

UNIT- II TYPICAL EMBEDDED SYSTEM

2.1) CORE OF THE EMBEDDED SYSTEM

Embedded systems are domain and application specific and are built around a central core. The core of the embedded system falls into any one of the following categories:

(1) General Purpose and Domain Specific Processors

1.1 Microprocessors

1.2 Microcontrollers

1.3 Digital Signal Processors

(2) Application Specific Integrated Circuits (ASICs)

(3) Programmable Logic Devices (PLDs)

(4) Commercial off-the-shelf Components (COTS)

If you examine any embedded system you will find that it is built around any of the core units mentioned above.

2.1.1) GENERAL PURPOSE AND DOMAIN SPECIFIC PROCESSORS

Almost 80% of the embedded systems are processor/controller based. The processor may be a microprocessor or a microcontroller or a digital signal processor, depending on the domain and application. Most of the embedded systems in the industrial control and monitoring applications make use of the commonly available microprocessors or microcontrollers whereas domains which require signal processing such as speech coding, speech recognition, etc. make use of special kind of digital signal processors supplied by manufacturers like, Analog Devices, Texas Instruments, etc.

2.1.1.1) Microprocessors

- A silicon chip representing a Central Processing Unit (CPU), which is capable of performing arithmetic as well as logical operations according to a pre-defined set of Instructions, which is specific to the manufacturer.
- In general the CPU contains the Arithmetic and Logic Unit (ALU), Control Unit and working registers
- Microprocessor is a dependant unit and it requires the combination of Hardware like Memory, Timer Unit, and Interrupt Controller etc for proper functioning.
- Intel claims the credit for developing the first Microprocessor unit Intel 4004 a 4- bit processor which was released in Nov 1971.

2.1.1.3) Microcontrollers:

- A highly integrated silicon chip containing a CPU, scratch pad RAM, Special and General Purpose Register Arrays, On Chip ROM/FLASH memory for program storage, Timer and Interrupt control units and dedicated I/O ports.
- Microcontrollers can be considered as a super set of Microprocessors
- Microcontroller can be general purpose (like Intel 8051, designed for generic applications and domains) or application specific (Like Automotive AVR from Atmel Corporation. Designed specifically for automotive applications)
- Since a microcontroller contains all the necessary functional blocks and independent working, they found greater place in the embedded domain in place of microprocessors
- Microcontrollers are cheap, cost effective and are readily available in the market Texas Instruments TMS 1000 is considered as the world's first microcontroller.

2.1.1.4) Microprocessor vs Microcontroller

The following table summarises the differences between a microcontroller and microprocessor.

Microprocessor	Microcontroller
It is a dependent unit. It requires the combination of other chips like timers, program and data memory chips, interrupt controllers, etc. for functioning.	It is a self-contained unit and it doesn't require external interrupt controller, timer, UART, etc. for its functioning.
Most of the time general purpose in design and operation	Mostly application-oriented or domain-specific
Doesn't contain a built in I/O port. The I/O port functionality needs to be implemented with the help of external programmable peripheral interface chips like 8255.	Most of the processors contain multiple built-in I/O ports which can be operated as a single 8 or 16 or 32 bit port or as individual port pins.
Targeted for high end market where performance is important.	Targeted for embedded market where performance is not so critical (At present this demarcation is invalid).
Limited power saving options compared to microcontrollers.	Includes lot of power saving features.

2.1.1.5) 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'. A typical Digital Signal Processor incorporates the following key units
 - **Program Memory:** Memory for storing the program required by DSP to process the data.
 - **Data Memory:** Memory for storing the program required by DSP to process the data.
 - **Computational Engine:** Performs the signal processing in accordance with the stored program memory. Computational Engine incorporates many specialised arithmetic units and each of them operates simultaneously to increase the execution speed. It also incorporates multiple hardware shifters for shifting operands and thereby saves execution time.
 - **I/O Unit:** Acts as an interface between the outside world and DSP. It is responsible for capturing signals to be processed and delivering the processed signals. Audio video signal processing, telecommunication, and multimedia applications are typical examples where DSP is employed. Digital signal processing employs a large amount of real-time calculations. Sum of products (SOP) calculation, convolution, fast fourier transform (FFT), discrete fourier transform (DFT), etc, are some of the operations performed by digital signal processors.

2.1.2) APPLICATION SPECIFIC INTEGRATED CIRCUITS (ASICS)

Application Specific Integrated Circuit (ASIC) is a microchip designed to perform a specific or unique application. It is used as replacement to conventional general purpose logic chips. It 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 a 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. ASIC based systems are profitable only for large volume commercial productions. Fabrication of ASICs requires a non refundable initial investment for the process technology and configuration expenses. This investment is known as Non-Recurring Engineering Charge (NRE) and it is a onetime investment. If the Non-Recurring Engineering Charges (NRE) is borne 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 is, but to a smaller number of

customers since it is for a specific application. “The ADE7760 Energy Metre ASIC developed by Analog Devices for Energy metering applications is a typical example for ASSP”. Since Application Specific Integrated Circuits (ASICs) are proprietary products, the developers of such chips may not be interested in revealing the internal details of it and hence it is very difficult to point out an example of it. Moreover it will create legal disputes if an illustration of such an ASIC product is given without getting prior permission from the manufacturer of the ASIC.

2.1.3) PROGRAMMABLE LOGIC DEVICES

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. As the name indicates, the circuits in a fixed logic device are permanent, they perform one function or set of functions—once manufactured, they cannot be changed. On the other hand, 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.

With programmable logic devices, designers use inexpensive software tools to quickly develop, simulate, and test their designs. There are no NRE costs and the final design is completed much faster than that of a custom, fixed logic device. Another key benefit of using PLDs is that during the design phase customers can change the circuitry as often as they want until the design operates to their satisfaction. That’s because PLDs are based on re-writable memory technology—to change the design, the device is simply reprogrammed. Once the design is final, customers can go into immediate production by simply programming as many PLDs as they need with the final software design file.

2.1.3.1) CPLDs and FPGAs

The two major types of programmable logic devices are Field Programmable Gate Arrays (FPGAs) and Complex Programmable Logic Devices (CPLDs). Of the two, FPGAs offer the highest amount of logic density, the most features, and the highest performance. The largest FPGA now shipping, part of the Xilinx Virtex™ line of devices, provides eight million “system gates” (the relative density of logic). These advanced 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. FPGAs are used in a wide variety of applications ranging from data processing and storage, to instrumentation, telecommunications, and digital signal processing. CPLDs, by contrast, offer much smaller amounts of logic—up to about 10,000 gates. But 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.

2.1.4) COMMERCIAL OFF-THE-SHELF COMPONENTS (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. The COTS component itself may be developed around a general purpose or domain specific processor or an Application Specific Integrated circuit or a programmable logic device. Typical examples of COTS hardware unit are remote controlled toy car control units including the RF circuitry part, high performance, high frequency microwave electronics (2–200 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. The major advantage of using COTS is that they are readily available in the market, are cheap and a developer can cut down his/her development time to a great extent. This in turn reduces the time to market your embedded systems. The TCP/IP plug-in module available from various manufactures like 'WIZnet', 'HanRun', 'Viewtool', etc. are very good examples of COTS product (Fig. 2.1).



Fig 2.1: An example of a COTS product for TCP/IP plug-in from WIZnet

This network plug-in module gives the TCP/IP connectivity to the system you are developing. There is no need to design this module yourself and write the firmware for the TCP/IP protocol and data transfer. Everything will be readily supplied by the COTS manufacturer. What you need to do is identify the COTS for your system and give the plugin option on your board according to the hardware plugin connections given in the specifications of the COTS. Though multiple vendors supply COTS for the same application, the major problem faced by the end-user is that there are no operational and manufacturing standards. This restricts the end-user to stick to a particular vendor for particular COTS. This greatly affects the product design.

The major drawback of using COTS components in embedded design is that the manufacturer of the COTS component may withdraw the product or discontinue the production of the COTS at any time if a rapid change in technology occurs, and this will adversely affect a commercial manufacturer of the embedded system which makes use of the specific COTS product.

2.2) MEMORY

Memory is an important part of a processor/controller based embedded systems. Some of the processors/controllers contain built in memory and this memory is referred as on-chip memory. Others do not contain any memory inside the chip and requires external memory to be connected with the controller/processor to store the control algorithm. It is called off-chip memory. Also some working memory is required for holding data temporarily during certain operations. This section deals with the different types of memory used in embedded system applications.

2.2.1) PROGRAM STORAGE MEMORY (ROM)

The program memory or code storage memory of an embedded system stores the program instructions and it can be classified into different types as per the block diagram representation given in Fig. 2.2.

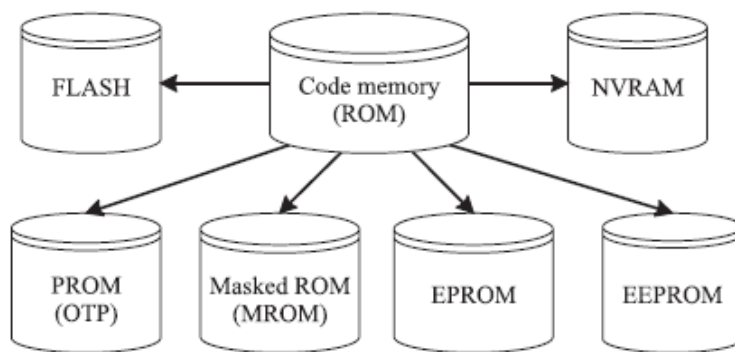


Fig. 2.2: Classification of Program Memory (ROM)

The code memory retains its contents even after the power to it is turned off. It is generally known as non-volatile storage memory. Depending on the fabrication, erasing, and programming techniques they are classified into the following types.

2.2.1.1) Masked ROM (MROM)

Masked ROM is a one-time programmable device. Masked ROM makes use of the hardwired technology for storing data. The device is factory programmed by masking and metallisation process at the time of production itself, according to the data provided by the end user. The primary advantage of this is low cost for high volume production. They are 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. Since the MROM is permanent in bit storage, it is not possible to alter the bit information.

2.2.1.2) Programmable Read Only Memory (PROM) / (OTP)

Unlike Masked ROM Memory, One Time Programmable Memory (OTP) or PROM is not pre-programmed by the manufacturer. The end user is responsible for programming these devices. This memory 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 represent logic “1” whereas 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 finalised. It is a low cost solution for commercial production. OTPs cannot be reprogrammed.

2.2.1.3) Erasable Programmable Read Only Memory (EPROM)

OTPs are not useful and worth for development purpose. During the development phase, the code is subject to continuous changes and using an OTP each time to load the code is not economical. Erasable Programmable Read Only Memory (EPROM) gives the flexibility to re-program the same chip. 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 ultraviolet 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 put in a UV eraser device for 20 to 30 minutes. So it is a tedious and time-consuming process.

2.2.1.4) Electrically Erasable Programmable Read Only Memory (EEPROM)

As the name indicates, 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 in-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).

2.2.1.5) FLASH

FLASH is the latest ROM technology and is the most popular ROM technology used in today's embedded designs. FLASH memory is a variation of EEPROM technology. It combines the reprogrammability of EEPROM and the high capacity of standard ROMs. FLASH memory is organised 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. SST39LF010 from Microchip (www.microchip.com) is an example of 1Mbit (Organised as 128K x8) Flash memory with typical endurance of 100,000 cycles.

2.2.1.6) NVRAM

Non-volatile RAM is a random access memory with battery backup. It contains static RAM based memory and a minute battery for providing supply to the memory in the absence of external power supply. The memory and battery are packed together in a single package. The life span of NVRAM is expected to be around 10 years. DS1644 from Maxim/Dallas is an example of 32KB NVRAM.

2.2.2) READ-WRITE MEMORY/ RANDOM ACCESS MEMORY (RAM)

RAM is the data memory or working memory of the controller/processor. Controller/processor can read from it and write to it. RAM is volatile, meaning when the power is turned off, all the contents are destroyed. RAM generally falls into three categories: Static RAM (SRAM), Dynamic RAM (DRAM), and Non-volatile RAM (NVRAM) (Fig. 2.3).

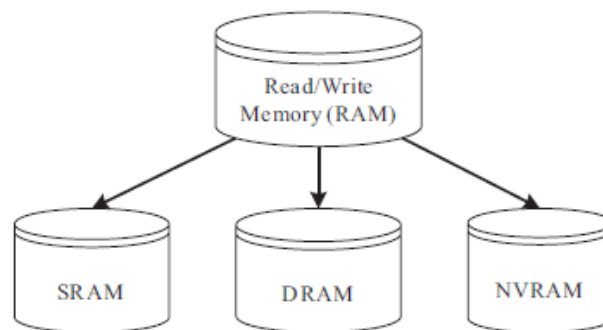


Fig. 2.3 Classification of Working Memory (RAM)

2.2.2.1) Static RAM (SRAM)

Static RAM stores data in the form of voltage. They are made up of flip-flops. Static RAM is the fastest form of RAM available. In typical implementation, an SRAM cell (bit) is realised using six transistors (or 6 MOSFETs). Four of the transistors are used for building the latch (flip-flop) part of the memory cell and two for controlling the access. SRAM is fast in operation due to its resistive networking and switching capabilities. In its simplest representation an SRAM cell can be visualised as shown in Fig. 2.4.

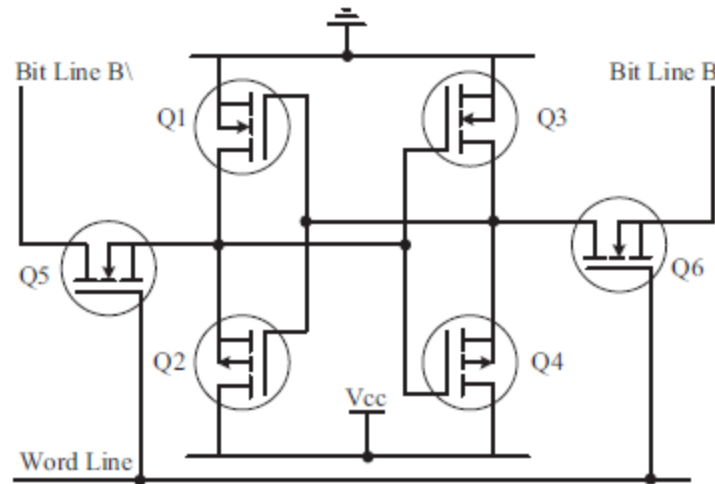


Fig. 2.4 SRAM cell implementation

This implementation in its simpler form can be visualised as two-cross coupled inverters with read/write control through transistors. The four transistors in the middle form the cross-coupled inverters. This can be visualised as shown in Fig. 2.5. From the SRAM implementation diagram, it is clear that access to the memory cell is controlled by the line Word Line, which controls the access transistors (MOSFETs) Q5 and Q6. The access transistors control the connection to bit lines B & B \bar . In order to write a value to the memory cell, apply the desired value to the bit control lines (For writing 1, make B = 1 and B \bar = 0; For writing 0, make B = 0 and B \bar = 1) and assert the Word Line (Make Word line high). This operation latches the bit written in the flip-flop. For reading the content of the memory cell, assert both B and B \bar bit lines to 1 and set the Word line to 1.

The major limitations of SRAM are low capacity and high cost. Since a minimum of six transistors are required to build a single memory cell, imagine how many memory cells we can fabricate on a silicon wafer.

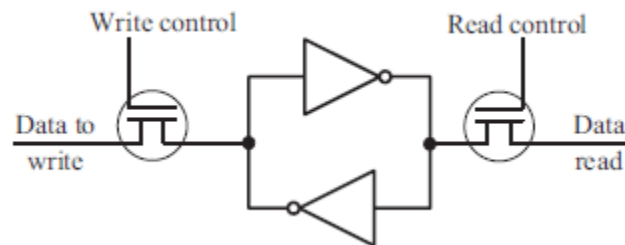


Fig. 2.5 Visualisation of SRAM cell

2.2.2.2) Dynamic RAM (DRAM)

Dynamic RAM stores data in the form of charge. They are made up of MOS transistor gates. The advantages of DRAM are its high density and low cost compared to SRAM. 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. Figure 2.6 illustrates the typical implementation of a DRAM cell. The MOSFET acts as the gate for the incoming and outgoing data whereas the capacitor acts as the bit storage unit. Table given below summarises the relative merits and demerits of SRAM and DRAM technology.

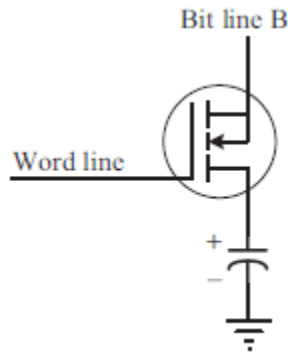


Fig. 2.6 DRAM cell implementation

2.2.2.3) NVRAM

Non-volatile RAM is a random access memory with battery backup. It contains static RAM based memory and a minute battery for providing supply to the memory in the absence of external power supply. The memory and battery are packed together in a single package. NVRAM is used for the non-volatile storage of results of operations or for setting up of flags, etc. The life span of NVRAM is expected to be around 10 years. DS1744 from Maxim/Dallas is an example for 32KB NVRAM.

2.2.3) MEMORY ACCORDING TO THE TYPE OF INTERFACE

The interface (connection) of memory with the processor/controller can be of various types. It may be a parallel interface [The parallel data lines (D0-D7) for an 8 bit processor/controller will be connected to D0-D7 of the memory] or the interface may be a serial interface like I2C (Pronounced as I Square C. It is a 2 line serial interface) or it may be an SPI (Serial peripheral interface, 2+n line interface where n stands for the total number of SPI bus devices in the system). It can also be of a single wire interconnection (like Dallas 1-Wire interface). Serial interface is commonly used for data storage memory like EEPROM. The memory density of a serial memory is usually expressed in terms of kilobits, whereas that of a parallel interface memory is expressed in terms of kilobytes. Atmel Corporations AT24C512 is an example for serial memory with capacity 512 kilobits and 2-wire interface. Please refer to the section 'Communication Interface' for more details on I2C, SPI, and 1-Wire Bus.

2.2.4) MEMORY SHADOWING

Generally the execution of a program or a configuration from a Read Only Memory (ROM) is very slow (120 to 200 ns) compared to the execution from a random access memory (40 to 70 ns). From the timing parameters it is obvious that RAM access is about three times as fast as ROM access. Shadowing of memory is a technique adopted to solve the execution speed problem in processor-based systems. In computer systems and video systems there will be a configuration holding ROM called Basic Input Output Configuration ROM or simply BIOS. In personal computer systems BIOS stores the hardware configuration information like the address assigned for various serial ports and other non-plug 'n' play devices, etc. Usually it is read and the system is configured according to it during system boot up and it is time consuming. Now the manufactures included a RAM behind the logical layer of BIOS at its same address as a shadow to the BIOS and the first step that happens during the boot up is copying the BIOS to the shadowed RAM and write protecting the RAM then disabling the BIOS reading.

You may be thinking that what a stupid idea it is and why both RAM and ROM are needed for holding the same data. The answer is: RAM is volatile and it cannot hold the configuration data which is copied from the BIOS when the power supply is switched off. Only a ROM can hold it permanently. But for high system performance it should be accessed from a RAM instead of accessing from a ROM.

2.2.5) MEMORY SELECTION FOR EMBEDDED SYSTEMS

List of Factors:

Speed: The time to read or write the data should be greatest consideration while selecting the memory. In general, speed will not be a greater issue for small embedded applications. But when you go for medium or high range applications, read/write access time should be faster.

Latency: It is the time between initiating the request of data until it is received. When executing the processor instructions, it request the data stored in the memory. The requested data must be retrieved by the processor for quick operation. Less latency, then more speed operation.

Memory Capacity: If your application need below 60 MB, it is advisable to choose the memory size as 64 MB. At the same time, running out of storage is the worst feeling we face with the digital camera. So it is important to choose the capacity as needed for the application.

Size: The size of the memory device should be compatible with the embedded system. For hand held devices, the size of the memory should be compact in nature. If it is a desktop

computer system, the size can be of medium sized. Proper size selection is also an essential criterion while selecting a memory.

Power Consumption: Memory needs power to read or write data. For high access speed, the power consumptions will be more, results in more power dissipation. More heat will reduce the lifetime of the embedded system. Hence the designers should go with optimum power consumption memory devices.

Cost: Cost plays a significant role in deciding any product. While planning to design an embedded system, similar importance should be given to memory as that of processor selection. Money should be allocated, considering the type of memory to be used in the embedded system.

2.3) SENSORS AND ACTUATORS

2.3.1) SENSORS

A sensor is a transducer device that converts energy from one form to another for any measurement or control purpose.

2.3.2) ACTUATORS

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

2.3.3) THE I/O SUBSYSTEM

The I/O subsystem of the embedded system facilitates the interaction of the embedded system with the external world. As mentioned earlier the interaction happens through the sensors and actuators connected to the input and output ports respectively of the embedded system. The sensors may not be directly interfaced to the input ports, instead they may be interfaced through signal conditioning and translating systems like ADC, optocouplers, etc. This section illustrates some of the sensors and actuators used in embedded systems and the I/O systems to facilitate the interaction of embedded systems with external world.

2.4) COMMUNICATION INTERFACE

Communication interface is essential for communicating with various subsystems of the embedded system and with the external world. For an embedded product, the communication interface can be viewed in two different perspectives; namely; Device/board level communication interface (Onboard Communication Interface) and Product level communication interface (External Communication Interface).

Embedded product is a combination of different types of components (chips/devices) arranged on a printed circuit board (PCB). The communication channel which interconnects the various components within an embedded product is referred as device/board level communication interface (onboard communication interface).

The 'Product level communication interface' (External Communication Interface) is responsible for data transfer between the embedded system and other devices or modules. The external communication interface can be either a wired media or a wireless media and it can be a serial or a parallel interface

2.4.1) ONBOARD COMMUNICATION INTERFACES

Onboard Communication Interface refers to the different communication channels/buses for interconnecting the various integrated circuits and other peripherals within the embedded system. The following section gives an overview of the various interfaces for onboard communication.

2.4.1.1) Inter Integrated Circuit (I2C) Bus

The Inter Integrated Circuit Bus (I2C—Pronounced 'I square C') is a synchronous bi-directional half duplex (one-directional communication at a given point of time) two wire serial interface bus. The concept of I2C bus was developed by 'Philips semiconductors' in the early 1980s.

The I2C bus comprise of two bus lines, namely; Serial Clock—SCL and Serial Data—SDA. SCL line is responsible for generating synchronisation clock pulses and SDA is responsible for transmitting the serial data across devices.

I2C supports multimasters on the same bus. The following bus interface diagram shown in Fig. 2.7 illustrates the connection of master and slave devices on the I2C bus.

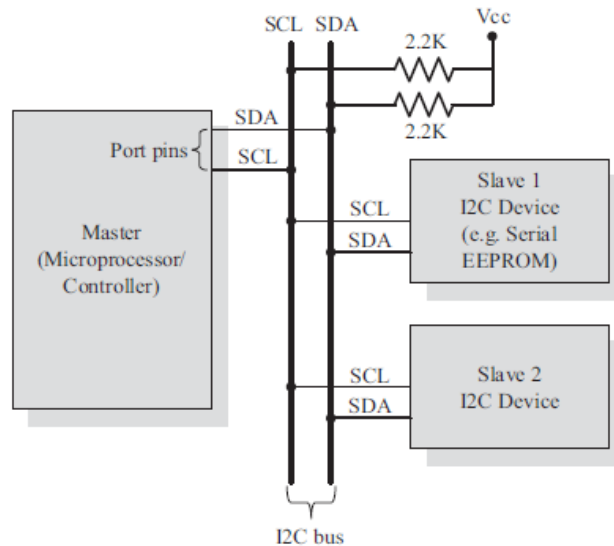


Fig. 2.7 I2C Bus Interfacing

The sequence of operations for communicating with an I2C slave device is listed below:

1. The master device pulls the clock line (SCL) of the bus to 'HIGH'
2. The master device pulls the data line (SDA) 'LOW', when the SCL line is at logic 'HIGH' (This is the 'Start' condition for data transfer)
3. The master device sends the address (7 bit or 10 bit wide) of the 'slave' device to which it wants to communicate, over the SDA line. Clock pulses are generated at the SCL line for synchronising the bit reception by the slave device. The MSB of the data is always transmitted first. The data in the bus is valid during the 'HIGH' period of the clock signal
4. The master device sends the Read or Write bit (Bit value = 1 Read operation; Bit value = 0 Write operation) according to the requirement.
5. The master device waits for the acknowledgement bit from the slave device whose address is sent on the bus along with the Read/Write operation command. Slave devices connected to the bus compares the address received with the address assigned to them
6. The slave device with the address requested by the master device responds by sending an acknowledge bit (Bit value = 1) over the SDA line
7. Upon receiving the acknowledge bit, the Master device sends the 8bit data to the slave device over SDA line, if the requested operation is 'Write to device'. If the requested operation is 'Read from device', the slave device sends data to the master over the SDA line
8. The master device waits for the acknowledgement bit from the device upon byte transfer complete for a write operation and sends an acknowledge bit to the Slave device for a read operation
9. The master device terminates the transfer by pulling the SDA line 'HIGH' when the clock line SCL is at logic 'HIGH' (Indicating the 'STOP' condition)

The first generation I2C devices were designed to support data rates only up to 100kbps.

I2C Categories based on Speed are:

Standard mode (Sm - Data rate up to 100kbit/sec),
 Fast mode (Fm - Data rate up to 400kbit/sec),
 Fast mode Plus (Fm+ - Data rate up to 1Mbit/sec),
 High-speed mode (Hs-mode - Data rate up to 3.4Mbit/sec) and
 Ultra Fast-mode (UFm), with a bit rate up to 5 Mbit/s for unidirectional I2C bus.

2.4.1.2) Serial Peripheral Interface (SPI) Bus

The Serial Peripheral Interface Bus (SPI) is a synchronous bi-directional full duplex four-wire serial interface bus. The concept of SPI was introduced by Motorola. SPI is a single master multi-slave system. It is possible to have a system where more than one SPI device can be master, provided the condition only one master device is active at any given point of time, is satisfied. SPI requires four signal lines for communication.

They are:

- i) Master Out Slave In (MOSI):** Signal line carrying the data from master to slave device. It is also known as Slave Input/Slave Data In (SI/SDI)
- ii) Master In Slave Out (MISO):** Signal line carrying the data from slave to master device. It is also known as Slave Output (SO/SDO)
- iii) Serial Clock (SCLK):** Signal line carrying the clock signals
- iv) Slave Select (SS):** Signal line for slave device select. It is an active low signal

The bus interface diagram shown in Fig. 2.8 illustrates the connection of master and slave devices on the SPI bus.

The master device is responsible for generating the clock signal. It selects the required slave device by asserting the corresponding slave device's slave select signal 'LOW'. The data out line (MISO) of all the slave devices when not selected floats at high impedance state.

SPI works on the principle of 'Shift Register'. The master and slave devices contain a special shift register for the data to transmit or receive. The size of the shift register is device dependent. Normally it is a multiple of 8. During transmission from the master to slave, the data in the master's shift register is shifted out to the MOSI pin and it enters the shift register of the slave device through the MOSI pin of the slave device.

At the same time the shifted out data bit from the slave device's shift register enters the shift register of the master device through MISO pin. In summary, the shift registers of 'master' and 'slave' devices form a circular buffer.

When compared to I2C, SPI bus is most suitable for applications requiring transfer of data in 'streams'. The only limitation is SPI doesn't support an acknowledgement mechanism.

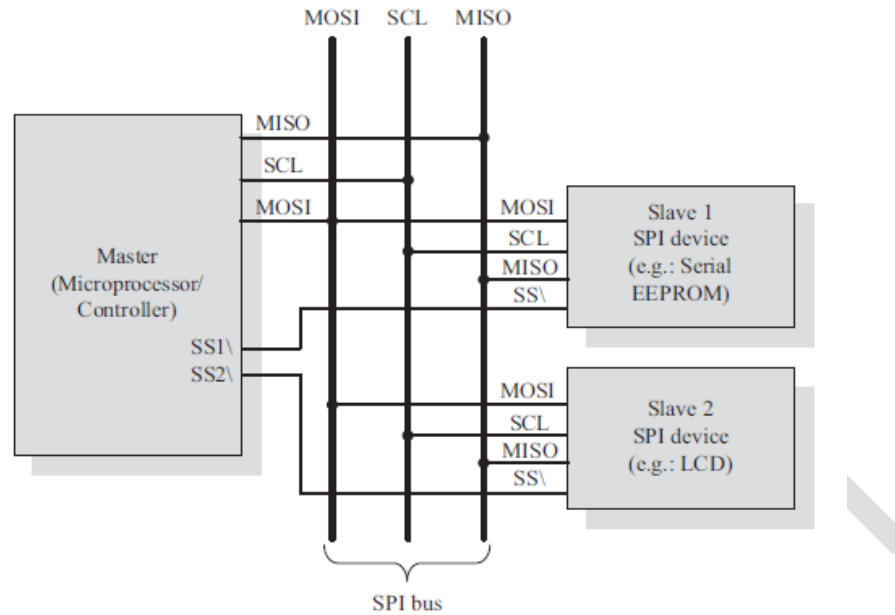


Fig. 2.8 SPI bus interfacing

2.4.1.3) Universal Asynchronous Receiver Transmitter (UART)

Universal Asynchronous Receiver Transmitter (UART) based data transmission is an asynchronous form of serial data transmission. UART based serial data transmission doesn't require a clock signal to synchronise the transmitting end and receiving end for transmission. Instead it relies upon the pre-defined agreement between the transmitting device and receiving device. The serial communication settings (Baud rate, number of bits per byte, parity, number of start bits and stop bit and flow control) for both transmitter and receiver should be set as identical.

The start and stop of communication is indicated through inserting special bits in the data stream. While sending a byte of data, a start bit is added first and a stop bit is added at the end of the bit stream. The least significant bit of the data byte follows the 'start' bit.

The 'start' bit informs the receiver that a data byte is about to arrive. If parity is enabled for communication, the UART of the transmitting device adds a parity bit. The UART of the receiving device calculates the parity of the bits received and compares it with the received parity bit for error checking.

The UART of the receiving device discards the 'Start', 'Stop' and 'Parity' bit from the received bit stream and converts the received serial bit data to a word. Figure 2.9 illustrates the same.

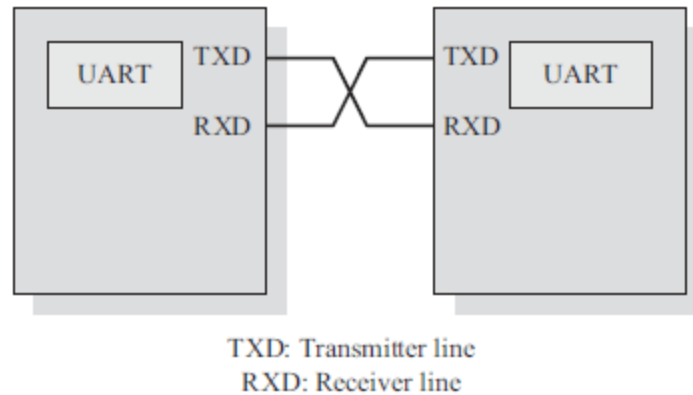


Fig. 2.9 UART Interfacing

2.4.1.4) 1-Wire Interface

1-wire interface is an asynchronous half-duplex communication protocol developed by Maxim Dallas Semiconductor (<http://www.maxim-ic.com>). It is also known as Dallas 1-Wire® protocol. It makes use of only a single signal line (wire) called DQ for communication and follows the master-slave communication model. The 1-wire interface supports a single master and one or more slave devices on the bus. The bus interface diagram shown in Fig. 2.10 illustrates the connection of master and slave devices on the 1-wire bus.

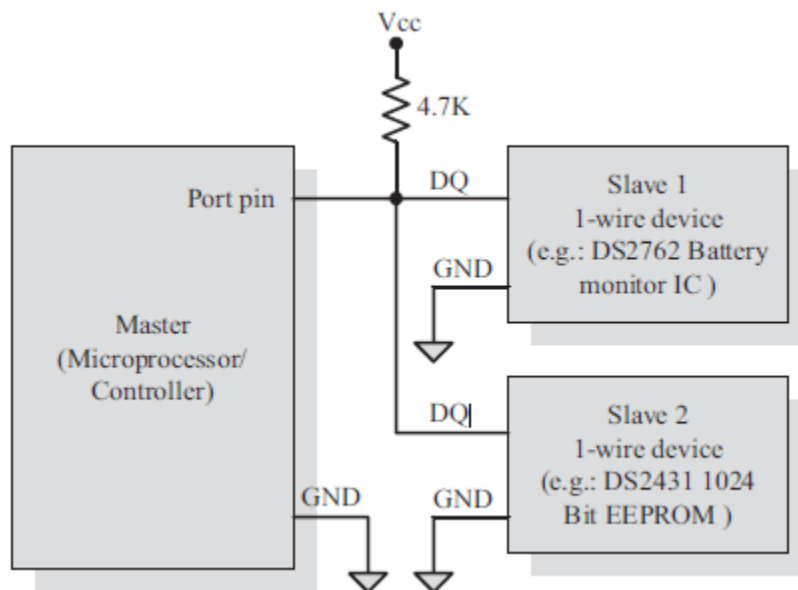


Fig. 2.10 1-Wire Interface bus

Every 1-wire device contains a globally unique 64bit identification number stored within it. This unique identification number can be used for addressing individual devices present on the bus in case there are multiple slave devices connected to the 1-wire bus. The identifier has three parts: an 8bit family code, a 48bit serial number and an 8bit Cyclic Redundancy Check (CRC) computed from the first 56 bits. The sequence of operation for communicating with a 1-wire slave device is listed below.

1. The master device sends a 'Reset' pulse on the 1-wire bus.
2. The slave device(s) present on the bus respond with a 'Presence' pulse.
3. The master device sends a ROM command (Net Address Command followed by the 64bit address of the device). This addresses the slave device(s) to which it wants to initiate a communication.
4. The master device sends a read/write function command to read/write the internal memory or register of the slave device.
5. The master initiates a Read data/Write data from the device or to the device.

All communication over the 1-wire bus is master initiated. The communication over the 1-wire bus is divided into timeslots of 60 microseconds for the regular speed mode of operation (16.3kbps). The 'Reset' pulse occupies 8 time slots. For starting a communication, the master asserts the reset pulse by pulling the 1-wire bus 'LOW' for at least 8 time slots (480ms). If a 'slave' device is present on the bus and is ready for communication it should respond to the master with a 'Presence' pulse, within 60ms of the release of the 'Reset' pulse by the master. The slave device(s) responds with a 'Presence' pulse by pulling the 1-wire bus 'LOW' for a minimum of 1 time slot (60ms). For writing a bit value of 1 on the 1-wire bus, the bus master pulls the bus for 1 to 15ms and then releases the bus for the rest of the time slot. A bit value of '0' is written on the bus by master pulling the bus for a minimum of 1 time slot (60ms) and a maximum of 2 time slots (120ms).

To Read a bit from the slave device, the master pulls the bus 'LOW' for 1 to 15ms. If the slave wants to send a bit value '1' in response to the read request from the master, it simply releases the bus for the rest of the time slot. If the slave wants to send a bit value '0', it pulls the bus 'LOW' for the rest of the time slot.

2.4.1.5) Parallel Interface

The on-board parallel interface is normally used for communicating with peripheral devices which are memory mapped to the host of the system. The host processor/controller of the embedded system contains a parallel bus and the device which supports parallel bus can directly connect to this bus system. The communication through the parallel bus is controlled by the control signal interface between the device and the host. The 'Control Signals' for communication includes 'Read/Write' signal and device select signal. The device normally contains a device select line and the device becomes active only when this line is asserted by the

host processor. The direction of data transfer (Host to Device or Device to Host) can be controlled through the control signal lines for 'Read' and 'Write'. Only the host processor has control over the 'Read' and 'Write' control signals. The device is normally memory mapped to the host processor and a range of address is assigned to it. An address decoder circuit is used for generating the chip select signal for the device. When the address selected by the processor is within the range assigned for the device, the decoder circuit activates the chip select line and thereby the device becomes active. The processor then can read or write from or to the device by asserting the corresponding control line (RD\ and WR\ respectively). Strict timing characteristics are followed for parallel communication. As mentioned earlier, parallel communication is host processor initiated. If a device wants to initiate the communication, it can inform the same to the processor through interrupts. For this, the interrupt line of the device is connected to the interrupt line of the processor and the corresponding interrupt is enabled in the host processor. The width of the parallel interface is determined by the data bus width of the host processor. The bus interface diagram shown in Fig. 2.11 illustrates the interfacing of devices through parallel interface.

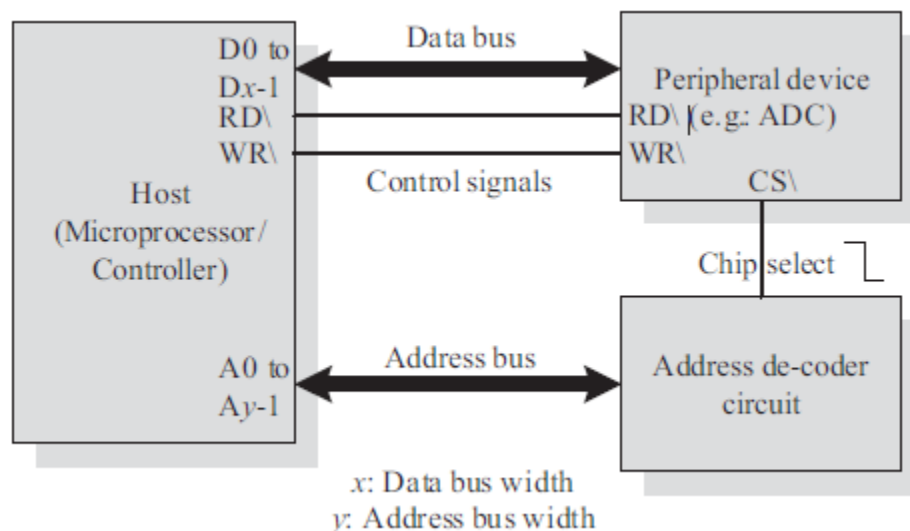


Fig. 2.11 Parallel Interface Bus

Parallel data communication offers the highest speed for data transfer.

2.4.2) EXTERNAL COMMUNICATION INTERFACES

The External Communication Interface refers to the different communication channels/buses used by the embedded system to communicate with the external world. The following section gives an overview of the various interfaces for external communication.

2.4.2.1) RS-232 C & RS-485

RS-232 C (Recommended Standard number 232, revision C from the Electronic Industry Association) is a legacy, full duplex, wired, asynchronous serial communication interface. The RS-232 interface is developed by the Electronics Industries Association (EIA) during the early 1960s. RS-232 extends the UART communication signals for external data communication.

In EIA standard, logic '0' is known as 'Space' and logic '1' as 'Mark'. The RS-232 interface defines various handshaking and control signals for communication apart from the 'Transmit' and 'Receive' signal lines for data communication. RS-232 supports two different types of connectors, namely; DB-9: 9-Pin connector and DB-25: 25-Pin connector. Figure 2.12 illustrates the connector details for DB-9 and DB-25.

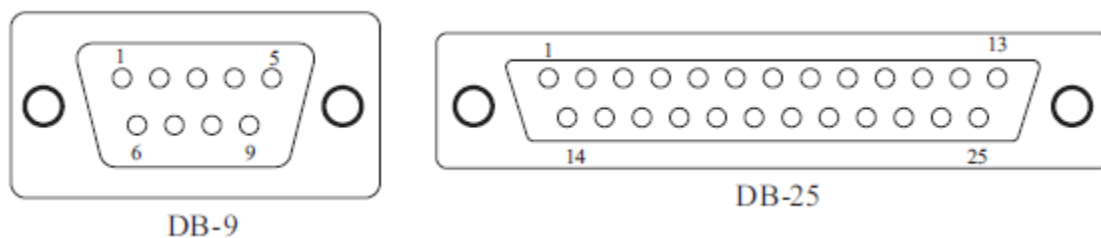


Fig. 2.12 DB-9 and DB-25 RS-232 Connector Interface

Table 2.1: The pin details for the two connectors are explained in the following table

Pin Name	Pin no: (For DB-9 Connector)	Pin no: (For DB-25 Connector)	Description
TXD	3	2	Transmit Pin for Transmitting Serial Data
RXD	2	3	Receive Pin for Receiving Serial Data
RTS	7	4	Request to send.
CTS	8	5	Clear To Send
DSR	6	6	Data Set Ready
GND	5	7	Signal Ground
DCD	1	8	Data Carrier Detect
DTR	4	20	Data Terminal Ready
RI	9	22	Ring Indicator
FG		1	Frame Ground
SDCD		12	Secondary DCD
SCTS		13	Secondary CTS
STXD		14	Secondary TXD
TC		15	Transmission Signal Element Timing
SRXD		16	Secondary RXD
RC		17	Receiver Signal Element Timing
SRTS		19	Secondary RTS
SQ		21	Signal Quality detector
NC		9	No Connection
NC		10	No Connection
NC		11	No Connection
NC		18	No Connection
NC		23	No Connection
NC		24	No Connection
NC		25	No Connection

RS-232 is a point-to-point communication interface and the devices involved in RS-232 communication are called 'Data Terminal Equipment (DTE)' and 'Data Communication Equipment (DCE)'. If no data flow control is required, only TXD and RXD signal lines and ground line (GND) are required for data transmission and reception. The RXD pin of DCE should be connected to the TXD pin of DTE and vice versa for proper data transmission. If hardware data flow control is required for serial transmission, various control signal lines of the RS-232 connection are used appropriately. The control signals are implemented mainly for modem communication and some of them may not be irrelevant for other type of devices. The Request To Send (RTS) and Clear To Send (CTS) signals co-ordinate the communication between DTE and DCE. Whenever the DTE has a data to send, it activates the RTS line and if the DCE is ready to accept the data, it activates the CTS line.

The Data Terminal Ready (DTR) signal is activated by DTE when it is ready to accept data. The Data Set Ready (DSR) is activated by DCE when it is ready for establishing a communication link. DTR should be in the activated state before the activation of DSR. The Data Carrier Detect (DCD) control signal is used by the DCE to indicate the DTE that a good signal is being received. Ring Indicator (RI) is a modem specific signal line for indicating an incoming call on the telephone line. The 25 pin DB connector contains two sets of signal lines for transmit, receive and control lines.

As per the EIA standard RS-232 C supports baudrates up to 20Kbps (Upper limit 19.2 Kbps) The commonly used baudrates by devices are 300bps, 1200bps, 2400bps, 9600bps, 11.52Kbps and 19.2Kbps. 9600 is the popular baudrate setting used for PC communication. The maximum operating distance supported by RS-232 is 50 feet at the highest supported baudrate.

2.4.2.2) Universal Serial Bus (USB)

Universal Serial Bus (USB) is a wired high speed serial bus for data communication. The first version of USB (USB1.0) was released in 1995 and was created by the USB core group members consisting of Intel, Microsoft, IBM, Compaq, Digital and Northern Telecom. The USB communication system follows a star topology with a USB host at the centre and one or more wUSB peripheral devices/USB hosts connected to it. A USB 2.0 host can support connections up to 127, including slave peripheral devices and other USB hosts. Figure 2.13 illustrates the star topology for USB device connection. USB transmits data in packet format. The USB communication is a host initiated one. The USB host contains a host controller which is responsible for controlling the data communication, including establishing connectivity with USB slave devices, packetizing and formatting the data packet.

The Pin details for the USB 2.0 Type A & B connectors are listed in the table given below.

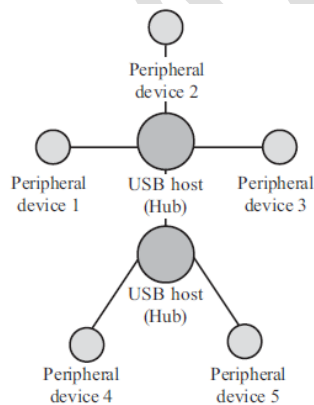


Fig. 2.13 USB Device Connection topology

Table 2.2: USB Pin configuration

Pin no.	Pin name	Description
1	V _{BUS}	Carries power (5V)
2	D ₋	Differential data carrier line
3	D ₊	Differential data carrier line
4	GND	Ground signal line

USB uses differential signals for data transmission. It improves the noise immunity. USB interface has the ability to supply power to the connecting devices. Two connection lines (Ground and Power) of the USB interface are dedicated for carrying power. A Standard Downstream USB 2.0 Port (SDP) can supply power up to 500 mA at 5 V, whereas a Charging Downstream USB 2.0 Port (CDP) can supply power up to 1500 mA at 5 V.

USB supports four different types of data transfers, namely;

Control transfer is used by USB system software to query, configure and issue commands to the USB device.

Bulk transfer is used for sending a block of data to a device. Bulk transfer supports error checking and correction. Transferring data to a printer is an example for bulk transfer.

Isochronous data transfer is used for real-time data communication. In Isochronous transfer, data is transmitted as streams in real-time. Isochronous transfer doesn't support error checking and re-transmission of data in case of any transmission loss. All streaming devices like audio devices and medical equipment for data collection make use of the isochronous transfer. Interrupt transfer is used for transferring small amount of data.

Interrupt transfer mechanism makes use of polling technique to see whether the USB device has any data to send. The frequency of polling is determined by the USB device and it varies from 1 to 255 milliseconds.

Presently USB supports different data rates namely; Low Speed (1.5Mbps), Full Speed (12Mbps), High Speed (480Mbps), SuperSpeed (5Gbps) and SuperSpeed + (or SuperSpeed USB 10 Gbps). Wireless USB combines the speed and security of wired USB technology with the ease-of-use of wireless technology.

2.4.2.3) IEEE 1394 (Firewire)

IEEE 1394 is a wired, isochronous high speed serial communication bus. It is also known as High Performance Serial Bus (HPSB). The research on 1394 was started by Apple Inc. in 1985 and the standard for this was coined by IEEE. Apple Inc's (www.apple.com) implementation of 1394 protocol is popularly known as Firewire. i.LINK is the 1394 implementation from Sony Corporation (www.sony.net) and Lynx is the implementation from Texas Instruments (www.ti.com). 1394 supports peer-to-peer connection and point- to- multipoint communication allowing 63 devices to be connected on the bus in a tree topology. 1394 is a wired serial interface and it can support a cable length of up to 15 feet for interconnection. The 1394 standard has evolved a lot from the first version IEEE 1394–1995 released in 1995 to the recent version IEEE 1394–2008 released in June 2008. The 1394 standard supports a data rate of 400 to 3200Mbits/second.

Table 2.3 Pin Configuration of IEEE 1394

Pin name	Pin no: (4 Pin Connector)	Pin no: (6 Pin Connector)	Pin no: (9 Pin Connector)	Description
Power		1	8	Unregulated DC supply. 24 to 30V
Signal Ground		2	6	Ground connection
TPB–	1	3	1	Differential Signal line for Signal line B
TPB+	2	4	2	Differential Signal line for Signal line B
TPA–	3	5	3	Differential Signal line for Signal line A
TPA+	4	6	4	Differential Signal line for Signal line A
TPA(S)			5	Shield for the differential signal line A. Normally grounded
TPB(S)			9	Shield for the differential signal line B. Normally grounded
NC			7	No connection

There are two differential data transfer lines A and B per connector. In a 1394 cable, normally the differential lines of A are connected to B (TPA+ to TPB+ and TPA–to TPB–) and vice versa. 1394 is a popular communication interface for connecting embedded devices like Digital Camera, Camcorder, Scanners to desktop computers for data transfer and storage.

Unlike USB interface (Except USB OTG), IEEE 1394 doesn't require a host for communicating between devices. For example, you can directly connect a scanner with a printer for printing. The data-rate supported by 1394 is far higher than the one supported by USB2.0 interface. The 1394 hardware implementation is much costlier than USB implementation.

2.4.2.4) Infrared (IrDA)

Infrared (IrDA) is a serial, half duplex, line of sight based wireless technology for data communication between devices. It is in use from the olden days of communication and you may be very familiar with it. The remote control of your TV, VCD player, etc. works on Infrared data communication principle. Infrared communication technique uses infrared waves of the electromagnetic spectrum for transmitting the data. IrDA supports point-point and point-to-multipoint communication, provided all devices involved in the communication are within the line of sight. The typical communication range for IrDA lies in the range 10 cm to 1 m. The range can be increased by increasing the transmitting power of the IR device. IR supports data rates ranging from 9600bits/second to 16Mbps. Depending on the speed of data transmission IR is classified into Serial IR (SIR), Medium IR (MIR), Fast IR (FIR), Very Fast IR (VFIR), Ultra Fast IR (UFIR) and GigaIR. SIR supports transmission rates ranging from 9600bps to 115.2kbps. MIR supports data rates of 0.576Mbps and 1.152Mbps. FIR supports data rates up to 4Mbps. VFIR is designed to support high data rates up to 16Mbps. The UFIR supports data rates up-to 96Mbps, whereas the GigaIR supports data rates 512 Mbps to 1 Gbps. IrDA communication involves a transmitter unit for transmitting the data over IR and a receiver for receiving the data. Infrared Light Emitting Diode (LED) is the IR source for transmitter and at the receiving end a photodiode acts as the receiver. Both transmitter and receiver unit will be present in each device

supporting IrDA communication for bidirectional data transfer. Such IR units are known as 'Transceiver'. Certain devices like a TV remote control always require unidirectional communication and so they contain either the transmitter or receiver unit (The remote control unit contains the transmitter unit and TV contains the receiver unit).

2.4.2.5) Bluetooth (BT)

Bluetooth is a low cost, low power, short range wireless technology for data and audio communication. Bluetooth was first proposed by 'Ericsson' in 1994. Bluetooth operates at 2.4GHz of the Radio Frequency spectrum and uses the Frequency Hopping Spread Spectrum (FHSS) technique for communication. Literally it supports a data rate of up to 1Mbps to 24Mbps (and a range of approximately 30 to 100 feet (Depending on the Bluetooth version – v1.2 supports data rate up to 1Mbps, v2.0 + EDR supports data rate up to 3Mbps, v3.0 + HS and v4.0 supports data rate up to 24Mbps)) for data communication.

Like IrDA, Bluetooth communication also has two essential parts; a physical link part and a protocol part. The physical link is responsible for the physical transmission of data between devices supporting Bluetooth communication and protocol part is responsible for defining the rules of communication.

Bluetooth communication follows packet based data transfer. Bluetooth supports point-to-point (device to device) and point-to-multipoint (device to multiple device broadcasting) wireless communication. The point-to-point communication follows the master slave relationship. A Bluetooth device can function as either master or slave. When a network is formed with one Bluetooth device as master and more than one device as slaves, it is called a Piconet. A Piconet supports a maximum of seven slave devices. Bluetooth is the favourite choice for short range data communication in handheld embedded devices. Bluetooth technology is very popular among cell phone users as they are the easiest communication channel for transferring ringtones, music files, pictures, media files, etc. between neighboring Bluetooth enabled phones. The specifications for Bluetooth communication is defined and licensed by the standards body 'Bluetooth Special Interest Group (SIG)'. For more information, please visit the website www.bluetooth.org.

2.4.2.6) Wi-Fi

Wi-Fi or Wireless Fidelity is the popular wireless communication technique for networked communication of devices. Wi-Fi follows the IEEE 802.11 standard. Wi-Fi is intended for network communication and it supports Internet Protocol (IP) based communication. It is essential to have device identities in a multipoint communication to address specific devices for data communication. In an IP based communication each device is identified by an IP address, which is unique to each device on the network.

Wi-Fi based communications require an intermediate agent called Wi-Fi router/Wireless Access point to manage the communications. The Wi-Fi router is responsible for restricting the access to a network, assigning IP address to devices on the network, routing data packets to the intended devices on the network. Wi-Fi enabled devices contain a wireless adaptor for transmitting and receiving data in the form of radio signals through an antenna. The hardware part of it is known as Wi-Fi Radio. Wi-Fi operates at 2.4GHz or 5GHz of radio spectrum and they co-exist with other ISM band devices like Bluetooth. Figure 2.14 illustrates the typical interfacing of devices in a Wi-Fi network. For communicating with devices over a Wi-Fi network, the device when its Wi-Fi radio is turned ON, searches the available Wi-Fi network in its vicinity and lists out the Service Set Identifier (SSID) of the available networks. If the network is security enabled, a password may be required to connect to a particular SSID. Wi-Fi employs different security mechanisms like Wired Equivalency Privacy (WEP) Wireless Protected Access (WPA), etc. for securing the data communication. Wi-Fi supports data rates ranging from 1Mbps to 1300Mbps (Growing towards higher rates as technology progresses), depending on the standards (802.11a/b/g/n/ac) and access/modulation method. Depending on the type of antenna and usage location (indoor/outdoor), Wi-Fi offers a range of 100 to 1000 feet.

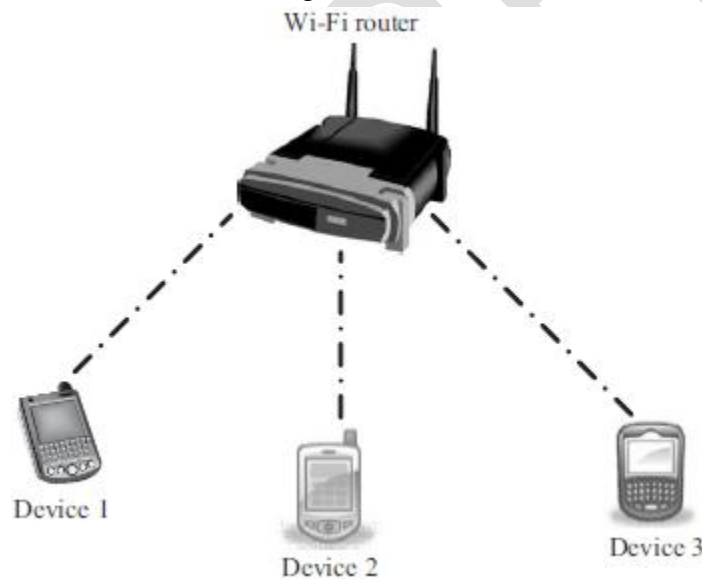


Fig. 2.14 Wi-Fi network

2.4.2.7) ZigBee

ZigBee is a low power, low cost, wireless network communication protocol based on the IEEE 802.15.4-2006 standard. ZigBee is targeted for low power, low data rate and secure applications for Wireless Personal Area Networking (WPAN). The ZigBee specifications support a robust mesh network containing multiple nodes. This networking strategy makes the network reliable by permitting messages to travel through a number of different paths to get from one node to another.

ZigBee operates worldwide at the unlicensed bands of Radio spectrum, mainly at 2.400 to 2.484 GHz, 902 to 928 MHz and 868.0 to 868.6 MHz. ZigBee Supports an operating distance of up to 100 metres and a data rate of 20 to 250Kbps.

In the ZigBee terminology, each ZigBee device falls under any one of the following ZigBee device category.

ZigBee Coordinator (ZC)/Network Coordinator: The ZigBee coordinator acts as the root of the ZigBee network. The ZC is responsible for initiating the ZigBee network and it has the capability to store information about the network.

ZigBee Router (ZR)/Full Function Device (FFD): Responsible for passing information from device to another device or to another ZR.

ZigBee End Device (ZED)/Reduced Function Device(RFD): End device containing ZigBee functionality for data communication. It can talk only with a ZR or ZC and doesn't have the capability to act as a mediator for transferring data from one device to another.

The diagram shown in Fig. 2.15 gives an overview of ZC, ZED and ZR in a ZigBee network. ZigBee is primarily targeting application areas like home & industrial automation, energy management, home control/security, medical/patient tracking etc

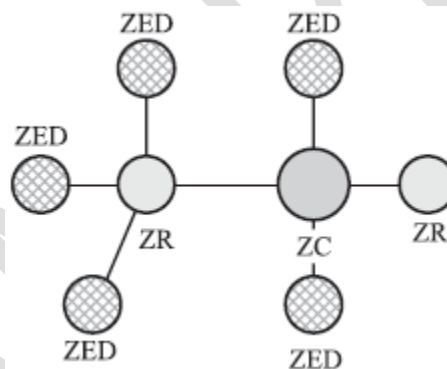


Fig. 2.15 A ZigBee network model

ZigBee PRO offers full wireless mesh, low-power networking capable of supporting more than 64,000 devices on a single network. It provides standardised networking designed to connect the widest range of devices, in any industry, into a single control network. ZigBee PRO offers an optional new and innovative feature, 'Green Power' to connect energy harvesting or self-powered devices into ZigBee PRO networks.

The 'Green Power' feature of ZigBee PRO is the most eco-friendly way to power battery-less devices such as sensors, switches, dimmers and many other devices and allows them to securely join ZigBee PRO networks. The specifications for ZigBee is developed and managed by the ZigBee alliance (www.zigbee.org), a non-profit consortium of leading semiconductor manufacturers, technology providers, OEMs and end-users worldwide.

2.4.2.8 General Packet Radio Service (GPRS)

3G, 4G, LTE General Packet Radio Service (GPRS), 3G, 4G and LTE are cellular communication technique for transferring data over a mobile communication network like GSM and CDMA. Data is sent as packets in GPRS communication. The transmitting device splits the data into several related packets. At the receiving end the data is re-constructed by combining the received data packets. GPRS supports a theoretical maximum transfer rate of 171.2kbps. In GPRS communication, the radio channel is concurrently shared between several users instead of dedicating a radio channel to a cell phone user. The GPRS communication divides the channel into 8 timeslots and transmits data over the available channel. GPRS supports Internet Protocol (IP), Point to Point Protocol (PPP) and X.25 protocols for communication. GPRS is mainly used by mobile enabled embedded devices for data communication.

The device should support the necessary GPRS hardware like GPRS modem and GPRS radio. To accomplish GPRS based communication, the carrier network also should have support for GPRS communication. GPRS is an old technology and it is being replaced by new generation cellular data communication techniques like 3G (3rd Generation), High Speed Downlink Packet Access (HSDPA), 4G (4th Generation), LTE (Long Term Evolution) etc. which offers higher bandwidths for communication. 3G offers data rates ranging from 144Kbps to 2Mbps or higher, whereas 4G gives a practical data throughput of 2 to 100+ Mbps depending on the network and underlying technology.