Chapter 6

Communication Interfaces

CHAPTER OBJECTIVES

After reading this chapter, you will be able to:

- Appreciate the need for communication interfaces
- Learn the details of various serial communication interfaces
- Understand the Ethernet and wireless LAN interfaces
- Grasp the details of IEEE 1394 interface
- Gain knowledge of infrared and Bluetooth wireless interfaces

Most of the embedded systems have to interface with the external world. This requirement may be to transmit the data to a PC or workstation, or to interact with another system for sharing the data. To meet this requirement, the embedded systems need to be provided with communication interfaces. In this chapter, we will study the various standard communication interfaces that can be provided to the embedded systems. We will discuss the details of RS232/UART, RS422/RS485, IEEE 1394, Universal Serial Bus (USB), Ethernet as well as wireless interfaces such as IrDA, IEEE 802.11 and Bluetooth.

6.1 Need for Communication Interfaces

The need for providing communication interfaces arises due to the following reasons:

- The embedded system needs to send data to a host (a PC or a workstation). The host will analyse of data and present the data through a Graphical User Interface.
- The embedded system may need to communicate with another embedded system to transmit/receive data. Providing a standard communication interface is preferable rather than providing a proprietary interface.
- A number of embedded systems may need to be networked to share data. Network interfaces need to be provided in such a case.

- An embedded system may need to be connected to the Internet so that anyone can access the embedded system. An example is a real-time weather monitoring system. The weather monitoring system can be Internet-enabled using TCP/IP protocol stack and HTTP server.
- Mobile devices such as cell phones and palmtops need to interact with other devices such as PCs and laptops for data synchronization. For instance, you need to ensure that the address book on your palmtop is the same as that on your laptop. When the palmtop comes near the laptop, automatically the two can form a network to exchange data.
- For some embedded systems, the software may need upgradation after it is installed in the field. The software can be upgraded through communication interfaces.

Due to these reasons, providing communication interfaces based on standard protocols is a must. Not surprisingly, many micro-controllers have on-chip communication interfaces such as a serial interface to meet these requirements. Now, we will discuss the following communication interfaces.

In Brief...

Embedded systems are provided with communication interfaces for monitoring and control by a host system or a node on a network. Through these interfaces, an embedded system can also be accessed over the Internet.

- RS 232/UART
- RS 422, RS 485
- Universal Serial Bus
- Infrared
- IEEE 1394 Firewire
- Ethernet
- IEEE 802.11 wireless interface
- Bluetooth

For each of these interfaces, we will discuss the hardware details and the protocol stack that needs to be implemented in software.

6.2 RS232/UART

RS232 is a standard developed by Electronic Industry Association (EIA). This is one of the oldest and most widely used communication interfaces. The PC will have two RS232 ports designated as COM1 and COM2. Most of the micro-controllers have an on-chip serial interface. The evaluation boards of the processors are also connected to the host system using RS232.

RS232 is used to connect a DTE (Data Terminal Equipment) to a Data Circuit Terminating Equipment (DCE). A DTE can be a PC, serial printer or a plotter. DCE can be a modem, mouse, digitizer or a scanner. RS232 interface specifies the physical layer interface only. The specifications describe the physical, mechanical, electrical, and procedural characteristics for serial communication.

In Brief...

RS232 standard is used for serial communication between two devices for speeds up to 115.2 Kbps over distances up to 100 meters.

RS 232 is a standard for serial communication, i.e. the bits are transmitted serially. The communication between two devices is in full duplex, i.e. the data transfer can take place in both directions.

6.2.1 RS232 Communication Parameters

When two devices have to communicate through RS232, the sending device sends the data character by character. The bits corresponding to the character are called data bits. The data bits are prefixed with a bit called start bit, and suffixed with one or two bits called stop bits. The receiving device decodes the data bits using the start bit and stop bits. This mode of communication is called asynchronous communication because no clock signal is transmitted. In addition to start bit and stop bits, an additional bit called parity bit is also sent. Parity bit is used for error detection at the receiving end.

For two devices to communicate with each other using RS232, the communication parameters have to be set on both the systems. And, for a meaninginful communication, these parameters have to be the same. The various communication parameters are listed below:

Data rate: The rate at which data communication takes place. The PC supports various data rates such as 50, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 and 115200 bps. The oscillator in the RS232 circuitry operates at 1.8432 MHz and it is divided by 1600 to obtain the 115200 data rate.

Data bits: Number of bits transmitted for each character. The character can have 5 or 6 or 7 or 8 bits. If you send ASCII characters, the number of bits is 7.

Start bit: The bit that is prefixed to the data bits to identify the beginning of the character.

Stop bits: These bits are appended to the data bits to identify the end of character. If the data bits are 7 or 8, one stop bit is appended. If the data bits are 5 or 6, two stop bits are appended.

Parity: The bit appended to the character for error checking. The parity can be even or odd. For even parity, the parity bit (1 or 0) will be added in such a way that the total number of bits will be even. For odd parity, the parity bit will make the total number of bits odd. If the parity is set to 'none', the parity bit is ignored. For example, if the data bits are 1010110, the parity bit is 0 if even parity is used; and the parity bit is 1 if odd parity is used. At the receiving end, the device will calculate the parity bit and if the received parity bit matches with the calculated parity bit, it can be assumed that the data is without errors. But this is not the reality. If two bits are in error, the receiver cannot detect that there is an error!

Flow Control: If one of the devices sends data at a very fast rate and the other device cannot absorb the data at that rate, flow control is used. Flow control is a protocol to stop/resume data transmission. This protocol is also known as handshaking. If you are sure that there will be no flow control problem, there is no need for handshaking. You can do hardware handshaking in RS232 using two signals: Request to Send (RTS) and Clear to Send (CTS). When a device has data to send, it asserts RTS and the receiving device asserts CTS. You can also do software handshaking — a device can send a request to suspend data transmission by sending the character Control S (0x13). The signal to resume data transmission is sent using the character Control Q (0x11). This software handshaking is also known as XON/XOFF.

In Brief...

The communication parameters to be set for serial communication are: data rate, number of data bits, number of stop bits, parity and flow control.

Notes...

Communication using RS232 involves lot of overhead due to the additional bits to be transmitted—for 7-bit data, the additional bits to be transmitted are 1 start bit, 1 stop bit and 1 parity bit.

6.2.2 RS 232 Connector Configurations

RS232 standard specifies two types of connectors: 25-pin connector and 9-pin connector. In the 25-pin configuration, only a few pins are used. The description of these pins is given in Table 6.1.

Pin number	Function (abbreviation)
1	Chassis ground
2	Transmit data (TXD)
3	Receive data (RXD)
4	Request To Send (RTS)
5	Clear To Send (CTS)
6	Data Set Ready (DSR)
7	Signal Ground (GND)
8	Carrier Detect (CD)
20	Data Terminal Ready (DTR)
22	Ring Indicator (RI)

Table 6.1: 25-pin Connector Pin Details

For the 9-pin connector, the pin details are given in Table 6.2.

Pin number	Function (abbreviation)
1	Carrier Detect (CD)
2	Receive Data (RXD)
3	Transmit Data (TXD)
4	Data Terminal Ready (DTR)
5	Signal Ground (GND)
6	Data Set Ready (DSR)
7	Request to Send (RTS)
8	Clear to Send (CTS)
9	Ring Indicator (RI)

Table 6.2: 9-pin Connector Pin Details

For transmission of 1's and 0's, the voltage levels are defined in the standard. The voltage levels are different for control signals and data signals. These voltage levels are given in Table 6.3. The voltage level is with reference to the local ground and hence RS232 uses unbalanced transmission.

Signal	Voltage Level
Data input	+3 volts and above for 0-3 volts and below for 1
Data output	+5 volts and above for 0-5 volts and below for 1
Control input	+3 volts and above for 1 (ON)-3 volts and below for 0 (OFF)
Control output	+5 volts and above for 1(ON)-5 volts and below for 0 (OFF)

Table 6.3: Voltage Levels for Data and Control Signals

In Brief...

Two types of connectors are used in RS232—25-pin connector and 9-pin connector. The three important pins are—2 for transmit, 3 for receive and 7 for signal ground in a 25-pin connector.

Note that the voltage levels used in RS232 are different from voltage levels used in embedded systems (as most chips use 5 volts and below only). Another problem is that the processor gives out the data in parallel format, not in serial format. These problems are overcome through the use of UART (Universal Asynchronous Receive Transmit) chips.

Notes...

RS232 uses unbalanced transmission i.e., the voltage levels are measured with reference to the local ground. Hence, this transmission is susceptible to noise.

6.2.3 UART

In Brief...

Universal Asynchronous Receive Transmit (UART) chip converts the parallel data received from the processor into serial format and sends it to RS232 level shifter. It also takes the serial data from the RS232 level shifter and converts into parallel format.

The processors process the data in parallel format, not in serial format. To bridge the processor and the RS232 port, Universal Asynchronous Receive Transmit (UART) chip is used. UART has two sections: receive section and transmit section. Receive section receives the data in serial format, converts it into parallel format and gives it to the processor. The transmit section takes the data in parallel format from the processor and converts it into serial format. The UART chip also adds the start bit, stop bits and parity bit. Many micro-controllers have on-chip UART. However, the necessary voltage level conversion has to be done to meet the voltage levels of RS232. This is achieved using a level shifter as shown in Fig. 6.1.

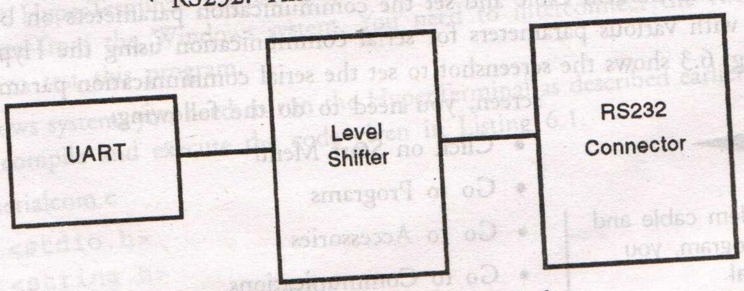


Fig. 6.1: Hardware for RS 232 Interface

UART chip operates at 5 Volts. The level conversion to the desired voltages is done by the level shifter, and then the signals are passed on to the RS232 connector.

RS232 standard specifies a distance of 19.2 meters. However, you can achieve distances up to 100 meters using RS232 cables. The data rates supported will be dependent on the UART chip and the clock used.

Most of the processors including Digital Signal Processors have on-chip UART. ICs such as MAX 3222, MAX 3241 of Maxim can be used as level shifters.

6.2.4 Null Modem Cable Connection

If you want to connect two DTEs such as two PCs, you need to interconnect the two RS232 ports using a null modem cable. The null modem cable connections are shown in Fig. 6.2. Fig. 6.2(a) shows the connections for 25-pin connectors and Fig. 6.2(b) shows the connections for 9-pin connectors. However, the minimal connections required are for TD, RD and GND. If you are using 25-pin connector, it is sufficient if you connect pin 2 to pin 3, pin 3 to pin 2 and pin 7 to pin 7. This is the minimal configuration.

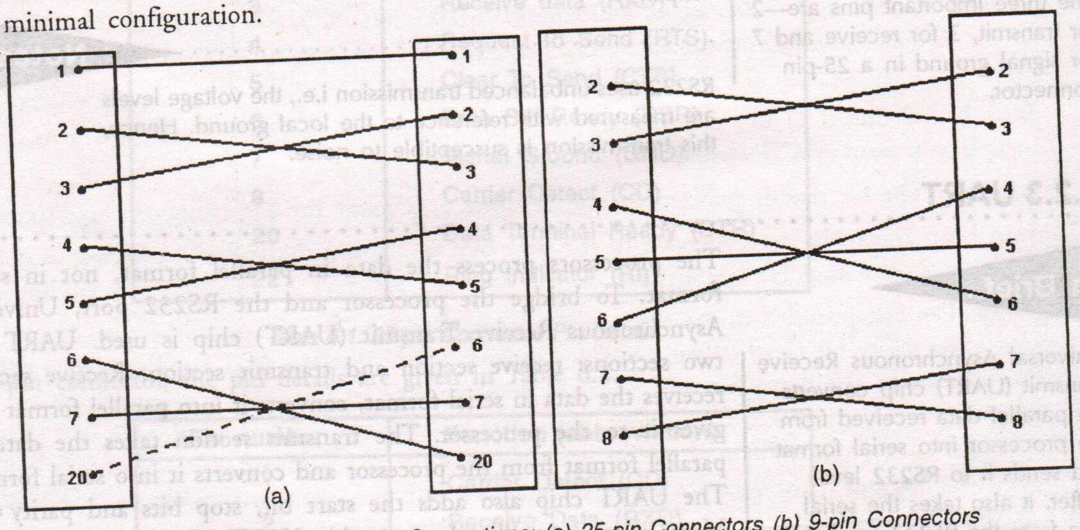


Fig. 6.2 Null Modem Cable Connections: (a) 25-pin Connectors (b) 9-pin Connectors

To make two devices communicate with each other using RS232 interface, you need to connect the two PCs using a null modem cable and set the communication parameters on both the devices. You can experiment with various parameters for serial communication using the HyperTerminal program on Windows. Fig. 6.3 shows the screenshot to set the serial communication parameters. To obtain this screen, you need to do the following:

In Brief...

Using a null modem cable and HyperTerminal program, you can establish serial communication between two machines running the Windows operating system.

- Click on Start Menu
- Go to Programs
- Go to Accessories
- Go to Communications
- Go to HyperTerminal
- Double click on HYPERTRM.EXE

- Enter a name for New Connection "xxxx" and Click OK
- You will get a screen with the title "Connect To". Select COM1 or COM2 from the "Connect Using" list box and click OK.

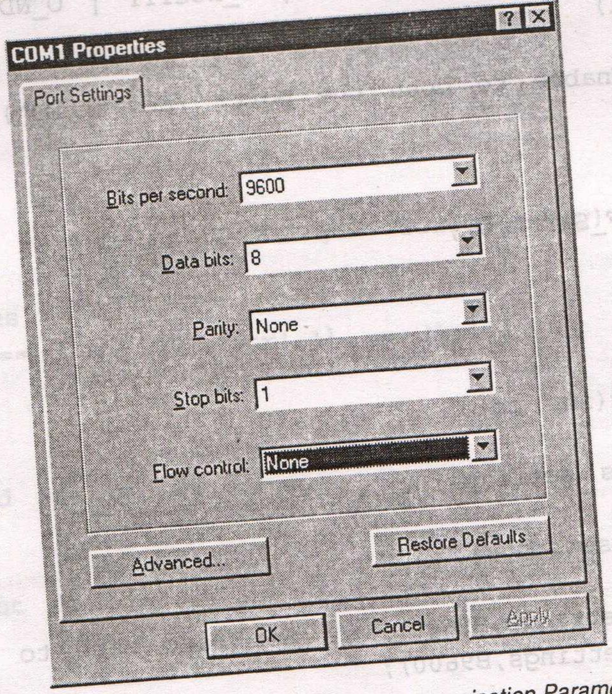


Fig. 6.3: Screenshot to set the Serial Communication Parameters

6.2.5 Serial Communication Programming

In this section, we will study how to establish communication between two PCs, one PC running the Windows operating system and another running the Linux operating system. On the Windows system, we will use the HyperTerminal program and on the Linux system we will write a C program to send and receive data from the Windows system. You need to interconnect the two PCs through a null modem cable to test this program.

On the Windows system, you need to run the HyperTerminal as described earlier. On the Linux system you need to compile and execute the code given in Listing 6.1.

Listing 6.1 serialcom.c

```
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#include <termios.h>
```

```
int port_open(void)
{
  int fd;
  fd = open("/dev/ttyS0",O_RDWR | O_NOCTTY | O_NDELAY);
  if(fd == -1)
  {
     perror("Unable to open the port: /dev/ttyS0");
  }
  else
  {
     fcntl(fd,F_SETFL,0);
  }
  return(fd);
}

void port_config(int fd)
{
   struct termios settings;

   tcgetattr(fd,&settings);

   cfsetispeed(&settings,B9600); // Set baud rate to 9600
   cfsetospeed(&settings,B9600);

   settings.c_cflag |= (CLOCAL | CREAD);// set to local mode

   settings.c_cflag &= ~PARENB; //no parity bit
   settings.c_cflag &= ~CSTOPB; // two stop bits
   settings.c_cflag &= ~CSIZE; //bit mask for data bits
   settings.c_cflag |= CS8;     //8 data bits

   tcsetattr(fd,TCSANOW,&settings);
}

void write_data(int fd,char *c)
{
   int n;
   n = write(fd, c, strlen(c));
   if(n<0)
   {
     fputs("write() of data failed! \n",stderr);
```

```
    }
}

void read_data(int fd)
{
   char array[255];
   char *ptr;
   int nbytes;

   ptr = array;
   while((nbytes = read(fd, ptr, array + sizeof(array)-ptr-1)) > 0)
   {
      ptr += nbytes;
      if((ptr[-1] == '\n') || (ptr[-1] == '\r'))
         break;
   }
   *ptr = '\0';
   printf("Received String: %s",array);
}

void port_close(int fd)
{
   close(fd);
}

int main(void)
{
  int fd;
  fd = port_open();
  port_config(fd);
  write_data(fd,"Hello World");
  read_data(fd);
  port_close(fd);
  return 0;
}
```

The function **port_open(void)** opens the serial port. **/dev/ttyS0** is the device file. You can find out the device file of your Linux installation in the /dev directory and use that file name instead of **/dev/ttyS0**.

The function **port_config(int fd)** is to configure the port parameters.

The function **write_data (int fd,** char *c) is to write data onto the port. The pointer to a character array is passed from this function.

The function `read_data(int  fd)` is to read the data from the port.

The function `port_close(int  fd)` closes the port.

The main(void) function calls the functions to open the port, configure the port parameters, write the data, read the data and then close the port.

In Brief...

Serial communication programming involves opening the device file, configuring the port by setting the communication parameters, reading/writing data and then closing the port.

You can compile this program using the following shell command. Note that the $ sign is the system prompt.

```
$gcc  serialcom.c
```

The file a.out will be created. You can execute this file by giving the command

```
$./a.out
```

The message "Hello World" will be displayed on the Windows system. You can type some text on the Windows system and it will be displayed on the Linux system.

6.3 RS422/RS485

RS422 standard for serial communication is used in noisy environments. Using this standard interface, the distance between two devices can be up to 1200 meters. Twisted copper cable is used as the transmission medium. Unlike RS232 in which the voltage levels are measured with reference to local ground, in RS422, voltage difference between the two copper wires represents the logic levels. Hence, RS422 uses balanced transmission. Two channels are used for transmit and receive paths. As compared to RS232, RS422 is better suited to work in noisy environments over longer distances because of balanced transmission.

In Brief...

In RS422 and RS485, twisted copper pair is used as transmission medium. The voltage difference between the two wires represents logic levels, i.e. balanced transmission is used and hence this transmission is immune to noise.

RS485 is a variation of RS422 to connect a number of devices in a network. An RS485 controller chip is used on each device. A network using RS485 protocols operates in a Master/Slave configuration. Up to 512 devices can be networked. Using one twisted pair, half-duplex communication can be achieved and using two twisted pairs, full-duplex communication can be achieved.

Chips such as MAX3488 are used for RS422. MAX 3483 is an RS 485 controller for half-duplex communication and MAX 3491 is for full-duplex communication.

6.4 USB

Universal Serial Bus has gained immense popularity in recent years. Desktops, laptops, printers, display devices, video cameras, hard disk drives, CDROM drives, audio equipment etc. are now available with USB interface. Using USB, a number of devices can be networked using Master/Slave architecture. A host, such as the PC, is designated as a master. As shown in Fig. 6.4, a number of devices, up to a maximum of 127, can be connected in the form of an inverted tree. On the host, such as a PC,

In Brief...

Universal Serial Bus (USB) is a serial communication protocol to connect up to 127 devices to a host. A USB device can be self-powered, or it can be powered by the bus.

there will be a host controller — a combination of hardware and software — to control all the USB devices. Devices can be connected to the host controller either directly or through a hub. A hub is also a USB device that extends the number of ports — from 2 to 8 — to connect other USB devices. A USB device can be self-powered, or powered by the bus. USB can supply 500 mA current to the devices.

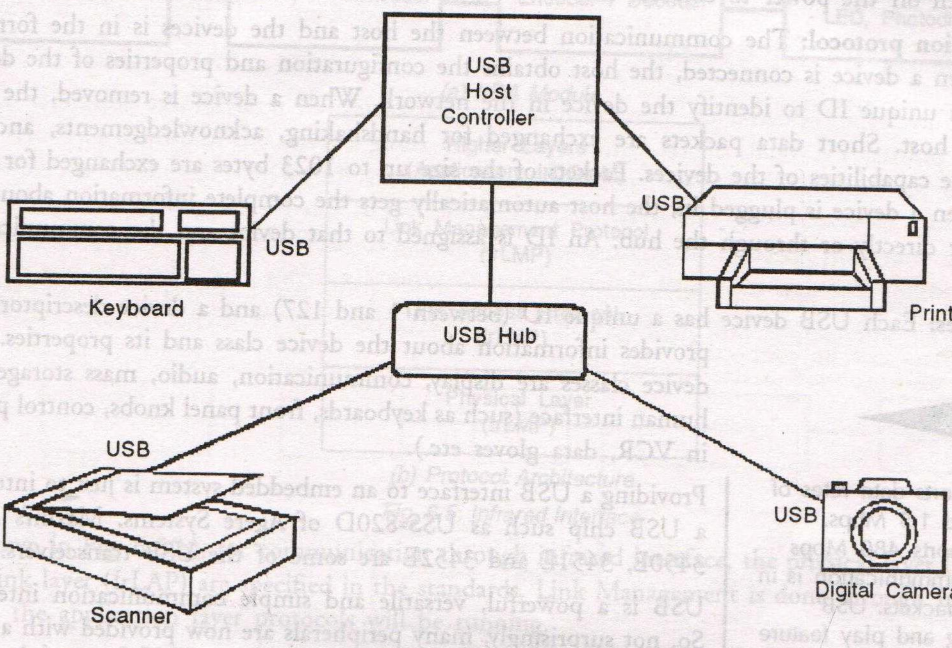


Fig. 6.4: USB Device Connection Hierarchy

6.4.1 USB Physical Interface

In Brief...

USB uses a twisted pair of wires as the medium. Data is transmitted over the twisted pair using differential data lines and hence balanced communication is used.

A shielded 4-wire twisted copper cable is used with the pin connections as shown in Table 6.4. Data is transmitted over a differential twisted pair of wires labeled D+ and D-.

Pin number	Function (abbreviation)
1	+5 V Power (VBUS)
2	Differential data line (D+)
3	Differential data line (D-)
4	Power and Signal ground (GND)

Table 6.4: Pin Connections for USB Interface

6.4.2 Features of USB

Data rates: USB 1.1 standard supports 12 Mbps data rate, and 1.5 Mbps for slower peripherals. USB2.0 supports data rates up to 480 Mbps.

Special features: USB supports plug and play, i.e. you can connect USB devices to the hub or the host without any need for configuration settings. The host will detect and identify the device by exchanging a set of packets. This is known as "Bus Enumeration". The devices are hot-pluggable, i.e. there is no need to switch off the power to connect the device.

Communication protocol: The communication between the host and the devices is in the form of packets. When a device is connected, the host obtains the configuration and properties of the device and assigns a unique ID to identify the device in the network. When a device is removed, the hub informs the host. Short data packets are exchanged for handshaking, acknowledgements, and for informing the capabilities of the devices. Packets of the size up to 1023 bytes are exchanged for data transfer. When a device is plugged in, the host automatically gets the complete information about the device, either directly or through the hub. An ID is assigned to that device and the communication can start.

Device classes: Each USB device has a unique ID (between 1 and 127) and a device descriptor that provides information about the device class and its properties. The device classes are display, communication, audio, mass storage and human interface (such as keyboards, front panel knobs, control panels in VCR, data gloves etc.).

Providing a USB interface to an embedded system is just to integrate a USB chip such as USS-820D of Agere Systems. Maxim's MAX 3450E, 3451E and 3452E are some of the USB transceivers.

USB is a powerful, versatile and simple communication interface. So, not surprisingly, many peripherals are now provided with a USB interface. RS232 will be confined to the technology museums.

In Brief...

USB1.1 supports data rates of 12 Mbps and 1.5 Mbps. USB2.0 supports 480 Mbps data rate. Communication is in the form of packets. USB supports plug and play feature as well as hot insertion.

6.5 Infrared

Infrared interfaces are used in remote control units of TV, VCR, air-conditioner, etc. However, these interfaces are all based on proprietary protocols. Infrared Data Association (IrDA), a non-profit industry association (www.irda.org) founded in 1993, released the specifications for low-cost infrared communication between devices. Infrared interfaces are now commonplace for a number of devices such as palmtops, cell phones, digital cameras, printers, keyboards, mice, LCD projectors, ATMs, smart cards etc. Infrared interface provides a low-cost, short range, point-to-point communication between two devices. The only drawback with infrared is that it operates in line of sight communication mode and it cannot penetrate through walls. It supports only data.

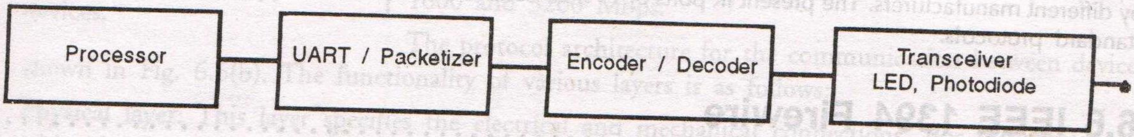
The block diagram of IrDA module is shown in Fig. 6.5(a) and the protocol architecture is shown in Fig. 6.5(b). As shown in Fig. 6.5(a), the device will have an infrared transceiver. The transmitter

In Brief...

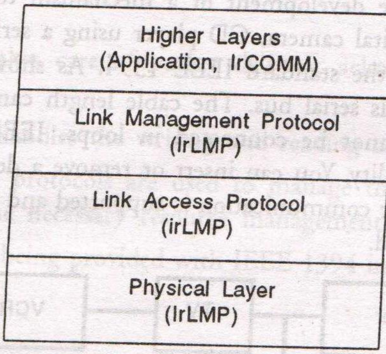
Infrared is a low-cost, point-to-point communication mechanism for short ranges.

is a LED and the receiver is a photodiode. Agilent's HSDL-1001 can be used as a transceiver. For low data rates, the processor of the embedded system itself can be used whereas for high data rates, a different processor may be needed.

The data to be sent on the infrared link is packetized and encoded as per the IrDA protocols and sent over the air to the other device. The receiving device will detect the signal, decode and depacketize the data.



(a) IrDA Module



(b) Protocol Architecture

Fig. 6.5: Infrared Interface

As shown in Fig. 6.5(b), for communication through infrared interface, the physical layer (IrPHY) and data link layer (IrLAP) are specified in the standards. Link Management is done through IrLMP, above which the application layer protocols will be running.

Physical layer: IrPHY specifies the data rates and the mode of communication. IrDA has two specifications viz., IrDA Data and IrDA control. IrDA Data has a range of 1 meter with bi-directional communication. Serial IR (SIR) supports data rate up to 115 Kbps and Fast IR (FIR) supports data rates up to 4 Mbps. IrDA Control has a range of 5 meters with bi-directional communication speed up to 75 Kbps. A host such as PC can communicate with 8 peripherals using IrDA protocols.

Data link layer: The data link layer is called the IrLAP. IrLAP is based on HDLC protocol. Master/Slave protocol is used for communication between two devices. The device that starts the communication is the master. The master sends the command and the slave sends a response.

Link management layer: This layer facilitates a device to query the capabilities of other devices. It also provides the software capability to share IrLAP between multiple tasks.

Higher layers: The higher layer protocols are application specific. IrCOMM protocol emulates the standard serial port. When two devices such as palmtop and mobile phone both fitted with infrared interface come face to face, they can exchange the data (say, the address book) using the application layer protocols.

6.7 Ethernet

Ethernet interface is now ubiquitous. It is available on every desktop and laptop. With the availability of low-cost Ethernet chips and the associated protocol stack, providing an Ethernet interface is very easy and useful to the embedded system. Through the Ethernet interface, the embedded system can be connected to the LAN. So, a number of embedded systems in a manufacturing unit can be connected as a LAN; and, another node on the LAN, a desktop computer, can monitor all these embedded systems. The data collected by an embedded system can be transferred to a database on the LAN.

The Ethernet interface provides the physical layer and data link layer functionality. Above the data link layer, the TCP/IP protocol stack and the application layer protocols will run. This protocol architecture is shown in Fig. 6.7.

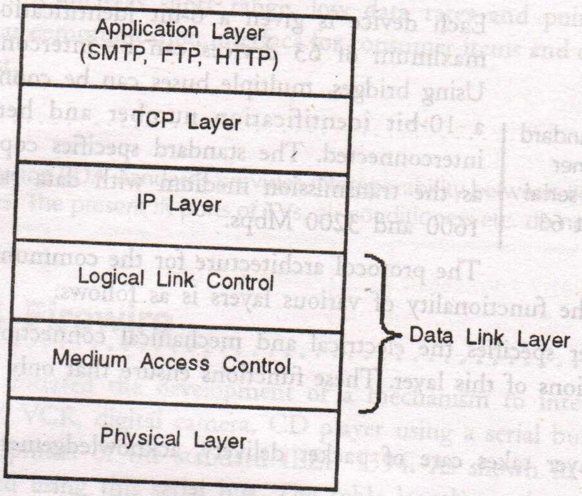


Fig. 6.7: Ethernet LAN Protocol Architecture

Physical layer: The Ethernet physical layer specifies a RJ 45 jack using which the device is connected to the Local Area Network. Unshielded twisted pair or coaxial cable can be used as the medium. Two pairs of wires are used for transmission, one for transmit path and one for receive path. Ethernet transmits balanced differential signals. In each pair, one wire carries signal voltage between 0 to +2.5 volts and the second wire carries signals with voltage between –2.5 volts and 0 volts, and hence the signal difference is 5 volts. The various pin connection details of RJ 45 connector are given in Table 6.5. Speeds of 10 Mbps and 100 Mbps are supported.

Pin number	Function (abbreviation)
1	Transmit Data (TD+)
2	Transmit Data (TD-)
3	Receive Data (RD+)
4	No connection (NC)
5	No connection (NC)
6	Receive Data (RD-)
7	No Connection (NC)
8	No Connection (NC)

Table 6.5: Pin Connections for Ethernet Interface

Data link layer: The data link layer is divided into Medium Access Control (MAC) layer and Logical Link Control (LLC) layer. The MAC layer uses the Carrier Sense Multiple Access/Collision Detection (CSMA/CD) protocol to access the shared medium. The LLC layer specifies the protocol for logical connection establishment, flow control, error control and acknowledgements. Each Ethernet interface will have a unique Ethernet address of 48 bits.

Due to the availability of low-cost Ethernet chips (such as CS 8900A of Cirrus Logic), with little additional cost, an embedded system can be provided with Ethernet connectivity. Even 8-bit micro-

> **In Brief...**
>
> Ethernet standard specifies the physical layer and data link layer standards to connect devices in a Local Area Network. Above the Ethernet protocols, TCP/IP protocol has to be run to make the embedded system Internet-enabled.

controller based embedded systems can be provided the Ethernet interface. To make the embedded system network-enabled, as shown in Fig. 6.7, the upper layer protocols viz., TCP/IP stack has to run above the Ethernet. The TCP/IP stack has to be embedded along with the Operating System and application software in the firmware. If the embedded system has to send mails, Simple Mail Transfer Protocol (SMTP) has to run. To support file transfer application, File Transfer Protocol (FTP) software has to be ported. If the embedded system has to work as a web server, the HTTP server software has to run on the system.

> **Notes...**
>
> In Ethernet, the data link layer is divided into two sub-layers—Medium Access Control (MAC) sub-layer and Logical Link Control (LLC) sub-layer. MAC sub-layer facilitates multiple nodes to access the medium using CSMA/CD protocol. LLC sub-layer defines the protocols for link establishment, flow control and error control.

6.8 IEEE 802.11

IEEE 802.11 family of standards is for Wireless Local Area Networks and Personal Area Networks. These standards cover the physical and MAC layers of Wireless LANs. The LLC layer is same as for the Ethernet LAN. The architecture of IEEE 802.11 standard for Wireless LAN is shown in Fig. 6.8. Each wireless LAN node has a radio and an antenna. All the nodes running the same MAC protocol and competing to access the same medium will form a Basic Service Set (BSS). This BSS can interface to a backbone LAN through Access Point (AP). The backbone LAN can be a wired LAN such as Ethernet LAN. Two or more BSSs can be interconnected through the backbone LAN. In trade magazines, the Access Points are referred as "Hotspots".

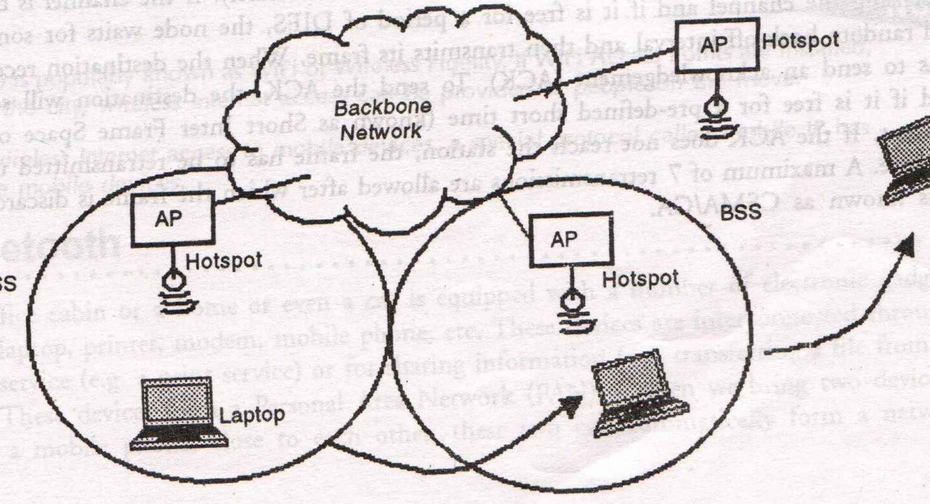


Fig. 6.8: IEEE 802.11 Wireless LAN

The physical medium specifications for 802.11 WLANs are:

- Diffused Infrared with anto operating wavelength between 850 and 950 nm. The data rate supported using this medium is 1 Mbps. 2 Mbps data rate is optional.
- Direct Sequence Spread Spectrum operating in 2.4 GHz ISM band. Up to 7 channels each with a data rate of 1 Mbps or 2 Mbps can be used.
- Frequency hopping spread spectrum operating in 2.4 GHz ISM band with 1 Mbps data rate. 2 Mbps data rate is optional.

ISM (Industrial, Scientific and Medical) band is a 'free' band and hence no government approvals are required to operate radio systems in this band. ISM band frequency range is 2400 – 2483.5 MHz.

Extensions to IEEE 802.11 have been developed to support higher data rates. 802.11b standard has been developed which supports data rates up to 22 Mbps at 2.4 GHz, with a range of 100 meters. Another extension, 802.11a, operates in the 5 GHz frequency band and can support data rates up to 54 Mbps, with a range of 100 meters. 802.11g supports 54 Mbps data rates in the 2.4 GHz band.

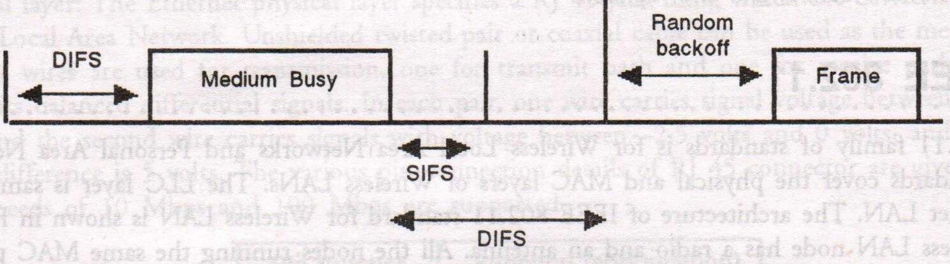
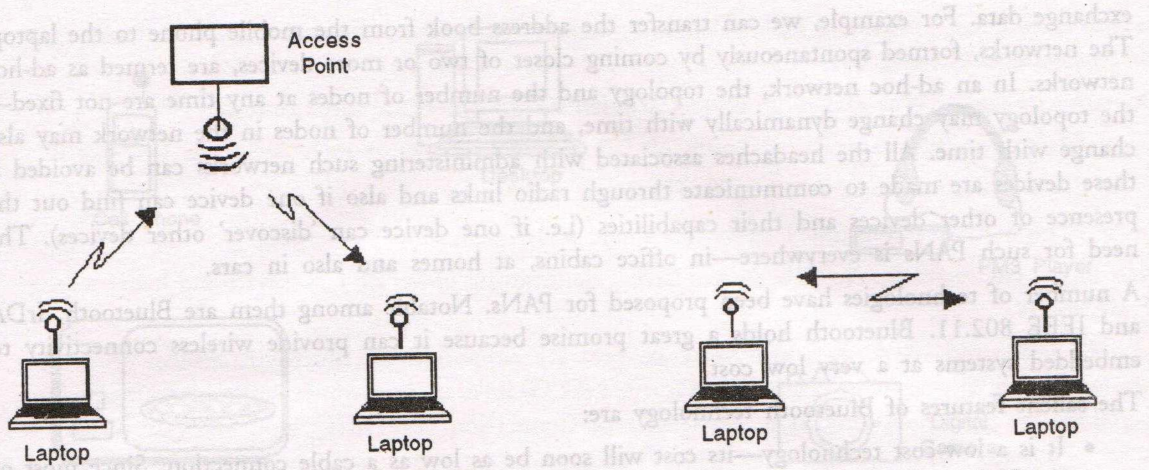


Fig. 6.9: CSMA/CA protocol

The MAC protocol used in 802.11 is called CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). The CSMA/CA operation is shown in Fig. 6.9. Before transmitting, a node senses the radio medium and if the channel is free for a period longer than a pre-defined value (known as Distributed Inter Frame Spacing or DIFS), the node transmits immediately. If the channel is busy, the node keeps sensing the channel and if it is free for a period of DIFS, the node waits for some more period called random back-off interval and then transmits its frame. When the destination receives the frame, it has to send an acknowledgement (ACK). To send the ACK, the destination will sense the medium and if it is free for a pre-defined short time (known as Short Inter Frame Space or SIFS), the ACK is sent. If the ACK does not reach the station, the frame has to be retransmitted using the above procedure. A maximum of 7 retransmissions are allowed after which the frame is discarded. This procedure is known as CSMA/CA.



(a) Communication through Access Point (b) Direct Communication

Fig. 6.10: Communication between nodes in Wireless LAN

An important feature of IEEE 802.11 wireless LAN is that two or more nodes can communicate directly also without the need for a centralized control. The two configurations in which the wireless LAN can operate are shown in Fig. 6.10. In Fig. 6.10(a), the configuration uses the Access Point as described earlier. In Fig. 6.10(b), direct communication between two devices is shown. When two or more devices form a network without the need for centralized control, they are called ad hoc networks. For instance, a mobile phone can form a network with a laptop and synchronize data automatically.

Embedded systems are now being provided with wireless LAN connectivity to exchange data. The main attraction of wireless connectivity is that it can be used in environments where running a cable is difficult such as in shop floors of manufacturing units.

In Brief...

IEEE 802.11 family standards specify the physical and data link layer protocols for providing wireless connectivity to devices. Infrared and ISM bands are used for transmission.

Notes...

IEEE 802.11b is popularly known as WiFi or Wireless Fidelity. If WiFi Access Points are installed, say all over the city, wireless Internet access can be provided to people on the move.

To provide wireless Internet access to mobile devices, a special protocol called Mobile IP has to run on the mobile devices.

6.9 Bluetooth

A typical office cabin or a home or even a car is equipped with a number of electronic gadgets such as desktop, laptop, printer, modem, mobile phone, etc. These devices are interconnected through wires for using a service (e.g. a print service) or for sharing information (e.g. transferring a file from desktop to laptop). These devices form a Personal Area Network (PAN). When we bring two devices, say a laptop and a mobile phone, close to each other, these two can automatically form a network and exchange data. For example, we can transfer the address book from the mobile phone to the laptop. The networks, formed spontaneously by coming closer of two or more devices, are termed as ad-hoc networks. In an ad-hoc network, the topology and the number of nodes at any time are not fixed—the topology may change dynamically with time, and the number of nodes in the network may also change with time. All the headaches associated with administering such networks can be avoided if these devices are made to communicate through radio links and also if one device can find out the presence of other devices and their capabilities (i.e. if one device can 'discover' other devices). The need for such PANs is everywhere—in office cabins, at homes and also in cars.

A number of technologies have been proposed for PANs. Notable among them are Bluetooth, IrDA and IEEE 802.11. Bluetooth holds a great promise because it can provide wireless connectivity to embedded systems at a very low cost.

The salient features of Bluetooth technology are:

- It is a low-cost technology—its cost will soon be as low as a cable connection. Since most of the Bluetooth-enabled devices have to operate through a battery, the power consumption is also very low.
- It is based on radio transmission in the ISM band. ISM band is not controlled by any government authority and hence no special approval is required to use Bluetooth radio systems.
- It caters to short ranges. The range of a Bluetooth device is typically 10 meters, though with higher power, the range can be increased to 100 meters.
- It is based on open standards formulated by a consortium of industries and a large number of equipment vendors are committed to this technology.

Bluetooth Special Interest Group (SIG) founded in February 1998 by Ericsson, Intel, IBM, Toshiba and Nokia released version 1.0 of Bluetooth specifications in July 1999. Version 1.1 of Bluetooth specifications was released in February 2001.

Most of electronic devices can be Bluetooth-enabled. These include a PC, laptop, PDA, digital camera, mobile phone, pager, MP3 player, headset, printer, keyboard, mouse, LCD projector, domestic appliances such as TV, microwave oven, music players etc. To make a device Bluetooth-enabled, a module containing the Bluetooth hardware and firmware is attacked to the device. And, a piece of software is run on the device. A Bluetooth-enabled device communicates with another Bluetooth-enabled device over the radio medium to exchange information or transfer data.

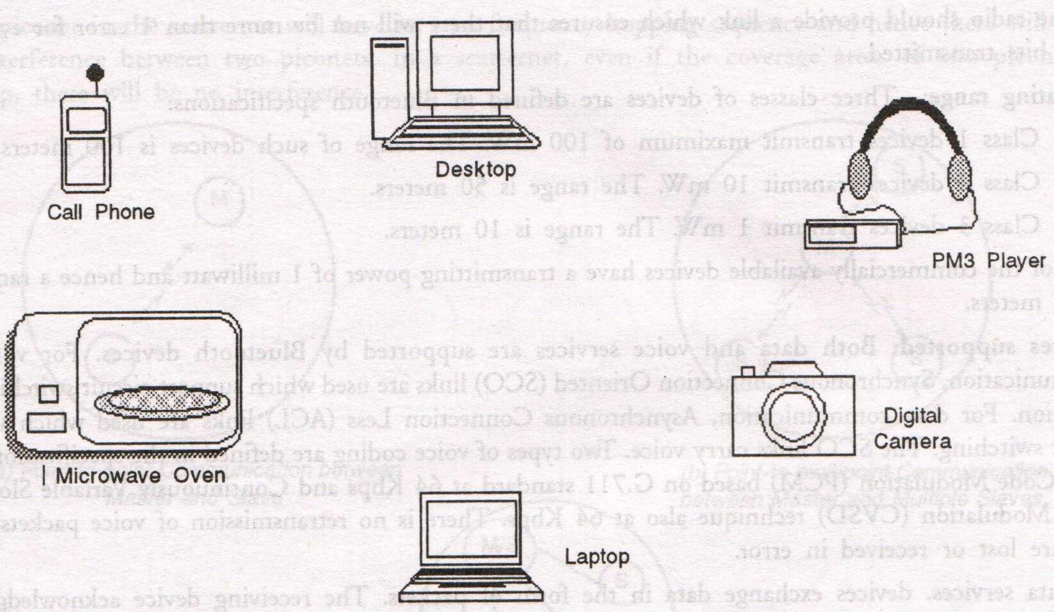


Fig. 6.11: Wireless Personal Area Network

As shown in Fig. 6.11, a set of devices can form a Personal Area Network if they are in the radio vicinity of each other (typically 10 meters radius). When a device comes in the vicinity of another device, Bluetooth protocols facilitate their forming a network. A device can find out what services are offered by the other device and then obtain that service. For example, a laptop can discover the printer automatically and then obtain the print service. Such networks are called ad-hoc networks as the network is formed on the fly and once the device gets out of sight, the network is no longer there. Such networks can be formed in office, at home, in cars and also in public places such as shopping malls, airports etc.

In Brief...

Bluetooth is a low-cost wireless technology to develop Personal Area Networks. Various devices within a vicinity of about 10 meters can form a network using Bluetooth wireless link.

6.9.1 Bluetooth System Specifications

The specifications of the Bluetooth system are as follows:

Frequency of operation: Bluetooth devices operate in the ISM band in the frequency range 2400 – 2483.5 MHz. This band consists of 79 channels each of 1 MHz bandwidth, with a lower guard band of 2 MHz and upper guard band of 3.5 MHz. When a device transmits its data, it uses frequency hopping, i.e. the device transmits each packet in a different channel. The receiving device has to switch to that channel to receive that packet. Though the radio design becomes complex when frequency hopping is used, the advantage is that it provides secure communication. Nominal frequency hop rate is 1600 hops per second.

Modulation: Gaussian Frequency Shift Keying (GFSK) is used as the modulation technique. Binary 1 is represented by a positive frequency deviation and 0 by negative frequency deviation. The radio receiver has to be designed in such a way that the Bit Error Rate (BER) of minimum 0.1% is ensured, i.e. the radio should provide a link which ensures that there will not be more than 1 error for every 1000 bits transmitted.

Operating range: Three classes of devices are defined in Bluetooth specifications:

- Class 1 devices transmit maximum of 100 mW. The range of such devices is 100 meters.
- Class 2 devices transmit 10 mW. The range is 50 meters.
- Class 3 devices transmit 1 mW. The range is 10 meters.

Most of the commercially available devices have a transmitting power of 1 milliwatt and hence a range of 10 meters.

Services supported: Both data and voice services are supported by Bluetooth devices. For voice communication, Synchronous Connection Oriented (SCO) links are used which support circuit switching operation. For data communication, Asynchronous Connection Less (ACL) links are used which use packet switching. The SCO links carry voice. Two types of voice coding are defined in the specifications: Pulse Code Modulation (PCM) based on G.711 standard at 64 Kbps and Continuously Variable Slope Delta Modulation (CVSD) technique also at 64 Kbps. There is no retransmission of voice packets if they are lost or received in error.

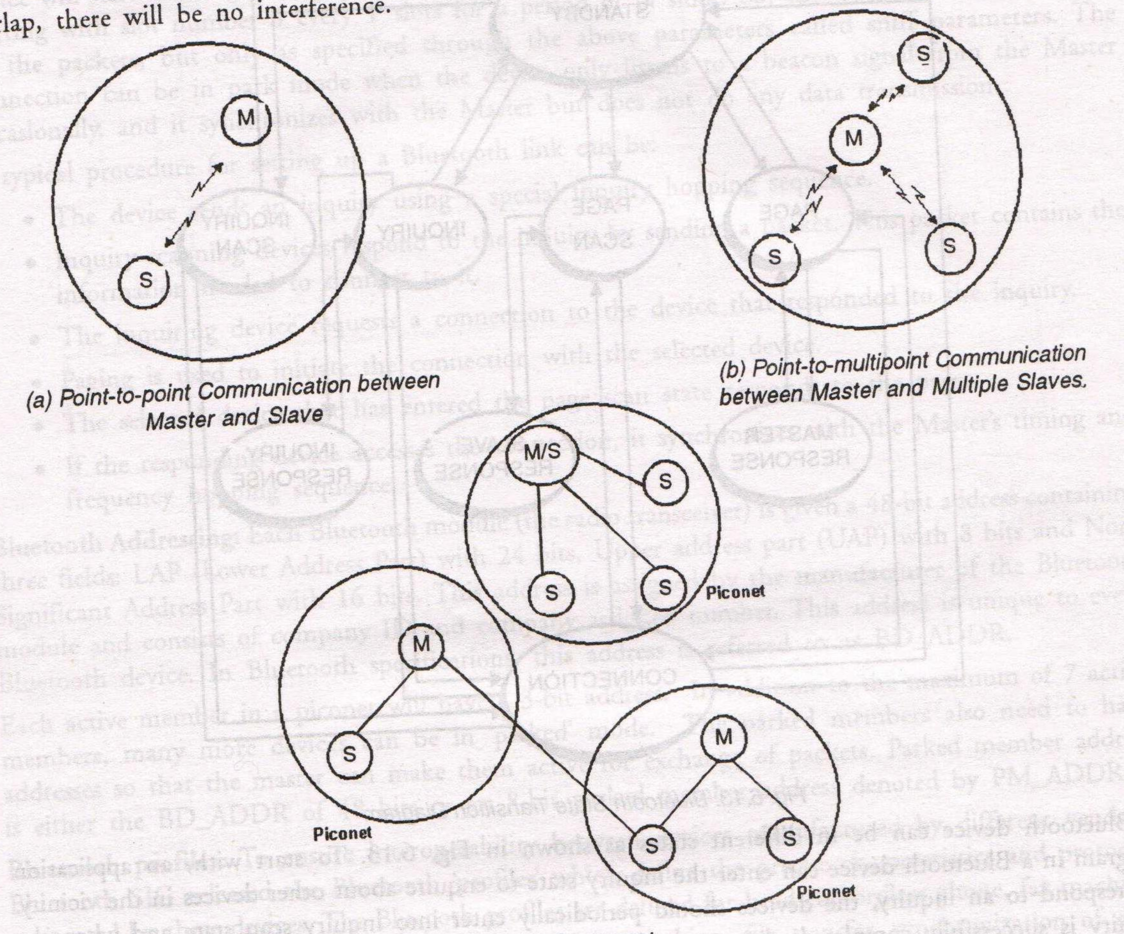
For data services, devices exchange data in the form of packets. The receiving device acknowledges the packet or reports that the packet is received in error. If a packet is received with errors, the packet is retransmitted. It is also possible to broadcast packets by one device to all the other devices in the network. However, in broadcast mode there is no acknowledgement or indication that the packet is received with errors. The broadcasting device informs the receiving devices how many times a broadcast packet will be transmitted so that at least once every device will receive the packet without errors.

Data rates: A Bluetooth device can support three synchronous voice channels and one asynchronous data channel. For voice communication, 64 Kbps data rate is used in both directions. For asynchronous links, two types of channels are defined with different data rates. In asymmetric channel, data rates are 723.2 Kbps in one direction and 57.6 Kbps in the other direction. In symmetric channel, data rate is 433.9 Kbps in both directions.

Network topology: In a PAN, a set of devices form a small network called piconet. In a piconet, there will be one Master and one or more Slaves. All the Slaves tune to the Master. The Master decides the hop frequency sequence and all the Slaves tune to these frequencies to establish communication links. Any device (desktop, mobile phone etc.) can be a Master or Slave. The Master/Slave terminology is only for the protocols, the device capabilities are not defined by this terminology. It is also possible for a Master and Slave to switch roles—a Slave can become a Master. A piconet can have maximum number of seven slaves which can actively communicate with the Master. In addition to these active slaves, a piconet can contain many Slaves that are in parked mode. These parked devices are synchronized with the Master, but they are not active on the channel. The communication between the Master and the Slave uses Time Division Duplex (TDD).

Fig. 6.12 shows the various topologies of a Bluetooth piconet. In Fig. 6.12(a), a piconet is shown with one Master and one Slave. It is a point-to-point communication mode. In Fig. 6.12(b), the piconet consists of a Master and a number of Slaves. It is a point-to-multipoint communication mode. Fig. 6.12(c) shows a scatternet which is formed by a number of piconets. In this scatternet, each piconet will have a Master and a number of Slaves. The Master of a piconet can be a Slave in another piconet.

Each piconet in the scatternet will have its own frequency hopping sequence and hence there will be no interference between two piconets. In a scatternet, even if the coverage areas of two piconets overlap, there will be no interference.



(a) Point-to-point Communication between Master and Slave

(b) Point-to-multipoint Communication between Master and Multiple Slaves.

(c) Scatternet

Fig. 6.12: Bluetooth Piconet & Scatternet

Communication between Master and Slave: The Master and Slave communicate in the form of packets. Each packet is transmitted in a time slot. Each time slot is of 625 microseconds duration. These slots are numbered from 0 to $2^{27}-1$. Master starts the transmission in even slots by sending a packet addressed to a slave and the slave sends the packets in odd numbered slots. A packet generally occupies one time slot, but can extend up to five slots. If a packet extends more than one slot, the hop frequency will be the same for the entire packet. If the Master starts the transmission in slot 0 using frequency f1, the slave transmits in slot 1 using frequency f2, master transmits in slot 2 using frequency f3, and so on.

In Brief...

Bluetooth-enabled devices form a network known as piconet. In a piconet, there will be one master and up to seven active slaves. Communication between the master and slaves can be point-to-point or point-to-multipoint.

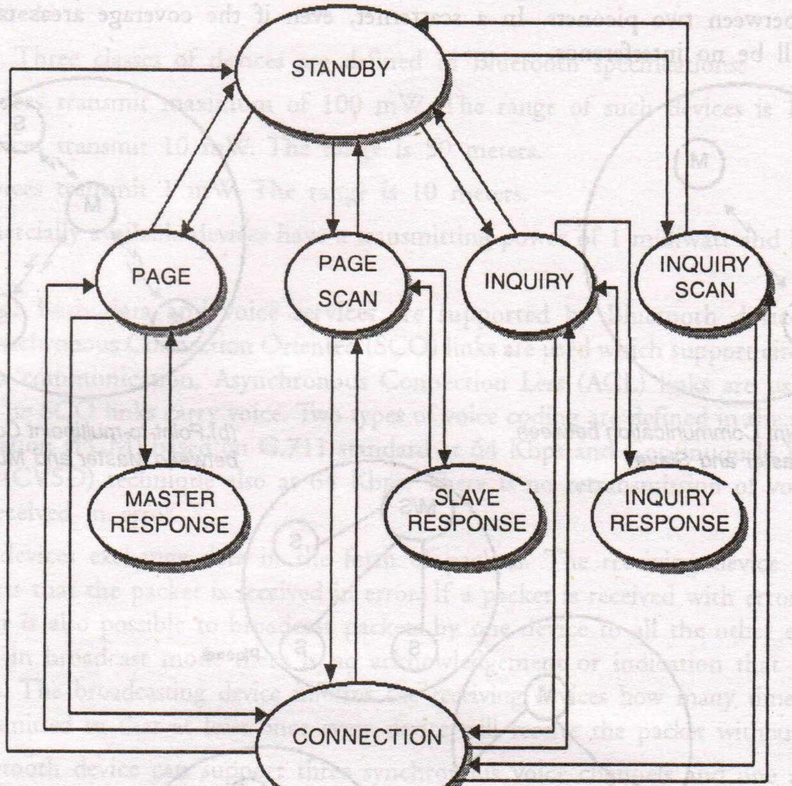


Fig. 6.13: Bluetooth State Transition Diagram

A Bluetooth device can be in different states as shown in Fig. 6.13. To start with, an application program in a Bluetooth device can enter the inquiry state to enquire about other devices in the vicinity. To respond to an inquiry, the devices should periodically enter into inquiry scan state and when the inquiry is successfully completed, they enter the inquiry response state. When a device wants to get connected to another device, it enters the page state. In this state, the device will become the Master and page for other devices. The command for this paging has to come from an application program running on this Bluetooth device. When the device pages for the other device, the other device may respond and the Master enters the master response state. Devices should enter the page scan state periodically to check whether other devices are paging for them. When device receives the page scan packet, it enters the slave response state.

In Brief...

Communication between the master and slaves is achieved using packet transmission. Each device sends its packets using frequency hopping. Master sends the packets in even numbered slots and the slaves send the packets in odd numbered slots.

Once paging of devices is completed, the Master and the Slave establish a connection. Thereafter, the connection is in active state, during which the packet transmission takes place. The connection can also be put in one of the three modes: 'hold' or 'sniff' or 'park' modes. In hold mode, the device will stop receiving the data traffic for a specific amount of time so that other devices in the piconet can use the channel. After the expiry of the specific time, the device will start listening to traffic again. In sniff mode, a slave will be given an instruction like 'listen starting with slot number S every T slots for a period of N slots'. So, the device need not listen to all the packets, but only as specified through the above parameters called sniff parameters. The connection can be in park mode when the device only listens to a beacon signal from the Master occasionally, and it synchronizes with the Master but does not do any data transmission.

A typical procedure for setting up a Bluetooth link can be:

- The device sends an inquiry using a special inquiry hopping sequence.
- Inquiry scanning devices respond to the inquiry by sending a packet. This packet contains the information needed to connect to it.
- The inquiring device requests a connection to the device that responded to the inquiry.
- Paging is used to initiate the connection with the selected device.
- The selected device that has entered the page scan state responds to the page.
- If the responding device accesses the connection, it synchronizes with the Master's timing and frequency hopping sequence.

Bluetooth Addressing: Each Bluetooth module (the radio transceiver) is given a 48-bit address containing three fields: LAP (Lower Address Part) with 24 bits, Upper address part (UAP) with 8 bits and Non-Significant Address Part with 16 bits. This address is assigned by the manufacturer of the Bluetooth module and consists of company ID and company assigned number. This address is unique to every Bluetooth device. In Bluetooth specifications, this address is referred to as BD_ADDR.

Each active member in a piconet will have a 3-bit address. In addition to the maximum of 7 active members, many more devices can be in 'parked' mode. The parked members also need to have addresses so that the master can make them active for exchange of packets. Parked member address is either the BD_ADDR of 48 bits or an 8-bit parked member address denoted by PM_ADDR.

Bluetooth profiles: To ensure interoperability between devices manufactured by different vendors, Bluetooth SIG released the Bluetooth 'profiles' which define the precise characteristics and protocols supported by these devices. The Blueooth profiles are defined for headset, cordless phone, fax machine, LAN Access Point, serial communication, dial-up networking, file transfer, synchronization of data between two devices, etc.

Notes...

To achieve interoperability of Bluetooth modules of different manufacturers, Bluetooth profiles are defined for different devices and services.

6.9.2 Bluetooth Protocol Architecture

The complete protocol stack of Bluetooth system is given in Fig. 6.14.

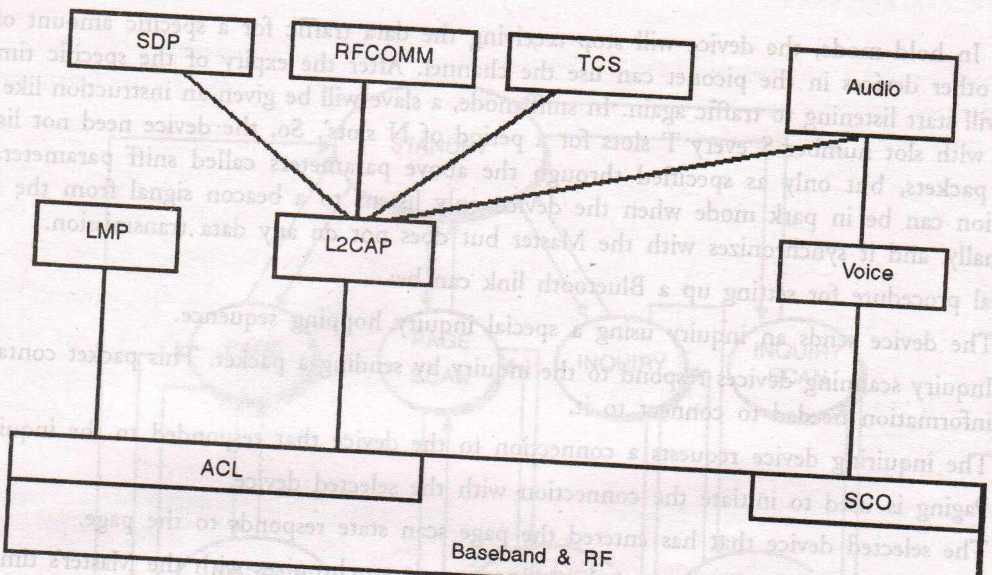


Fig. 6.14: Bluetooth Protocol Architecture

Baseband and RF

The baseband layer is for establishing the links between devices based on the type of service required—ACL for data services and SCO for voice services. This layer also takes care of addressing and managing the different states of the Bluetooth device. The RF portion provides the radio interface.

Link Manager Protocol (LMP)

The Link Manager Protocol (LMP) is used to set up and control links. The three layers—RF, Link controller and the Link manager—will be on the Bluetooth module attached to the device. The link manager on one device exchanges messages with the link manager on the other device. These messages, known as LMP messages, are not sent to higher layers. Link messages have higher priority compared to data. LMP messages are sent as single slot packets, with a header of 1 byte. The functions of the LMP are as follows:

- **Authentication:** When two devices have to communicate with each other, one has to verify the other device. So, one device is called verifier and the other is called claimant. The verifier sends a message, a packet containing a random number, which is called a challenge. The claimant calculates the response which is a function of challenge and sends the response along with its Bluetooth address (48-bit address) and secret key. This is known as Challenge-Response scheme—you throw a challenge and check whether the other device can correctly respond to that challenge.
- **Encryption:** To maintain confidentiality of data over the radio link, the data is encrypted. The Master sends a key with which the data is encrypted to all the slaves, through an LMP message.
- **Clock offset request:** Synchronizing the clocks between the Master and the Slaves is a must for proper exchange of data. If the clock has to be offset, the LMP exchanges messages to ensure clock synchronization.

- **Timing accuracy information request:** To ensure synchronization, the master can request the slaves for timing accuracy information.
- **LMP version:** It needs to be ensured that both the devices use the same version of LMP. To achieve this, version number of the LMP protocol is exchanged.
- **Type of packets supported:** As different Bluetooth-enabled devices may support different features, LMP features request and response are exchanged between the devices.
- **Switching Master/Slave role:** In a piconet, a device will act as a Master and other devices will act as Slaves. The Master and the Slave in a piconet can switch roles using the LMP messages. The Master or the Slave can initiate the switching operation.
- **Name request:** Each device can be given a user-friendly name having a maximum of 248 bits in ASCII format. A device can request for the name through an LMP message and obtain the response.
- **Detach:** Messages exchanged to close a connection.
- **Hold mode:** To place ACL link in hold for a specified time when there is no data to send. This feature is mainly to save power.
- **Park mode:** To be in synchronization with the Master but not participate in data exchange.
- **Power control:** To request for transmitting less power. This is useful particularly for class 1 devices which are capable of transmitting 100 mW power.
- **Quality of Service (QoS) parameters exchange:** In applications that require good quality transmission link, quality of service parameters can be specified. These parameters include number of repetitions for broadcast packets, delay and bandwidth allocation.
- **Request SCO link:** To request for an SCO link after the ACL link is established.
- **Multi-slot packet control:** To control the procedure when data is sent in consecutive packets.
- **Link supervision:** To monitor link when device goes out of range (through a time-out mechanism).

In Brief...

The important functions of Link Management Protocol are: link establishment and disconnection, switching between master and slave roles, authentication, encryption and controlling quality of service parameters.

- **Connection establishment:** After paging is successfully completed, to establish the connection.

A Bluetooth device will implement the base band, RF and LMP/ layers in a hardware/firmware combination. A 16-bit processor based system is used to implement these three layers of protocols. However, to reduce the cost, single chip solutions are now available which will reduce the cost to make a device Bluetooth-enabled. These three layers ensure establishment of a connection and managing the connection for transfer of voice or data. But to ensure that the whole application runs as per user requirements, we need lot of other protocols.

In Brief...

To Bluetooth-enable a device, a Bluetooth module is attached to the device. This module contains the hardware and firmware that implements the baseband, RF and link management protocols.

Logical Link Control and Adaptation Protocol (L2CAP)

L2CAP runs above the baseband and carries out the data link layer functionality. L2CAP layer is only for ACL links. L2CAP data packets can be up to 64 Kilobytes long. L2CAP protocol runs on hosts such as laptop, cellular phone or other wireless devices.

When L2CAP messages are exchanged between two devices, it assumes that an ACL link is already established between two devices. It also assumes that packets are delivered in sequence. L2CAP does not do any checksum calculation. Note that L2CAP does not support SCO links for voice communication. L2CAP does not support multicasting.

The functions of L2CAP layer are:

- **Protocol multiplexing:** In the protocol stack given in Fig. 6.14, above L2CAP, a number of other protocols can be running. A packet received by L2CAP has to be passed onto the correct higher layer. This is protocol multiplexing.
- **Segmentation and reassembly:** Baseband packets are limited in size. Large L2CAP packets are segmented into small baseband packets and sent to the baseband layer. Similarly, the small packets received from the baseband layer are reassembled and sent to higher layers.
- **Quality of Service:** Quality of Service (QoS) parameters such as delay can be specified, and this layer ensures that the QoS constraints are honored.

In Brief...

The functions of Logical Link Control and Adaptation Protocol (L2CAP) are protocol multiplexing, segmentation and reassembly and ensuring that the quality of service parameters are honored.

L2CAP layer sends connection request and QoS request messages from the application programs through the higher layers. It receives from the lower layers the responses for these requests. The responses can be: connection indication, connection confirmation, connect confirmation negative, connect confirmation pending, disconnection indication (from remote), disconnect confirmation, timeout indication and quality of service violation indication.

Service Discovery Protocol (SDP)

The Service Discovery Protocol (SDP) provides the Bluetooth environment the capability to create ad-hoc networks. This protocol is used for discovering the services offered by a device. SDP offers the following services:

- A device can search for the service needed by it in the piconet.
- A device can discover a service based on a class of services (e.g. A laptop wants a print service, and it can find out the different printers available in the piconet—dot matrix printer, laser printer etc., and subsequently select the desired print service).
- Browsing of services.
- Discovery of new services when devices enter in the radio range of other devices.
- Mechanism to find out when a service becomes unavailable when the device goes out of radio range.
- The details of services such as classes of services and the attributes of services.
- To discover services on another device without consulting the third device.

When a device wants to discover a service, the application software initiates the request (which is the client) and the SDP client sends SDP request to the server (the device which can provide the required service). SDP client and server exchange SDP messages. Note that the server and client can be any two devices—the server is the device that can provide the service being requested by the client.

The server maintains list of service records. Each record is identified by a unique 32-bit number. Service record will have a number of attributes. The attributes can be service class ID list (type of service), service ID, protocol description list (protocol used for using the service), provider name, Icon URL (an iconic representation of the service), service name and service description. Each attribute will have two components: attribute ID and attribute value.

For instance, consider a device (a laptop) that requires a print service. The laptop is a client looking for a print service in a Bluetooth environment. The procedure for obtaining this service is as follows:

- Client sends a service search request specifying the print service class ID to the Server.
- Server sends a service search response to the client indicating that two print services are provided.
- Client sends a service attribute request, protocol descriptor list to the server, asking for the details of the service.
- Server sends the response to the client indicating that PostScript print service is provided.
- If the client wants to use the service, it sends a command to print the desired document.

The SDP is the heart of the Bluetooth system as it provides the capability to discover availability of services and the details of the services along with the necessary information such as protocols to access the service.

In Brief...

Service Discovery Protocol (SDP) provides the capability to discover the availability of services in a Bluetooth network and to access these services.

Notes...

Service Discovery Protocol (SDP) is the heart of any network that supports ad-hoc networking of computers/devices. Sun Microsystems' Jini technology also has SDP that facilitates development of 'administration-free' networking of devices.

RFCOMM

RFCOMM is a transport protocol to emulate serial communication (RS232 serial port) over L2CAP. Through RFCOMM, two devices can communicate using serial communication protocols over Bluetooth radio. To achieve this, RFCOMM emulates the 9 signals of RS 232. These signals are:

102 Signal Ground (GND)

103 Transmit Data (TD)

104 Received Data (RD)

105 Request to Send (RTS)

106 Clear to Send (CTS)

107 Data Set Ready (DSR)

108 Data Terminal Ready (DTR)

109 Data Carrier Detect (DTR)

125 Ring Indicator (RI)

In Brief...

RFCOMM emulates serial communication protocol, i.e. using radio as the transmission medium, data can be transmitted using the serial communication protocol such as RS232 standard.

RFCOMM is derived from the GSM (Global System for Mobile Communications) specifications TS 07.10 for serial emulation. It supports two types of devices. Type 1 devices are communication end-points such as computers and printers. Type 2 devices are part of communication segment such as modems.

Telephony Control Protocol Specifications (TCS)

To establish voice communication between two Bluetooth devices, we need the SCO links. SCO links are not handled by L2CAP protocol. However, L2CAP handles the signaling required for establishing voice connections through Telephony Control Protocol Specification abbreviated TCS. Note that it is not abbreviated as TCP: TCP stands for Transmission Control Protocol used in the Internet protocol architecture. TCS defines call control signaling for establishing speech and data calls between Bluetooth devices and mobility management procedures. This protocol is based on the International Telecommunications Union (ITU) standard Q.931, which is the standard for signaling in Integrated Services Digital Network (ISDN). TCS messages are exchanged between devices to establish and release connections and to provide supplementary services such as calling line identification (to identify the telephone number of the calling subscriber).

In Brief...

Telephony Control Protocol Specification (TCS) handles signaling information to establish voice and data calls between Bluetooth devices. This protocol is based on Q.931 standard using in ISDN.

Host Control Interface

If we have to Bluetooth-enable a laptop computer, we can connect a small Bluetooth module to the USB port of the laptop and run the protocol stack on the laptop (called the host). Bluetooth device will have two parts: a module implementing the lower layers (LMP and below) and a software module implementing higher layers stack (L2CAP and above). The software module runs on the laptop (the host). The Host Controller Interface (HCI) provides a standard interface between the Bluetooth module and the host software, so that we can buy the hardware module from one vendor and software module from another vendor. HCI uses three types of packets:

- Commands which are sent from the host to the module,
- Events which are sent from the module to the host, and
- Data packets which are exchanged between the host and the module.

The functions of HCI are

- Setting up and disconnection of the links and configuring the links.
- Control of baseband features such as timeouts.

- Retrieving of status information of the module.
- Invoking the test modes to test the module for local testing of Bluetooth devices.

HCI provides command interface to the baseband controller and link manager as well as access to hardware status and control registers. HCI has to reside in the Bluetooth module connected to the laptop as well as the host. In the Bluetooth module firmware, HCI commands are implemented so that the host can access the baseband commands, link manager commands, hardware status registers, control registers and event registers. The Bluetooth module is connected to the USB port (say, of the laptop). Three interfaces are defined to get HCI packets from host to the Bluetooth module: (a) USB; (b) RS232; and (c) UART.

In the host, the bus driver is implemented as software above which the HCI driver software and other higher layer protocol softwares are implemented. The HCI commands can be categorized as

- Link control commands to establish piconets and scatternets
- Link policy commands to put devices in hold mode or sniff mode
- Commands to get information about the local hardware
- Commands to get the status parameters
- Commands to test the local Bluetooth module.

In Brief...

Host Control Interface (HCI) provides a standard interface between the Bluetooth module attached to a device and the software running on the device. Through the HCI, the Bluetooth device obtains the commands to establish piconets and scatternets, to obtain status information and test the Bluetooth module.

Each Bluetooth module is itself an embedded system built around a 16-bit processor. To make low-cost Bluetooth module, single-chip solutions are now becoming available. Bluetooth-enabling consumer electronic items as well as embedded systems will facilitate low-cost networking.

Summary

In this chapter, the important communication interfaces for embedded systems are presented. RS232 interface provides a serial communication interface for data rates up to 115.2 Kbps and distances up to about 100 meters. It is one of the most widely used interfaces. For longer distances and for noisy environments, RS422 and RS485 interfaces are used. These serial interfaces are now being replaced by Universal Serial Bus which provides very high data rates. USB 1.1 supports up to 12 Mbps and USB 2.0 up to 480 Mbps. Up to 127 devices can be connected in hierarchical fashion using USB. Infrared interface provides low-cost point-to-point communication between devices. With the availability of low-cost Ethernet chips, Ethernet interfaces can be provided to embedded systems to make them network-enabled. By incorporating Ethernet interface and running the TCP/IP protocol stack in the embedded software, the embedded system can be Internet-enabled and can be accessed over a LAN or over the public Internet. Wireless LAN interfaces based on IEEE 802.11 family standards provide wireless communication interface to the embedded systems. Bluetooth is another communication interface that uses wireless technology. Using Bluetooth, two or more devices can form an ad hoc network and share data. In fact, Ethernet, IEEE 802.11 and Bluetooth interfaces are embedded systems and when these embedded systems are integrated into the embedded devices, they become powerful communicating devices.