CO326: Industrial Networks
Lab 07 - General Purpose Interface Bus (GPIB)

Introduction

IEEE 488 is a short-range digital communications 8-bit parallel multi-master interface bus specification developed by Hewlett-Packard as **HP-IB** (**Hewlett-Packard Interface Bus**). It subsequently became the subject of several standards and is generically known as **GPIB** (**General Purpose Interface Bus**). It can be used as an interface between the computers and measuring instruments. GPIB is used to connect PCs with the measuring instruments. IEEE 488 and IEEE 488.2 are the two standards related to GPIB.

The benefits of using the GPIB can be listed as,

- GPIB makes the connecting and the configuring of devices easy. Many devices can be connected to a single PC.
- It ensures reliable data transfer.
- Devices with different speeds can be connected using GPIB.

GPIB card and the interface

GPIB cards can be PCI, PCI-e, or USB to connect to the PCs. It uses the GPIB port interface to connect the devices to the GPIB card.

Figure 1: GPIB Card

Figure 2: GPIB Interface

Cabling of GPIB

GPIB devices can be connected to form two different topologies as Bus topology or Star Topology

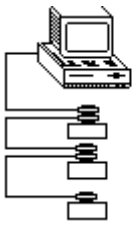

Figure 1: BUS Topology

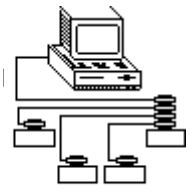

Figure 2: STAR Topology

GPIB Programming

GPIB Programming can be done in many ways,

- MATLAB
- C
- LabVIEW
- Measurement and Automation Explorer from NI - used in this lab

SCPI - Standard Commands for Programmable Instruments

SCPI (Standard Commands for Programmable Instruments) is a universal programming language for electronic test and measurement instruments, based on the IEEE 488 and IEEE 488.2 standards. It is an ASCII-based Instrument Command Language.

For IEEE 488.2 there are two types of command sets,

- Standard/Common command set
  - Common SCPI commands, defined by IEEE488.2, control and manage generic system functions such as reset, self-test, configuration storage, and device identification
  - They start with an asterisk mark (*)
  - Length is only 3 characters
  - Can have zero or more parameters

    E.g: *CLS , *ESE , *OPC?

- Device-specific Subsystem Commands
  - SCPI subsystem commands are either measurement-related or other device-specific commands

Lab/Demonstration

Required Devices and Tools

- Personal Computer with a GPIB card connected to it
- GPIB supported device - Agilent 33600A Series Signal Generator
- GPIB Cables to connect devices to the card
- Measurement and Automation Explorer tool

Commands Used for the Demonstration

The following example commands are issued for the Signal Generator connected via GPIB

1. **Identification query**

   ***IDN?**

   ➢ Outputs the device details in 4 comma-separated fields
   - Manufacture Name
   - Model Number
   - Serial Number
   - Revision Code

2. **Clear status**

   ***CLS**

   ➢ Clearing the Error Queue

3. **Check system date**

   SYSTem:DATE? or **SYST:DATE?**

   ➢ Outputs the system date of the device

4. **Change the properties of a waveform**

   SOURce:FUNCtion SINusoid or **SOUR:FUNC SIN**

   ➢ Changes the type of the waveform generated

   ➢ Types of waveforms can be,
   - SIN, RAMP, TRI, NOIS, PRBS, ARB

   SOURce:FREQuency 1 KHz or **SOUR:FREQ 1 KHz**

   ➢ Sets the frequency to the specified frequency

SOURce:VOLTage 200 mV or **SOUR:VOLT 200 mV**

➢ Sets the amplitude of the waveform to the specified value

SOURce:VOLTage:OFFSet 100 mV or **SOUR:VOLT:OFFS 100 mV**

➢ Sets the DC offset of the waveform to the specified value

SOURce:PHASe +90.0 or **SOUR:PHAS +90.0**

➢ Sets the phase of the waveform to the specified value

Note: You can query the above commands with a question mark at the end to obtain the current setting of the respective parameter (e.g. **SOURce:FREQuency?**)

5. Turn on/off output from the Signal Generator

OUTPut ON or OUTP ON

➢ Turns on the output

OUTPut OFF or OUTP OFF

➢ Turns off the output

Practise Exercises

1. Produce a sine wave with 1.5KHz frequency, 5V amplitude, and a +90 degrees phase shift.
2. Produce a ramp wave with a 2KHz frequency, 3V amplitude, and a 1V DC offset
3. Write and issue commands to the Signal Generator so that it produces the following signal