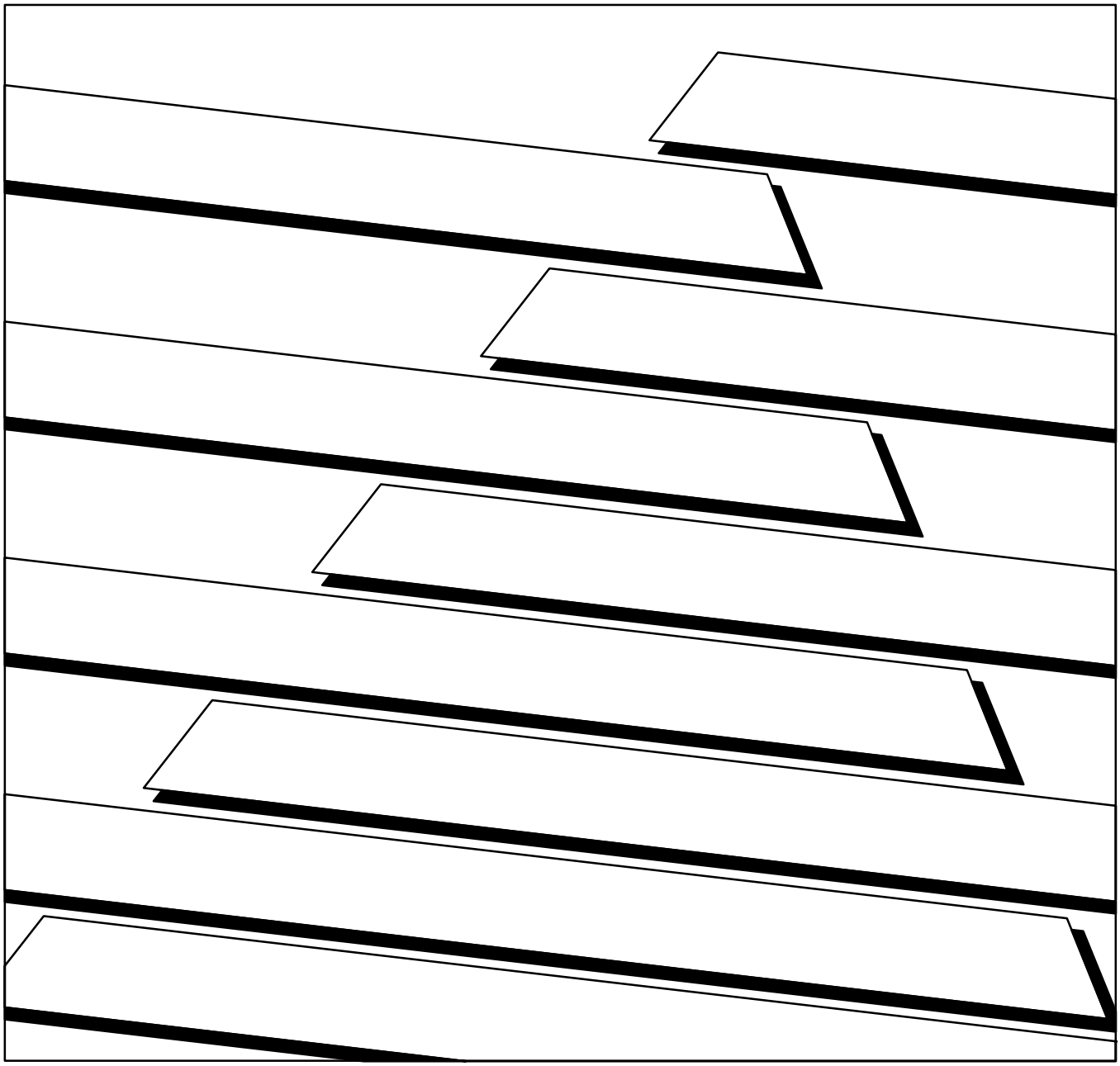
ALLEN-BRADLEY

# SPI Protocol Interface Module
# Cat. No. 1771-SPI

Concepts Manual



## Important User Information

Because of the variety of uses for the products described in this publication, those responsible for the application and use of this control equipment must satisfy themselves that all necessary steps have been taken to assure that each application and use meets all performance and safety requirements, including any applicable laws, regulations, codes, and standards.

The illustrations, charts, sample programs, and layout examples shown in this guide are intended solely for purposes of example. Since there are many variables and requirements associated with any particular installation, Allen-Bradley does not assume responsibility or liability (to include intellectual property liability) for actual use based upon the examples shown in this publication.

Allen-Bradley publication SGI-1.1, Safety Guidelines For The Application, Installation and Maintenance of Solid State Control (available at your local Allen-Bradley office) describes some important differences between solid-state equipment and electromechanical devices which should be taken into consideration when applying products such as those described in this publication.



Throughout this manual we make notes to alert you to possible injury to people or damage to equipment under specific circumstances.

**ATTENTION:** Identifies information about practices or circumstances that can lead to personal injury or death, property damage or economic loss.

Attention helps you:

- Identify a hazard
- Avoid the hazard
- Recognize the consequences

**Important:** Identifies information that is critical for successful application and understanding of the product.

# Table of Contents

# Using This Manual

## Purpose of This Manual

This concepts manual helps you apply the SPI module to your application. It serves as a supplement to the SPI user manual (publication 1771-6.5.97). The purposes of this concepts manual are to help you:

- understand SPI terminology and concepts of SPI module operation
- understand relationships among the various “custom” data blocks for
  – module-specific configuration
  – device-specific configuration
  – the transfer of data to and from the device via the SPI module

- apply the custom-configured scheme of SPI communication
- program the correct protocol for block transfers to / from the SPI module
- map the data table for the various command and status blocks
- wire the primary and secondary machine connections to the SPI network
- troubleshoot with diagnostics that indicate the source of the fault

To help achieve these objectives, we provide example diagrams and worksheets.

## Who Should Use This Manual

This manual assumes that you are a first-time user of the SPI module, but that you are experienced with the following:

- programming a PLC-5 processor with 6200 Series Software
- programming the block transfer of multiple data blocks
- creating operational interface software for a PLC-5 processor
- using the hexadecimal numbering system

## Minimum Hardware and Software Requirements

To set up a minimal SPI network, you should have at least the following:

- 1771-SPI module
- 1771 I/O chassis and power supply
- PLC-5 processor
- SPI-specified device
- personal computer or programming terminal
- 6200 Series Software for programming a PLC-5 processor

Chapter 1

# Overview of an SPI Communication Network

## What This Chapter Contains

This overview chapter contains:

- purpose of the SPI network
- types of data communicated on the SPI network
- role of the SPI module
- role of the host PLC-5 processor
- what you program
- how the SPI module polls its network devices

## Purpose of the SPI Network

The protocol developed by the Society of Plastics Industry provides a standard for transferring command and status data between a master interface (SPI module) and slave devices on a network. This provides for compatible network communication among devices made by various manufacturers.

## Types of Data Communicated on the SPI Network

Typically, a master interface can use SPI protocol to communicate with slave devices on the network to:

- send them command data such as
  - bit-level commands
  - setpoints
  - high/low deviation alarm limits
- receive from them status data such as:
  - bit-level indicators of detected alarm conditions
  - current operating parameters (process actuals)

## Role of the SPI Module

The SPI module is the network master that coordinates the flow of data between the PLC-5 host processor and slave devices on the network. The SPI module:

- stores configuration information corresponding to each device
- returns configuration status to the host PLC-5 processor
- receives command data destined for devices on the network
- temporarily stores the data while it polls the target device
- sends data serially to and receives status serially from the target device
- repeats the data transfer for each device in polled order
- returns status data from network devices to the host PLC-5 processor

## Role of the Host PLC-5 Processor

The host processor stores information for transfer to and received from devices on the SPI network via the SPI module. The processor executes commands written in the instructions of your program.

You program the host processor to communicate with the SPI module. Your programming should:

- prepare command data that contain setpoints and control bits destined for each device on the SPI network, and use status data received from each device to report operating conditions.
- move command and status data (for each device) between data table files and block-transfer buffers
- transfer command data to the SPI module via block transfer write (BTW) instructions, and transfer status data from the SPI module via block transfer read (BTR) instructions
- prepare configuration for the SPI module and for each device on the SPI network. Configuration defines such items as:
  - ID number and address of each device on the SPI network
  - polling order to communicate with devices on the SPI network
  - number of command words transferred by BTW instructions
  - number of status words transferred by BTR instructions

## Files That You Map in the Data Table

When you program the processor to communicate with devices on the SPI network via the SPI module, you must first map the data table for the following data files (defined in chapter 3):

| Name of File | Quantity | Purpose |
|---|---|---|
| Block-transfer Buffers | 1 BTR | buffer data coming from the SPI module |
| | 1 BTW | buffer data going to the SPI module |
| Module Configuration Command (MCC) | 1 MCC | send module-specific configuration to the SPI module |
| System Status (SYS) | 1 SYS | receive module and network status from the SPI module |
| Custom Configuration Block (CCB) | 1 CCB for each device type | send device-specific configuration to the SPI module |
| Custom Data Block (CDB) | 1 CDB for each device type | send commands to the SPI module destined for the device |
| Custom Configuration Status (CCS) | 1 CCS for each device type | receive device-configuration status from the SPI module |
| Custom Data Status (CDS) | 1 CDS for each device type | receive data from the device via the SPI module |

Then, you must write ladder logic that writes commands to and reads status from the SPI module.

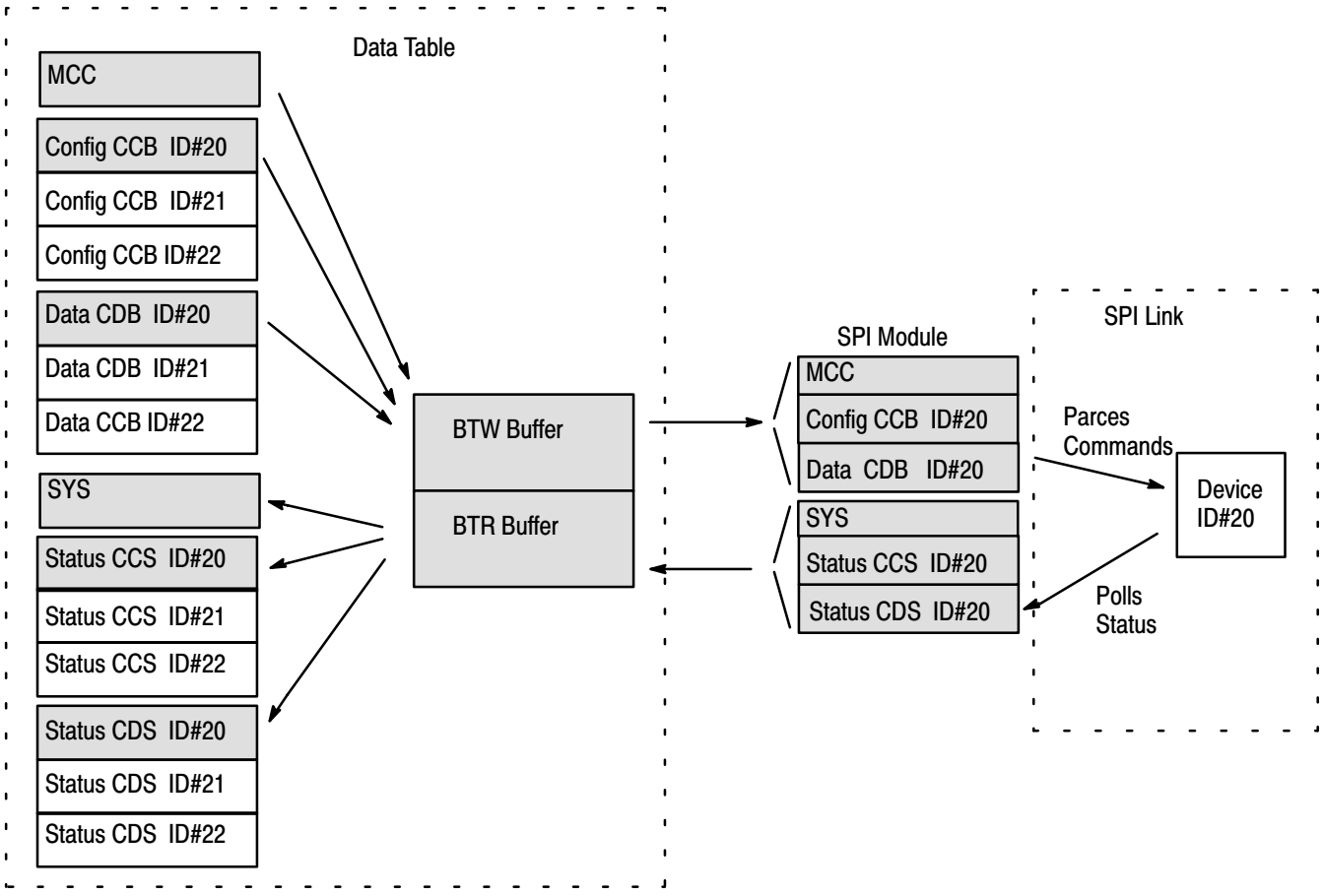
## Overview of the Data Transfer Scheme

When you program the processor to communicate with devices on the SPI network via the SPI module, you must follow a strict procedure of data transfers that:

- receives status from the SPI module
- configures the SPI module with module-specific configuration (MCC)
- configures the SPI module with device-specific configuration (CCB)
- receives configuration status from the SPI module (SYS and CCS)
- sends commands to the SPI module destined for target devices (CDB)
- receives status from target devices via the SPI module (CDS)

We show the general flow of configuration and command data transferred to the SPI module, and status received from it. Once configured, the SPI module serves as the master interface with slave devices on the link: the module parces commands to target devices and polls status from them.

**Transferring Configuration and Command Data to One of Three Devices on an SPI Network**



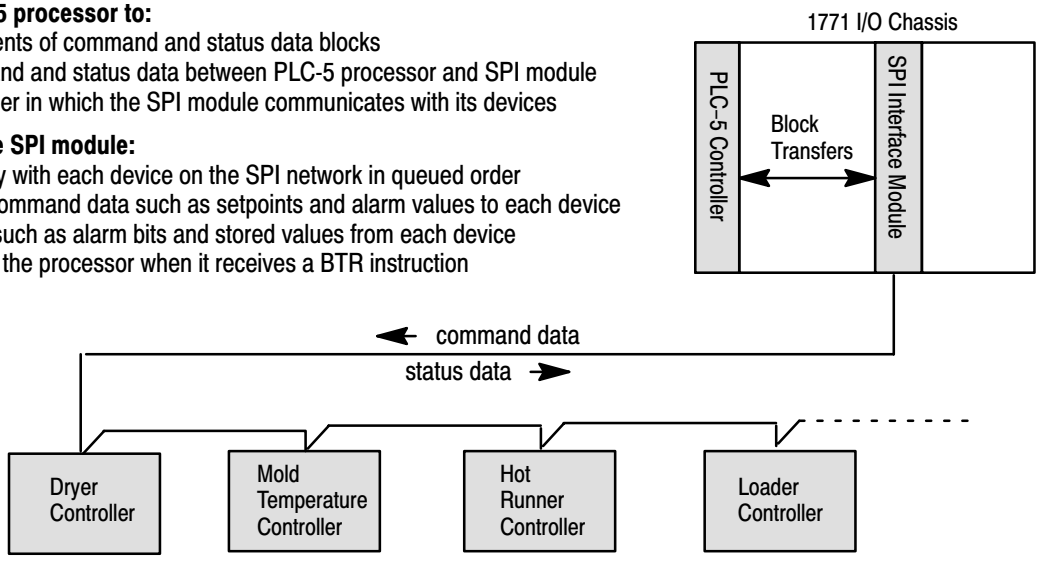
You must program block transfer instructions according to an exact protocol.

### Programming the Transfer of Configuration, Command, and Status Data

| Programming That You Develop | Data Blocks Transferred | Automatic Response of SPI Module |
|---|---|---|
| Until your ladder logic tranafers a valid MCC to the SPI module, it can only return SYS status in response to a BTR request. | BTR SYS | The SPI module returns the current SYS status to the processor. |
| To set the SPI module into operation, send (BTW) a valid MCC to it.<br>Your ladder logic moves SYS from the BTR buffer into the data table. | BTW MCC<br>BTR SYS | After storing a valid MCC, the SPI module sets the power-up bit in SYS and returns SYS to the processor. Now it is ready for custom configuration. |
| To configure the SPI for a custom device type, move device-specific custom configuration (CCB) to the BTW buffer for transfer to the SPI module.<br>Your ladder logic moves the CCS into the data table where your program checks its validity. | BTW CCB<br>BTR CCS | The SPI module formats and stores the CCB. Then it returns the corresponding CCS to acknowledge receipt of the CCB. Store a CCB for each type of device. |
| Your ladder logic moves CDS into the data table for use in your application program. | BTR CDS | After storing CCBs, the SPI module begins automatic polling: it reads status from the first device and returns CDS to the processor. Then repeats for each device on the network in queued order as listed in the MCC. |
| To communicate with target custom devices, move device-specific data (CDBs) to the BTW buffer for transfer to the SPI module.<br>Your ladder logic moves the CDS into the data table for use in your application program. | BTW CDB<br>BTR CDS | The SPI module formats and stores CDBs. Then it interrupts automatic polling and sends data to target devices in queue order.<br>The SPI module returns status (CDS) to the processor each time it sends a CDB to a target device. |

**You program the PLC-5 processor to:**
* set the size and contents of command and status data blocks
* block transfer command and status data between PLC-5 processor and SPI module
* specify the queue order in which the SPI module communicates with its devices

**When commanded, the SPI module:**
* communicates serially with each device on the SPI network in queued order
* sends programmed command data such as setpoints and alarm values to each device
* receives status data such as alarm bits and stored values from each device
* returns status data to the processor when it receives a BTR instruction




**When polled in queued order, each device on the SPI network:**
* stores command bits, setpoints, and alarm values received from the SPI module
* returns alarm bits and requested status to the SPI module

10579–2

## How the SPI Module Communicates With Its Network Devices

You establish the order (polling queue) in which the SPI module communicates with devices on the network. You do this by listing device ID codes in the module configuration command (MCC) block. When entering this list, you may also choose to have:

- each device polled once
- one or more devices polled more frequently in the queue
- the SPI module return system status (SYS) if desired

The SPI module communicates with its devices under two circumstances:

- automatic polling
- in response to a CDB command

### Automatic Polling

Once the SPI module has successfully stored a custom configuration block (CCB) for each device on its network (and for each device listed in queued order in the MCC), the SPI module begins automatic polling to return device status to the PLC processor as follows:

**1.** receives (polls) status from the first device

**2.** formats the status into a CDS and returns it to the processor by BTR

**3.** if it detects an error,
- enters an error code in CDS word 3
- clears status words of their values

**4.** goes to the next device

**5.** returns SYS (only if you listed SYS in the polling sequence)

**6.** when it receives a CDB command during automatic polling, executes the command only after completing the current polling sequence

### In Response to CDB Commands

When the SPI module receives CDB commands addressed to one or more devices on the SPI network, it stores them until it completes the current automatic polling sequence. Then it:

**1.** sends the CDB to the first target device

**2.** if it detects no errors, returns the device's CDS to the processor

**3.** if it detects an error or that it cannot find the device,
- enters an error code in CDS word 3
- clears CDS status words of their values

**4.** sends the next CDB to the next target device and repeats steps 2 and 3

**5.** returns to automatic polling after servicing the last target device

## How Many Different Devices Can the SPI Network Handle?

The SPI module can store the configuration blocks of:

- up to 10 different types of devices (device types) such as mold temperature controller, hot runner, and/or dryer controller
- no more than one device (of a given device type) at a time (for example, only one mold temperature controller) unless they are identical with respect to:
  - configuration, data, and status blocks
  - command bits, such as for turning the device on or off
  - presets, such as setpoints and alarm limits
  - status bits and words

  Exception : If you must network to different devices of the same type (mold temperature controllers (MTCs) from different manufacturers), then your program may become very complex because it must:

  a. clear the previous configuration and data blocks (for the MTC from company A)

  b. replace them with a new set of configuration and data blocks (for the MTC from company B)

  c. communicate with the device from company B

  d. repeat the three steps for each different device of the same type (for the MTCs from companies C, D, etc.)

## Definition of Terms

The SPI specification and previous revisions of the SPI module user manual use terms that may be unfamiliar to you. We attempt to translate SPI terms to ones more commonly used on the shop floor so you can more easily understand concepts of SPI module operation.

### SPI Terminology

| SPI Term | How it works or what it means | Shop-floor Term |
|---|---|---|
| ASCII Polls | ASCII strings such as SPI Revision returned by BTR | ASCII status word (32-bit word or 2 PLC words) |
| ASCII Selects | ASCII strings that you send to the SPI module | ASCII string (32-bit word or 2 PLC-5 words) |
| Bit Polls | bit-level status words returned by BTR | bit-status words (16-bit words) |
| Bit Selects | bit-level command words that you send to the SPI module | bit-command words (16-bit words) |
| Device ID | SPI-specified code for an SPI-specified device | 8-bit code such as 00100010 (22) for a dryer |
| Device Type | SPI-specified device | mold temperature controller, dryer, hot runner, etc. |
| Numeric Polls | process actuals such as pressure or flow returned by BTR | numeric status words (32-bit word or 2 PLC-5 words) |
| Numeric Selects | setpoints or alarm limits that you send to the SPI module | numeric setpoints (32-bit word or 2 PLC-5 words) |
| Polls (noun) | status returned to the processor by BTR | Numeric status, or words containing status bits |
| Selects (noun) | commands transferred to the SPI module by BTW | setpoints, or words containing control bits |
| Station Address | address that you select for a device on the SPI network | device address |
| Version | latest update | revision level |
| Zone | applies to temperature-control devices such as hot runner: independently controlled area, slave to master controller such as the barrel, nozzle, or runner, on a molding machine | zone or temperature zone |
| Zone Addresses | slave addresses for a single temperature control device | zone address |

### Other Important Terms

| Term | Definition |
|---|---|
| Command | bit-level such as to turn ON a device, or word-level such as a setpoint or alarm |
| Decimal Data | the PLC-5's base-10 (integer) numbering system with a range of –32,768 to +32,767, stored in N files as 2's-complement binary in 16-bit words |
| Polling Sequence or Automatic Polling | The manner in which the SPI module pulls status from each device on the SPI link, formats it, and sends it to the PLC processor by BTR. The SPI module services the devices (completes one before going to the next) in the order that you listed them in the MCC polling sequence. |
| Status | bit-level such as the ON/OFF status of a device, or word-level such as a process temperature |

## Notes

Chapter 2

# Important Information that You Need to Set Up the SPI Network

## Objective

This chapter tells you what information you need from the device manufacturer, and how to wire an SPI network.

## Required Information from the Device Manufacturer

To use a device on the SPI network, you need to know the following information about the device from the device manufacturer:

- Device-type IDs
- Address code to identify the device on the SPI network
- How to set the address code in the device
- Communication rate
- Data Types and Data Codes for Transfer to/from Devices on the Network
- Information that You Need to configure the SPI module
- Information that you Send to / Receive From Devices Via the SPI Module

### Device-type IDs

Each device manufactured for use on the SPI network must have an SPI-specified device-type identification number (device ID). Currently, there are about 10 SPI-specified device ID codes. Examples of device types and their ID codes include:

| Type of Device (Device Type) | ID | ID Code |
|---|---|---|
| mold temperature controller | 20 | 00100000 |
| chiller | 21 | 00100001 |
| dryer | 22 | 00100010 |
| loader | 23 | 00100011 |
| melt pump | 24 | 00100100 |
| additive feeder | 25 | 00100101 |
| self-tuning temperature controller | 26 | 00100110 |
| general purpose temperature controller | 27 | 00100111 |
| volumetric blender | 28 | 00101000 |
| continuous-weigh blender | 29 | 00101001 |
| batch-weigh blender | 2A | 00101010 |
| robot | 2B | 00101011 |
| reserved for future devices | 2C | 00101100 to 11111110 |

**Important:** When using any of these devices with a custom configuration (as described in this manual) rather than a standard configuration (described in the SPI Protocol Interface Manual), you use the above ID codes, but you must set the Configuration Select (CS) bit (bit 14 in MCC word 5, 7, 9, etc) when you enter the device polling list in the MCC. (We describe the MCC and other data files in chapter 3.)

## Address Code to Identify the Device on the SPI Network

The SPI module uses the address code that you set on the device, in combination with the SPI-assigned device ID code, to locate the device on the SPI network. This lets the SPI module differentiate between two or more devices having the same device ID. However, we recommend that your program be structured to communicate with no more than one device of a given type at a time. The use of two or more devices of the same type may require special considerations described elsewhere in this manual.

The address code is an 8-bit code with a decimal range of 32-255.

## How to Set the Address Code in the Device

Most devices will typically use jumpers or a switch assembly to set the address code. More complex devices such as a multi-zone temperature controller may use a keypad to set the device address and zone addresses. You must get this information from the manufacturer if not included in the user manual that accompanied the device.

## Communication Rate

Typical communication rates vary from 1200 to 19200 baud. You select a communication rate that suits all devices on your SPI network. The device with the *lowest* maximum rate establishes the upper limit for the network. The device manufacturer must tell you the range of selectable rates, and how to select a given rate.

## Data Types and Data Codes for Transfer to and from Devices on the SPI Network

You need to know the number, type, codes, and purpose of data words that the device will respond to:

- how many bit-command words you can send to the device
- the SPI-assigned code (such as 20 49) of each bit-command word
- the purpose of each bit-command word such as
  – used to turn the device on or off

- how many numeric-command words you can send to the device
- the SPI-assigned code (such as 20 31) of each numeric-command word
- the title of each numeric-command word such as
  – setpoint
  – high deviation alarm
  – low deviation alarm

- the same for ASCII strings (though seldom used) to send to the device

Then, to receive status from the device via the SPI module:

- how many bit-status words you can receive from the device
- the SPI-assigned code (such as 20 40) of each bit-status word
- the purpose of each bit-status word such as
  – report if the device is turned on or off

- how many numeric-status words you can receive from the device
- the SPI-assigned code (such as 20 30) of each numeric-status word
- the title of each numeric-status word such as
  – process temperature or pressure
  – stored high deviation alarm
  – stored low deviation alarm

- the same for ASCII strings (seldom used) received from the device

The SPI protocol requires the transfer of at least five words (four header words + one data word) to transfer as little as a bit command to a device on the SPI network. This is in addition to the initial transfer of configuration words that prepare the SPI module to handle data transfers. The SPI protocol is relatively inefficient.

## Information to Configure the SPI Module

Configuration information that you initially transfer to the SPI module with the MCC does several things:

- identifies the devices with which it will communicate on the network
- establishes the order in which it will poll the devices
- sets the network's communication rate

Then you transfer device-specific configuration to the SPI module with one CCB for each device on the SPI network to set up the SPI module's memory to store specified command and status words. Each CCB defines:

- the number of command words to be sent to the device via the SPI module
- the number of status words to be sent to the device via the SPI module
- word maps to designate the type and location of all command words
- word maps to designate the type and location of all status words

## Information that you Send to / Receive From Devices Via the SPI Module

Once you have successfully configured the SPI module, you transfer data to and from the device via the SPI module with a CDB and a CDS, respectively.

The types of information that you can send to the device include:

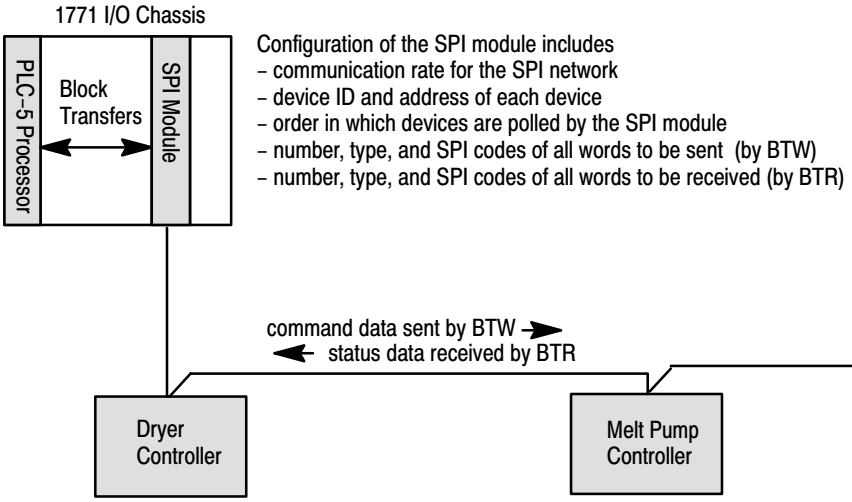
- bit commands to turn a machine on or off
- setpoints such as for temperature and pressure
- alarms to tell when temperature or pressure setpoints have been exceeded

The types of information that you can receive from the device include:

- status bits such as to indicate if the machine is on or off
- process temperatures or pressures
- pressure and temperature alarm limits currently stored by the device

## Informational Requirements for an Example SPI Network

To summarize the informational requirements, we present an example 3-device SPI network.




SPI-assigned device type ID = 22
Device address that you select = uu
Communication rate that you select = xxxx

Command words (by BTW) that set
- device on or off
- temperature setpoint
- high deviation alarm
- low deviation alarm

Status words (by BTR) that report
- device on or off
- process temperature
- high deviation alarm
- low deviation alarm

SPI-assigned device type ID = 24
Device address that you select = vv
Communication rate that you select = xxxx

Command words that set
- device on or off
- pressure setpoint
- high deviation alarm

Status words that report
- device on or off
- process pressure
- high deviation alarm

SPI-assigned device type ID = 26
Device address that you select = ww
Zone 1 address that you select = yy
Zone 2 address that you select = zz
Communication rate that you select = xxxx

Command words that set
- device on or off
- temperature setpoint, zone 1
- temperature setpoint, zone 2
- high deviation alarm
- low deviation alarm

Status words that report
- device on or off
- process temperature, zone 1
- process temperature, zone 2

## Wiring the SPI Network

The device manufacturer should provide a wiring diagram that shows the RS-485 data connections to the device, or at least a pin-out table of connections to the 9-pin D-shell connector on the device. Use this information with the following wiring considerations.

### Wiring Considerations

We recommend that you observe the following, because omission of one or more considerations may prevent the network from working correctly.

| SPI Terminal | Function | Wiring Considerations |
|---|---|---|
| A and 0 | RS-485 Data Connections | • Daisy chain to each device via the SPI network and secondary connection cable.<br>• SPI module revision C (and later) has an internal termination resistor. If replacing an earlier revision SPI module, discard the external termination resistor.<br>• You must terminate the other end of the SPI network with a 120 ohm resistor. |
| 0 and 1 | R1 Pullup Resistor | (Optional) Used with pulldown resistor R2. (See notes 1 and 2.) |
| 1 | Signal Common | Connect it to chassis ground at one end only. |
| 2 | Floating Terminal | Can be used as a tie point for wiring |
| 3, 6, 7, B | Not Used | Make no connections to these terminals. |
| 4 and 5 | R2 Pulldown Resistor | (Optional) Used with pullup resistor R1. (See notes 1 and 2.) |
| SPI Cable | Drain & Shield | Daisy chain to each of the primary machine connectors, then to earth ground. |

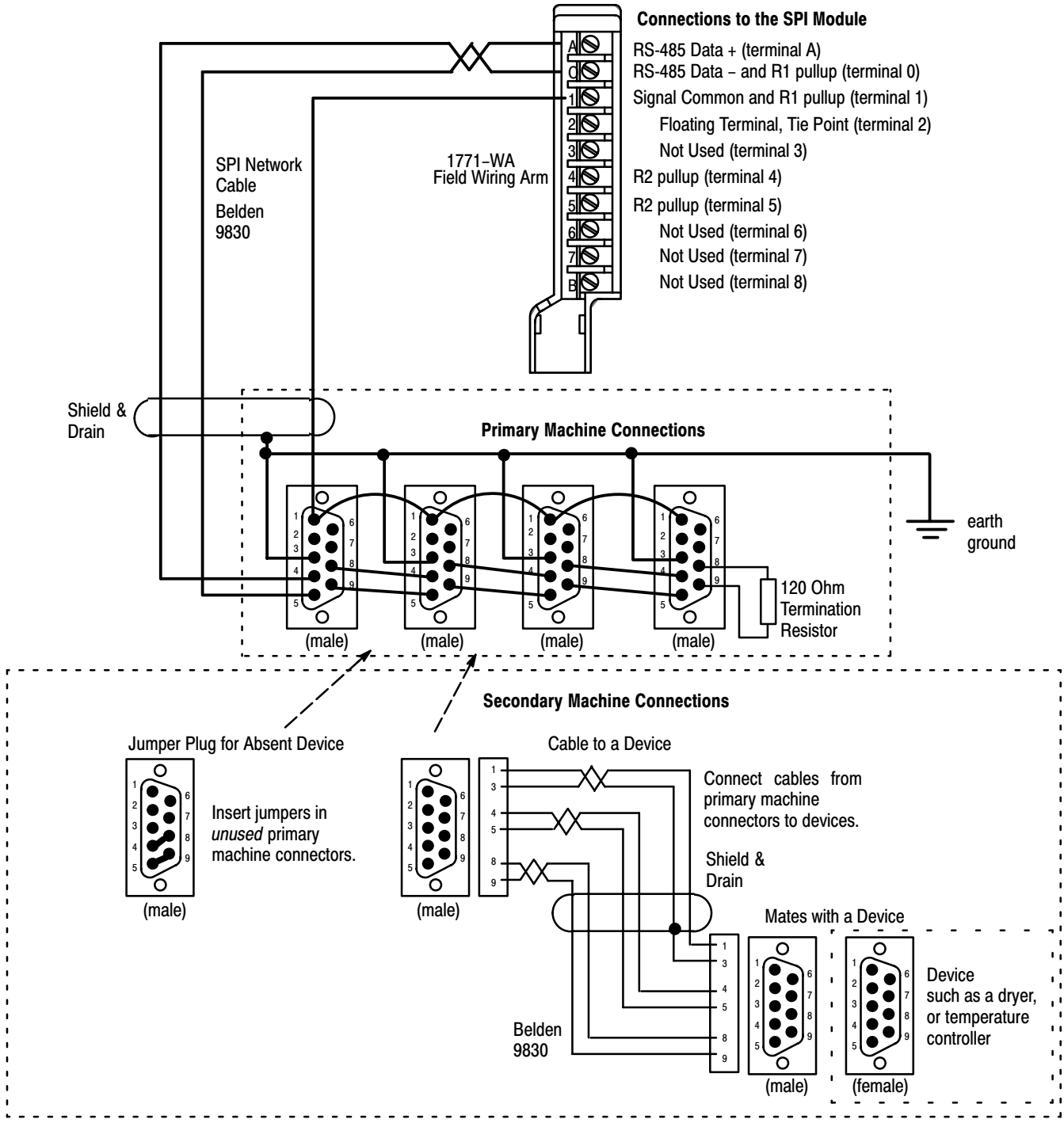
Note 1 Use R1 and R2 to trim the SPI module's output impedance when you need to boost the output-driver signal strength. Adding pullup and pulldown resistors increases the output current by loading the output. The nominal output impedance without these resistors is 100K ohms. R1 must equal R2. As an example, we compute the effective impedance for R1 = R2 = 40K ohms as follows:

$$R1 = R2 = R = \frac{100K \times 40K}{100K + 40K} = \frac{4000K}{140K} \simeq 28.6K$$

Note 2 Removing or adding a single device on the SPI network may change the network impedance sufficiently to require computing and substituting a different pair of pullup/pulldown resistors. We recommend that you determine and record the input impedance of each device. Then, compute the required value of the pullup/pulldown resistor pair for any desired combination of devices on the network.

## Example Diagram with Primary and Secondary Machine Connections

In the following example, a machine has a junction box (primary machine connections) to which you may connect up to four devices (secondary machine connections). Use jumper plugs in the *absence* of connected devices.



## Notes

Chapter 3

# Defining Command and Status Blocks

## Chapter Objectives

You communicate with devices on the SPI network by transferring data blocks between a PLC processor and an SPI module. In this chapter, we present the structure of data blocks and show relationships between them. In this chapter we cover:

- command and status blocks
- module communication command block (MCC)
- system status block (SYS)
- custom configuration block (CCB)
- custom data block (CDB)
- custom configuration status (CCS)
- custom data status (CDS)

## Command and Status Blocks

The processor uses command and status blocks to communicate with the SPI module. The processor writes command blocks to the module with block-transfer write (BTW) instructions, and reads status blocks from the module with block-transfer read (BTR) instructions. The status block can partially or totally reflect the data transferred in the command block.

SPI protocol uses three types of command and status blocks:

- module-specific configuration block sent to the SPI module to configure it, and an optional system status block returned by the SPI module
- device-specific configuration blocks sent to the SPI module to configure target devices on the SPI network, and configuration status blocks returned by the device via the SPI module
- device-specific data blocks sent to the SPI module destined for target devices on the SPI network, and device status blocks returned by the device via the SPI module

The SPI specification defines standard device types (types of devices) for use on the SPI network by ID code. Command and status blocks for each type of device have unique acronyms such as MCC for Module Configuration Command (block) or CDB for Custom Data Block.

## Module Configuration Command Block (MCC)

Your ladder logic must send (BTW) a valid MCC to the SPI module after power-up to prepare it for operation. This can be an MCC header (4 words) or a complete MCC block (up to 64 words). Only after the SPI module receives it, can the module operate as follows:

- accept any valid command block sent to it by BTW
- poll devices on the SPI network
- return device-specific status blocks by BTR
- return optional system status (SYS) by BTR

Until it receives a valid MCC block, the SPI module can only return the system status (SYS) block in response to a BTR request.

**Important:** Your ladder program must always initiate communication with the SPI module at power-up with the MCC.

**Table 3.A**
**Bit/Word Map of the Module Configuration Command Block (MCC)**

| Word | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | CNE | Baud Select | | | CNS | NF | | DF | MCC Block ID | | | | | | | |
| 2 | 0 | | | | 0 | | | | Page Number | | | | Total Pages | | | |
| 3 and 4 | 0 | | | | 0 | | | | 0 | | | | 0 | | | |
| The following word pairs (5&6, 7&8, etc.) define SPI network devices.<br>**Important:** List them in the order in which you want them polled (queue order). | | | | | | | | | | | | | | | | |
| 5 | DS | CS | PM | Reserved | | | | | Device ID (for first queued BTR) | | | | | | | |
| 6 | Zone Address (for temperature control devices, only) | | | | | | | | Device Address | | | | | | | |
| 7 | DS | CS | PM | Reserved | | | | | Device ID (for 2nd queued BTR) | | | | | | | |
| 8 | Zone Address (for temperature control devices, only) | | | | | | | | Device Address | | | | | | | |
| | ⇓ | | ⇓ | | | | | | ⇓ | | ⇓ | | | | | |
| 63 | DS | CS | PM | Reserved | | | | | Device ID (for 30th queued BTR) | | | | | | | |
| 64 | Zone Address (for temperature control devices, only) | | | | | | | | Device Address | | | | | | | |
| [ 1 ] | DS | CS | PM | Reserved | | | | | SYS Block ID | | | | | | | |
| | Reserved | | | | | | | | SPI Module Address | | | | | | | |

[ 1 ]: You must add the SYS Block ID (11111111) to the list of device IDs if you want the SPI module to return system status. If so, add it last in the list.

**Table 3.B**
**Word/Bit Definitions of the Module Configuration Command Block (MCC)**

| Word Number | Bit Number | Title | Range / Description | |
|---|---|---|---|---|
| 1 | 0–7 | Block ID | 11111111 MCC block ID | |
| | 8 | Data Format<br>DF | 0 = decimal data (required for PLC-5)<br>1 = do not use (for other processors) | **Note**: Determines the format of device ID and addresses for ALL data blocks |
| | 9–10 | Numeric Format<br>NF | 00 = floating point<br>01 = decimal data (preferred) | **Note:** Determines numeric format for ALL data blocks |
| | 11 | Cmd not Supported<br>CNS | 0 = SPI module reports Command not Supported errors from all devices<br>1 = SPI module does NOT report these errors | |
| | 12–14 | Baud Select | Bits 14 13 12 Baud<br>0 0 0 1200<br>0 0 1 2400<br>0 1 0 4800<br>0 1 1 9600<br>1 0 0 19200 | **Note:** Determines the SPI module's communication rate with ALL devices on the SPI network |
| | 15 | Cmd not Executed<br>CNE | 0 = SPI module reports Command not Executed errors from all device.<br>1 = SPI module does NOT report these errors | |
| 2 | 0–3 | Number of MCC pages | 0 = one page<br>9 = 10 pages | **Note:** If the MCC requires more than one page, download pages in numeric order. Last page must match page number (bits 4-7). |
| | 4–7 | Page Number | 0 = page 1<br>9 = page 10 | **Note:** If the MCC requires more than one page, your ladder logic must toggle these bits to identify each successive page number. |
| | 8–15 | Not used | zero | |
| 3 and 4 | | Not used | zero | |
| Each pair of words below defines a device on the SPI network.<br>The SPI module will poll them in listed order. You may repeat a device for more frequent polling. | | | | |
| 5 | 0–7 | Device Type | The 8-bit code defines the device type. See | |
| | 8–12 | Reserved | | |
| | 13 | Protected Mode<br>PM | 0 = not used<br>1 = Use machine protected mode | **Note:** Tells the SPI module to use the protected machine mode word: Poll 20 4A and Select 20 4B |
| | 14 | Configuration Select<br>CS | 0 = not used<br>1 = Use custom configuration | **Important:** Tells the SPI module that:<br>* this device will be configured by a custom CCB.<br>* data will be sent to it by CDB. |
| | 15 | Device Select<br>DS | 0 = normal BTR of status<br>1 = disable BTR reporting | **Note**: *Disable* removes the device from the SPI network queue. Lets you read other devices selectively |
| 6 | 0–7 | Device Address | 32–255 (decimal data) | **Note:** Use the address that you selected with switches on the device. |
| | 8–15 | Zone Address | 48–255 (decimal data) | **Note:** If the device has multiple temperature zones, use assigned zone addresses. If not, zero the digits. |
| Use word pairs 7 & 8, 9 & 10, 11 & 12, up to 63 & 64 as needed for multiple devices on the SPI network. | | | | |

## System Status Block (SYS)

The system status (SYS) block is returned by the SPI module on power-up with the power up bit reset (to zero) to indicate that the module is ready for a valid MCC. Until the SPI module is configured by a valid MCC, the module returns the SYS with each BTR request regardless of the device type or station address specified in the BTR request. Until the SPI module stores a valid MCC, it returns the SYS with the following diagnostics to help you correct data entry errors in the MCC:

- a diagnostic code in word 3 (see 9000 series codes in chapter 6)
- invalid device identification from the MCC polling queue list
  – invalid device ID in SYS word 2, upper byte
  – invalid device address in SYS word 4, lower byte
  – invalid zone address (if used) in SYS word 4 upper byte

As soon as the SPI module receives a valid MCC, it returns the SYS with the power-up bit set. Then, it will respond to BTR requests and poll devices on the SPI network in the queue order established by the MCC.

**Important:** After you have successfully configured the SPI module with a valid MCC, the use of the SYS is optional. If you want network status returned by the SPI module as a result of polling devices in the SPI network (one device-OK status bit per device), add the SYS block ID at the end of the polling list of device IDs that you enter in the MCC. If using custom command and status blocks such as CDB and CDS, you probably will have little reason to use SYS after configuring the SPI module with a valid MCC.

**Table 3.C**
**Bit/Word Map of System Status Block (SYS)**

| Word \| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | PU | Baud Select | | | AI | NF | | DF | SYS Block ID | | | | | | | |
| 2 | Device ID (returned for an invalid entry in MCC polling queue list) | | | | | | | | Page Number | | | | Total Pages | | | |
| 3 | Diagnostic Word (returned for an Invalid MCC) | | | | | | | | | | | | | | | |
| 4 | Zone Address for temperature control devices, only (returned for an invalid entry in MCC polling queue list) | | | | | | | | Device Address (returned for an invalid entry in MCC polling queue list) | | | | | | | |
| Communication Status Bits (OK Bits) for Devices on the SPI Network | | | | | | | | | | | | | | | | |
| 5 | Devices 32-47 indicated by word 5, bits 00-15 (1 bit per device) | | | | | | | | | | | | | | | |
| 6 | Devices 48-63 indicated by word 6, bits 00-15 (1 bit per device) | | | | | | | | | | | | | | | |
| | : : : | | | | | | | | | | | | | | | |
| 18 | Devices 240-255 indicated by word 18, bits 00-15 (1 bit per device) | | | | | | | | | | | | | | | |

**Table 3.D**
**Word/Bit Definitions of the System Status Block (SYS)**

| Word Number | Bit Number | Title | Range / Description | |
|---|---|---|---|---|
| 1 | 0–7 | Block ID | 11111111 SYS block ID | **Mirrors the MCC when the entry is valid.** |
| | 8 | Data Format DF | 0 = decimal data | **Mirrors the MCC when the entry is valid.** |
| | 9–10 | Numeric Format NF | 00 = floating point<br>01 = decimal data | **Mirrors the MCC when the entry is valid.** |
| | 11 | Command not Supported CNS | 0 = SPI module reports Command not supported errors.<br>1 = SPI module does NOT report these errors. | **Mirrors the MCC when the entry is valid.** |
| | 12–14 | Baud Select | Bits 14 13 12 Baud<br>0 0 0 1200<br>0 0 1 2400<br>0 1 0 4800<br>0 1 1 9600<br>1 0 0 19200 | **Mirrors the MCC when the entry is valid.** |
| | 15 | Power Up PU | 0 = expecting MCC block, or invalid MCC<br>1 = MCC block stored OK | **Note:** Tells the processor that the SPI module stored a valid MCC block. |
| 2 | 0–3 | Number of MCC pages | 0 = one page<br>9 = 10 pages | **Mirrors the MCC when the entry is valid.** |
| | 4–7 | Page Number | 0 = page 1<br>9 = page 10 | **Mirrors the MCC when the entry is valid.** |
| | 8–15 | Device Type | returned with word 3 for invalid entry in the MCC polling queue list, or zero if MCC is OK | |
| 3 | all | Diagnostic Word | returned to help you diagnose an invalid MCC, or zero if MCC is OK | |
| 4 | 0–7 | Device Address | returned with word 3 for invalid entry in the MCC polling queue list, or zero if MCC is OK | |
| | 8–15 | Zone Address | returned with word 3 for invalid entry in the MCC polling queue list , or zero if MCC is OK | |
| 5 thru 18 | all | SPI Network (OK) | 0 = device inactive on SPI network<br>1 = communication is OK | **Note:** One bit per device starting with word 5 bit 00 = device address 32 up to word 18 bit 15 = device address 255 |

## Custom Configuration Block (CCB)

The custom configuration block (CCB) tells the SPI module the required configuration for a single device, specified by the device ID in its header. Your program must transfer a CCB to the SPI module for each custom device on the SPI network. Then, your program must use custom *data* blocks (CDBs) to transfer data to the devices.

Transferring *custom* configuration, data, and status blocks to/from the SPI module uses fewer words and a simpler format than SPI-specified *standard* blocks. However, you must still configure the SPI module with the MCC.

The SPI module stores custom configuration and data in RAM. If the module loses power this information is lost, requiring replacement downloads. The SPI module can store:

- up to 10 CCBs for that number of devices
- up to 60 bit-command words and/or setpoints per device

**Table 3.E**
**Bit/Word Map of the Custom Configuration Block (CCB)**

| Word | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | | | | CA | DC | | | SPI-specified Device ID | | | | | | | |
| 2 | Number of numeric setpoints | | | | | | | | Number of bit-command words | | | | | | | |
| 3 | Number of bit-status words | | | | | | | | Number of ASCII strings | | | | | | | |
| 4 | Number of ASCII status strings | | | | | | | | Number of numeric status words | | | | | | | |
| starting at 5 | CMD1 Code (such as 20) | | | | | | | | CMD2 Code (such as 49) | | | | | | | |
| | SPI codes CDM1 and CDM2 define the specific bit-command word that your program will load into word 5 *of the CDB.*<br>If you have more codes of this type, load them in the next consecutive words. Leave no blank words.<br>The number of bit-command words that you entered in word 2 bits 0-7 must be identical to the number of:<br># code words (such as word 5 of this block)<br># bit-command words that your program will load *into the CDB* starting with word 5. | | | | | | | | | | | | | | | |
| | ⇓ | | ⇓ | | ⇓ | | ⇓ | | ⇓ | | ⇓ | | ⇓ | | ⇓ | |
| 6 (or higher) | CMD1 Code (such as 20) | | | | | | | | CMD2 Code (such as 31) | | | | | | | |
| | SPI codes CDM1 and CDM2 define the specific numeric setpoint that your program will load next *into the CDB.*<br>If you have more codes of this type, load them in the next consecutive words. Leave no blank words.<br>The number of numeric setpoints that you entered in word 2 bits 8-15 must be identical to the number of:<br># code words (such as word 6 of this block)<br># numeric setpoints that your program will load *into the CDB* following the bit-command words. | | | | | | | | | | | | | | | |
| | ⇓ | | ⇓ | | ⇓ | | ⇓ | | ⇓ | | ⇓ | | ⇓ | | ⇓ | |
| 7 (or higher) | CMD1 Code (such as 20) | | | | | | | | CMD2 Code (such as 22) | | | | | | | |
| | SPI codes CDM1 and CDM2 define the specific ASCII string that your program will load next *into the CDB.*<br>If you have more codes of this type, load them in the next consecutive words. Leave no blank words.<br>The number of ASCII strings that you entered in word 3 bits 0-7 must be identical to the number of:<br># code words (such as word 7 of this block)<br># ASCII strings that your program will load *into the CDB* following the numeric setpoints. | | | | | | | | | | | | | | | |
| ⇓ **Repeat for bit-,numeric-,and ASCII-*status* codes that you want returned in the CDS during automatic polling.** ⇓ | | | | | | | | | | | | | | | | |
| 64 | CMD1 Code (such as 20) | | | | | | | | CMD2 Code (such as 3C) | | | | | | | |

**Table 3.F**
**Word/Bit Definitions of the Custom Configuration Block (CCB)**

| Word Number | Bit Number | Title | Range | Notes |
|---|---|---|---|---|
| 1 | 0–7 | Device ID | 00100000 to 11111110 | Use the SPI-specified device ID. |
| | 8–10 | Determine Configuration DC (ID-specific) | 100 = Store configuration ><br>101 = Read configuration >><br>110 = Overwrite configuration >><br>111 = Clear configuration > | If device ID matches a stored ID, SPI returns an error.<br>If SPI finds no matching ID, it returns an error.<br>If SPI finds matching ID, new config overrides old.<br>SPI returns a flag, and ignores all other data in CCB. |
| | 11 | Clear All CA | 0 = No action<br>1 = Clear all configurations | The SPI module clears all stored configurations. |
| | 12–15 | Not used | | |
| 2 | 0–7 | Number of Bit-command Words | 0 to 60 decimal data | Tells the SPI module how many words containing bit commands to expect in the CDB. |
| | 8–15 | Number of Numeric Setpoints | 0 to 30 decimal data | Tells the SPI module how many setpoints and/or alarm limits to expect in the CDB. |
| 3 | 0–7 | Number of ASCII Strings | 0 to 30 decimal data | Tells the SPI module how many ASCII strings to expect in the CDB. |
| | 8–15 | Number of Bit-status Words | 0 to 30 decimal data | Tells the SPI module how many words containing status bits to return to the processor in the CDS. |
| 4 | 0–7 | Number of Numeric-status Words | 0 to 30 decimal data | Tells the SPI module how many numeric status words such as temperature or flow to return in the CDS. |
| | 8–15 | Number of ASCII-status Strings | 0 to 30 decimal data | Tells the SPI module how many ASCII strings such as SPI Revision to return to the processor in the CDS. |
| Starting at word 5 | 8–15 | CMD1 | starts at 00100000 20 | SPI code pair for the specified bit-command word sent to the SPI module.<br>For example: 20 49 for Mode, Machine |
| | 0–7 | CMD2 | starts at 00100010 22 | |
| Note 1 | 8–15 | CMD1 | starts at 00100000 20 | SPI code pair for the specified numeric setpoint sent to the SPI module.<br>For example: 20 31 for Temperature Setpoint |
| | 0–7 | CMD2 | starts at 00100010 22 | |
| Note 1 | 8–15 | CMD1 | starts at 00100000 20 | SPI code pair for the specified ASCII string sent to the SPI module.<br>For example, 20 22 for SPI Revision |
| | 0–7 | CMD2 | starts at 00100010 22 | |
| Note 1 | 8–15 | CMD1 | starts at 00100000 20 | SPI code pair for the specified bit-status word sent to the processor.<br>For example, 20 44 for Status, Machine |
| | 0–7 | CMD2 | starts at 00100010 22 | |
| Note 1 | 8–15 | CMD1 | starts at 00100000 20 | SPI code pair for the specified numeric status word returned to the processors.<br>For example, 20 72 for Process Temperature |
| | 0–7 | CMD2 | starts at 00100010 22 | |
| Note 1 | 8–15 | CMD1 | starts at 00100000 20 | SPI code pair for the specified ASCII status word returned to the processor.<br>For example, ?? ?? |
| | 0–7 | CMD2 | starts at 00100010 22 | |

Note 1: Load command-code pairs in the order given. Leave no blank words. If not using a specified type of command code, the SPI module expects the next type instead. Command codes for status specify the type of status that you want returned in the CDS.

## Custom Data Block (CDB)

Custom data blocks (CDBs) are used with custom configuration blocks (CCBs) blocks to communicate with devices on the SPI network via the SPI module. For a specified device on the SPI network, the custom configuration block (CCB) tells the SPI module:

- the type of data in each word, specified by SPI command-code pairs
- the location of data, specified by word number
- the order in which it will store the data

Custom data blocks must obey the "specs" of custom configuration blocks.

**Table 3.G**
**Word/Bit Map of the Custom Data Block (CDB)**

| Word | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | | | | | DC | | | SPI-specified Device ID | | | | | | | |
| 2 | Zone Address (for temperature control devices, only) | | | | | | | | Device Address | | | | | | | |
| 3 and 4 | 0 | | | | 0 | | | | 0 | | | | 0 | | | |
| starting at 5 | Load bit-command words for BTW to the SPI module whose code(s) you specified in the CCB block. (If you specified none in the CCB block, start loading numeric setpoints here.) | | | | | | | | | | | | | | | [ 1 ] |
| | : | | | | | | | | : | | | | | | | : |
| two words | Load numeric presets for BTW to the SPI module whose code(s) you specified in the CCB block. (Numeric setpoints are 32-bit words, equivalent to 2 PLC words) | | | | | | | | | | | | | | | [ 1 ] |
| | : | | | | | | | | : | | | | | | | : |
| two words | Load ASCII strings for BTW to the SPI module whose code(s) you specified in the CCB block. (ASCII strings are 32-bit words, equivalent to 2 PLC words) | | | | | | | | | | | | | | | [ 1 ] |

[ 1 ]: Command words that you load in this CDB must match the SPI command codes that you loaded in the CCB. They must match in type, quantity and order. Otherwise, the SPI module declares a fault.

**Table 3.H**
**Word/Bit Definitions for the Custom Data Block (CDB)**

| Word Number | Title | Range | Description |
|---|---|---|---|
| 1 | SPI Device ID | starts at 00100000 20 | Must use the SPI-specified device ID. |
| | Determine Configuration DC | 000= Standard<br>**001 = Custom**<br>010 = Random Data<br>011 = Random Data Open | Use 001 for custom configuration stored in SPI RAM. |
| | Not used | | |
| 2 | Device Address | 32 to 255 decimal data | Use the address that you selected on the device with switches. |
| | Zone Address | 48 to 255 decimal data | Use for temperature control devices, only. Otherwise, zero. |
| 3 and 4 | Not used | | Not used |
| stating at 5 | Bit-command word | 0 to 255 decimal data | Bit-command words must match the corresponding SPI codes that you loaded in the CCB in type, quantity and order. |
| | : | : | : : |
| two words per | Numeric preset (32-bit) | 0 to 32,767 decimal data | Numeric setpoints, such as alarm limits, must match the corresponding SPI codes that you loaded in the CCB in type, quantity and order. |
| | : | : | : : |
| two words per | ASCII string (32-bit) | 0 to 32,767 decimal data | ASCII strings such as SPI Revision must match the corresponding SPI codes that you loaded in the CCB in type, quantity and order. |

## Custom Configuration Status (CCS)

The SPI module responds to the custom configuration block (CCB) by reflecting it back to the PLC processor as custom configuration status (CCS) *unless* the SPI module detects:

- an error in the CCB
- it cannot store the CCB

Upon detecting either of the above, the SPI module inserts error information in the CCB and returns it as the CCS as follows:

- sets the configuration error CE bit to 1
- loads fault diagnostics into word 3

**Table 3.I**
**Bit/Word Map of Custom Configuration Status (CCS)**

| Word | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | CE | CA | DC | | | Device Identification | | | | | | | |
| 2 | Number of Numeric Setpoints or 0 | | | | | | | | Number of Bit-command Words or 0 | | | | | | | |
| 3 | Number of Bit-status Words or Fault Diagnostics | | | | | | | | Number of ASCII Strings or Fault Diagnostics | | | | | | | |
| 4 | Number of ASCII-status Strings or 0 | | | | | | | | Number of Numeric-status Words or 0 | | | | | | | |
| 5 | CMD1 (for example, 20) | | | | | | | | CMD2 (for example, 49) | | | | | | | |
| : | CMD1 (for example, 20) | | | | | | | | CMD2 (for example, 31) | | | | | | | |
| : | CMD1 (for example, 20) | | | | | | | | CMD2 (for example, 22) | | | | | | | |
| 64 | CMD1 (for example, 20) | | | | | | | | CMD2 (for example, 44) | | | | | | | |

**Table 3.J**
**Bit/Word Definitions of Custom Configuration Status (CCS)**

| Word Number | Bit Number | Title | Range | SPI Module: |
|---|---|---|---|---|
| 1 | 0–7 | Device ID | 00100000 to 11111110 | mirrors the SPI-specified device ID. |
| | 8–10 | Determine Configuration DC (ID-specific) | 100 = Store configuration >><br>**101 = Read configuration** >><br>110 = Overwrite configuration >><br>111 = Clear configuration >> | If device ID matches a stored ID, SPI returns an error.<br>If SPI finds no matching ID, it returns an error.<br>If SPI finds matching ID, new config overrides old.<br>SPI returns a flag, and ignores all other data in CCB. |
| | 11 | Clear All CA | 0 = No action<br>1 = Clear all configurations | mirrors the CA command |
| | 12 | Command Executed CE | 0 = SPI module detected an error<br>1 = SPI executed the command | sets this bit to tell the PLC processor that the custom configuration block (CCB) was stored without error. |
| | 13–15 | Not used | | |
| If DC (Determine Configuration) bits 8-10 above = 101 = Read Configuration, then the SPI module returns words 2 up to 64 as follows. (If the SPI module finds no matching ID (cannot read specified configuration), it indicates an error by zeroing words 2 up to 64.) | | | | |
| 2 | 0–7 | Number of Bit-command Words | 0 to 60 decimal data | mirrors the number of words containing bit commands |
| | 8–15 | Number of Numeric Setpoints | 0 to 30 decimal data | mirrors the number of setpoints and/or alarm limits<br>If it detected an error, zeros this byte. |
| 3 | 0–7 | Number of ASCII Strings | 0 to 30 decimal data | reflects number of ASCII strings<br>If it detected an error, returns fault diagnostics. |
| | 8–15 | Number of Bit-status Words | 0 to 30 decimal data | mirrors the number of words containing bit status<br>If it detected an error, returns fault diagnostics. |
| 4 | 0–7 | Number of Numeric-status Words | 0 to 30 decimal data | mirrors the number of numeric-status words such as temperature. If it detected an error, zeros this byte. |
| | 8–15 | Number of ASCII-status Strings | 0 to 30 decimal data | mirrors the number of ASCII-status strings<br>If it detected an error, zeros this byte. |
| Starting at word 5 | 8–15 | CMD1 | starts at 00100000 20 | mirrors the SPI code pair for the specified bit-command word sent to the SPI module.<br>For example: 20 49 for Mode, Machine |
| | 0–7 | CMD2 | starts at 00100010 22 | |
| word pair | 8–15 | CMD1 | starts at 00100000 20 | mirrors the SPI code pair for the specified numeric setpoint sent to the SPI module.<br>For example: 20 31 for Temperature Setpoint |
| | 0–7 | CMD2 | starts at 00100010 22 | |
| word pair | 8–15 | CMD1 | starts at 00100000 20 | mirrors the SPI code pair for the specified ASCII string sent to the SPI module.<br>For example, 20 22 for SPI Revision |
| | 0–7 | CMD2 | starts at 00100010 22 | |
| word pair | 8–15 | CMD1 | starts at 00100000 20 | mirrors the SPI code pair for the specified bit-status word returned to the processor.<br>For example, 20 44 for Status, Machine |
| | 0–7 | CMD2 | starts at 00100010 22 | |
| word pair | 8–15 | CMD1 | starts at 00100000 20 | mirrors the SPI code pair for the specified numeric status word returned to the processors.<br>For example, 20 72 for Process Temperature |
| | 0–7 | CMD2 | starts at 00100010 22 | |
| word pair | 8–15 | CMD1 | starts at 00100000 20 | mirrors the SPI code pair for the specified ASCII status word returned to the processor. |
| | 0–7 | CMD2 | starts at 00100010 22 | |

## Custom Data Status (CDS)

The SPI module returns custom data status (CDS) from a custom-configured device whenever the SPI module:

- reads status from the device during a polling sequence
- acknowledges a CDB command addressed to the device

The status contained in CDS is different for each of the above.

| If the SPI module: | Then it: |
|---|---|
| detects no faults | 1. Sets the command executed (CE) bit 12 to 1.<br>2. Indicates the CDS origin with command response (CR) bit 13:<br>CR = 0 = response to a polling sequence<br>CR = 1 = response to a CDB command<br>3. Loads CDS according to CR status:<br>CR = 0 (polling sequence) = loads CDB with status that you specified in CCB<br>CR = 1 (response to CDB) = reflects the CDB as CDS |
| detects a fault | 1. Sets the command executed (CE) bit 12 to 0.<br>2. Loads fault diagnostics into word 3 of the CDS.<br>3. Zeroes follow-on specified status words in the CDS. |

### CDS from Polling Sequence

Command codes that you defined in the CCB determine the device status returned by the SPI module after polling devices on the SPI network (polling sequence). These command codes specify the bit status, numeric status, and/or ASCII status strings that you want returned in the CDS starting at word 5 (Table 3.K).

**Table 3.K**
**Custom Data Status (CDS)**

| Word | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | | CR | CE | 0 | DC | | | Device Identification | | | | | | | |
| 2 | Zone Address | | | | | | | | Station Address | | | | | | | |
| 3 | Diagnostic Word: If the SPI module detects a fault, it sets an error code to help identify the cause of the fault | | | | | | | | | | | | | | | |
| 4 | 0 | | | | 0 | | | | 0 | | | | 0 | | | |
| 5 | Returns the first bit-status word that you specified by a command-code pair in CCB, followed by others if specified. Reports zero for a fault. | | | | | | | | | | | | | | | |
| two words, per | Returns the numeric-status word(s) that you specified by command-code pairs in CCB. Reports zero for a fault. | | | | | | | | | | | | | | | |
| two words, per | Returns the ASCII-status word(s) that you specified by command-code pairs in CCB. Reports zero for a fault. | | | | | | | | | | | | | | | |

## CDS Resulting from CDB

The SPI module responds to CDB commands by reflecting the CDB back to the PLC processor as the CDS (Table 3.L):

**Table 3.L**
**Custom Data Status (CDS)**

| Word | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | | CR | CE | 0 | DC | | | Device Identification | | | | | | | |
| 2 | Zone Address | | | | | | | | Station Address | | | | | | | |
| 3 | Diagnostic Word: If the SPI module detects a fault, it sets an error code to help identify the cause of the fault | | | | | | | | | | | | | | | |
| 4 | 0 | | | | 0 | | | | 0 | | | | 0 | | | |
| starting at 5, up to 64 | If no communication fault, reflects all the command-code pairs that you entered in CCB. Reports zero for a fault. | | | | | | | | | | | | | | | |

**Table 3.M**
**Bit/Word Definitions for Custom Data Status (CDS)**

| Word Number | Bit Number | Title | Range |
|---|---|---|---|
| 1 | 0-7 | Block ID | 00100000 to 11111111 |
| | 8-10 | Determine Configuration DC | 000= Standard Configuration<br>**001 = Custom Configuration**<br>010 = Random Data Configuration<br>011 = Random Data Open Config<br>100 = Store Configuration<br>101 = Read Configuration<br>110 = Overwrite Configuration<br>111 = Clear Configuration |
| | 11 | Reserved | zero |
| | 12 | Command Executed CE | Bit 12 = 0 – Command not executed<br>Bit 12 = 1 – Command executed |
| | 13 | Command Response CR | Bit 13 = 0 – response to a queued read<br>Bit 13 = 1 – response to a CDB |
| | 14-15 | Reserved | zero |
| 2 | 0-7 | Station Address | 32 to 255 decimal data |
| | 8-15 | Zone Address (temperature control) | 48 to 255 decimal data |
| 3 | 0-15 | Diagnostic Word | error code if SPI module detected a fault |
| 4 | | Not used | zero |
| 5 up to 64 | 0-15 or 0-32 | Reflected command, specified status, or zero | For a CDB, reflects the command-code pair.<br>For a polling sequence, returns specified status.<br>For a fault, clears the word. |

## Summary of SPI Block Acronyms, Device Types, and ID Codes

The following table helps you identify:

- acronyms of various command and status blocks that your ladder logic transfers to and from the SPI module
- SPI-specified ID codes of devices that you can place on the SPI network

**Table 3.N**
**Standard Device Types with ID Code, and Acronyms for Command and Status Blocks**

| Type of Command or Status Block | ID Code HEX | ID Code Binary | Block Acronym Command | Block Acronym Status |
|---|---|---|---|---|
| SPI-specified Module Configuration Command | FF | 11111111 | MCC | |
| SPI-specified System Status | FF | 11111111 | | SYS |
| Custom Configuration Block | Select the SPI-specified device-type ID from the ID codes listing below. Allowable range is 20-FE. | | CCB | |
| Custom Configuration Status | | | | CCS |
| Custom Data Block | | | CDB | |
| Custom Data Status | | | | CDS |
| **Type of Device** | **ID Code HEX** | **ID Code Binary** | **Block Acronym Command** | **Block Acronym Status** |
| Mold Temperature Controller | 20 | 00100000 | MTC | MTS |
| Chiller | 21 | 00100001 | CHC | CHS |
| Dryer | 22 | 00100010 | DRC | DRS |
| Loader | 23 | 00100011 | LDC | LDS |
| Melt Pump | 24 | 00100100 | MPC | MPS |
| Additive Feeder | 25 | 00100101 | AFC | AFS |
| Self-tuning Temperature Controller (Hot Runner) | 26 | 00100110 | STC | STS |
| General-purpose Temperature Controller | 27 | 00100111 | GPC | GPS |
| Volumetric Blender | 28 | 00101000 | BVC | BVS |
| Continuous-weigh Blender | 29 | 00101001 | BCW | BCS |
| Batch-weigh Blender | 2A | 00101010 | BBW | BBS |
| Robot | 2B | 00101011 | ? | ? |

Chapter 4

# Creating Command and Status Blocks to Match Your Application

## Objective

This chapter helps you:

- map your data table for command and status blocks
- enter initial conditions in command blocks for transfer to the SPI module
- determine where you expect to find status returned from the SPI module

You must do this for the following blocks:

- MCC and SYS
- CCB and CCS
- CDB and CDS

## Task Overview

Map your data table for command and status blocks as follows:

- For Command Blocks MCC, CCB, and CDB
  Use command-block worksheets to write down all required information that your ladder logic must transfer to the SPI module. You must enter it into command blocks in the data table after you map their addresses.

  We show an example worksheet followed by a blank worksheet that you can use to write down required information for your own application.



- For Status Blocks SYS, CCS, and CDS
  Use bit/word maps of status blocks to determine where you can expect to find status returned from the SPI module. After mapping their addresses in the data table, use the maps to determine source addresses for move instructions so your ladder logic can use the status.

If necessary, refer the the corresponding command or status block in chapter 3 for explanations and locations of parameters.

## Module Configuration Command Block, MCC, to Configure the SPI Module

Use this worksheet for the MCC to write down:

- data format of all addresses and data
- communication rate of the SPI link
- number of pages of the MCC *(above* 1 page of 64 words)
- queue order in which the SPI module will poll devices on the link
- device ID and address code of devices in the polling queue list

### Example MCC Worksheet

| Word | Write your entry in this column. *(example in italics)* | Notes |
|---|---|---|
| 1 | link baud = *9600*<br>address format = *integer* numeric format = *integer* | Enter this data by setting bits. See MCC in chapter 3. |
| 2 | # of MCC pages above 1 = *0* | One MCC page = 64 words. |
| 3-4 | zero | |
| 5 & 6 | 1st device *Dryer* *ID = 22 (00100010)* *Addr = 20* | Polling Order:<br>Enter types of devices, ID and address codes in the order you want them polled.<br><br>You may enter the same device more than once if you want it polled more frequently. |
| 7 & 8 | 2nd device *GP Temp Controller* *ID = 27 (00100111)* *Addr = 21* | |
| 9 & 10 | 3rd device *Melt Pump* *ID = 24 (00100100)* *Addr = 22* | |
| 11 & 12 | 4th device *Loader* *ID = 23 (00100011)* *Addr = 23* | |
| : | | |
| last pair or 63, 64 | SYS returned? *no SYS returned after polling* | Enter SYS if you want SYS returned to the processor after each polling sequence. |

## Blank MCC Worksheet for Your Application

Write your own entries in this blank worksheet to suit your application.

### MCC Worksheet

| Word | Write your entries in this column. |
|---|---|
| 1 | baud rate = _____ SPI reports: Commands not Executed Y N . . . not Supported Y N<br>address format [1] = _____ numeric format [1] = _____ |
| 2 | # of MCC pages above 1 = _____ |
| 3-4 | zero |
| 5 & 6 | 1st device ID = Addr = |
| 7 & 8 | 2nd device |
| 9 & 10 | 3rd device |
| 11 & 12 | 4th device |
| 13 & 14 | 5th device |
| 15 & 16 | 6th device |
| 17 & 18 | 7th device |
| 19 & 20 | 8th device |
| 21 & 22 | 9th device |
| 23 & 24 | 10th device |
| 25 & 26 | 11th device |
| 27 & 28 | 12th device |
| 29 & 30 | 13th device |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| last pair or 63, 64 | SYS returned? |

[1] Your format selection here determines the format for all command and status bocks for the network.

**Important:** If the number of devices on the SPI link exceeds 60 (requires another MCC page), continue on another sheet of paper and enter the number of MCC pages *above one page* in word 2, above. For example, if you need two MCC pages, enter *1* in word 2 above.

## Custom Configuration Block CCB, to Configure the SPI Module for a Device

You must create one CCB for each device type on the link to define:

- data layout of the CCB
- number of bit-command words and their command codes
- number of numeric preset words and their command codes
- number of ASCII string words and their command codes
- same information for status (independent of data sent)

**Important:** To obtain command-code pairs CMD1 and CMD2 for specifying commands and setpoints to be sent to the device and status returned from it, refer to the SPI Communication Protocol manual or the device manufacturer. Specify them according to your application requirements. In this example, command words 5-8 define data to be sent to the device via the SPI module. Status words 9-12 define data to be returned via the SPI module.

### Example CCB Worksheet

| Word | Write your entries in this column. *(example entries in italics)* | | | | Notes |
|---|---|---|---|---|---|
| 1 | device ID: name *Dryer* | | code *22* | | Enter the device-type ID. |
| 2 | # numeric preset words = *3* | # bit-command words = *1* | | | Numbers that you enter in words 2-4 must match corresponding number of entries in words 5 to end of file. |
| 3 | # bit status words = *2* | # ASCII strings = *0* | | | |
| 4 | # ASCII status words = *1* | # numeric status words = *1* | | | |
| **Word** | **Write your entries in this column.** | | **CMD1** | **CMD2** | **Notes** |
| 5 | bit command 1 | *Machine mode: turn machine on/off* | *20* | *49* | Enter names of bit commands and their specified command codes CMD1 and CMD2. [1] |
| 6 | numeric preset 1 | *Temperature Setpoint* | *20* | *31* | Enter names of numeric presets and their specified command codes CMD1 and CMD2. [1] |
| 7 | numeric preset 2 | *Alarm, Hi-temperature Deviation* | *20* | *33* | |
| 8 | numeric preset 3 | *Alarm, Lo-temperature Deviation* | *20* | *35* | |
| | ASCII string 1 | not used in this example | | | Enter ASCII strings and command codes CMD1 and CMD2. [1] |
| **Word** | **Write your entries in this column.** | | **CMD2** | **CMD1** | **Notes** |
| 9 | bit status word 1 | *Process Status* | *20* | *40* | Enter specified status codes CMD1 and CMD2 for bit-status words. [1] |
| 10 | bit status word 2 | *Machine Status* | *20* | *42* | |
| 11 | numeric status wrd 1 | *Process Temperature* | *20* | *72* | Enter status codes CMD1 and CMD2 for numeric-status words. [1] |
| 12 | ASCII status word 1 | *Device Revision Level* | *20* | *22* | Enter specified status codes CMD1 and CMD2 for ASCII status [1] |

[1] Leave no blank words between groups of entries. If not using a group, put the next group in its space.

**Important:** Status data (words 9-12 in this example) are returned to the processor from the target device by the SPI module in Custom Data Status (CDS), NOT in the Custom Configuration Status (CCS). They are returned only as a result of the SPI module's polling of devices on the link.

## Blank CCB Worksheet for Your Application

Write your own entries in this blank worksheet to suit your application.

**Important:** Because we cannot predict the number of bit, numeric, or ASCII entries in your application, we give you a blank word column and extra rows for each type of entry. Enter the number of the file word for each entry, and use blank rows as needed. Enter device ID and data in the formats that you specified in MCC word 1 bits 8-10. See notes [1] and [2] below.

## CCB Worksheet

| Word | Write your entries in this column. | | |
|---|---|---|---|
| 1 | device ID: name _____ code _____ | | |
| 2 | # numeric preset words = _____ # bit-command words = _____ | | |
| 3 | # bit status words = _____ # ASCII strings = _____ | | |
| 4 | # ASCII status words = _____ # numeric status words = _____ | | |
| **Word** | **Write your entries in this column. [1]** | **CMD1** | **CMD2** |
| 5 | bit command 1 | | |
| | | | |
| | numeric preset 1 | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | ASCII strings 1 | | |
| | | | |
| **Word** | **Write your entries in this column. [1] [2]** | **CMD2** | **CMD1** |
| | bit status word 1 | | |
| | | | |
| | | | |
| | numeric status word 1 | | |
| | | | |
| | | | |
| | | | |
| | ASCII status word 1 | | |
| | | | |

[1] Leave no blank words between groups of entries. If not using a group, put the next group in its space.
[2] Status words that you specify here are returned in CDS whenever the SPI module polls its devices.

## Blank CCB Worksheet for Your Application

Copy this worksheet as needed.
Write your own entries in this blank worksheet to suit your application.

## CCB Worksheet

| Word | Write your entries in this column. | | |
|---|---|---|---|
| 1 | device ID: name _____ code _____ | | |
| 2 | # numeric preset words = _____ # bit-command words = _____ | | |
| 3 | # bit status words = _____ # ASCII strings = _____ | | |
| 4 | # ASCII status words = _____ # numeric status words = _____ | | |
| **Word** | **Write your entries in this column. [1]** | **CMD1** | **CMD2** |
| 5 | bit command 1 | | |
| | | | |
| | numeric preset 1 | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | ASCII strings 1 | | |
| | | | |
| **Word** | **Write your entries in this column. [1] [2]** | **CMD2** | **CMD1** |
| | bit status word 1 | | |
| | | | |
| | | | |
| | numeric status word 1 | | |
| | | | |
| | | | |
| | | | |
| | ASCII status word 1 | | |
| | | | |

[1] Leave no blank words between groups of entries. If not using a group, put the next group in its space.
[2] Status words that you specify here are returned in CDS whenever the SPI module polls its devices.

## Blank CCB Worksheet for Your Application

Copy this worksheet as needed.
Write your own entries in this blank worksheet to suit your application.

## CCB Worksheet

| Word | Write your entries in this column. | | |
|---|---|---|---|
| 1 | device ID: name _____ code _____ | | |
| 2 | # numeric preset words = _____ # bit-command words = _____ | | |
| 3 | # bit status words = _____ # ASCII strings = _____ | | |
| 4 | # ASCII status words = _____ # numeric status words = _____ | | |
| **Word** | **Write your entries in this column. [1]** | **CMD1** | **CMD2** |
| 5 | bit command 1 | | |
| | | | |
| | numeric preset 1 | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | ASCII strings 1 | | |
| | | | |
| **Word** | **Write your entries in this column. [1] [2]** | **CMD2** | **CMD1** |
| | bit status word 1 | | |
| | | | |
| | | | |
| | numeric status word 1 | | |
| | | | |
| | | | |
| | | | |
| | ASCII status word 1 | | |
| | | | |

[1] Leave no blank words between groups of entries. If not using a group, put the next group in its space.
[2] Status words that you specify here are returned in CDS whenever the SPI module polls its devices.

## Custom Data Block, CDB, to Transfer Data to the Device

You must create a CDB to send data to the device.

The purpose of this worksheet is to help you enter values into the words that you defined in the CCB. We help you define:

- data layout of the CDB
- bit status of bit-command words
- values of numeric preset words
- ASCII strings

**Important:** Your entries in this CDB must match those entries for command words (to be sent to the device) that you specified in the CCB, entry for entry, in the same order, with no blank words.

**Important:** This example matches the CCB for the dryer that we previously described.

### Example CDB Worksheet

| Word | Write your entries in this column. *(example in italics)* | | Notes |
|---|---|---|---|
| 1 | device ID: *22 (Dryer)* | | Enter the device type ID. |
| 2 | device address: *(what you set with switches on the device)* | | Enter 2-digit device address. |
| 3 & 4 | zero | | |
| 5 | bit command 1 | *0000 0000 0000 000x*<br>0 = machine off, 1 = machine on | Set specified bits in each bit-command word. [1] |
| 6, 7 | numeric preset 1 | *490* ºF (temperature setpoint in TCS) | Enter preset values, **two words** per preset. Use second word of pair for floating-point numbers. Otherwise, leave it zero. [1] |
| 8, 9 | numeric preset 2 | *30* ºF (Hi-temp deviation alarm, TCS) | |
| 10, 11 | numeric preset 3 | *30* ºF (Lo-temp deviation alarm, TCS) | |
| pair | ASCII 1 | not used in this example | Enter ASCII strings, **two words** per string. [1] [2] |

[1] Your CDB entries here must match identically the command codes you entered in the CCB: entry for entry, in the same order, with no blank words
[2] One character per byte, four characters maximum.

**Important:** The SPI module returns status from the target device in the device's Custom Data Status (CDS).

## Blank CDB for Your Application

Write your own entries in this blank worksheet to suit your application.

**Important:** Because we cannot predict the number of bit, numeric, or ASCII entries in your application, we give you a blank word column and extra rows for each type of entry. Enter the number(s) of the file word(s) for each entry, and use blank rows as needed. Enter data, ID, and addresses in the formats that you specified in MCC word 1, bits 8-10. See notes [1] and [2] below.

## CDB Worksheet

| Word | Write your entries in this column. |
|---|---|
| 1 | device ID = _____ |
| 2 | device address = _____ |
| 3 & 4 | zero |
| 5 | bit command 1 [1] |
| | bit command 2 |
| | bit command 3 |
| | numeric preset 1 [1] |
| | numeric preset 2 |
| | numeric preset 3 |
| | numeric preset 4 |
| | numeric preset 5 |
| | numeric preset 6 |
| | numeric preset 7 |
| | numeric preset 8 |
| | numeric preset 9 |
| | ASCII string 1 [1] [2] |

[1] Your CDB entries here must match identically the command codes you entered in the CCB: entry for entry, in the same order, with no blank words

[2] One character per byte, four characters maximum.

## Blank CDB for Your Application

Copy this worksheet as needed.
Write your own entries in this blank worksheet to suit your application.

**Important:** Because we cannot predict the number of bit, numeric, or ASCII entries in your application, we give you a blank word column and extra rows for each type of entry. Enter the number(s) of the file word(s) for each entry, and use blank rows as needed. Enter data, ID, and addresses in the formats that you specified in MCC word 1, bits 8-10. See notes [1] and [2] below.

## CDB Worksheet

| Word | Write your entries in this column. |
|---|---|
| 1 | device ID = _____ |
| 2 | device address = _____ |
| 3 & 4 | zero |
| 5 | bit command 1 [1] |
| | bit command 2 |
| | bit command 3 |
| | numeric preset 1 [1] |
| | numeric preset 2 |
| | numeric preset 3 |
| | numeric preset 4 |
| | numeric preset 5 |
| | numeric preset 6 |
| | numeric preset 7 |
| | numeric preset 8 |
| | numeric preset 9 |
| | ASCII string 1 [1] [2] |

[1] Your CDB entries here must match identically the command codes you entered in the CCB: entry for entry, in the same order, with no blank words
[2] One character per byte, four characters maximum.

## Blank CDB for Your Application

Copy this worksheet as needed.
Write your own entries in this blank worksheet to suit your application.

**Important:** Because we cannot predict the number of bit, numeric, or ASCII entries in your application, we give you a blank word column and extra rows for each type of entry. Enter the number(s) of the file word(s) for each entry, and use blank rows as needed. Enter data, ID, and addresses in the formats that you specified in MCC word 1, bits 8-10. See notes [1] and [2] below.

## CDB Worksheet

| Word | Write your entries in this column. |
|---|---|
| 1 | device ID = _____ |
| 2 | device address = _____ |
| 3 & 4 | zero |
| 5 | bit command 1 [1] |
| | bit command 2 |
| | bit command 3 |
| | numeric preset 1 [1] |
| | numeric preset 2 |
| | numeric preset 3 |
| | numeric preset 4 |
| | numeric preset 5 |
| | numeric preset 6 |
| | numeric preset 7 |
| | numeric preset 8 |
| | numeric preset 9 |
| | ASCII string 1 [1] [2] |

[1] Your CDB entries here must match identically the command codes you entered in the CCB: entry for entry, in the same order, with no blank words
[2] One character per byte, four characters maximum.

## System Status Block, SYS

Use this SYS bit/word map to identify:

- data layout of the SYS
- location of the diagnostic word to describe detected MCC errors
- location of invalid device ID and/or device address
- number of bits in word 5 to indicate “device OK” on link (in last row)

We identify fault diagnostic information in bold print.

### SYS Bit/Word Map

| Word \| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | PU [1] | Baud Select [2] | | | AI [2] | NF [2] | | DF [2] | SYS Block ID [2] | | | | | | | |
| 2 | Device ID **(returned for an invalid entry in MCC polling list)** | | | | | | | | Page Number | | | | Total Pages | | | |
| 3 | Diagnostic Word **(Returned for an invalid MCC. See 9000 series error codes in chapter 6.)** | | | | | | | | | | | | | | | |
| 4 | Zone Address for temperature control devices, only **(Returned for an invalid entry in MCC polling list.)** | | | | | | | | Device Address **(Returned for an invalid entry in MCC polling list.)** | | | | | | | |
| Communication Status Bits (OK Bits) for Devices on the SPI Network | | | | | | | | | | | | | | | | |
| 5 | Devices 32-47 indicated by word 5, bits 00-15 (1 bit per device) | | | | | | | | | | | | | | | |
| 6 | Devices 48-63 indicated by word 6, bits 00-15 (1 bit per device) | | | | | | | | | | | | | | | |
| | : : : | | | | | | | | | | | | | | | |
| 18 | Devices 240-255 indicated by word 18, bits 00-15 (1 bit per device) | | | | | | | | | | | | | | | |

[1] Power Up bit **0 = no MCC, or invalid MCC**
1 = SPI module stored a valid MCC

[2] Mirrors the same MCC bit

**Important:** The SPI module returns the SYS after your program downloads the MCC. For diagnostic data returned in SYS word 3, refer to chapter 6, Troubleshooting. The return of SYS after each polling sequence is optional.

## Custom Configuration Status, CCS, to Return Configuration Status From the SPI Module

Once you have successfully configured the SPI module with the MCC and whenever you transfer the CCB to the SPI module, it responds by returning Custom Configuration Status (CCS).

You must correct any detected errors as indicated by the CCS before you can transfer data to devices whose configuration has been successfully stored in the SPI module.

The CCS is returned to the PLC processor by the SPI module. To map the CCS in your data table, use the same number of words as its corresponding CCB. We repeat the CCS bit/word map from chapter 3.

**Important:** When the SPI module detects a fault in the CCB, it returns the CCS with its status cleared, containing information only in words 1 and 3:

- word 1 identifies the device having the invalid CCB
- word 3 contains the 5000 series error codes described in chapter 6.

We identify fault diagnostic information in bold print.

### CCS Bit/Word Map

| Word | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | CE | CA | DC | | | **Device ID** | | | | | | | |
| 2 | Number of Numeric Setpoints **or 0** | | | | | | | | Number of Bit-command Words **or 0** | | | | | | | |
| 3 | Number of Bit-status Words, **or 5000 series error codes** | | | | | | | | Number of ASCII Strings, **or 5000 series error codes** | | | | | | | |
| 4 | Number of ASCII-status Strings **or 0** | | | | | | | | Number of Numeric-status Words **or 0** | | | | | | | |
| 5 | CMD1 (for example, 20) **or 0** | | | | | | | | CMD2 (for example, 49) **or 0** | | | | | | | |
| : | CMD1 (for example, 20) **or 0** | | | | | | | | CMD2 (for example, 31) **or 0** | | | | | | | |
| : | CMD1 (for example, 20) **or 0** | | | | | | | | CMD2 (for example, 22) **or 0** | | | | | | | |
| 64 | CMD1 (for example, 20) **or 0** | | | | | | | | CMD2 (for example, 44) **or 0** | | | | | | | |

## Custom Data Status, CDS, to Return Device Status From the SPI Module

The SPI module receives status from the target device and returns Custom Data Status (CDS) to the PLC processor whenever the target device:

- receives the Custom Data Block (CDB) from the SPI module
- is polled by the SPI module

The contents of the CDS are different for each condition:

| For this origin: | CDS contains |
|---|---|
| CDB command | the reflected command-code pairs that you entered in CCB |
| polling | numeric values (status bits and process variables)<br>specified by the command-code pairs and located in their addresses |

The size is the same for both conditions and is equal to the sum of:

- four header words
- number of command-code pairs for *status* that you entered in CCB

If the SPI module detects a fault, it returns the CDS with only the first three words containing fault diagnostics:

- words 1 and 2 identify the target device having the fault
- word 3 contains the 70xy series error codes described in chapter 6.

We identify fault diagnostic information in bold print.

### CDS Bit/Word Map

| Word | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | | CR | CE | 0 | DC | | | **Device ID** | | | | | | | |
| 2 | **Zone Address (if used)** | | | | | | | | **Device Address** | | | | | | | |
| 3 | **If the SPI module detects a fault, it sets an error code (70xy series) to help identify the cause of the fault** | | | | | | | | | | | | | | | |
| 4 | 0 | | | | 0 | | | | 0 | | | | 0 | | | |
| starting at 5, up to 64 | If no communication fault:<br>If origin is a CDB, CDS reflects all the command-code pairs that you entered in CCB.<br>If origin is polling, CDS returns numeric values specified by the command-code pairs.<br>**If SPI detects a communication fault, it**<br>**clears all status words starting at word 4, and returns words 1-3 with a diagnostic code (70xy) in word 3.** | | | | | | | | | | | | | | | |

Chapter 5

# Interpreting Diagnostic Information from LEDs and Error Codes

## Chapter Objectives

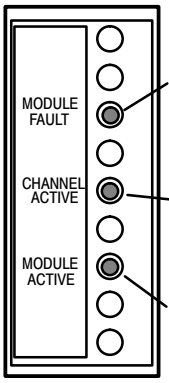
This chapter helps you troubleshoot the SPI module using:

- diagnostic indicators on the front of the module
- diagnostic codes returned by the SPI module

## Troubleshooting with Diagnostic Indicators

Diagnostic indicators (Figure 5.1) and (Table 5.A) indicate the SPI module's operating status. Use them to troubleshoot your module.

**Figure 5.1**
**How to Read Individual LED Indicators**



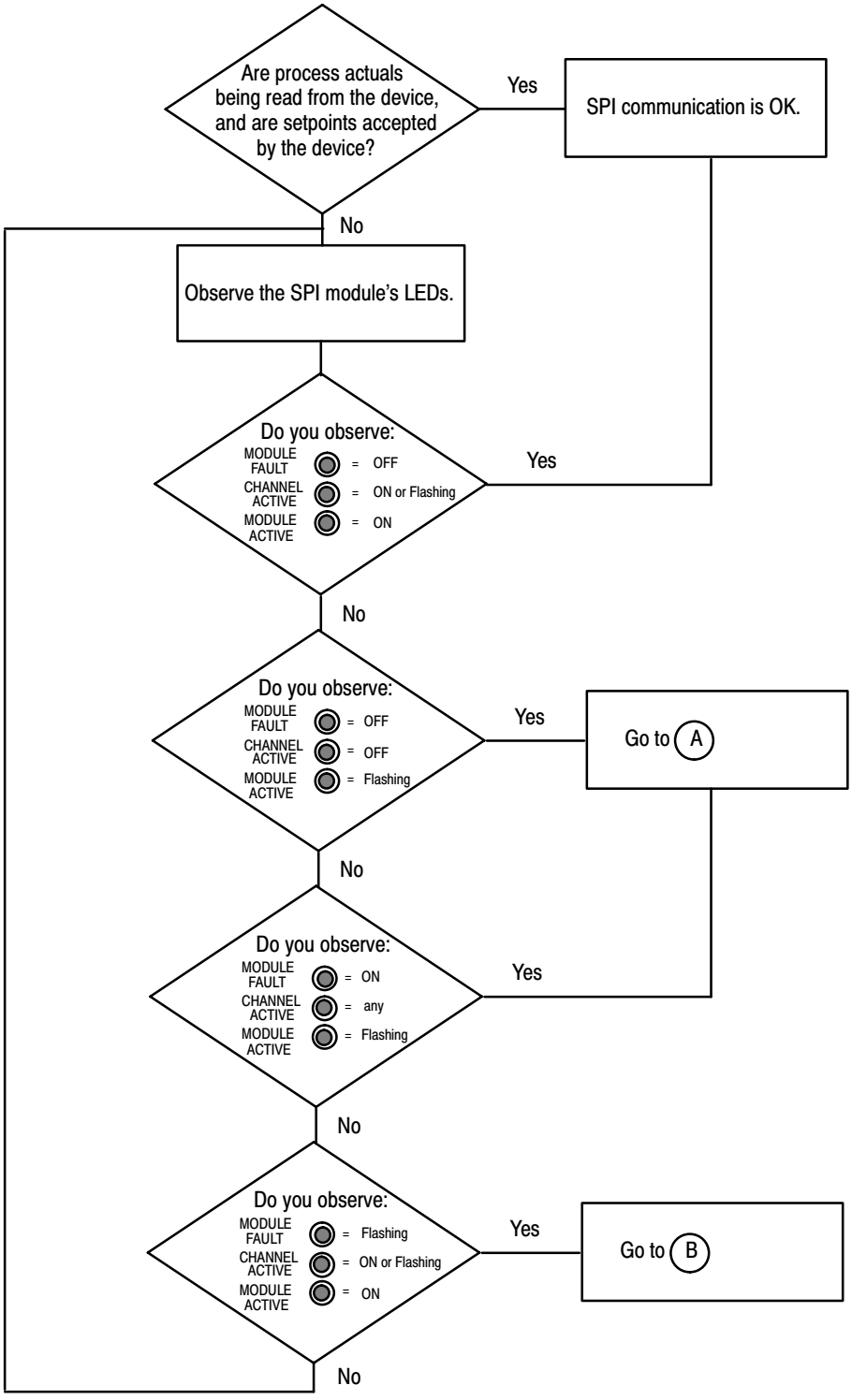

| When this LED: | Is: | The SPI Module Has Detected the Following: | Take this Corrective Action: |
|---|---|---|---|
| MODULE FAULT | Flashing | SPI module is not receiving from a device | Refer to 8000 series error codes in CDS. |
| | ON | an invalid MCC | Refer to 9000 series error codes in SYS. |
| | | module failure | Replace SPI module. |
| | OFF | none of the above | See other LEDs. |
| CHANNEL ACTIVE | ON or Flash | SPI module is sending data on the SPI network | n/a |
| | OFF | SPI module is not sending data on the network | See other LEDs. |
| MODULE ACTIVE | Flashing | SPI module is powered but not configured | Download a valid MCC to SPI module. |
| | ON | SPI module is configured | n/a |
| | OFF | SPI module is not powered or is bad | Apply power or replace SPI module. |

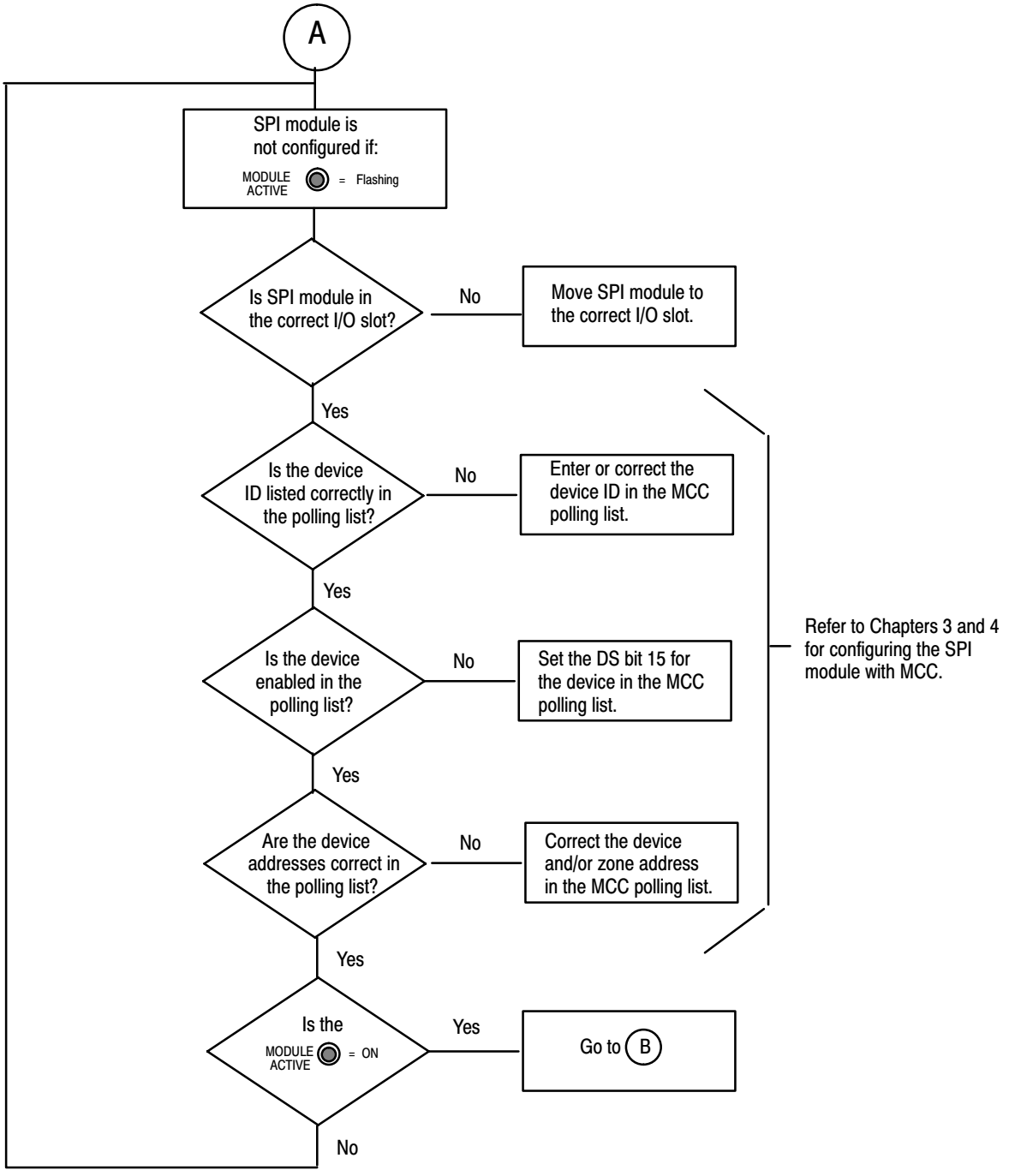
**Table 5.A**
**Troubleshooting with LED Indicators**

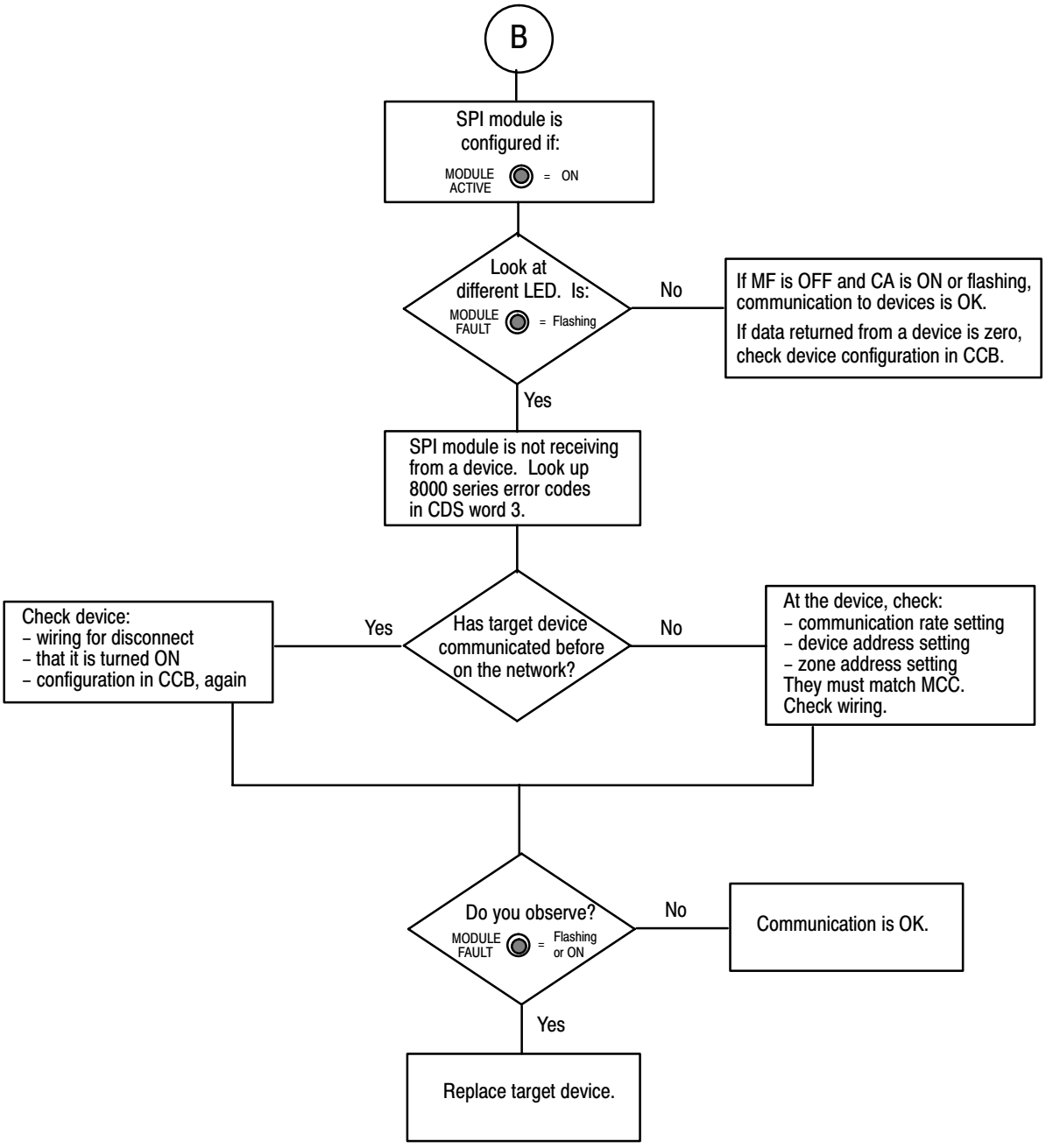
| Indicator: | Possible Cause: | Possible Correction: | See Flow Chart at: |
|---|---|---|---|
| MF = Red = OFF<br>CA = Yellow = ON or FLASH<br>MA = Green = ON | Module is operating correctly: powered and communicating on the SPI network. | | n/a |
| MF = Red = ON or OFF<br>CA = Yellow = OFF<br>MA = Green = OFF | RAM or PROM failure, or module is not powered. | Turn off power. Re-seat the module in the chassis. Re-power. If problem persists, replace SPI module. | n/a |
| MF = Red = OFF<br>CA = Yellow = OFF<br>MA = Green = FLASHING | Module is powered but not configured. | Download a valid MCC to the SPI module. | A |
| MF = Red = ON<br>CA = Yellow = any<br>MA = Green = FLASHING | Valid MCC not yet loaded into module, or valid MCC in module was corrupted. | Correct and download a valid MCC to the SPI module. | A |
| MF = Red = FLASHING<br>CA = Yellow = ON or FLASH<br>MA = Green = ON | Module is not receiving from a device. Communication OK with other devices. | Check field wiring arm connections.<br>Check D–shell connectors on primary machine.<br>Check that device settings match MCC entries. | B |

## Use this Flow Chart to Troubleshoot Your SPI Network

Locate the SPI module in the I/O chassis and observe its LED indicators. Use the following flow charts to troubleshoot the cause of the fault:




A
SPI module is not configured if:
MODULE ACTIVE = Flashing
Is SPI module in the correct I/O slot?
No
Move SPI module to the correct I/O slot.
Yes
Is the device ID listed correctly in the polling list?
No
Enter or correct the device ID in the MCC polling list.
Yes
Is the device enabled in the polling list?
No
Set the DS bit 15 for the device in the MCC polling list.
Yes
Are the device addresses correct in the polling list?
No
Correct the device and/or zone address in the MCC polling list.
Yes
Is the MODULE ACTIVE = ON
Yes
Go to B
No
Refer to Chapters 3 and 4 for configuring the SPI module with MCC.


B
SPI module is configured if:
MODULE ACTIVE = ON
Look at different LED. Is:
MODULE FAULT = Flashing
No
If MF is OFF and CA is ON or flashing, communication to devices is OK.
If data returned from a device is zero, check device configuration in CCB.
Yes
SPI module is not receiving from a device. Look up 8000 series error codes in CDS word 3.
Has target device communicated before on the network?
Yes
Check device:
– wiring for disconnect
– that it is turned ON
– configuration in CCB, again
No
At the device, check:
– communication rate setting
– device address setting
– zone address setting
They must match MCC.
Check wiring.
Do you observe?
MODULE FAULT = Flashing or ON
No
Communication is OK.
Yes
Replace target device.

## Using Diagnostic Error Codes

When the SPI module detects an error, it indicates the source of the error in word 3 returned in the SYS, CCS, or CDS status block.

Error codes are 4–digit hexadecimal values 0001-FFFF, grouped as follows:

| Error Code: | Source: | Returned in: | Description: | Refer to: |
|---|---|---|---|---|
| 9000 series | MCC | SYS | MCC Data Entry Errors | |
| 5000 series | CCB | CCS | CCS Data Entry Errors | |
| 7000 series | CDB | CDS | Command Not Executed/Supported | |
| 8000 series | | | Communication Network | |

## 9000 Series Error Codes for the MCC (Returned in SYS)

When you attempt to download the MCC to the SPI module, it returns SYS with an error code each time it detects a data entry error.

We list 9000 series error codes (Table 5.B) and the corresponding MCC bit/word map to show the data fields where your entered data was detected as invalid. Locate the circled number in the MCC bit/word map corresponding to the 9000 series error code.

**Table 5.B**
**9000 Series Error Codes for the MCC (Returned in SYS)**

| Error Code: | Your entry in the MCC was invalid because you entered: | In this Entry Field: | In MCC at: |
|---|---|---|---|
| 9000 | a page number greater than 9 | Page Number | (1) |
| 9001 | a page number greater than your total page number entry | | |
| 9002 | an A-F hex character, but the format (MCC word 1, bit 8) specified decimal | Device Address | (2) |
| 9003 | a device address less than 32 for the data format selected in word 1, bit 8 | | |
| 9004 | a device ID less than 32 for the data format selected in word 1, bit 8 | Device ID | (3) |
| 9005 | more than one device ID for the same device address (can have only one) | (duplicate address) | (2) (3) |
| 9006 | an ID for a device not supported by the SPI module | Device ID | (3) |
| 9007 | a page number, programmed or downloaded out of sequence | Page Number | (1) |
| 9008 | a total page number greater than 9 | Total Pages | (4) |
| 9009 | a communication rate that does not match allowed selections | Baud Select | (5) |
| 900A | a device ID that does not match device IDs stored in custom configuration | Device ID | (3) |
| 900B | a zone address less then 31 for the data format selected in word 1, bit 8 | Zone Address | (6) |

Locate the circled number in the MCC bit/word map

**MCC Bit/Word Map**

| Word | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | CNE | Baud Select (5) | | | CNS | NF | | DF | MCC Block ID | | | | | | | |
| 2 | 0 | | | | 0 | | | | Page Number (1) | | | | Total Pages (4) | | | |
| 3 and 4 | 0 | | | | 0 | | | | 0 | | | | 0 | | | |
| Error codes may refer to any of the following word pairs (5&6, 7&8, etc.) which define SPI network devices, listed in BTR queue order. | | | | | | | | | | | | | | | | |
| n | DS | CS | PM | Reserved | | | | | Device ID (for first queued BTR) (3) | | | | | | | |
| n+1 | Zone Address (for temperature control devices, only) (6) | | | | | | | | Device Address (2) | | | | | | | |

## 5000 Series Error Codes for the CCB (Returned in CCS)

When you attempt to download each CCB to the SPI module to configure the module for each device on the SPI network, the module returns CCS with an error code each time it detects a data entry error.

We list 5000 series error codes (Table 5.C) and the corresponding CCB bit/word map to show the data fields where your entered data was detected as invalid. Locate the circled number in the CCB bit/word map corresponding to the 5000 series error code.

**Table 5.C**
**5000 Series Error Codes for Custom Configuration CCB (Returned in CCS)**

| Error Code | Your entry in the CCB was invalid because you entered: | In This Entry Field: | in CCB at |
|---|---|---|---|
| 5000 | an A-F hex character, but the format (MCC word 1, bit 8) specified decimal | Words 2-4 | (1) |
| 5001 | a number of command and status words greater than 60, in words 2 thru 4 | | |
| 5002 | an ID that is not stored in custom configuration | Device ID | (2) |
| 5003 | an ID that matches an ID stored in custom configuration (can have only one of a kind at a time) | | |
| 5004 | you exceeded the SPI module's storage limit for custom configuration | n/a | n/a |
| 5005 | too many bit-command words, numeric presets, and/or ASCII strings, the total of which exceeded the maximum for one CCB page | Words 2-4, and Starting at Word 5 | (1) (3) |
| 5006 | a device ID less than 32 for the data format selected in MCC word 1, bit 8 | Device ID | (2) |

Locate the circled number in the CCB bit/word map

### CCB Bit/Word Map

| Word | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | | | | CA | DC | | | Device ID (2) | | | | | | | |
| 2 | (3) Number of numeric presets (1) | | | | | | | | (3) Number of bit-command words (1) | | | | | | | |
| 3 | (3) Number of bit-status words (1) | | | | | | | | (3) Number of ASCII strings (1) | | | | | | | |
| 4 | (3) Number of ASCII status strings (1) | | | | | | | | (3) Number of numeric status words (1) | | | | | | | |
| starting at 5 (3) | CMD1 Code (such as 20) | | | | | | | | CMD2 Code (such as 49) | | | | | | | |

(3) SPI codes CDM1 and CDM2 define the specific bit-command word that your program will load into word 5 *of the CDB.*
If you have more codes of this type, load them in the next consecutive words. Leave no blank words.
The number of bit-command words that you entered in word 2 bits 0-7 must be identical to the number of:
# code words (such as word 5 of this block)
# bit-command words that your program will load *into the CDB* starting with word 5.

## Error Codes Returned in CDS

Error codes returned in CDS deal with communication with devices on the SPI network, and fall into two categories:

- commands not supported or not executed by the device (return of 7000 series codes)
- communication network error (return of 8000 series codes)

### 7000 Series Error Codes for Commands Not Supported or Not Executed

A device on the SPI link can detect either of these conditions and indicate them to the SPI module. In response, the SPI module can return a 7000 code to the processor if you set corresponding enable bits in MCC word 1:

| Bit # | Title | SPI Module Reports |
|---|---|---|
| 11 | Cmd Not Supported (CNS) | 0 = Command Not Supported errors from all devices<br>1 = errors are not reported |
| 15 | Cmd Not Executed (CNE) | 0 = Command Not Executed errors from all devices<br>1 = errors are not reported |

Each error code is defined as listed in Table 5.D.

**Table 5.D**
**7000 Series Error Codes for Commands Not Supported or Not Executed**

This table describes errors in data coming from the device to the SPI module.

| Code | Type of Error | The SPI module returns this bit set = 1 when it detects: |
|---|---|---|
| 7007 | invalid data | out-of-range data from the device (SPI module discards it.) |
| 7006 | device not ready | device temporarily cannot execute the command. Try later. |
| 7005 | reserved for future use | always set = 1 |
| 7004 | | always returned reset = 0 |
| 7003 | command not supported | that the device on the link cannot support the command |
| 7002 | command not executed | that the device on the link cannot execute the supported command (typically returned with another error bit) |
| 7001 | invalid header | an error in the header from the device (may be due to noise) |
| 7000 | communication error | a framing error, a CRC error, or a protocol violation |

### 8000 Series Communication-network Error Codes

The SPI module returns these protocol-related error codes (Table 5.E) in the CDS when it detects network-related errors.

**Table 5.E**
**Communication-network Error Codes**

| Error Code | Type of Error | The SPI module detected: |
|---|---|---|
| 8000 | no response, timed-out | that the device did not respond to the command within the 1 sec time limit |
| 8001 | no status, timed-out | that the device accepted the command but did not return status within the 0.1 sec time limit |
| 8002 | numeric over-range | a numeric command value in CDB exceeded 32,767 decimal |
| 8003 | buffer over-run | the command was too large and overran the buffer |
| 8004 | communication | an unspecified communication error on the SPI link |
| 8005 | checksum | a checksum error in status received from a device |
| 8006 | device termination | early termination of status received from the device |
| 8007 | device address | an A-F hex character, but the format (MCC word 1, bit 8) specified decimal |
| 8008 | numeric over-range | a numeric status value from a device exceeded 32767 decimal |
| 8009 | illegal address/ID | the device/ID address of the CDB did not match any of the device/ID addresses stored in the SPI module's configuration |
| 800A | numeric word | an A-F hex character, but the format (MCC word 1, bit 8) specified decimal |
| 800B | illegal ID | the ID of a CDB is not currently stored in custom configuration |
| 800C | | the ID of a CDB is not a SPI-specified ID |
| 800D | link time-out | there has been no communication on the link within the 1.2 sec time limit |

## Notes

## Customer Support

If you need additional assistance on using your software, Allen-Bradley offers telephone and on-site product support at Customer Support Centers worldwide.

For technical assistance on the telephone, first contact your local sales office, distributor, or system integrator. If additional assistance is needed, then contact your local Customer Support Center or contact System Support Services.

### In the United States and Canada

If you have a SupportPlus agreement or your software is under warranty, you can contact System Support Services at: 1-800-289-2279. Have your support contract or software registration number available.

For assistance that requires on-site support, contact your local sales office, distributor, or system integrator. During non-office hours, contact the Allen-Bradley 24-hour Hot Line at 1-800-422-4913.

Rockwell Automation
Allen-Bradley

Allen-Bradley, a Rockwell Automation Business, has been helping its customers improve productivity and quality for more than 90 years. We design, manufacture and support a broad range of automation products worldwide. They include logic processors, power and motion control devices, operator interfaces, sensors and a variety of software. Rockwell is one of the world's leading technology companies.

## Worldwide representation.

Argentina • Australia • Austria • Bahrain • Belgium • Brazil • Bulgaria • Canada • Chile • China, PRC • Colombia • Costa Rica • Croatia • Cyprus • Czech Republic • Denmark • Ecuador • Egypt • El Salvador • Finland • France • Germany • Greece • Guatemala • Honduras • Hong Kong • Hungary • Iceland • India • Indonesia • Ireland • Israel • Italy • Jamaica • Japan • Jordan • Korea • Kuwait • Lebanon • Malaysia • Mexico • Netherlands • New Zealand • Norway • Pakistan • Peru • Philippines • Poland • Portugal • Puerto Rico • Qatar • Romania • Russia–CIS • Saudi Arabia • Singapore • Slovakia • Slovenia • South Africa, Republic • Spain • Sweden • Switzerland • Taiwan • Thailand • Turkey • United Arab Emirates • United Kingdom • United States • Uruguay • Venezuela • Yugoslavia

Allen-Bradley Headquarters, 1201 South Second Street, Milwaukee, WI 53204 USA, Tel: (1) 414 382-2000 Fax: (1) 414 382-4444

Publication 1771-6.5.122 – November 1995

PN 955122-23