Module

5

Embedded Communications

Lesson

24

Parallel Data Communication

Instructional Objectives

After going through this lesson the student would be able to

- Explain why a parallel interface is needed in an embedded system
- List the names of common parallel bus standards along with their important features
- Distinguish between the GPIB and other parallel data communication standards
- Describe how data communication takes place between the controller, talker and listener devices connected via a GPIB interface

Questions

Question	Visual (If any)	A	B	C	D	Ans.
Parallel Data Communication is preferred when the following conditions are satisfied:						D
i) distance between the devices is small		T	T	F	T	
ii) the volume of traffic is small		T	F	T	F	
iii) the required data rate is high		T	F	T	T	
The IEEE 488 standard was originally developed by		Intel	IBM	HP	Sun	C
The devices connected in a GPIB system are classified into the following types of categories		1	2	3	4	C
Each device connected in a GPIB system has an n-bit address where n=		3	5	6	7	B

Parallel Data Communication

Data processed by an embedded processor need to be conveyed to other components in the system, namely, an instrument, a smart actuator, a hard disk or a communication network for onward transmission to a central data warehouse. Similarly data may have to be fetched from a digital oscilloscope, a CD-ROM Drive or a sensor from the field. Typically, when the physical distance between the processor and the other component is small, say a within a few meters and a high volume of data need to be conveyed in a short time, parallel bus interfaces are used.

In this lesson, we first learn about one of the most popular parallel bus standards, namely the IEEE 488 standard, also known as the GPIB (formerly HPIB). Next we compare and contrast it with the other similar standards. Finally we discuss about its future particularly in view of the recently emerging high-serial bus standards like the USB.

go to top

The IEEE 488 (GPIB, HPIB) Standard

This BUS-SYSTEM was designed by Hewlett Packard’s (Currently known as Agilent Technologies) ***Test & Measurement Division***, in 1960s and was named as HPIB, a short form for ***<u>H</u>ewlett <u>P</u>ackard <u>I</u>nterface <u>B</u>us,*** to control programmable instruments that were manufactured by the company. It was a short-range digital communications cable standard.

Because of its success and proven reliability, in 1973 the HPIB bus became an American Standard, adopted by the IEEE and renamed as GPIB, for <u>G</u>eneral <u>P</u>urpose <u>I</u>nterface <u>B</u>us. The standard's number is IEEE488.1.

In parallel, the International Electronic Commission (IEC), responsible for the international standardization outside the U.S., approved the standard and called it IEC625.1. Due to introduction of a new naming scheme for all standards, it was renamed to IEC60625.1 later.

There was a slight difference between the IEEE488.1 and IEC625.1. The IEC625.1 standard used a 25 pin DSUB connector for the bus, the IEEE488.1 standard favored a Centronics-like 24 pin connector. Today, the 24-pin connector is always used, but there are also adaptors available in case older instruments are equipped with a 25-pin DSUB connector.

The '.1’ extension of IEEE488.1 / IEC60625.1 indicates that there are several layers of interface standards. In fact, there is a whole 'family' of standards:

- IEEE488.1 / IEC60625.1 defines the physical layer of the bus system.
- IEEE488.2 / IEC60625.2 is not a revision of the '.1' standard, it extends its functionality. A command language (syntax) is defined and common properties of instruments are defined. Same command names result in similar actions. In contrast to the '.1' standard that defines physical means like cables, timing and so on, the '.2' standard focuses on the instrument model.
- An application of IEEE488.2 / IEC60625.2 is IEEE1174. It is currently adopted. Briefly stated, it translates GPIB functionality to a serial RS232 line, albeit without networking capability. It is intended for low cost instruments.

go to top

go to top

Thus GPIB has several versions and makes which reflect the same thing, courtesy to the various developments pertaining to its history.

GPIB Electrical and Mechanical Specifications:

The BUS actually comprises a 24 Wire Cable with both MALE and FEMALE Connectors at each of the individual ends to facilitate the connectivity in a daisy-chain network topology.

Standard TTL level signals are assumed for the ACTIVE, INACTIVE and TRANSITION states both for Control and Communication.
Specified Transfer Rate: 1 Mega Byte per second.
Cable length:
Twenty meters between Controller and one Device or
Two meters between two devices

Device fanout : Number of instruments may range from Eight to Ten.

CLASSIFICATION of Instruments or Devices (as are called in the Standard) connected through this bus system:

- TALKER: Designated to send data to other instruments eg., Tape Readers, Data Recorders, Digital Voltmeters, Digital Oscilloscopes etc.
- LISTENER:Designated to receive data from other instruments or Controllers, eg., Printers, Display devices, Programmable Power Supplies, Programmable Signal Generators etc.
- CONTROLLER: Decision maker for the designation of an instrument either as a TALKER or a LISTNER. Usually this role is carried out by a computer.

go to top

All the Talkers, Listeners and the Controller are connected to each other via the following three different SYSTEM BUSES:

(Also see A TYPICAL SEQUENCE of DATA FLOW)

- Bidirectional Databus
- Bus management Lines
- Handshake Lines

Eight BI-DIRECTIONAL DATALINES have the following functionalities. These are used to transfer Data, Addresses, Commands and Status information in the form of Bytes.

- DATA : Transferred as BYTES with the reception of each data byte being duly acknowledged.
- ADDRESSES :Instruments intended for use on a GPIB usually have some switches which allow a selection of 5-bit address the instrument will be assuming on the BUS. Addresses are characterized as :
    - TALK ADDRESSES
    - LISTEN ADDRESSES
- CONTROL and COMMAND: BYTES containing information for orienting the devices to perform the functions like listen, talk etc. These commands can be referred to as the CONTROL WORDs necessary for establishing efficient communication between the Controller and the other class of devices.

  The various commands are: (also see the COMMAND TABLE)

    - UNIVERSAL Commands
    - UNLISTEN Commands
    - UNTALK Commands
    - SECONDARY Commands

Note: The Commands are sent by the Controller to the instruments.

Five BUS MANAGEMENT LINES have the functionalities as follows:

- IFC : Interface Clear
- ATN : Attention
- SRQ : Service Request
- REN : Remote Enable
- EOI : Identify

go to top

go to top

Three HANDSHAKE LINES having the functionality of coordinating the transfer of data bytes on the data bus.These functions can be elaborated as :

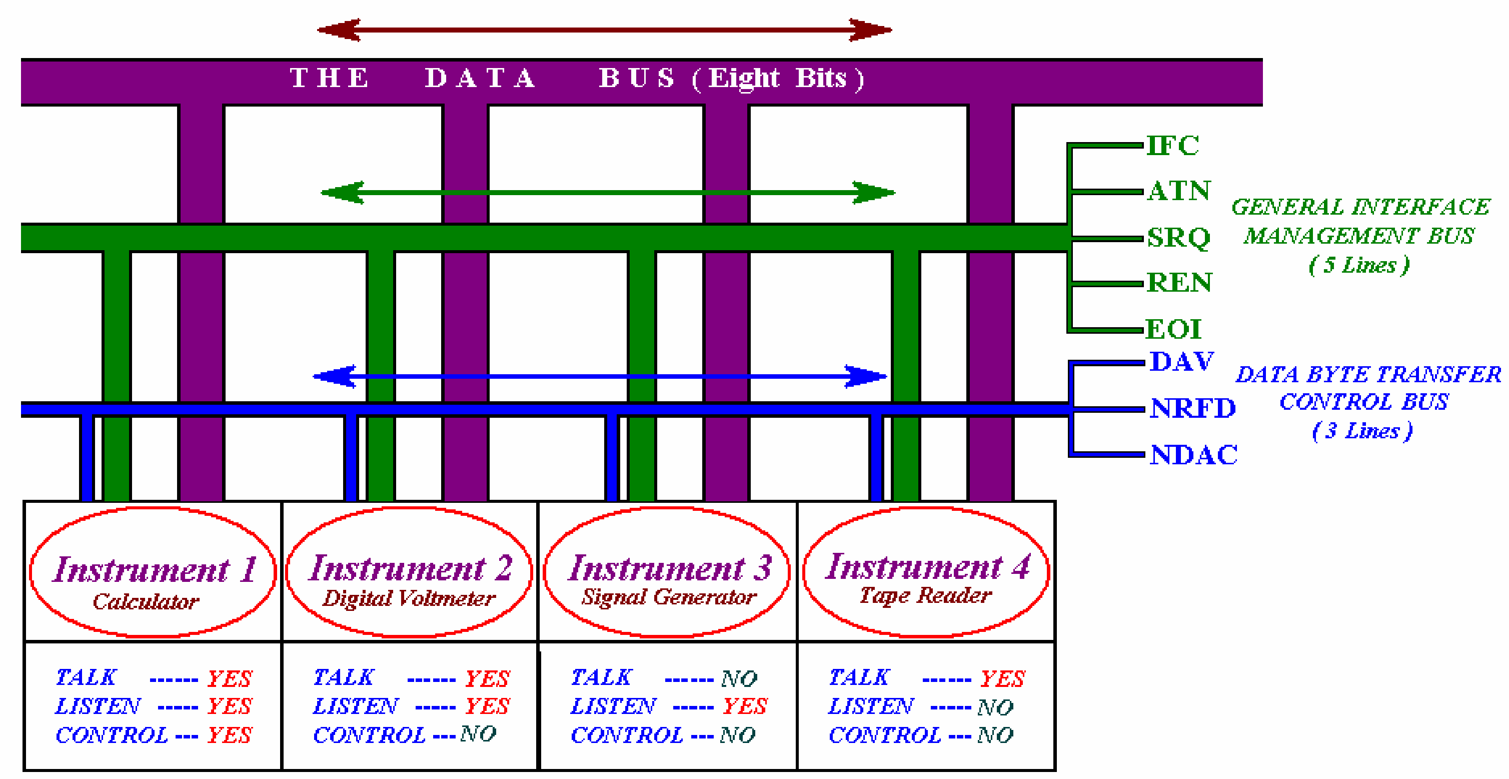
- DAV : Data Valid
- NRFD : Not Ready For Data
- NDAC : Not Data Accepted

Note: The Handshake Signals are necessary to facilitate transmission at different BANDWIDTHS (Data Rates).

go to top

go to top

The Block Diagram


THE DATA BUS (Eight Bits)
IFC
ATN
SRQ
REN
EOI
GENERAL INTERFACE MANAGEMENT BUS (5 Lines)
DAV
NRFD
NDAC
DATA BYTE TRANSFER CONTROL BUS (3 Lines)

Instrument 1 Calculator	Instrument 2 Digital Voltmeter	Instrument 3 Signal Generator	Instrument 4 Tape Reader
TALK ------ YES	TALK ------ YES	TALK ------ NO	TALK ------ YES
LISTEN ----- YES	LISTEN ----- YES	LISTEN ----- YES	LISTEN ----- NO
CONTROL --- YES	CONTROL --- NO	CONTROL --- NO	CONTROL --- NO

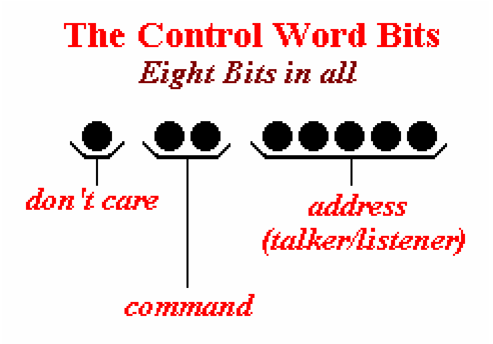
The GPIB "Bus-Structure"

go to top

go to top

The SEQUENCE of events pertaining to the actual communication is as follows:

- Power On: Controller takes up the Control of Buses and sends out the IFC signal to set all instruments on the bus to a known state.
- Controller starts performing the desired series of measurements or tests.
- Controller asserts the ATN line low and starts sending the command – address codes to the talkers and the listeners.
- The CONTROL WORD Structure:



The Control Words are given in brief in the Command Table:

The Command Table

COMMAND	**CONTROL WORD**
Ignored	X1111111
Listen Command	X01 + 5 LSBs (actual address)
Talk Command	X10 + 5 LSBs (actual address)
Universal Command	X000 + 4 LSBs (16 Commands)
Unlisten Command	X0111111
Untalk Command	X1011111
Secondary Commands	X11 + 5 LSBs (actual address)

Note:

- All the Commands Control words are activated only if the ATN line is asserted low; otherwise, they are in a disabled state.
- ‘X’ here represents the don’t care condition.
- ‘+’ here represents the NEXT indicated number of LSBs.

The following are the most important features:

- The Universal Commands go to all the Listeners and Talkers.
- The Untalk or Unlisten Commands are for TURNING ‘on’ or ‘off’ the indicated device.
- In addition to all the above-indicated tasks the controller checks for the SRQ line in the context of SERVICE REQUEST.

  - On finding it as LOW, it POLLS each device on the bus in a se
fashion, that is,‘one-by-one’ or in parallel.
  - It then determines the source of the SRQ, and asserts the ATN line low.
  - It then sends the relevant information or command to all the listners and the talkers depending on the data utility.

go to top

- The controller again asserts the ATN line high and data is transferred di
from the TALKER to the LISTENERS using a double-handshake-signal sequence.

go to top

Some information about DAV, NRFD, and NDAC are given below:

- All are OPEN-COLLECTOR.
  - A listener can hold NRFD low to indicate that it is not ready for data.
  - A listener can hold NDAC low to indicate that it has not yet accepted a data byte.
- An Instance for the above two points can be sited as follows:
  - All Listeners release the NRFD line indicating that they are ready to receive data.
  - The Talker assets the DAV low to indicate that a valid data is on the bus.
  - All the addressed listeners then pull NRFD low and start accepting the data, NDAC line being asserted as high.
  - The talker, on sensing the NDAC line getting high unasserts the corresponding DAV signal. The listeners pull NDAC low again, and the sequence is repeated until the talker has sent all the data bytes it has to send.
  - The Data Transfer Rate depends on the rate at which the slowest listener can accept the data.
  - On completion of the data transfer the talker pulls the EOI line of the management group of signals low to indicate the transfer completion.
  - Finally, the controller takes control of all the data bus and sends Untalk and Unlisten commands to all the talkers and the listeners, and continues executing its pre-specified internal instructions.

Other Parallel Bus Standards

The following are some other popular Parallel bus standards. They have been designed mainly for a particular type of application, namely, within a processor mother-board to interface various peripherals.

1. ISA (IBM Standard Architecture) Bus. This was primarily designed for the IBMPC (8086 / 186 / 286 – Processor based) and uses a 16 bit data bus. It allows only up to 1024 port addresses. An extension EISA (Extended ISA) allows upto 32 bit data and addresses.

2. PCI (Peripheral Systems Interconnect), PCI /X and PCI Super Buses. This is an advanced version of the IBM-PC bus designed for the Pentium range of processors. It has 32/33 and 64/66 MHz versions (64/100 MHz in the PCI / X). A

current standard PCI Super allows upto 800 Mbps on a 64-bit bus. It supports automatic detection of devices via a 64- byte configuration register which makes it easy to interface ‘plug-and-play’ devices in a system.

3. IEEE-796 (Multi bus): Originally introduced by Intel as a means of connecting multiple processors on the system board, this bus is no longer very popular. It works with 16 bit data & 24 bit address buses.
4. VME Bus: (Euro-standard) Introduced for the same purpose as Intel Multibus it works with 24 bit address 8/16/32 bit data buses.
5. SCSI Bus (Small Computer System Interface): This standard was originally designed for use with Apple Mcintosh computers and then popularized by the Workstation Vendors. The main purpose is to interface peripherals like harddisks, CD-ROM Drives and similar relatively slow peripheral which use a data rate less than 100Mbps. The following varieties of SCSI are currently implemented:

- **SCSI-1:** Uses an 8-bit bus, and supports data rates of 4 Mbps
- **SCSI-2:** Same as SCSI-1, but uses a 50-pin connector instead of a 25-pin connector, and supports multiple devices. This is what most people mean when they refer to plain *SCSI*.
- **Wide SCSI:** Uses a wider cable (168 cable lines to 68 pins) to support 16-bit transfers.
- **Fast SCSI:** Uses an 8-bit bus, but doubles the clock rate to support data rates of 10 Mbps.
- **Fast Wide SCSI:** Uses a 16-bit bus and supports data rates of 20 Mbps.
- **Ultra SCSI:** Uses an 8-bit bus, and supports data rates of 20 Mbps.
- **SCSI-3:** Uses a 16-bit bus and supports data rates of 40 Mbps. Also called *Ultra Wide SCSI.*
- **Ultra2 SCSI:** Uses an 8-bit bus and supports data rates of 40 Mbps.
- **Wide Ultra2 SCSI:** Uses a 16-bit bus and supports data rates of 80 Mbps.

However, for the kind of applications targeted by GPIB, it is now facing a very strong competition from the recently introduced high speed serial bus standards. Currently there are four major candidates for future bus systems in Test & Measurement:

- The Universal Serial bus (USB) is now very popular. The current implementation provides transfer rates of up to 12MBit/s. From that viewpoint, there is no speed enhancement in comparison to GPIB; in fact, it is a drawback.
- USB II is an enhanced USB bus capable of transferring up to 480MBit/s. It is backwards compatible to USB. The IEC SC65C Working group 3 (that developed also the IEC625.1 and IEC625.2 standards) is planning to work on this.

- IEEE1394 (Fire Wire) is now available with transfer rates up to 400MBit/s. A specification to simulate GPIB was developed by a working group inside the IEEE1394 Trade Association. It is called IICP (Industrial and Instrumentation Control Protocol).
- Ethernet and related networks using TCP/IP protocol. Transfer rates up to 1GBit/s are possible. For simulating GPIB, a specification called VXI-11, introduced by the VXI plug play alliance, exists.