.

5. Embedded systems need to be highly reliable. Once in a while, pressing ALT-CTRL-DEL is OK on our desktop,but we cannot afford to reset our embedded system.

6. Some Embedded systems have to operate in extreme environmental conditions such as very high temparature and humidity.

7. Embedded systems that address the consumer market are very cost sensitive. Choosing the right platform(hardware and software) is the most complex task.

APPLICATION AREAS

Embedded system market is one of the highest growth areas as these systems are used in very market segment- consumer electronics, office automation, industrial automation, biomedical engineering, wireless communication, data communication, telecommunications, transportation,military and so on.

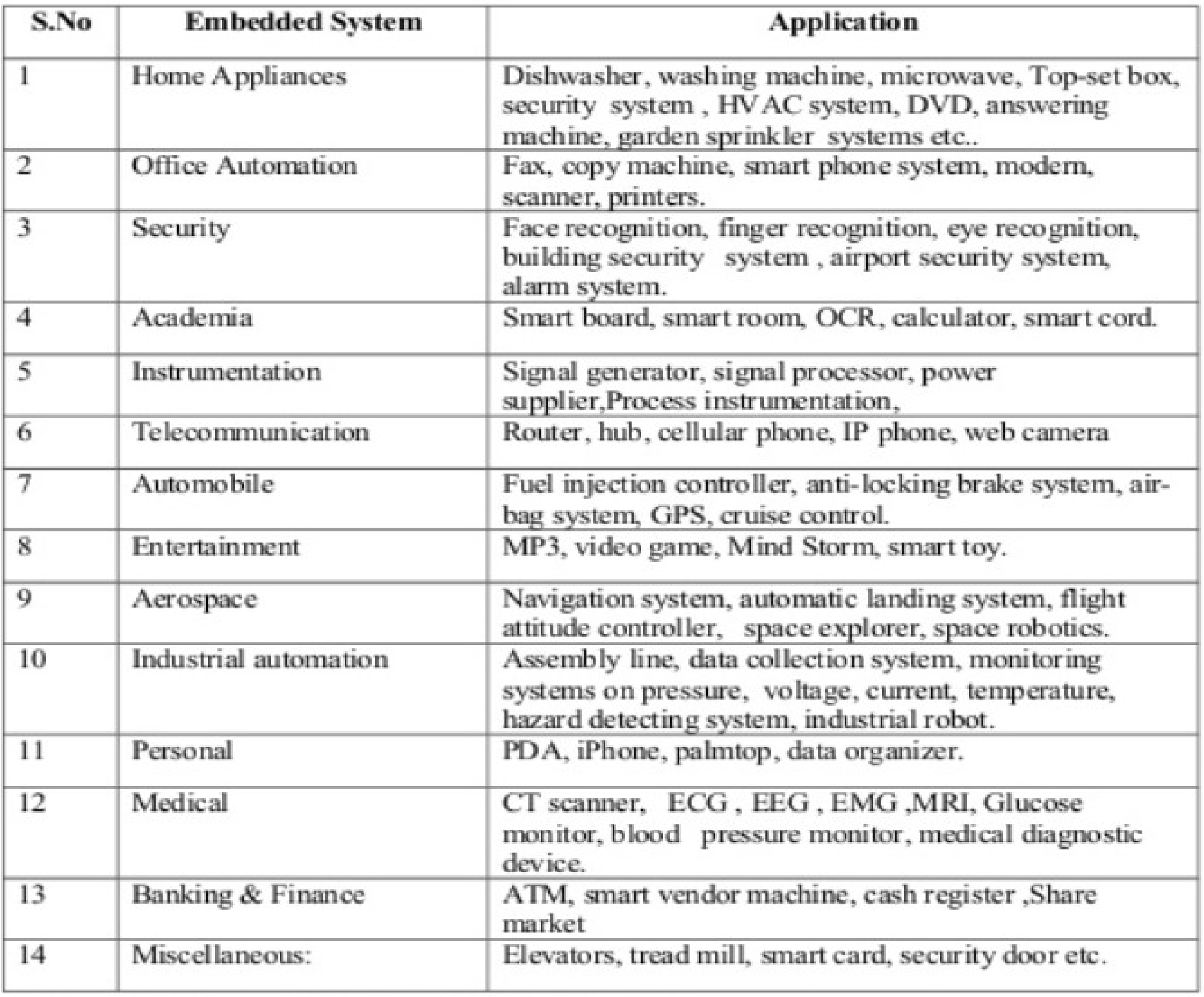
Consumer Appliances – Digital camera, digital diary, DVD player, electronic toys,microwave oven. Todays high tech car has has a lot of embedded systems for transmission control, engine spark control, air conditioning ,navigation etc.

Office automation – Copying machine,fax machine, keytelephone,modem,printer,scanner etc,
Industrial automation – A lot of industries use embedded systems for process control. These include pharmaceutical,cement,sugar,oil exploration, nuclear energy, electricity generation and transmission.
The Embedded systems for industrial use are designed to carry out specific tasks such as monitoring the temparature, pressure, humidity,voltage , current etc, and then take appropriate action based on the monitored level to control other devices or to send information to a centralized monitoring station.
In hazardous industrial environment, where human presence has to be avoided ,robots are used,which are programmed to do specific jobs. The robots are now becoming very powerful and carry out many interesting and complicated tasks such as hardware assembly.
Medical electronics – Almost every medical equipment in the hospital is an embedded system.Like ECG, EEG, Blood pressuring measuring devices, X-ray, Scanners, Equipment used in blood analysis, radiation, etc'

Computer Networking – Computer networking products such as bridges, routers, Integrated Services Digital Networks(ISDN), Asynchronous Transfer Mode (ATM), etc. are uses embedded systems.
Telecommunications - embedded systems are used in subscriber terminals and network equipment.Subscriber terminals include like keytelephones, ISDN phones, terminal adapters and the network equipment includes multiplexers ,satellite modems.

Wireless technologies – Advances in mobile communications are paving are paving way for many interesting applications using embedded systems. The Personal Digital Assistants(PDA) and the palmtops can now be used to access multimedia services over the internet. Mobile communication infrastructure such as base station controllers,mobile switching centers are also powerful embedded systems.
Instrumentation – Test equipments such as oscilloscope, spectrum analyzer, logic analyzer, radio communication test set etc are embedded systems built around powerful processors.
Security – Security devices at homes, offices, airports etc for authentication and verification are embedded systems. Biometric systems using fingerprint and face recognition are now being extensively used for user authentication in banking applications as well as for access control in high security buildings

Finance – Financial dealing through cash and cheques are now slowly paving way for transactions using smart cards and ATM. Smart card has a small micro controller and memory, and it interacts with the smartcard reader/ATM machine and acts as an electronic wallet.



S.No	Embedded System	Application
1	Home Appliances	Dishwasher, washing machine, microwave, Top-set box, security system , HVAC system, DVD, answering machine, garden sprinkler systems etc..
2	Office Automation	Fax, copy machine, smart phone system, modem, scanner, printers.
3	Security	Face recognition, finger recognition, eye recognition, building security system , airport security system, alarm system.
4	Academia	Smart board, smart room, OCR, calculator, smart cord.
5	Instrumentation	Signal generator, signal processor, power supplier,Process instrumentation,
6	Telecommunication	Router, hub, cellular phone, IP phone, web camera
7	Automobile	Fuel injection controller, anti-locking brake system, air-bag system, GPS, cruise control.
8	Entertainment	MP3, video game, Mind Storm, smart toy.
9	Aerospace	Navigation system, automatic landing system, flight attitude controller, space explorer, space robotics.
10	Industrial automation	Assembly line, data collection system, monitoring systems on pressure, voltage, current, temperature, hazard detecting system, industrial robot.
11	Personal	PDA, iPhone, palmtop, data organizer.
12	Medical	CT scanner, ECG , EEG , EMG ,MRI, Glucose monitor, blood pressure monitor, medical diagnostic device.
13	Banking & Finance	ATM, smart vendor machine, cash register ,Share market
14	Miscellaneous:	Elevators, tread mill, smart card, security door etc.

SPECIALITIES OF EMBEDDED SYSTEMS

1. Reliability

Reliability is of paramount important in embedded systems.They should continue to work for thousands of hours without break.The design of the embedded system should be such that in case the system has to be reset, the reset should be done automatically.

Special hardware/software needs to be built into the system to take care of it. This special module is known as watchdog timer.

Many embedded systems used in industrial automation and defence equipment need to work in extreme environmental conditions such as very high/low temparatures,high humidity. Besides, they should be able to withstand bump and vibrations. Hence, very stringent environmental specifications have to be met by such systems. The ability to work reliably in extreme environmental conditions is known as ruggedness.

2. Performance

The system must meet some deadlines like, temparature, time etc. If the deadlines are missed, it may result in a catastropohe. So it must work against all deadlines.

3.Power Consumption

Most of the embedded systems operate through a battery. To reduce the battery drain and avoid frequent recharging of the battery, the power consumption of the embedded system has to be very low. To reduce power consumption such hardware components should be used that consume less power. So the component count must be minimum.Reducing the component count apart from reducing the power consumption also increases the reliability of the system.

4. Cost

The cost of the embeded system must be low.

5. Size

The size and weight are the important parameters in embedded systems used in somany areas like aircraft, spacecraft etc.

6. Limited User Interface.

Embedded systems do not have sophisticated interfaces for input and output.Some embedded systems do not have any user interface at all.

They take electrical signals as input and produce electrical signals as output. In many embedded systems, the input is through a small function keyboard or a set of buttons. The output is displayed either on a set of LEDs or a small LCD.

7. Software Upgradation Capability.

Embedded systems are meant for a very specific task.So, once the software is transferred to the embedded system, the same software will run throughout its life. However, in some cases, it may be necessary to upgrade the software, like Public Call Office. In such cases software upgradation can be done by downloading the software on to the embedded system through a network connection.

EMBEDDED SYSTEM ARCHITECTURE

Every embedded system consists of custom-built hardware built around a Central Processing Unit (CPU).

This hardware also contains memory chips onto which the software is loaded. The software residing on the memory chip is also called the firmware.

The operating system runs above the hardware, and the application software runs above the operating system.

It is not compulsory to have an operating system in every embedded system.

For applications involving complex processing, it is advisable to have an operating system.In such a case, you need to integrate the application software with the operating system and then transfer the entire software on to the memory chip.

Once the software is transferred to the chip, the software will continue to run for a long time and no need to reload new software.

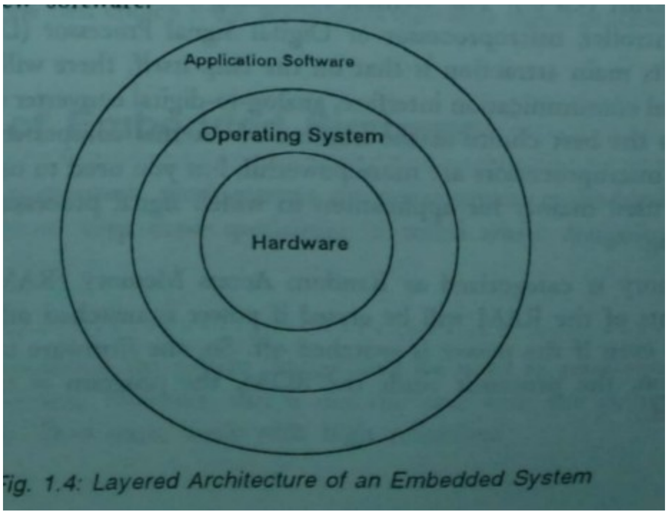
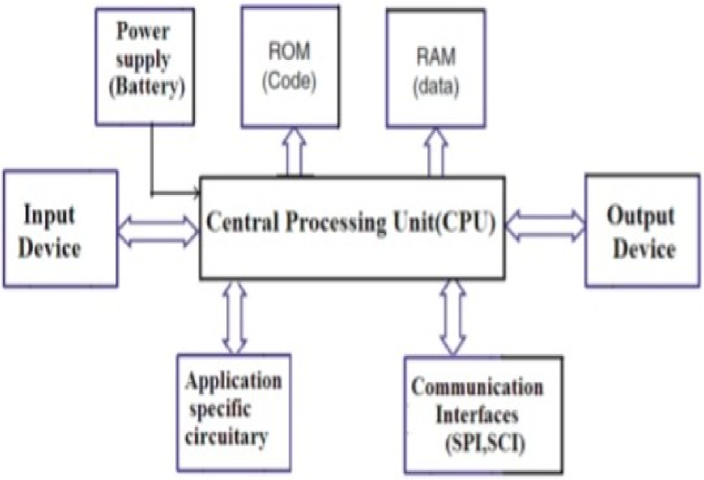


Fig. 1.4: Layered Architecture of an Embedded System

The basic building blocks of embedded systems are
1. Central Processing Unit(CPU)
2. Memory(ROM and RAM)
3. Input Devices
4. Out Devices
5. Communication interfaces
6. Application – specific circuitry.


Power supply (Battery)
ROM (Code)
RAM (data)
Input Device
Central Processing Unit(CPU)
Output Device
Application specific circuitary
Communication Interfaces (SPI,SCI)

Central Processing Unit(CPU) – The CPU can be any of the following ; microcontroller, microprocessor or Digital Signal Processor(DSP).

A microcontroller is a low-cost processor. For small applications, a microcontroller is the best choise as the number of external components required will be very less.

Microprocessors are more powerful, but need to use many external components with them.

DSP is used mainly for applications in which signal processing is involved such as audio and video processing.

Memory – It is categorized as RAM and ROM.The firmware is stored in the ROM. When power is switched on,the processor reads the ROM,the programis transferred to RAM and the program is executed.

Input Devices – The input devices to an embedded system have very limited capability. Many embedded systems will have a small keypad. A keypad may be used to input only the digits. Many embedded systems used in process control do not have any input device for user interaction; they take inputs from sensors or transducers and produce electrical signals that are in turn fed to other systems.

Output devices – The output devices of the embedded systems also have very limited capability. Some embedded systems will have a few LEDs to indicate the health status of the system modules, or for visual indication of alarms. A small LCD may also be used to display some important parameters.

Communication Interfaces – The embedded systems may need to interact with other embedded systems or they may have to transmit data to a desktop.To facilitate this, the embedded systems are provided with one or a few communication interfaces such as RS232,USB,Ethernet etc.

Application – specific circuitry – Sensors, transducers , Actuators special processing and control circuitry may be required for an embedded system, depending on its application. This circuitry interacts with the processor to carry out the necessary work.

The entire hardware has to be given power supply either through the 230V mains supply or through a battery. The hardware has to be designed in such a way that the power consunption is minimised.

TYPES OF OPERATING SYSTEMS

Based on the capabilities, operating systems are divided into

1. Single tasking and multi tasking

In a Single tasking OS, only one task is carried out at a time. eg word processor

In a multi tasking OS, multiple tasks can be run simultaneously.
eg. Play game and downloading internet at a time

In a multi tasking system, there may be only one CPU, but the CPU is shared by all the tasks.
MS DOS is a single OS. Windows and UNIX are multitasking OS.

2. Single User OS and Multi-User OS

In a Single User OS, only one user can use the system at a time.

In a Multi-User OS, multiple users can share the system simultaneously. Here, the application run by each user is given a small slice of time. However each user feels as though he is alone using the system. This is known as time-sharing.

MS-DOS and Windows are single user OS, whereas UNIX is a multi-user OS.

3.Command-driven and GUI-based OS.

An important function of the OS is to provide an interface to the user to access the computers resources. The user has to give instructions to the computer to carry out various tasks.

One method of giving these instructions is to give commands such as “copy x.c y.c” through the keyboard. Operating system working on commands are said to be command-driven. MS DOS is a command driven OS.

On, the other hannd, the OS can provide a GUI such as in Windows OS and give instructions through mouse click. This is a GUI-driven OS.

Special Requirements of Embedded Operating Systems.

1. Reliability

The OS in an embedded system has to be very reliable.

OS will manage the overall resources effectively.

2. Multi tasking with time constrints.

Embedded systems need to support multi-tasking.The task management has to be done efficiently to meet the real time performance requirements. Embedded systems must meet definite deadlines for some tasks.

3. Small footprint

As the memory devices have limited capacity in embedded systems, they have little memory for the OS.The memory occupied by the operating system is known as the footprint. The footprint should be very small for an embedded OS.

4. Support diskless systems

Unlike the desktop computers, embedded systems may not have secondary storage such as hard disk or CDROM. The embedded OS along with application software will reside on a memory chip.

File system management is not mandatory in embedded systems.If the application demands a file system,we can create a small file system on the flash memory.

5. Portability

A variety of processors are available for developing embedded systems. So,an important requirement of embedded operating system is portability.

6. Scalability

The embedded OS may be used on an 8-bit microcontroller or a powerful 64-bit microprocessor. So, scalability is very important for embedded OS.

7. Support for standard API

Application software is developed using the Application Programming Interface (API) of the OS. API is a set of function calls. An application developed for one OS may not be portable to another OS. To achieve portability, IEEE standardized the API called Portable Operating System Interface(POSIX). OS used in embedded systems must comply with this standard.

ARCHITECTURE OF EMBEDDED OPERATING SYSTEMS

The software in an ES can be divided into the operating system and the application software.The inbetween layer, the Application Progrmming Interface(API), used by the developers to write the application software.API provides the function calls to access the OS services.

The Operating System consists of
1.Kernel
2.Device manager

3.Networking protocol software
4.Libraries
5.File System(Optional)

Kernel

Kernel manages the tasks to achieve the desired performance of the ES. To manage the tasks, the important requirements are schedule the tasks and to provide inter-task communication facilities. To achieve these two requirements, Kernel objects are defined such as tasks, mutexes,ISRs, events, message boxes, mailboxes, pipes and timers. Kernel provides the memory management services, time management services, interrupt handling services and device management services.
Kernel is the heart of the OS. Its main fuction is to manage the tasks. A task scheduling algorithm decides which task has to run next.

Device Manager

The I/O devices are used to send/receive data from the embedded system. The OS manage the I/O devices through interrupts and device drivers.Device drivers provide the necessary interface between the applicatio and the hardware.

Communication Protocol Software

If the embedded system has communication interfaces such as Ethernet, USB etc. The upper layer protocols such as TCP/IP stack need to be integrated with the OS. Then the embedded system can be network enabled.

Libraries

The OS may have some C/C++ library filesin object code, which can be used through the API calls.

File System

Most of the Embedded Systems do not have a secondary storage. In such cases, The ROM is used to store the program. In case a file system is required, a small file system can be developed on flash memory. Some ES may use a secondary storage just for booting.

Important Steps of developing embedded software

1. Identifying the tasks and assigning priorities to each task

2.Identifying the the time critical tasks

3. Scheduling the tasks in such a way that all the tasks are completed and and the tasks which are time-critical meet the deadlines.

4. Working out when to send interrupts to the processor.

5. Identifying the shared resources and working out mechanisms for sharing the resourse by multiple tasks.

6.Working out the strategies for inter-task communication.

7.Keeping track of the time RTC

Functions of OS kernel

The kernel is the heart of the OS, and its main functions are given below.

1. Task scheduling
2. Inter-task communication
3. Interrupt handling
4. Providing timer service
5. device management and
6. Memory management.

Tasks

The work to be done by an ES is divided into a number of tasks. Each task copetes for the CPU time independently.Each task will have its own stack area.Each tasks runs forever and hence it is implemented as an infinite loop. Tasks are similar to processes in the sense that both compete independently for CPU time. However, the difference is that process have better memory protection features. A task/process is further divided into threads. Each thread runs independently, but threads in a task can share variables.,

To meet the special requirements ,the embeddedsystems need to have mechanisms to manage the tasks and to make the tasks communicate with one another.

Task Scheduling

Since only one CPU has to handle multiple tasks, the tasks have to share the CPU time in a discipled way so that one task does not get lot of time while others wait for unduly long time. Important time critical tasks have to be given higher priority; and a mechanism for deciding which task will get the CPU time next has to be worked out. This is known as task scheduling.

If the ES has to meet realtime requirements, the task scheduling has to be done in such a way that the time deadlines are met. An operating system in which the time that is required to execute a task can be estimated is called a determinstic OS – we should be able to determine the worst case timings for completing a task.

In all embedded/real time OS, a scheduling algorithm known as priority based pre-emptive scheduling algorithm is used. In this algorithm,each task is given a priority. The highest priority task will always be given the CPU time first.

Context Switching

Suppose a low priority task is presently being executed by the processor but a high priority task has to run. In this case, CPU will be interrupted through an interrupt signal. The CPU will save the current tasks information in a stack and execute the high priority task. The mechanism of storing the current CPU registers in a stack to run the other task is known as context switching.

Mutual exclusion

Different tasks may have to share same resources,like printer.Then the tasks should maintain discipline to share the resources. Ensuring that two or more tasks access a shared resource without corrupting the data is called mutual exclusion. ES provide disciplined access to shared resources through special objects such as mutexes and semaphores. Mutex and semaphore are like keys. To access a resource, a task will obtain the key, use the resource and release the key.

Inter task Communication

Tasks may need to exchange data themselves.For instance, a task may write some data to a file and another task has to read that data. This mechanism is known as Inter task Communication. The task which has to read the data has to obtain information that data is written by another task. So the two tasks have to synchronize their activities. Special OS objects such as mailboxes,message queues, pipes, status registers, and event flags are used to achieve Inter task Communication and Inter task synchronization.

Memory Management

The memory of an ES is to be shared by a number of tasks. So, memory management is another service provided by the OS.

Timer Services.

The OS needs to keep track of the time for activities such as the time for which a particular task is running, the time for which a task can wait for a shared resource, etc. The OS also provides the timer service.

In addition to the application tasks, the kernel has its own own system tasks with its own priorities. One important system task is the idle task. If no other task is running, the idle task is executed.

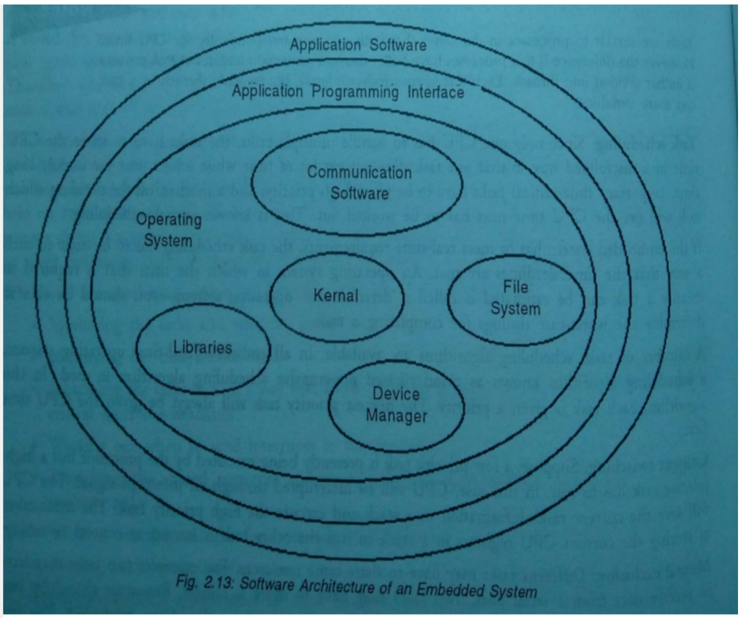


Fig. 2.13: Software Architecture of an Embedded System

The kernel is the central module of an operating system (OS).

Typically, the kernel is responsible for memory management, process and task management, and disk management.

The kernel connects the system hardware to the application software.

Every operating system has a kernel.

Categories of Embedded Operating Systems

It is broadly divided into the following categories.

1. Non-Real time embedded operating systems

These OS have a small footprint but they are not suitable for hard real time applications. Examples are Embedded Linux, Embedded NT, Windows XP Embedded. These OS are used in consumer appliances such as DVD players, Set Top Boxes etc. Which are soft soft real time embedded systms.

2. Real time embedded operating systems

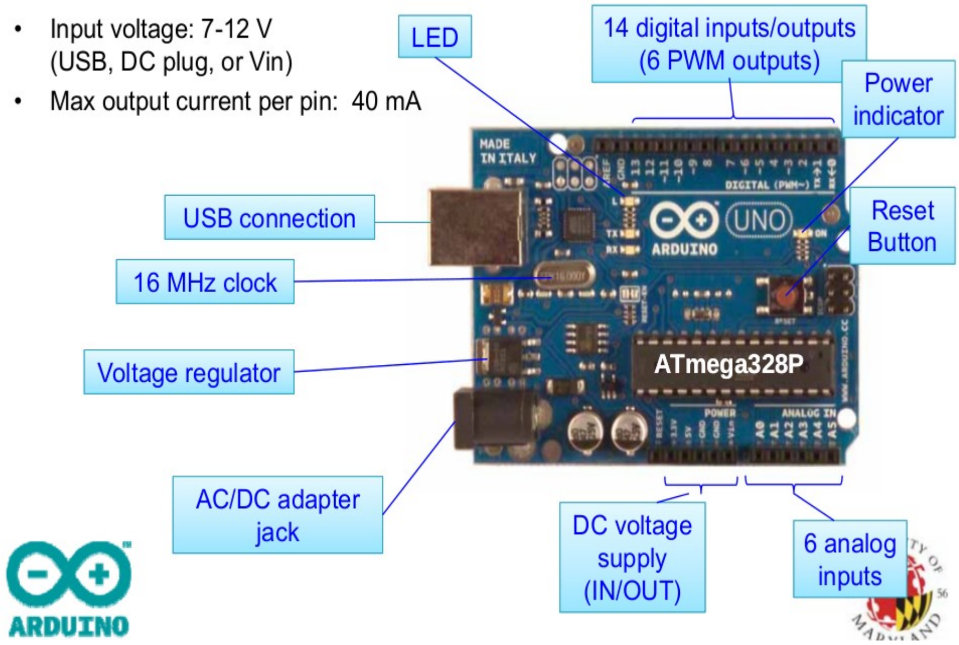
These are suitable for hard real-time embedded applications. Examples are QNX Neutrino, VxWorks, RTLinux, MicroC/OS-II and OS/9.

3. Mobile/handheld operating systems

The operating systems used in mobile devices such as palmtops, PDAs, smartphones are known asymbian OS, Windows CE are the major Oss in this category. Embedded Linux, VxWorks, OS/9 are also now being used in hand held computers.

Arduino

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

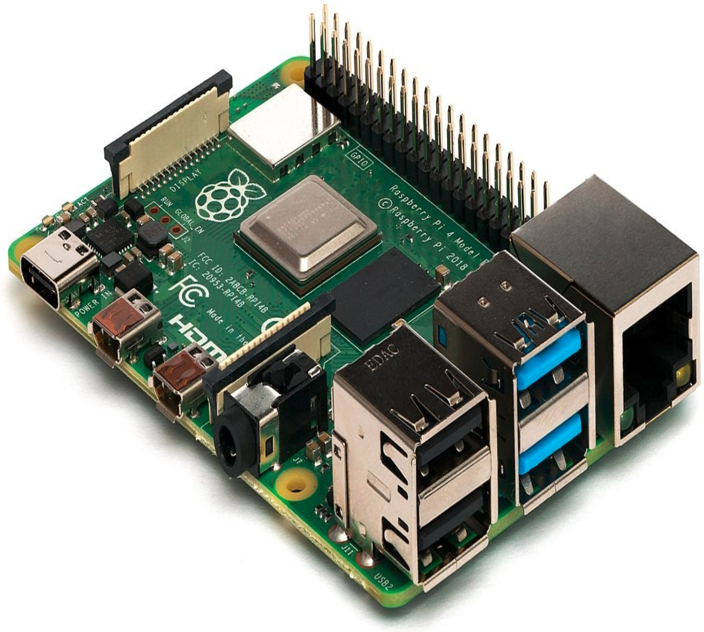

• Input voltage: 7-12 V (USB, DC plug, or Vin)
• Max output current per pin: 40 mA
LED
14 digital inputs/outputs (6 PWM outputs)
Power indicator
USB connection
16 MHz clock
Reset Button
Voltage regulator
ATmega328P
AC/DC adapter jack
DC voltage supply (IN/OUT)
6 analog inputs
ARDUINO
UNO

The Arduino Uno is a microcontroller board based on the ATmega328 .
It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button.
It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.
The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.
Latest version -R3

The Raspberry Pi

The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation. Early on, the Raspberry Pi project leaned towards the promotion of teaching basic computer science in schools and in developing countries. It is now widely used in many areas, such as for weather monitoring, because of its low cost and high portability.

It does not include peripherals (such as keyboards and mouce) or cases. However, some accessories have been included in several official and unofficial bundles.



The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It's capable of doing everything you'd expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.

Several generations of Raspberry Pis have been released. All models feature a Broadcom system on a chip (SoC) with an integrated ARM-compatible central processing unit (CPU) and on-chip graphics processing unit (GPU).

The Raspberry Pi launched in 2012, and there have been several iterations and variations released since then. The original Pi had a single-core 700MHz CPU and just 256MB RAM, and the latest model has a quad-core 1.4GHz CPU with 1GB RAM.

Raspberry Pi 4 Model B was released in June 2019 with a 1.5 GHz 64-bit quad core ARM Cortex-A72 processor, on-board 802.11ac Wi-Fi, Bluetooth 5, full gigabit Ethernet , two USB 2.0 ports, two USB 3.0 ports, and dual-monitor support via a pair of micro HDMI (HDMI Type D) ports for up to 4K resolution. The Pi 4 is also powered via a USB-C port, enabling additional power to be provided to downstream peripherals, when used with an appropriate PSU.