Document Type: Tutorial
NI Supported: Yes
Publish Date: Apr 30, 2008

GPIB Hardware and Software Specifications

Overview

This document describes the three General-Purpose Interface Bus (GPIB) specifications: IEEE 488.1, IEEE 488.2, and SCPI.

Table of Contents

The IEEE 488.1 Specification

The GPIB is a digital 8-bit parallel communications interface with data transfer rates up to 1 Mbyte/s. The bus supports one System Controller, usually a computer, and up to 15 additional instruments. Because the GPIB is an 8-bit parallel interface with fast data transfer rates, it gained popularity in other applications such as intercomputer communication and peripheral control.

The IEEE 488.2 Specification

IEEE Standard 488.2-1987 encouraged a new level of growth and acceptance of the IEEE 488 bus or GPIB by addressing problems that had arisen from the original IEEE 488 standard. IEEE 488.2 was drafted on the premise that it stay compatible with the existing IEEE 488.1 standard. The overriding concept used in the IEEE 488.2 specification for the communication between Controllers and instruments is that of "precise talking" and "forgiving listening." In other words, IEEE 488.2 exactly defined how both IEEE 488.2 Controllers and IEEE 488.2 instruments talk so that a completely IEEE 488.2-compatible system can be highly reliable and efficient. The standard also required that IEEE 488.2 devices be able to work with existing IEEE 488.1 devices by accepting a wide range of commands and data formats as a Listener. You obtain the true benefits of IEEE 488.2 when you have a completely IEEE 488.2-compatible system.

Controllers

Although IEEE 488.2 had less impact on Controllers than it did on instruments, there are several requirements and optional improvements for Controllers that made an IEEE 488.2 Controller a necessary component of test systems. IEEE 488.2 precisely defined the way IEEE 488.2 Controllers send commands and data and added functionality. Because of these IEEE 488.2 Controller requirements, instrument manufacturers can design more compatible and efficient instruments. The benefits of this standardization for the test system developer are reduced development time and cost, because it solves the problems caused by instrument incompatibilities, varying command structures, and data formats.

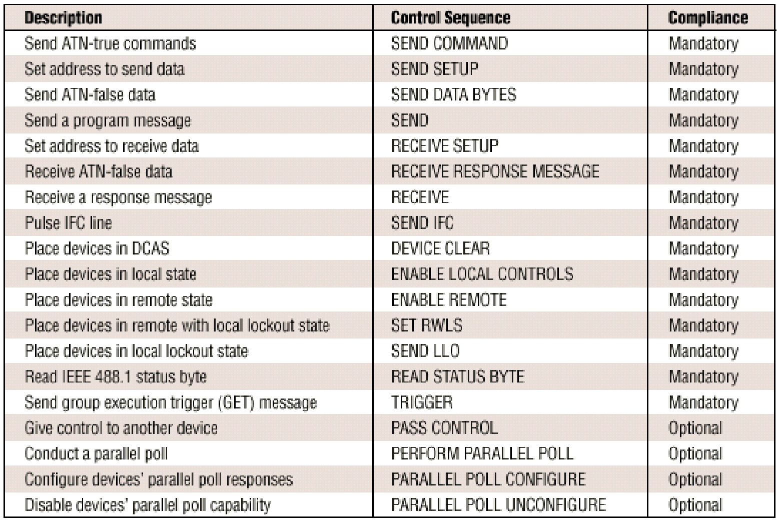
Requirements of IEEE 488.2 Controllers

IEEE 488.2 defined a number of requirements for a Controller, including an exact set of IEEE 488.1 interface capabilities, such as pulsing the interface clear line for 100 µs, setting and detecting EOI, setting/asserting the REN line, sensing the state and transition of the SRQ line, sensing the state of NDAC, and timing out on any I/O transaction. Other key requirements for Controllers are bus control sequences and bus protocols.

IEEE 488.2 Control Sequences

The IEEE 488.2 standard defined control sequences that specify the exact IEEE 488.1 messages that are sent from the Controller as well as the ordering of multiple messages. IEEE 488.2 defined 15 required control sequences and four optional control sequences, as shown in Table 1. The IEEE 488.2 control sequences describe the exact states of the GPIB and the ordering of command messages for each defined operation. IEEE 488.2 control sequences remove the ambiguity of the possible bus conditions, so instruments and Controllers are much more compatible. By exactly defining the state of the bus and how devices should respond to specific messages, IEEE 488.2 solves such system development problems.

Table 1. IEEE 488.2 Required and Optional Control Sequences



Description	Control Sequence	Compliance
Send ATN-true commands	SEND COMMAND	Mandatory
Set address to send data	SEND SETUP	Mandatory
Send ATN-false data	SEND DATA BYTES	Mandatory
Send a program message	SEND	Mandatory
Set address to receive data	RECEIVE SETUP	Mandatory
Receive ATN-false data	RECEIVE RESPONSE MESSAGE	Mandatory
Receive a response message	RECEIVE	Mandatory
Pulse IFC line	SEND IFC	Mandatory
Place devices in DCAS	DEVICE CLEAR	Mandatory
Place devices in local state	ENABLE LOCAL CONTROLS	Mandatory
Place devices in remote state	ENABLE REMOTE	Mandatory
Place devices in remote with local lockout state	SET RWLS	Mandatory
Place devices in local lockout state	SEND LLO	Mandatory
Read IEEE 488.1 status byte	READ STATUS BYTE	Mandatory
Send group execution trigger (GET) message	TRIGGER	Mandatory
Give control to another device	PASS CONTROL	Optional
Conduct a parallel poll	PERFORM PARALLEL POLL	Optional
Configure devices' parallel poll responses	PARALLEL POLL CONFIGURE	Optional
Disable devices' parallel poll capability	PARALLEL POLL UNCONFIGURE	Optional

IEEE 488.2 Protocols

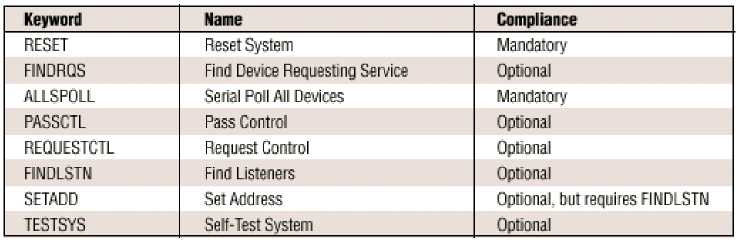
Protocols are high-level routines that combine a number of control sequences to perform common test system operations. IEEE 488.2 defines two required protocols and six optional protocols, as shown in Table 2. These protocols reduce development time because they combine several commands to execute the most common operations required by any test system. The RESET protocol ensures that the GPIB has been initialized and all devices have been cleared and set to a known state. The ALLSPOLL protocol serial polls each device and returns the status byte of each device.

The PASSCTL and REQUESTCTL protocols pass control of the bus between a number of different devices. The TESTSYS protocol instructs each device to run its own self-tests and report back to the Controller whether it has a problem or is ready for operation.

Perhaps the two most important protocols are FINDLSTN and FINDRQS. The FINDLSTN protocol takes advantage of the IEEE 488.2 Controller capability of monitoring bus lines to locate listening devices on the bus. The Controller implements the FINDLSTN protocol by issuing a particular listen address and then monitoring the NDAC handshake line to determine if a device exists at that address. The result of the FINDLSTN protocol is a list of addresses for all the locate devices. FINDLSTN is used at the start of an application program to ensure proper system configuration and to provide a valid list of GPIB devices that can be used as the input parameter to all other IEEE 488.2 protocols. The bus line monitoring capability of an IEEE 488.2 Controller is also useful to detect and diagnose problems within a test system.

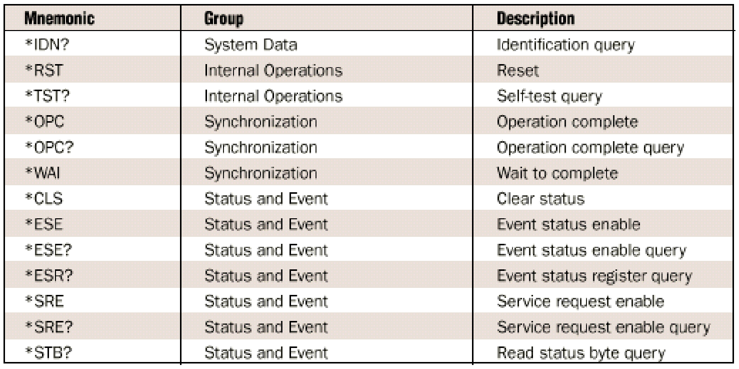
The FINDRQS protocol is an efficient mechanism for locating and polling devices that are requesting service. It uses the IEEE 488.2 Controller capability of sensing the FALSE to TRUE transition of the SRQ line. You prioritize the input list of devices so that the more critical devices receive service first. If the application program can jump to this protocol immediately upon the assertion of the SRQ line, you increase program efficiency and throughput.

Table 2. IEEE 488.2 Controller Protocols



Keyword	Name	Compliance
RESET	Reset System	Mandatory
FINDRQS	Find Device Requesting Service	Optional
ALLSPOLL	Serial Poll All Devices	Mandatory
PASSCTL	Pass Control	Optional
REQUESTCTL	Request Control	Optional
FINDLSTN	Find Listeners	Optional
SETADD	Set Address	Optional, but requires FINDLSTN
TESTSYS	Self-Test System	Optional

Table 3. IEEE 488.2 Mandatory Common Commands



Mnemonic	Group	Description
*IDN?	System Data	Identification query
*RST	Internal Operations	Reset
*TST?	Internal Operations	Self-test query
*OPC	Synchronization	Operation complete
*OPC?	Synchronization	Operation complete query
*WAI	Synchronization	Wait to complete
*CLS	Status and Event	Clear status
*ESE	Status and Event	Event status enable
*ESE?	Status and Event	Event status enable query
*ESR?	Status and Event	Event status register query
*SRE	Status and Event	Service request enable
*SRE?	Status and Event	Service request enable query
*STB?	Status and Event	Read status byte query

IEEE 488.2 Instruments

The IEEE 488.2 specifications for instruments can require major changes in the firmware and possibly the hardware. However, IEEE 488.2 instruments are easier to program because they respond to common commands and queries in a well defined manner using standard message exchange protocols and data formats. The IEEE 488.2 message exchange protocol is the foundation for the SCPI standard that makes test system programming even easier. IEEE 488.2 defines a minimum set of IEEE 488.1 interface capabilities that an instrument must have. All devices must be able to send and receive data, request service, and respond to a device clear message. IEEE 488.2 defines precisely the format of commands sent to instruments and the format and coding of responses sent by instruments. All instruments must perform certain operations to communicate on the bus and report status.

Because these operations are common to all instruments, IEEE 488.2 defined the programming commands used to execute these operations and the queries used to receive common status information. These common commands and queries are shown in Table 3. Because IEEE 488.2 standardizes status reporting, the Controller knows exactly how to obtain status information from every instrument in the system. This status reporting model builds on the IEEE 488.1 status byte to provide more detailed status information. The status reporting model is shown in Figure 1.

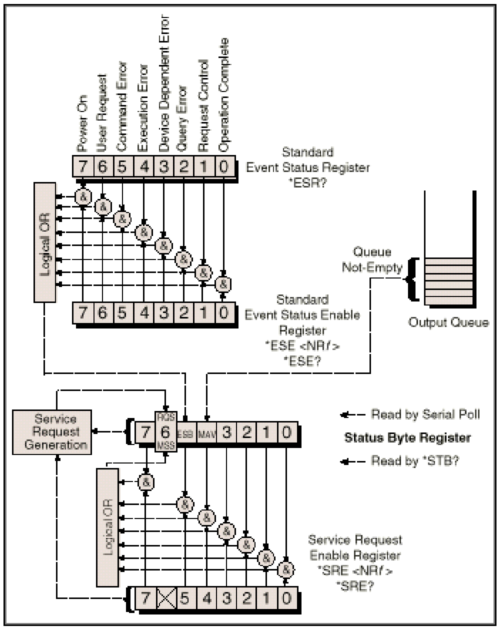


Figure 1. Status Reporting Model

The SCPI Specification

The SCPI specification expanded the IEEE 488.2 common command set by defining a single, comprehensive command set suitable for all instruments. For example, all SCPI-compatible voltmeters, regardless of manufacturer or model, respond to the same command for reading AC voltage. Their response format is also the same.

SCPI embraces many of the commands and protocols that the hardware-independent portion of the IEEE 488.2 standard defines. Figure 2 shows the structure of the GPIB standards.

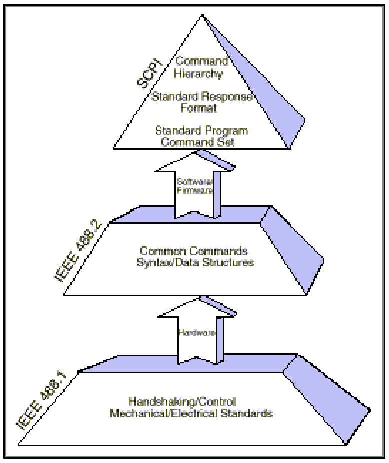

SCPI
Command Hierarchy
Standard Response Format
Standard Program Command Set
Software Firmware
IEEE 488.2
Common Commands Syntax/Data Structures
Hardware
IEEE 488.1
Handshaking/Control Mechanical/Electrical Standards

Figure 2. Structure of the GPIB Standards

Combining IEEE 488.2 and SCPI leads to greater productivity, because of software command standards and instant interchangeability. Rather than learning a different command set for each instrument, you focus on solving measurement problems.

Although you can mix SCPI and non-SCPI instruments in a system, your complete system must adhere to IEEE 488.2 for you to benefit from the standards.

Related Links:

NI GPIB Home Page