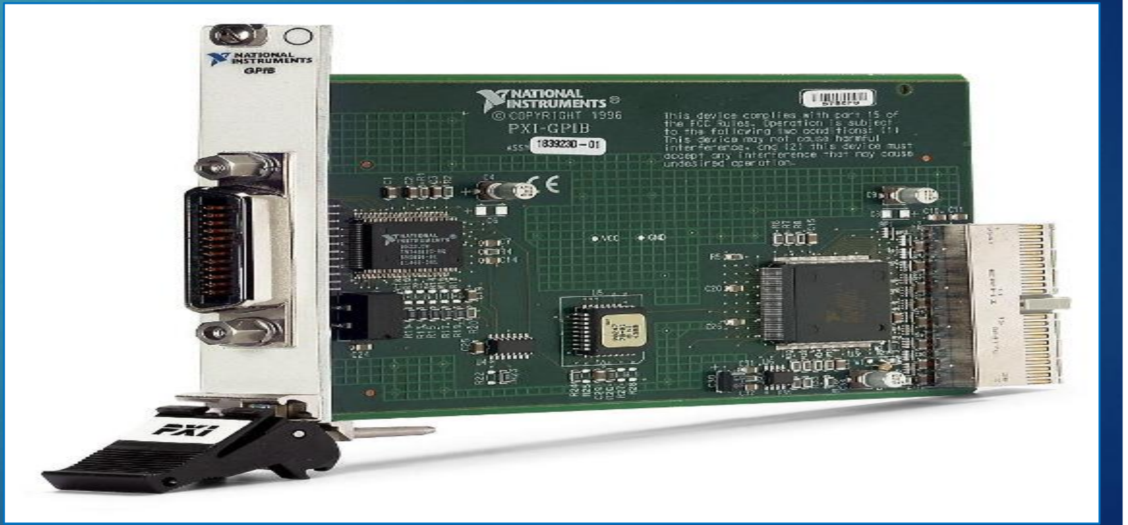
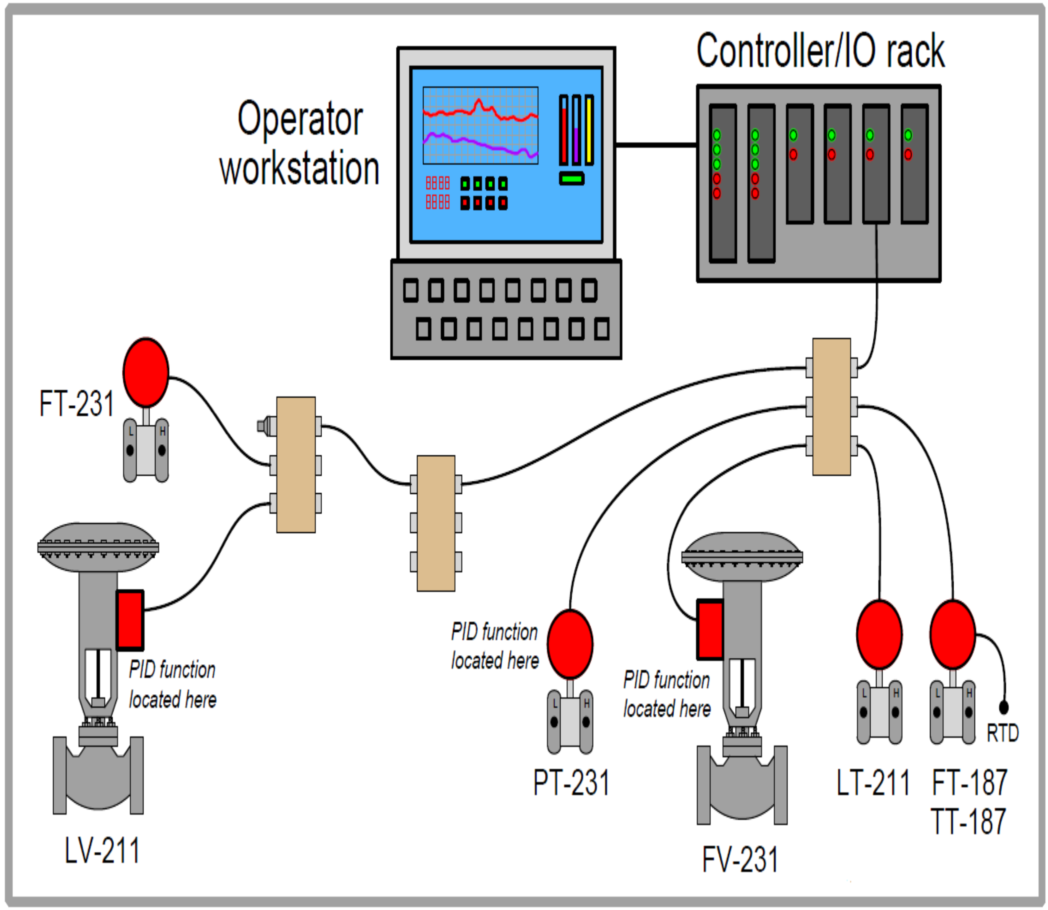
Instrumentation Bus
IEEE-488 & HART Protocol



Introduction

- In computer nomenclature, a *bus* is a collection of unbroken signal lines that interconnect computer modules (the connections are made by taps on the lines).
- In general, a bus contains data lines, control and signaling lines, and, in some cases, address lines.
- There are many *parallel bus architectures* and protocols that have been developed since the advent of the microcomputer.
- These include, but are not limited to, the Intel Multibus (IEEE-796 bus), the VMEbus, the Centronics printer bus, the DEC Unibus, the LSI-11 bus, the IBM PC bus (IBM PC/XT/AT), the Macintosh II NuBus, the MicroVAX Q-bus, the IBM PS2 Micro Channel Architecture (MCA) bus, the Intel iSBX bus, the ISA and EISA PC buses, and, of course, the IEEE-488 Instrumentation Bus (GPIB).



IEEE-488 GPIB (General Purpose Interface Bus)

- The IEEE-488 bus was developed by HP in the early 1970s as a standard, 8-bit, bidirectional, asynchronous bus (HP-IB) to enable a number of HP-IB-compatible instruments to communicate with a controlling computer and with each other.
- Not only can measured data be sent to the host computer for storage and processing, but in certain GPIB-compatible instruments, the computer can be used to set the instrument's front panel controls and to control the measurements.
- The original IEEE-488 standard was defined by the IEEE Standards Committee in 1975, and it was revised in 1978 and again in 1987 as IEEE Std. 488.2-1987.
- The IEEE 488.2 standard is also called the *Standard Commands for Programmable Instruments* (SCPI)

Key specifications

- This allows 15 devices to be shared over a single physical bus.
- The maximum data rate is about 1 MB/sec .
- Communication is digital & messages are sent one byte (8 bits) at a time.
- Manage transactions are hardware handshake.
- Total bus length may be up to 20m & the distance between devices may be up to 2m.
- The IEEE 488 has a 24-pin connector and is used for double headed design.
- The IEEE 488 has 16 signal lines.
- Eight lines are dedicated for bi-directional communication, five lines are used for bus management.
- The remaining three lines are dedicated for handshakes.

IEEE – 488 GPIB - General Purpose Interface Bus

Devices exist on the bus in any one of 3 general forms:

1. Controller
2. Talker
3. Listener

- A single device may incorporate all three options, although only one option may be active at a time.
- The Controller makes the determination as to which device becomes active on the bus.
- The GPIB can handle only 1 'active' controller at a time on the bus, although it may pass operation to another controller.
- Any number of active listeners can exist on the bus with an active talker as long as no more than 15 devices are connected to the bus.

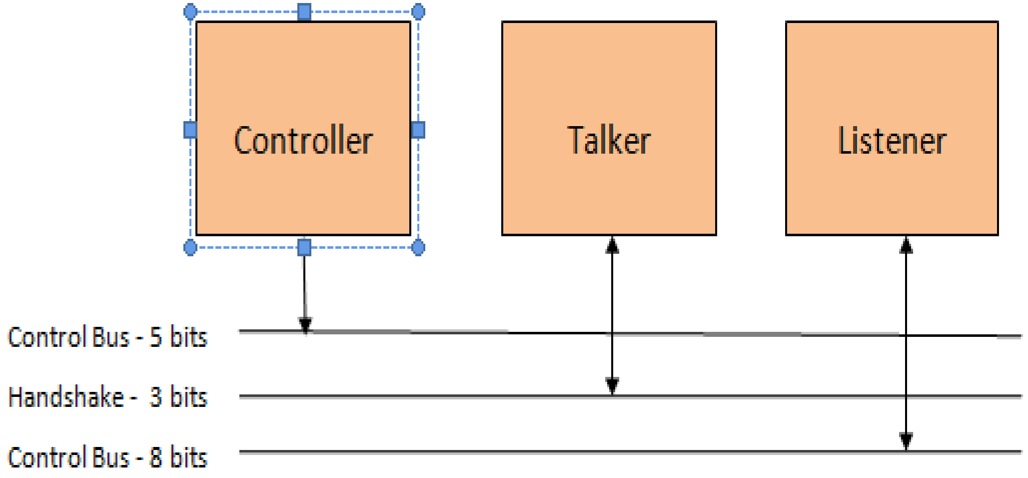
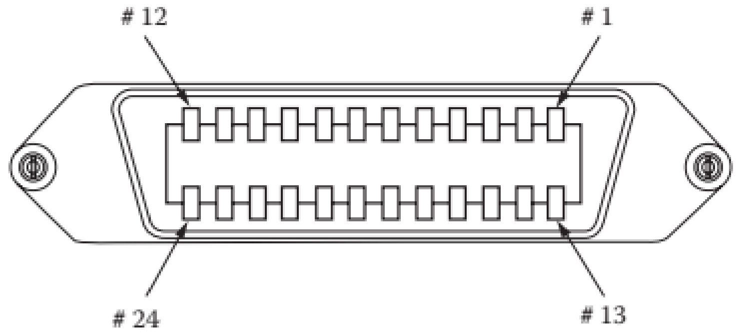


Figure: Architecture of GPIB

GPIB Bus Structure

- The GPIB consists of eight, tristate, bidirectional data lines and eight control lines that select, deselect, and otherwise coordinate the asynchronous communications between the host computer and satellite instruments.



Pin signal	Pin signal
1. Data 1	13. Data 5
2. Data 2	14. Data 6
3. Data 3	15. Data 7
4. Data 4	16. Data 8
5. EOI	17. REN
6. DAV	18. Gnd
7. NRFO	19. Gnd
8. NDAC	20. Gnd
9. IFC	21. Gnd
10. SRQ	22. Gnd
11. ATN	23. Gnd
12. Shield	24. Logic ground

Figure : Pinout of GPIB

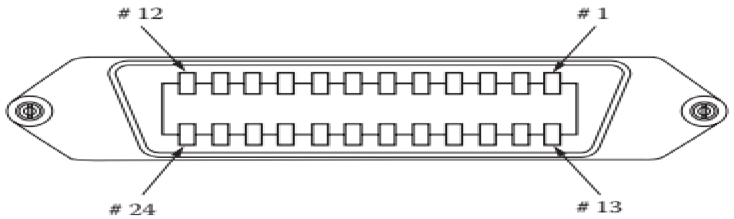
Control lines

- The eight control lines subdivided into *handshake lines* and *interface management lines Handshake lines* .
- *DAV* (data valid): When a selected device on the GPIB supplies an 8-bit word to the data lines (i.e., is a talker), DAV is set LOW (or TRUE) to indicate to all on the bus that the data byte on the bus is ready to be read to a listener device.
- DAV is one of the three *handshaking* lines that control data transmission on the GPIB.
- *NRFD* (not ready for data): This second handshaking line is pulled LOW by a selected listener to indicate it is ready to accept data (NRFD would be better called RFD).
- *NDAC* (not data accepted): This third handshaking line is set LOW by the selected listener device when the data byte transmitted has been accepted and that new data may now be supplied

Interface management lines

- *ATN* (attention): This line is pulled LOW by the bus controller (computer) to signal that it is sending a command. It drives ATN HI to signal that a talker can send it data messages.
- *IFC* (interface clear): The GPIB controller drives this line LOW to reset the status of all other devices on the GPIB and to become controller in charge (CIC).
- *SRQ* (service request): This is the GPIB equivalent of an interrupt line. Any device on the bus (except of course the controller) can pull the SRQ line LOW, signaling to the controller that it requires *service*.
- Devices can be programmed to signal SRQ for various reasons, such as being overloaded and out of paper and having completed an internal task such as computing an FFT spectrum.
- The controller, upon sensing SRQ LOW, must poll the devices on the GPIB to determine which one sent the SRQ and then take appropriate control action (service).
- *REN* (remote enable): This controller output line is set LOW to allow the controller to take over front-panel controls of a selected instrument on the GPIB (if that instrument has the capability of having its front-panel settings taken over by the controller).
- *EOI* (end or identify): This line has two functions—a talker uses EOI LOW to mark the end of a multibyte data message. EOI LOW is also used by the bus controller to initiate a parallel poll.
- When the controller sets both EOI and ATN LOW, a parallel poll causing devices configured for a parallel poll to present status bits on the data bus, which are read by the controller.

GPIB/IEEE-488 Protocol



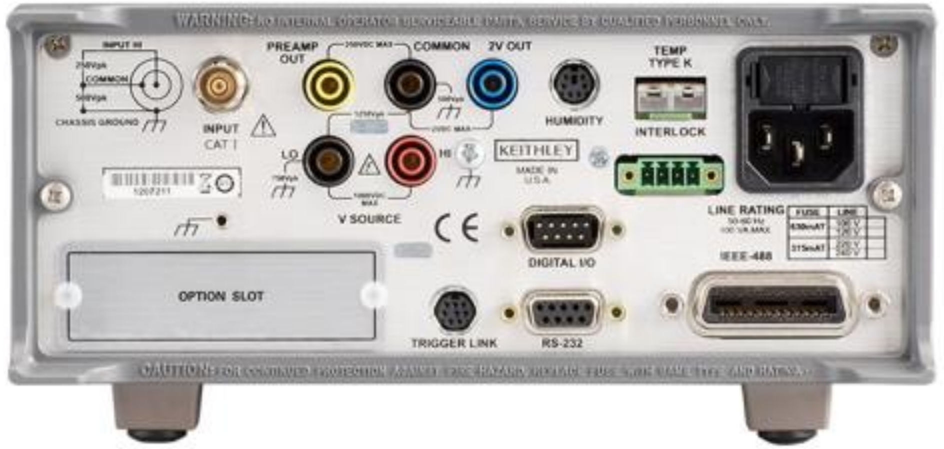
- GPIB is based around the handshaking protocol.
- Three lines
  - DAV (Data valid)
  - NDAC (Not data accepted)
  - NRFD (Not ready for data) control this the 8-bit data bus can carry data or commands.
- The controller pulls the ATN line LOW, to signal it is sending commands. There are *four types of command: addressed, listen, talk, and universal*.
- Commands are generally represented by 7-bit ASCII characters on the bus. The type of command being sent is signaled by bits 5, 6, and 7 on the data bus.
- If bits 5, 6, and 7 are HI voltages (logic 0), then the command is an addressed command.
- If bit 5 is LOW and 6 and 7 are HI, then the command is universal.
- If bit 6 is LOW and 7 is HI, then the command is a listen command, and if bit 6 is HI and 7 is LOW, then the command is a talk command.
- If both bits 6 and 7 are LOW, the command byte is a *secondary command*.
- Secondary commands are used for sending secondary addresses or setting up devices for a parallel poll

Advantages

- Well established & mature, widely supported.
- Rugged connectors held in place by screws, means cables can't easily be accidentally removed as they can with five wire & USB.
- Rugged cables (usually).
- In some locations large heavily protected cables are an advantages.
- Simple hardware interface.
- Ease of connecting multiple devices to a single host.
- Allows mixing of slow & fast devices.

Disadvantages

- High cost.
- Limited availability.
- Mechanically bulky connectors & cables
- Lack of command protocol standards
- Non mandatory galvanic isolation between bus & devices.
- Many manufacturers of interface cards for PCs and MACs offer IEEE-488.2-compatible cards and controlling software.
- Some of the newer GPIB systems have hardware and software that permit one to make a very fast and flexible measurement control system.



HART (Highway Addressable Remote Transducer)

- As intelligent devices developed over the years, the need arose for network protocols that could provide for the necessary digital communications to and from such devices.
- HART (**H**ighway **A**ddressable **R**emote **T**ransducer) is a well-known bus-based networking protocol that satisfies this need.
- Over the years, this has gained widespread international use, and has now become a *de facto* standard, with HART-compatible devices being available from all major instrument manufacturers.

Modes of HART

- In *hybrid mode*, status/command signals are digital but data transmission takes place in analogue form (usually in 4–20 mA format).
- One serious limitation of this mode is that it is not possible to transmit multiple measurement signals on a single bus, since the analogue signals would corrupt each other.
- Hence, when HART is used in hybrid mode, the network must be arranged in a star configuration, using a separate line for each field device rather than a common bus .
- In *fully digital mode*, data transmission is digital as well as status/command signals. This enables one cable to carry signals for up to 15 intelligent devices.
- Therefore, the main application of the HART protocol has been to provide a communication capability with intelligent devices when existing analogue measurement signal transmission has to be retained because conversion to fully digital operation would be too expensive.

The Protocol

- The communication between devices using the HART protocol is of the Master-Slave type, in which case the slave device transmits when the master requests, in general the field devices are slaves.
- In this case the protocol allows the insertion of up to two masters, where one is in standby, so if there is any problem in the communication of the master with the field, the other assumes the communication to immediately.
- Typically, the following situation arises:
- The master sends a command and waits for a response
- A slave waits for a command and sends a reply
- A slave usually has a unique address, this address is entered in the message by the master. The address can be from 4 bits up to 38 bits.
- Slaves can also be addressed through tags (identifier associated by the user)
- There are three types of commands:
- Universal
- Common
- Owner

Advantages of HART Protocol

- Improved plant operations
- Operational flexibility
- Instrumentation investment protection
- Digital communication

References

- *Mathivanan, N., Microprocessor PC Hardware and Interfacing, Prentice Hall of India Private Limited (2007).*
- *Anand, M.M.S., Electronic Instruments and Instrumentation Technology, Prentice Hall of India Private Limited (2004).*
- *Murthy, D.V.S., Transducers and Instrumentation, Prentice Hall of India Private Limited (2006).*