

CLS200 Communications Specification

Includes CLS200, MLS300 and CAS200



1241 Bundy Boulevard, Winona, Minnesota USA 55987
Phone: +1 (507) 454-5300, Fax: +1 (507) 452-4507 <http://www.watlow.com>

Table of Contents

| | |
|--|-----------|
| Overview | 3 |
| Chapter 1: Anafaze/AB Protocol | 3 |
| Anafaze/AB Protocol Basics | 3 |
| Protocol Syntax..... | 3 |
| ErrorChecking..... | 7 |
| Examples | 8 |
| MessageData | 12 |
| Anafaze/AB Data Table Summary | 13 |
| Chapter 2: Modbus® RTU Protocol..... | 17 |
| Overview | 17 |
| Examples | 25 |
| Modbus® RTU Data Table Summary..... | 27 |
| Chapter 3: Controller Parameter Descriptions..... | 31 |
| Correlating Menu Items withParameters | 31 |
| Parameters (by number) | 35 |
| Glossary | 62 |

Overview

This reference guide is designed to help applications software programmers interface with Watlow® CLS200 and MLS300 controllers and CAS200 alarm scanners via serial communication.

The following chapters are included in this guide:

- *Anafaze/AB Protocol*—gives an overview and explanation of the Anafaze/Allen Bradley communications protocol
- *Modbus®-RTU Protocol*—gives an overview and explanation of the Modbus®-RTU communications protocol
- *Data Table Summary*—provides standard controller data table maps for the parameters (one for each protocol)
- *Parameters Description*—describes each parameter
- *Glossary*—explanations of commonly used terms and acronyms

1

Chapter 1: Anafaze/AB Protocol

This chapter explains the ANAFAZE/Allen Bradley protocol.

Anafaze/AB Protocol Basics

This protocol is used with a serial communications link (EIA/TIA-232 or EIA-TIA-485) configured as follows:

- 2400 or 9600 baud
- 8 data bits
- One or 2 stop bits
- No parity

Protocol Syntax

The controllers use a half-duplex (master-slave) protocol to interface to high-level software. The host software is considered the “master” and the controller is considered the “slave.” In other words, the software can request information from the controller or download information to the controller. The controller can only respond to communications transactions initiated by the host software. The controller cannot initiate communications.

Control Codes

The controller and host software communicate by sending and receiving information in a “packet” format. A packet consists of a sequence of bytes in a specific format; it can be as large as 256 bytes of data. (For more information about packets, see the Packet Format section later in this chapter.)

The numbers in the packet are sent in binary format. However, our examples show bytes in hexadecimal format.

Control code abbreviations in this manual

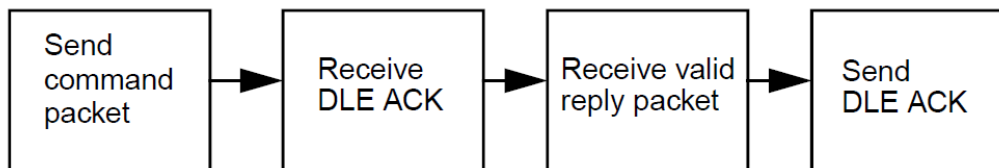
| Code | Meaning | Decimal Value | Hex Value |
|------|---|---------------|-----------|
| DLE | Escape code Signals the start of the other control code character sequences. | 16 | 10 |
| STX | Start Text Begins a transmission. | 02 | 02 |
| ETX | End Text Ends a transmission. | 03 | 03 |
| ENQ | Request Resend Tells the controller to resend its last ACK or NAK. Host software sends this command, and the controller responds to it. | 05 | 05 |
| ACK | Acknowledged Signals that a syntactically correct packet has been received. | 06 | 06 |
| NAK | Not Acknowledged Signals that an incorrect, invalid packet has been received. | 21 | 15 |

Transaction Sequence

Here are the four steps in a transaction between the host software and the controller. The following example shows the transaction as an exchange of packets. The example also assumes that there are no communication errors in the exchange.

1. The host software sends a packet that contains a read command or write command.
2. The controller sends a DLE ACK to the host software.
3. The host software receives a reply packet from the controller.
4. The host software sends a DLE ACK.

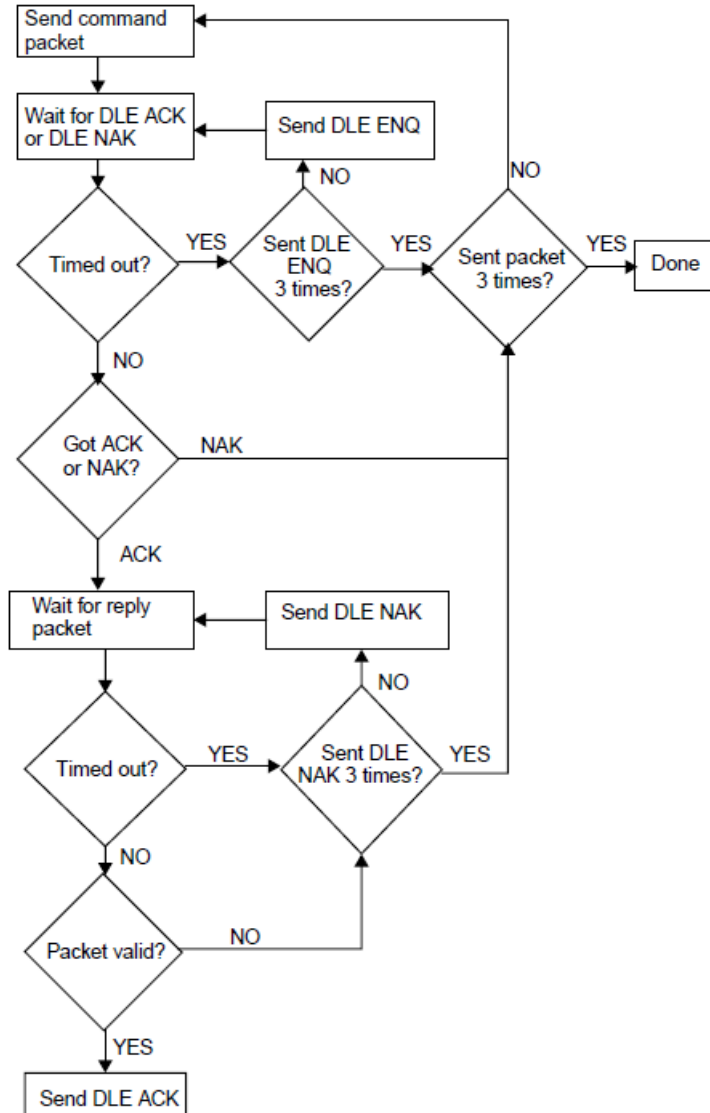
Transaction flow with no error handling



NOTE Due to the difference between the processing speeds of the controller and PC, it may be necessary to delay the computer's acknowledgement (ACK) in order for the controller to receive it. A delay of 200ms should suffice.

The flowchart below shows one way for the host software to handle error checking. If you are writing simple software, you don't necessarily need to use error handling routines as complete as these.

Transaction flow with error handling



Packet Format

Messages are transmitted in the form of packets. Command and reply packets specify the source and destination addresses, whether to read or write, the block of data to read or write, etc.

A packet contains a sequence of binary bytes formatted this way:

Sending Control Codes

To send a control code, send a DLE before the control code to distinguish it from data.

Sending a DLE as Data

When you send a byte with 0x10, (a DLE), the controller and software interpret it as a command. Therefore, to send a DLE as data, send another DLE immediately before it (DLE DLE).

Codes in a Packet

This section describes the sequence of bytes in a packet, in the order the host software or controller sends them.

DLE STX (byte) signals the beginning of a transmission. Every packet of information starts with the control codes DLE STX.

DST (byte) the address of the destination device (usually a controller; the first CLS200 controller is at 0x08).

NOTE: When host software communicates with a CLS200 controller via the ANAFAZE or AB protocol, it does not send the controller's actual address. Since the protocol reserves device addresses 0 to 7, the host software sends the value (controller address + 7), instead of the actual device address.

SRC (byte) the device address of the packet's source. The host software is usually designated address 0x00.

CMD (byte) indicates the command that the host software sends to the controller. The software sends a read (0x01) or write (0x08). When the controller replies, it returns the read or write command with the 7th bit set—in other words, it sends 0x41 or 0x48.

STS (byte) indicates the controller's general status and error flags to the host software. The controller ignores the status byte in the host software's command packet. The table below lists status byte values and definitions.

An “n” in the status bytes below indicates that the associated nibble may contain additional information. In most cases, the status byte is composed of two independent nibbles. Each nibble is independent so that two codes can return at once. For example, status code F1 indicates that data has changed (Fn) and the controller is being updated through the front panel (0x1).

| Status in Hex | Description |
|---------------|---|
| 00 | The controller has nothing to report, or AB protocol is selected. |
| 01 | Access denied for editing. The controller is being updated through the front panel. |
| 02 | AIM Comm failure. |
| A0 | A controller reset occurred. |
| Cn | The controller received a command that was not a block read or block write. (Command Error) |
| Dn | The block write command attempted to write beyond a particular parameter block boundary, or the host software attempted to access a data table block that does not exist. (Data Boundary Error) |
| En | The Alarm_Status variable has changed. The software should query the alarm status block to determine the particular alarm flag that changed. |
| Fn | The controller altered shared data, either internally (from the firmware) or externally (from the keyboard). The host software should read the Data Changed Register to determine which data has been altered and update its own run-time memory. |

TNSL least significant byte of the transaction number. This is the first half of a “message stamp.” The controller sends back the TNSL and TNSH exactly as it received them, so host software can use the TNSL and TNSH bytes to keep track of message packets.

TNSH most significant byte of the transaction number. This is the second half of the “message stamp.”

ADDL the low byte of the beginning data table address of the block of data to read or write.

ADDH the high byte of the beginning data table address of the block of data to read or write.

DATA the new values to be set with a write command, or the requested data in a response to a read command.

DLE ETX ends every packet of information. Signals the end of a transmission.

BCC or **CRC** one or two-byte error check at the end of the packet. There are two error check methods: Block Check Character (BCC), which requires 1 byte, and Cyclic Redundancy Check (CRC), which requires 2 bytes.

Error Checking

The default error check method, BCC is easier to implement than CRC, and is acceptable for most applications.

Select one error check method and configure both software and controller for that method, or they will be unable to communicate.

The error check methods work this way:

Block Check Character (BCC)

BCC checks the accuracy of each message packet transmission. It provides a medium level of security. The BCC is the 2's complement of the 8-bit sum (modulo-256 arithmetic sum) of the data bytes between the DLE STX and the DLE ETX. (1's complement +1)

- BCC does not detect transposed bytes in a packet.
- BCC cannot detect inserted or deleted 0 values in a packet.
- If you have sent 0x10 as data (by sending DLE 0x10) only one of the DLE data bytes is included in the BCC's sum (the DLE = 0x10 also).

For instance, the block read example shown in the examples section, adds 0x08 00 01 00 00 80 02 10. Note that the 0x10 representing DLE has been left out of the calculation. The sum should come to 0x9B.

$$1001\ 1011 = 0x9B$$

$$0110\ 0100 = 1's\ complement$$

$$\underline{\hspace{2cm}} + 1 = 2's\ complement$$

$$0110\ 0101 = 0x65$$

Cyclic Redundancy Check (CRC)

CRC is a more secure error check method than BCC. It provides a very high level of data security. It can detect:

- All single-bit and double-bit errors.

- All errors of odd numbers of bits.
- All burst errors of 16 bits or less.
- 99.997% of 17-bit error bursts.
- 99.998% of 18-bit and larger error bursts.

The CRC is calculated using the value of the data bytes and the ETX byte. At the start of each message packet, the transmitter must clear a 16-bit CRC register.

When a byte is transmitted, it is exclusive-ORed with the right 8 bits of the CRC register and the result is transferred to the right 8 bits of the CRC register. The CRC register is then shifted right 8 times by inserting 0's on the left.

Each time a 1 is shifted out on the right, the CRC register is Exclusive-ORed with the constant value 0xA001. After the ETX value is transmitted, the CRC value is sent, least significant byte (LSB) first.

Structured English procedure from AB Manual

```

data_byte = all application layer data, ETX
CLEAR CRC_REGISTER

FOR each data_byte
    GET data_byte
    XOR (data_byte, right eight bits of CRC_REGISTER)
    PLACE RESULT in right eight bits of CRC_REGISTER

    DO 8 times
        Shift bit right, shift in 0 at left
        IF bit shifted =1
            XOR (CONSTANT, CRC_REGISTER)
            PLACE RESULT in CRC_REGISTER
        END IF
    END DO
END FOR
TRANSMIT CRC_REGISTER as 2-byte CRC field

```

Examples

The host software sends two kinds of commands: block reads and block writes. This section shows examples of both commands.

Note: If you read data from a loop set to SKIP, the controller will send an empty packet for that loop.

This section does not show how to calculate the error check value included with every packet. For help calculating the error check value, see the section on BCC or CRC.

Block Read

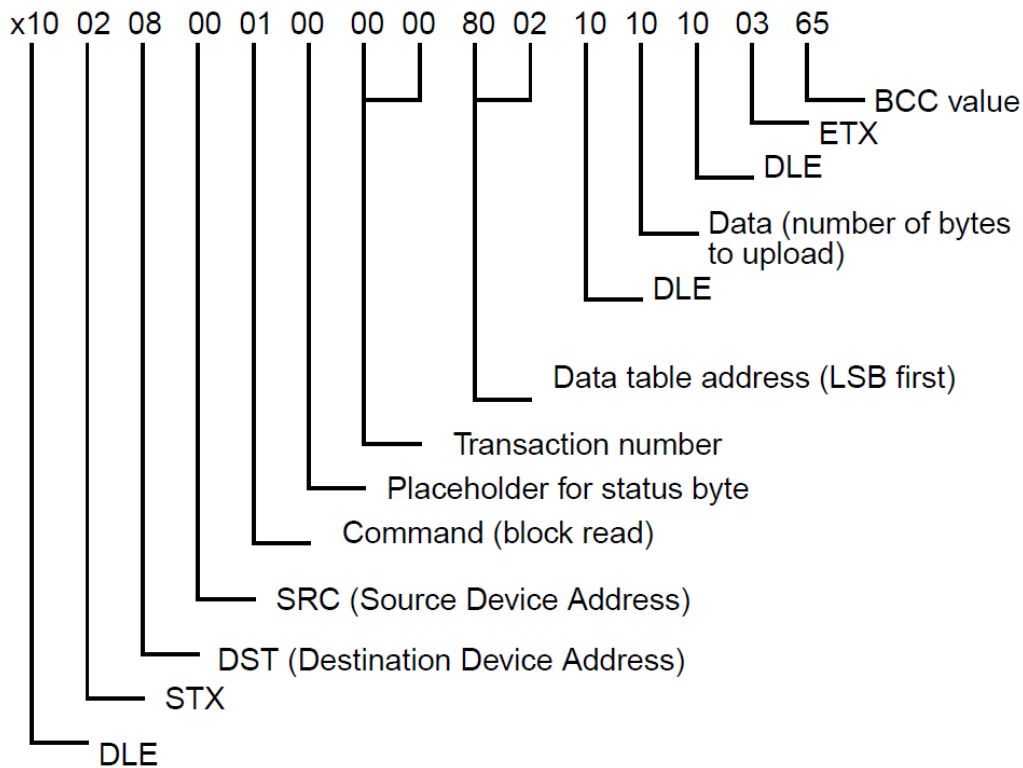
This example shows the block read command the host software sends, the controller's responses, and the software's acknowledgment.

Situation: Read process variables for loops 1 to 8.

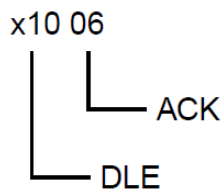
- 8 process variables 2 bytes each = 16 bytes from data table address 0x0280.

- Character values are represented in hex.
- The sender is device address 0.
- The destination is device address 0x8 (controller address 1).
- The software sends transaction number 00.

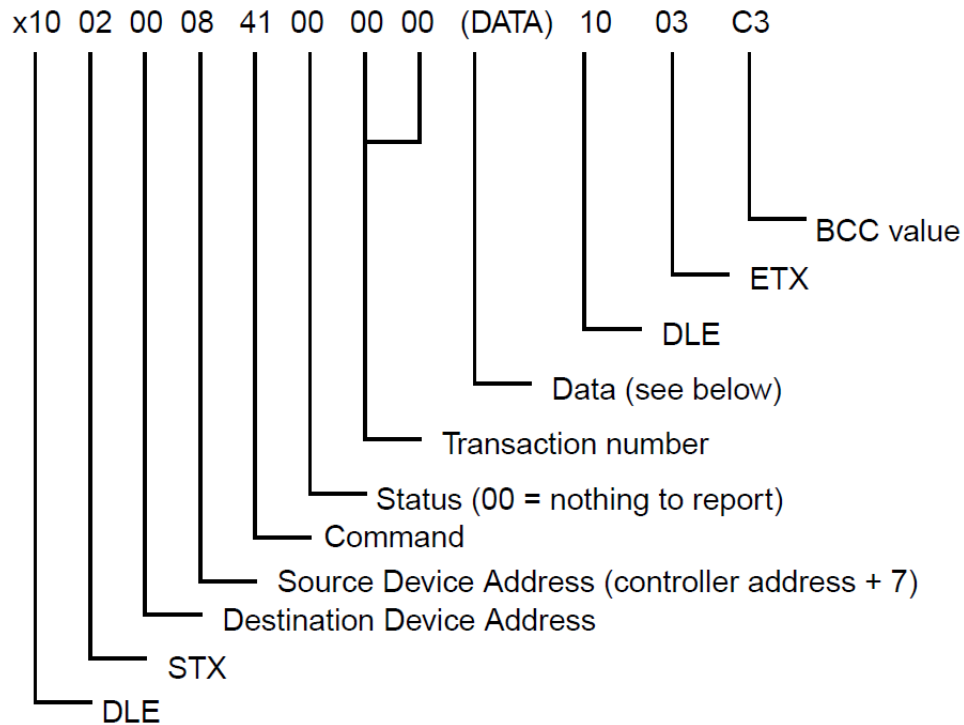
Structure of the read command



Controller acknowledges the message

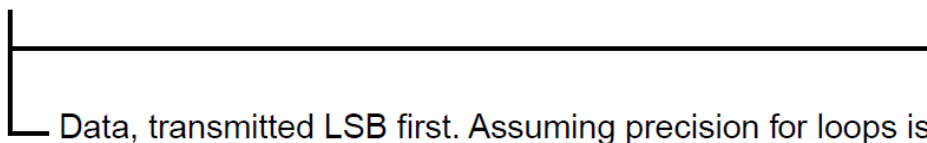


Controller reply



DATA

xE2 01 09 02 E4 01 09 02 F1 01 DF 01 28 3C E4 01



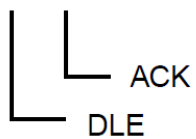
Data, transmitted LSB first. Assuming precision for loops is -1:

Loop 1 PV = x01E2 = 482, displayed as 48

Loop 2 PV = x0209 = 521, displayed as 52, etc.

Host software acknowledges the reply

x10 06



Block Write

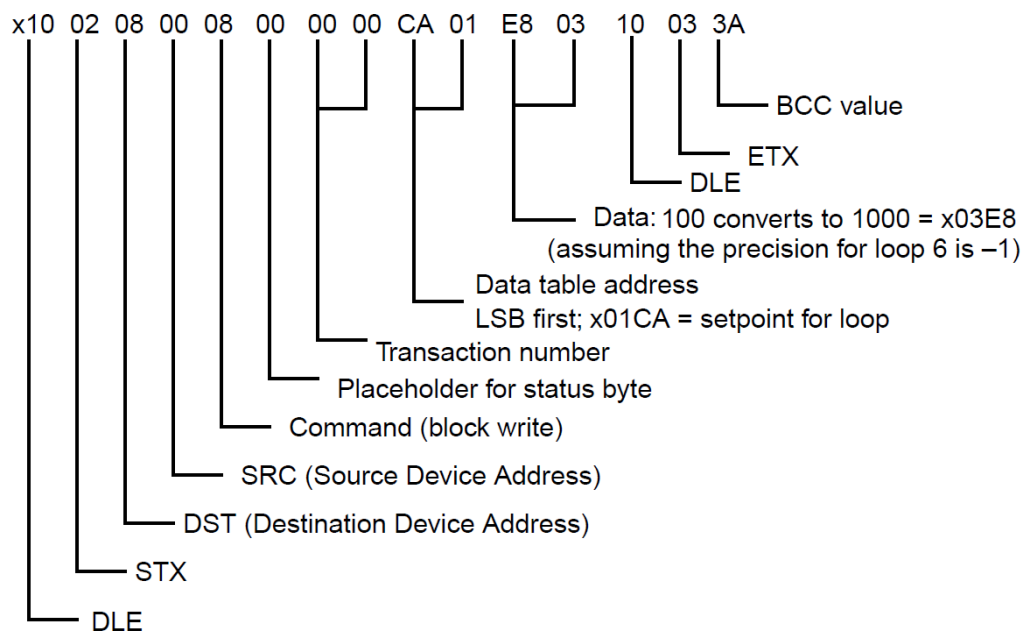
This section describes the block write command.

This example shows the block write command the master sends, the controller's responses, and the master's acknowledgment:

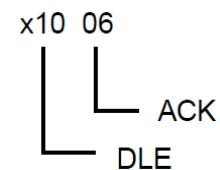
Situation: Write setpoint of 100 to loop 6.

- 1 setpoint 2 bytes per setpoint = 2 bytes to address 0x01CA (0x01C0 + xA, a 10-byte offset).
- Character values are represented in hexadecimal.
- The sender is device address 0.
- The destination is device address 8 (controller address 1).
- The software sends transaction number 00.

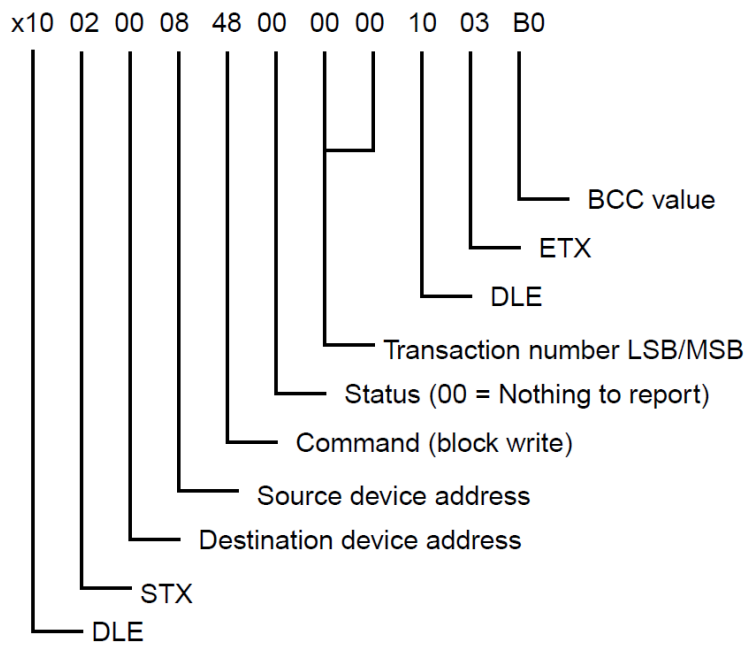
Structure of the write command



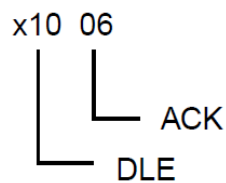
Controller acknowledges the message



Controller reply



Host software acknowledges the reply



Message Data

Some messages contain data. What the data is and how much depends on the command used and the purpose of the message.

Data for a Read Command

For a block read command, the data block consists of one byte that indicates the number of bytes to read (up to 244 bytes of data). The controller sends back a packet with a data block that contains the requested bytes.

Data for a Write Command

For a block write command, the block contains the bytes to write (up to 242 bytes of data). The controller sends back a message packet without data.

Two-Byte Data Types

For two-byte data types, like process variable and setpoint, the controller or host software sends the data in two-byte pairs with the least significant byte first.

Figuring Block Size

To read parameter values, you must know how many bytes to request. Parameter values are stored contiguously such that the setpoints for all the loops are stored together and in loop number order. For

example, to read the deviation alarm deadband value for loops one to five, you would read five bytes starting at 0x05A0. Some parameters, such as setpoint, require two bytes of memory to store. So, for example, if you want to read the setpoint for four loops, you must read eight bytes.

Calculate total block size in bytes for most loop parameters this way (do not forget the pulse loop):

$$(\text{Data Size}) * (\text{Number of Loops})$$

Some parameters have values for both heat and cool. Calculate block size for such a parameter this way:

$$2 * (\text{Data Size}) * (\text{Number of Loops})$$

One exception is the units for each loop. Calculate the data size for the units this way:

$$3 * (\text{Number of Loops})$$

Parameters that are not loop parameters (like system status, digital inputs, or digital outputs) have specific data sizes. These data sizes are listed in the data table in the next section.

Anafaze/AB Data Table Summary

Each address holds one byte of data. Each parameter value requires one or two addresses to store depending on the type of data. The table below indicates the number of bytes for each data type. The data type for each parameter is indicated in the tables on the following pages.

| Data Type and Symbol | Data Size |
|----------------------|-----------|
| Unsigned char (UC) | 1 byte |
| Signed char (SC) | 1 byte |
| Unsigned int (UI) | 2 bytes |
| Signed int (SI) | 2 bytes |

Because each loop is individually configurable, the number of instances of many parameters depends on the number of loops in the controller. Therefore, the number of bytes for these parameters is listed in the tables on the following pages in terms of the number of loops in the controller.

The storage requirements for some parameters depend on the number of digital inputs or digital outputs in the controller (MAX_DIGIN_BYTES and MAX_DIGOUT_BYTES). The storage of ramp-soak profile parameters depend on the number of profiles (MAX_RSP), the number of segments per profile (MAX_SEG), the number of triggers per segment (MAX_TRIG) and the number of events per segment (MAX_EVENT).

The table below shows the values for each of these factors. Use them to calculate the number of bytes for each parameter.

| | |
|--|----|
| MAX_CH: | |
| CLS204 (4 loops + 1 pulse loop) | 5 |
| CLS208 (8 loops + 1 pulse loop) | 9 |
| CLS216 / MLS316 / CAS200 (16 loops + 1 pulse loop) | 17 |
| MLS332 (32 loops + 1 pulse loop) | 33 |
| MAX_DIGIN_BYTES | 1 |
| MAX_DIGOUT_BYTES | 8 |
| MAX_RSP | 17 |
| MAX_SEG | 20 |

| | |
|-----------|---|
| MAX_TRIG | 2 |
| MAX_EVENT | 4 |

Ordering of Heat and Cool Channel Parameters

For parameters that have both heat and cool settings the heat values are stored in the first registers and the cool values are stored in the registers starting at the listed address plus MAX_CH.

Note: Data table parameters 46 to 60 and 100 are ramp-soak parameters. They are only used in controllers with the ramp-soak option. Parameters 81 to 95 are enhanced features and only available in controllers with the enhanced features option.

Ordering of Ramp-Soak Profile Parameters

Ramp-soak profile parameters are ordered first by profile, then by segment where applicable. So, for example, the first eight bytes of the Ready Events parameter are the ready segment event states for the first profile (profile A) and the next eight bytes are for profile B and so on. In the case of the segment triggers, the first byte contains the first trigger setting for the first segment of profile A, the second byte contains the settings for the second trigger for the first segment of profile A, the third byte contains the settings for the first trigger for the second segment of profile A and so on.

Anafaze/AB Protocol Data Table

| Number | Description | Address in Hex | Type | Number of Bytes |
|--------|-----------------------------|----------------|------|-----------------|
| 0 | Proportional Band/Gain | 0020 | UC | MAX_CH * 2 |
| 1 | Derivative Term | 0060 | UC | MAX_CH * 2 |
| 2 | Integral Term | 00A0 | UI | MAX_CH * 4 |
| 3 | Input Type | 0120 | UC | MAX_CH |
| 4 | Output Type | 0180 | UC | MAX_CH * 2 |
| 5 | Setpoint | 01C0 | SI | MAX_CH * 2 |
| 6 | Process Variable | 0280 | SI | MAX_CH * 2 |
| 7 | Output Filter | 0340 | UC | MAX_CH * 2 |
| 8 | Output Value | 0380 | UI | MAX_CH * 4 |
| 9 | High Process Alarm Setpoint | 0400 | SI | MAX_CH * 2 |
| 10 | Low Process Alarm Setpoint | 04C0 | SI | MAX_CH * 2 |
| 11 | Deviation Alarm Band Value | 05A0 | UC | MAX_CH |
| 12 | Alarm Deadband | 0600 | UC | MAX_CH |
| 13 | Alarm Status | 0660 | UI | MAX_CH * 2 |
| 14 | Not used | 06A0 | | 128 |
| 15 | Ambient Sensor Readings | 0720 | SI | 2 |
| 16 | Pulse Sample Time | 0730 | UC | 1 |
| 17 | High Process Variable | 0790 | SI | MAX_CH * 2 |
| 18 | Low Process Variable | 0850 | SI | MAX_CH * 2 |
| 19 | Precision | 0910 | SC | MAX_CH |
| 20 | Cycle Time | 09D0 | UC | MAX_CH * 2 |
| 21 | Zero Calibration | 0A10 | UI | 2 |
| 22 | Full-scale Calibration | 0A16 | UI | 2 |

| Number | Description | Address in Hex | Type | Number of Bytes |
|--------|------------------------------------|----------------|------|----------------------------------|
| 23 | Job Select Dig Inputs | 0A1C | UC | 1 |
| 24 | Job Sel Dig Ins Active | 0A20 | UC | 1 |
| 25 | Digital Inputs | 0A60 | UC | MAX_DIGIN_BYTES |
| 26 | Digital Outputs | 0A70 | UC | MAX_DIGOUT_BYTES |
| 27 | Reserved | 0A80 | UC | MAX_DIGOUT_BYTES |
| 28 | Override Digital Input | 0AA0 | UC | 1 |
| 29 | Override Polarity | 0AC0 | UC | 1 |
| 30 | System Status | 0AC8 | UC | 4 |
| 31 | System Command Register | 0ACC | UC | 1 |
| 32 | Data Changed Register | 0ACE | UC | 1 |
| 33 | Input Units | 0AD0 | UC | MAX_CH * 3 |
| 34 | EPROM Version Code | 0BF0 | UC | 12 |
| 35 | Options Register | 0BFC | UC | 1 |
| 36 | Process Power Digital Input | 0C00 | UC | 1 |
| 37 | High Reading | 0C60 | SI | MAX_CH * 2 |
| 38 | Low Reading | 0D20 | SI | MAX_CH * 2 |
| 39 | Heat/Cool Spread | 0DE0 | UC | MAX_CH |
| 40 | Startup Alarm Delay | 0E20 | UC | 1 |
| 41 | High Process Alarm Output Number | 0E30 | UC | MAX_CH |
| 42 | Low Process Alarm Output Number | 0E90 | UC | MAX_CH |
| 43 | High Deviation Alarm Output Number | 0EF0 | UC | MAX_CH |
| 44 | Low Deviation Alarm Output Number | 0F50 | UC | MAX_CH |
| 45 | Not used | 0F60 | | MAX_CH |
| 46 | Channel Profile and Status | 1000 | UC | MAX_CH |
| 47 | Current Segment | 1020 | UC | MAX_CH |
| 48 | Segment Time Remaining | 1040 | UI | MAX_CH * 2 |
| 49 | Current Cycle Number | 1080 | UI | MAX_CH * 2 |
| 50 | Tolerance Alarm Time | 10C0 | UI | MAX_RSP * 2 |
| 51 | Last Segment | 1100 | UC | MAX_RSP |
| 52 | Number Cycles | 1120 | UC | MAX_RSP |
| 53 | Ready Setpoint | 1140 | SI | MAX_RSP * 2 |
| 54 | Ready Event States | 1180 | UC | MAX_RSP * MAX_DIGOUT_BYTES |
| 55 | Segment Setpoint | 1280 | SI | MAX_RSP * 2 * MAX_SEG |
| 56 | Triggers and Trigger States | 1780 | UC | MAX_RSP* MAX_SEG* MAX_TRIG |
| 57 | Segment Events and Event States | 1C80 | UC | MAX_RSP * MAX_SEG * MAX_EVENT |
| 58 | Segment Time | 2680 | UI | MAX_RSP * 2 * MAX_SEG |
| 59 | Tolerance | 2B80 | SI | MAX_RSP * 2 * MAX_SEG |
| 60 | Ramp/Soak Flags | 3080 | UC | MAX_CH |
| 61 | Output Limit | 3200 | SI | MAX_CH * 4 |
| 62 | Output Limit Time | 3280 | SI | MAX_CH * 4 |

| Number | Description | Address in Hex | Type | Number of Bytes |
|--------|-------------------------------------|----------------|------|-----------------|
| 63 | Alarm_Control | 3300 | UI | MAX_CH * 2 |
| 64 | Alarm_Acknowledge | 33C0 | UI | MAX_CH * 2 |
| 65 | Alarm_Mask | 3480 | UI | MAX_CH * 2 |
| 66 | Alarm_Enable | 3540 | UI | MAX_CH * 2 |
| 67 | Output Override Percentage | 3600 | SI | MAX_CH * 4 |
| 68 | AIM Failure Output | 3690 | UC | 1 |
| 69 | Output Linearity Curve | 3700 | UC | MAX_CH * 2 |
| 70 | SDAC Mode | 3740 | UC | MAX_CH * 2 |
| 71 | SDAC Low Value | 3780 | SI | MAX_CH * 4 |
| 72 | SDAC High Value | 3800 | SI | MAX_CH * 4 |
| 73 | Save Setup to Job | 3880 | UC | 1 |
| 74 | Input Filter | 3890 | UC | MAX_CH |
| 75 | Loop Alarm Delay | 38D0 | UI | MAX_CH * 2 |
| 76 | Not used | 3990 | | 16 |
| 77 | Loop Names | 39A0 | UI | MAX_CH * 2 |
| 78 | T/C Failure Detection Flags | 3A30 | UC | MAX_CH |
| 78 | Not used | 3994 | UC | MAX_CH * 8 |
| 79 | Restore PID Digital Input | 4130 | UC | MAX_CH |
| 80 | Manufacturing Test | 4160 | UI | 1 |
| 81 | PV Retransmit Primary Loop Number | 4200 | UC | MAX_CH * 2 |
| 82 | PV Retransmit Maximum Input | 4250 | SI | MAX_CH * 4 |
| 83 | PV Retransmit Maximum Output | 42E0 | UC | MAX_CH * 2 |
| 84 | PV Retransmit Minimum Input | 4330 | SI | MAX_CH * 4 |
| 85 | PV Retransmit Minimum Output | 43C0 | UC | MAX_CH * 2 |
| 86 | Cascade Primary Loop Number | 4410 | UC | MAX_CH |
| 87 | Cascade Base Setpoint | 4440 | SI | MAX_CH * 2 |
| 88 | Cascade Minimum Setpoint | 4490 | SI | MAX_CH * 2 |
| 89 | Cascade Maximum Setpoint | 44E0 | SI | MAX_CH * 2 |
| 90 | Cascade Heat/Cool Span | 4530 | SI | MAX_CH * 4 |
| 91 | Ratio Control Master Loop Number | 45C0 | UC | MAX_CH |
| 92 | Ratio Control Minimum Setpoint | 45F0 | SI | MAX_CH * 2 |
| 93 | Ratio Control Maximum Setpoint | 4640 | SI | MAX_CH * 2 |
| 94 | Ratio Control Control Ratio | 4690 | UI | MAX_CH * 2 |
| 95 | Ratio Control Setpoint Differential | 46E0 | SI | MAX_CH * 2 |
| 96 | Loop Status | 4730 | UC | MAX_CH |
| 97 | Output Type/Disable | 4760 | UC | MAX_CH * 2 |
| 98 | Output Reverse/Direct | 47B0 | UC | MAX_CH * 2 |
| 99 | Controller Type | 47F0 | UC | 1 |
| 100 | Ramp/Soak Profile Number | 4800 | UC | MAX_CH |
| 101 | Controller Address | 4830 | UC | 1 |
| 102 | Baud Rate | 4840 | UC | 1 |

2

Chapter 2: Modbus® RTU Protocol

Overview

This chapter explains the use of the Modbus® RTU protocol in the CLS200, MLS300 and CAS200 products

Transactions on Modbus® RTU Networks

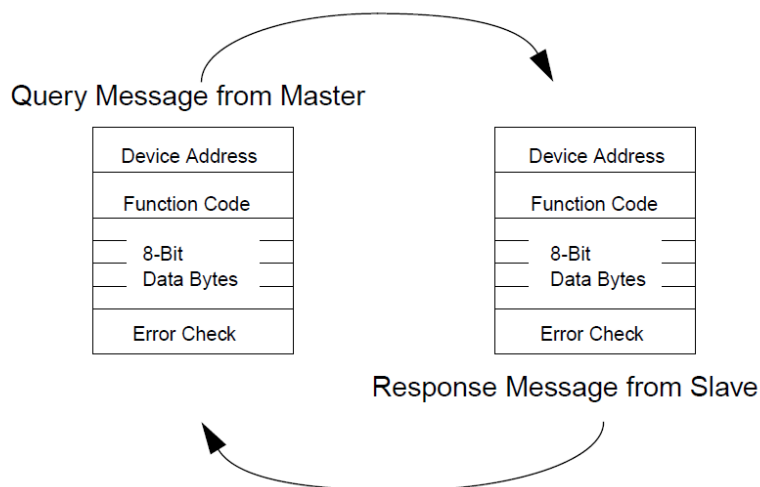
Standard Modbus® RTU ports use an EIA/TIA-232C or EIA/TIA-485/ 422-compatible serial interface that defines connector pinouts, cabling, signal levels, transmission baud rates, and parity checking.

Controllers communicate using a master-slave technique, in which only one device (the master) can initiate transactions (called “queries”). The other devices (slaves) respond by supplying the requested data to the master, or by taking the action requested in the query. Typical master devices include host processors and programming panels.

The master can address individual slaves, or initiate a broadcast message to all slaves. Slaves return a message (called a “response”) to queries that are addressed to them individually. Responses are not returned to broadcast queries from the master.

The Modbus® RTU protocol establishes the format for the master’s query by placing into it the device (or broadcast) address, a function code defining the requested action, any data to be sent, and an error-checking field. The slave’s response message is also constructed using Modbus® RTU protocol. It contains fields confirming the action taken, any data to be returned, and an error-checking field. If an error occurred in receipt of the message, or if the slave is unable to perform the requested action, the slave will construct an error message and send it as its response.

The Query-Response Cycle



The Query

The function code in the query tells the addressed slave device what kind of action to perform. The data bytes contain any additional information that the slave will need to perform the function. For example, function code 03 will query the slave to read holding registers and respond with their contents. The data field must contain the information telling the slave which register to start at and how many registers to read. The error check field provides a method for the slave to validate the integrity of the message contents.

The Response

If the slave makes a normal response, the function code in the response is an echo of the function code in the query. The data bytes contain the data collected by the slave, such as register values or status. If an error occurs, the function code is modified to indicate that the response is an error response, and the data bytes contain a code that describes the error. The error check field allows the master to confirm that the message contents are valid.

Serial Transmission

Each 8-bit byte in a message contains two 4-bit hexadecimal characters. This high character density allows better data throughput than ASCII for the same baud rate. Each message must be transmitted in a continuous stream.

Coding System

- 8-bit binary, hexadecimal 0 to 9, A to F
- 2 hexadecimal characters contained in each 8-bit field of the message

Bits per Byte

- 1 start bit
- 8 data bits, least significant bit sent first
- 2 stop bits
- No parity

Error Check Field

Cyclical Redundancy Check (CRC)

Message Framing

Messages start with a silent interval of at least 3.5 character times. This is most easily implemented as a multiple of character times at the baud rate that is being used on the network (shown as T1-T2-T3-T4 in the figure below). The first field then transmitted is the device address.

The allowable characters transmitted for all fields are hexadecimal 0 to 9, A to F. Networked devices monitor the network bus continuously, including during the silent intervals. When the first field (the address field) is received, each device decodes it to find out if it is the addressed device.

Following the last transmitted character, a similar interval of at least 3.5 character times marks the end of the message. A new message can begin after this interval.

Similarly, if a new message begins earlier than 3.5 character times following a previous message, the receiving device will consider it a continuation of the previous message. This will set an error, as the value in the final CRC field will not be valid for the combined messages.

Typical message frame

| START | ADDRESS | FUNCTION | DATA | CRC CHECK | END |
|-------------|---------|----------|--------------|-----------|-------------|
| T1-T2-T3-T4 | 8 Bits | 8 Bits | $n * 8$ Bits | 16 Bits | T1-T2-T3-T4 |

Handling the Address Field

The address field of a message frame contains 8 bits. Valid slave device addresses are in the range of 0 to 247 decimal. The individual slave devices are assigned addresses in the range of 1 to 247 decimal. A master addresses a slave by placing the slave address in the address field of the message. When the slave sends its response, it places its own address in this address field of the response to let the master know which slave is responding.

Handling the Function Field

The function code field of a message frame contains 8 bits. Valid codes are in the range of 1 to 255 decimal. Not all these codes are applicable to all controllers. Current codes are described in the *Function Codes* section.

When a message is sent from a master to a slave device, the function code field tells the slave what kind of action to perform. Examples are to read the On/Off states of a group of discrete coils or inputs; to read the data contents of a group of registers; or to read the diagnostic status of a slave.

When the slave responds to the master, it uses the function code field to indicate either a normal (error-free) response or that some kind of error occurred (called an exception response). For a normal response, the slave simply echoes the original function code. For an exception response, the slave returns a code that is equivalent to the original function code with its most significant bit set to a logic 1.

For example, a message from the master to slave to read a group of holding registers would have the following function code:

0000 0011 0x3

If the slave device takes the requested action without error, it returns the same code in its response. If an exception occurs, it returns:

1000 0011 0x83

In addition to its modification of the function code for an exception response, the slave places a unique code into the data field of the response message. This tells the master what kind of error occurred, or the reason for the exception.

The master device's application program has the responsibility of handling exception responses. Typical processes are to post subsequent retries of the message, to try diagnostic messages to the slave, and to notify operators.

Contents of the Data Field

The data field is constructed using sets of two hexadecimal numbers, in the range of x00 to xFF.

The data field of messages sent from a master to slave devices contains additional information that the slave must use to take the action defined by the function code. This can include items like discrete and register addresses, the quantity of items to be handled, and the count of actual data bytes in the field.

For example, if the master requests a slave to read a group of holding registers (function code 03), the data field specifies the starting register and how many registers are to be read.

If no error occurs, the data field of a response from a slave to a master contained the data requested. If an error occurs, the field contains an exception code that the master application can use to determine the next action to be taken.

The data field can be nonexistent (of zero length) in certain kinds of messages, where the function code alone specifies the action.

Contents of the Error Checking Field

The error checking field contains a 16-bit value implemented as two 8-bit bytes. The error check value is the result of a Cyclical Redundancy Check (CRC) calculation performed on the message contents.

The CRC field is appended to the message as the last field in the message. When this is done, the low-order byte of the field is appended first, followed by the high-order byte. The CRC high-order byte is the last byte to be sent in the message.

How Characters are Transmitted Serially

When messages are transmitted on standard Modbus® RTU serial networks, each character or byte is sent in this order (left to right):

Least Significant Bit (LSB).....Most Significant Bit (MSB)

The bit sequence

| | | | | | | | | | | |
|-------|---|---|---|---|---|---|---|---|------|------|
| Start | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Stop | Stop |
|-------|---|---|---|---|---|---|---|---|------|------|

CRC Error Checking

All messages include an error-checking field that is based on a Cyclical Redundancy Check (CRC) method. The CRC field checks the contents of the entire message. It is applied regardless of any parity check method used for the individual characters of the message.

The CRC field is two bytes, containing a 16-bit binary value. The CRC value is calculated by the transmitting device, which appends the CRC to the message. The receiving device recalculates a CRC during receipt of the message and compares the calculated value to the actual value it received in the CRC field. If the two values are not equal, an error results.

The CRC is started by first preloading a 16-bit register to all 1's. Then a process begins of applying successive 8-bit bytes of the message to the current contents of the register. Only the 8 bits of data in each character are used for generating the CRC. Start and stop bits do not apply to the CRC.

During generation of the CRC, each 8-bit character is exclusive ORed with the register contents. Then the result is shifted in the direction of the least significant bit (LSB), with a zero filled into the most

significant bit (MSB) position. The LSB is extracted and examined. If the LSB was a 1, the register is then exclusive ORed with a preset, fixed xA001. If the LSB was a 0, no exclusive OR takes place.

This process is repeated until eight shifts have been performed. After the last shift, the next 8-bit byte is exclusive ORed with the register's current value, and the process repeats for eight more shifts as described above. The final contents of the register, after all the bytes of the message have been applied, is the CRC value.

Function Codes

The listing below shows the function codes supported by the CLS200 controllers. Codes are listed in decimal.

| Code in Hex | Name |
|-------------|---------------------------|
| 01 | Read Coil Status |
| 02 | Read Input Status |
| 03 | Read Holding Registers |
| 04 | Read Input Registers |
| 05 | Force Single Coil |
| 06 | Preset Single Register |
| 08 | Diagnostics |
| 0F | Force Multiple Coils |
| 10 | Preset Multiple Registers |

0x01 Read Coil Status

Reads the On/Off status of discrete outputs (registers 00001 to 09999, the coils) in the slave. Broadcast is not supported.

0x02 Read Input Status

Reads the On/Off status of discrete inputs (registers 10001 to 19999) in the slave. Broadcast is not supported.

0x03 Read Holding Registers

Reads the binary contents of holding registers (registers 40001 to 49999) in the slave. Broadcast is not supported.

0x04 Read Input Registers

Reads the binary contents of input registers (registers 30001 to 39999) in the slave. Broadcast is not supported.

0x05 Force Single Coil

forces a single coil (registers 00001 to 09999, the coils) to either On or Off. When broadcast, the function forces the same coil reference in all attached slaves. This function overrides the controller's memory protect state and the coil's disable state. The forced state will remain valid until the controller's logic next solves the coil. The coil will remain forced if it is not programmed in the controller's logic.

0x06 Preset Single Register

presets a value into a single holding register (registers 40001 to 49999). When broadcast, the function presets the same register reference in all attached slaves. The function overrides the controller's memory protect state. The preset value will remain valid in the register until the controller's logic next solves the register contents. The register's value will remain if it is not programmed in the controller's logic.

0x08 Diagnostics

provides a series of tests for checking the communication system between the master and slave, or for checking various internal error conditions within the slave. Broadcast is not supported. This function uses a 2-byte subfunction code field in the query to define the type of test to be performed. The slave echoes both the function code and subfunction code in a normal response. Most of the diagnostic queries use a 2-byte data field to send diagnostic data or controller information to the slave. Some of the diagnostics cause data to be returned from the slave in a data field of a normal response.

Diagnostic Subfunctions

0x00 Return Query Data

The data passed in the query data field is to be returned (looped back) in the response. The entire response message should be identical to the query.

| Subfunction | Data Field (Query) | Data Field (Response) |
|-------------|--------------------|-----------------------|
| 0x00 00 | Any | Echo Query Data |

0x01 Restart Communications Option

The slave's peripheral port is to be initialized and restarted, and all of its communications event counters are to be cleared. If the port is currently in Listen Only Mode, no response is returned. This function is the only one that brings the port out of Listen Only Mode. If the port is not currently in Listen Only Mode, a normal response is returned. This occurs before the restart is executed.

| Subfunction | Data Field (Query) | Data Field (Response) |
|-------------|--------------------|-----------------------|
| 0x00 01 | 0x00 00 | Echo Query Data |
| 0x00 01 | 0xFF 00 | Echo Query Data |

0x02 Return Diagnostic Register

The contents of the slave's 16-bit diagnostic register are returned in the response.

| Subfunction | Data Field (Query) | Data Field (Response) |
|-------------|--------------------|---------------------------------|
| 0x00 02 | 0x00 00 | Diagnostic Register Contents |

0x04 Force Listen Only Mode

Forces the addressed slave to its Listen Only Mode for Modbus[®] RTU communications. This isolates it from the other devices on the network, allowing them to continue communicating without interruption from the addressed slave. No response is returned. When the slave enters its Listen Only Mode, all active communication controls are turned off. The ready watchdog timer is allowed to expire, locking the controls off. While in this mode, any Modbus[®] RTU messages addressed to the slave or broadcast are monitored, but no actions will be taken and no responses will be sent. The only function that will be processed after the mode is entered will be the Restart Communications Option function (function code 8, subfunction 1).

| Subfunction | Data Field (Query) | Data Field (Response) |
|-------------|--------------------|-------------------------|
| 0x00 04 | 0x00 00 | No Response Returned |

0x0A Clear Counters

Clears all Communication Event counters. Counters are also cleared upon power-up.

| Subfunction | Data Field (Query) | Data Field (Response) |
|-------------|--------------------|-----------------------|
| 0x00 0A | 0x00 00 | Echo Query Data |

0x0B Return Bus Message Count

The response data field returns the quantity of messages that the slave has detected on the communications system since its last restart, clear counters operations, or power-up.

| Subfunction | Data Field (Query) | Data Field (Response) |
|-------------|--------------------|-----------------------|
| 0x00 0B | 0x00 00 | Total Message Count |

0x0C Return Bus Communication Error Count

The response data field returns the quantity of CRC errors encountered by the slave since its last restart, clear counters operation, or power-up.

| Subfunction | Data Field (Query) | Data Field (Response) |
|-------------|--------------------|-----------------------|
| 0x00 0C | 0x00 00 | CRC Error Count |

0x0D Return Bus Exception Error Count

The response data field returns the quantity of Modbus[®] RTU exception responses returned by the slave since its last restart, clear counters operation, or power-up.

| Subfunction | Data Field (Query) | Data Field (Response) |
|-------------|--------------------|-----------------------|
| 0x00 0D | 0x00 00 | Exception Error Count |

0x0E Return Slave Message Count

The response data field returns the quantity of messages addressed to the slave, or broadcast, that the slave has processed since its last restart, clear counters operation, or power-up.

| Subfunction | Data Field (Query) | Data Field (Response) |
|-------------|--------------------|-----------------------|
| X000 0E | 0x00 00 | Slave Message Count |

0x0F Return Slave No Response Count

The response data field returns the quantity of messages addressed to the slave for which it returned a no response (neither a normal response nor an exception response), since its last restart, clear counters operation, or power-up.

| Subfunction | Data Field (Query) | Data Field (Response) |
|-------------|--------------------|-------------------------|
| 0x00 0F | 0x00 00 | Slave No Response Count |

0x0F Force Multiple Coils

Forces each coil (registers 00001 to 09999, the coils) in a sequence of coils to either ON or OFF. When broadcast, the function forces the same coil references in all attached slaves.

0x10 Preset Multiple registers

Presets values into the sequence of holding registers (registers 40001 to 49999). When broadcast, the function presets the same register references an all attached slaves.

Note: The function will override the controller's memory protect state. The preset values will remain valid in the registers until the controller's logic next solves the register contents. The register values will remain if they are not programmed in the controller's logic.

Writing Data

The CLS200 controller's memory is divided into approximately 100 parameters with unique control functions, such as temperature, set point, etc. Each parameter can have several Modbus® RTU addresses associated with it. When a Modbus® RTU host writes data to a controller parameter, the data sent may “command” controller firmware to perform specific functions. While host writes to multiple registers are permitted within a parameter, controller function hierarchies necessitate that only one parameter may be written at a time. Any data written past a parameter boundary is simply rejected.

Reading Data

The same parameter-based model is used for reading data from the controller's Modbus® RTU interface. Modbus-RTU allows multiple register block reads for all types of registers. While Modbus® RTU allows this type of read function, unexpected results may occur when reading registers that span across the boundaries of more than one parameter. A query from a Modbus® RTU host starts specific firmware processes that allow data to be formatted properly for the Modbus® RTU host.

The sequence proceeds as follows:

1. A query for data comes in from a Modbus® RTU host.
2. The controller determines what parameter among the hundred or so is being queried.
3. The data format from internal memory is converted for a Modbus® RTU interface.
 - a. Data for registers 40001 to 49999 and registers 30001 to 39999 are always two bytes in length. If the internal controller memory is in bit or byte size, the data is padded appropriately.
 - b. The byte order from internal memory is LSB to MSB (least significant byte to most significant byte. For example, the LSB first value for 550.0° would be x7C 15. The MSB first value would be x15 7C.) while the Modbus® RTU is MSB to LSB. Bytes are swapped as needed.
4. Parameter data is sent to the Modbus® RTU interface.

When data from adjacent parameters is read and the parameters share the same data type (bit, byte or integer), the information is formatted correctly. However, if they do not share the same data type, host software could be written to pad and/or swap bytes as necessary.

Examples

Read Examples

The data read must be sequentially located. If you're reading a coil rather than a register, you must offset the address by the location of the bit you wish to read.

Sample Packet for Host Transmission

| Example | Slave Address in Hex | Function in Hex | Start Address High in Hex | Start Address Low in Hex | Number of Points High in Hex | Number of Points Low in Hex | CRC High in Hex | CRC Low in Hex |
|--|----------------------|-----------------|---------------------------|--------------------------|------------------------------|-----------------------------|-----------------|----------------|
| 1. Reading PV of loop 2 (1600), controller 1 (single-point read) | 01 | 03 | 01 | 6C | 00 | 01 | 45 | EB |
| 2. Reading loops 4 and 5 heat outputs, of controller 3 (multipoint read) | 03 | 03 | 01 | D1 | 00 | 02 | 94 | 2C |
| 3. Reading digital input 4, controller 1 (input status read) | 01 | 02 | 03 | 82 | 00 | 10 | D9 | AA |

Sample Packet for Slave Transmission

| Example | Slave Address in Hex | Function in Hex | Byte Count in Hex | Data in Hex | CRC High in Hex | CRC Low in Hex |
|---|----------------------|-----------------|-------------------|-------------|-----------------|----------------|
| 1. Reading PV of loop 2 (1600), controller 1 (single-point read) | 01 | 03 | 02 | 3E 80 | 84 | 1B |
| 2. Reading loops 4 and 5 (50%, 60%) outputs (heat), of controller 3 (multipoint read) | 03 | 03 | 04 | 3F DE 4C 4A | 2D | 41 |
| 3. Reading digital input 4, controller 1 (input status read) | 01 | 02 | 02 | 08 00 | BE | 78 |

Write Examples

The data written is echoed back to the controller.

Sample Packet for Host Transmission, a Single-Point Write

| Example | Slave Address in Hex | Function in Hex | Address High in Hex | Address Low in Hex | Data High in Hex | Data Low in Hex | CRC High in Hex | CRC Low in Hex |
|--|----------------------|-----------------|---------------------|--------------------|------------------|-----------------|-----------------|----------------|
| 4. Writing loop 1 gain (20), controller 4 (single-point write) | 04 | 06 | 00 | 00 | 00 | 14 | 89 | 90 |
| 5. Writing digital output 30 (on), controller2 (single coil write) | 02 | 05 | 03 | A8 | FF | 00 | 0D | AD |

Sample Packet for Slave Transmission, a Single-Point Write

| Example | Slave Address in Hex | Function in Hex | Address High in Hex | Address Low in Hex | Data High in Hex | Data Low in Hex | CRC High in Hex | CRC Low in Hex |
|--|----------------------|-----------------|---------------------|--------------------|------------------|-----------------|-----------------|----------------|
| 4. Writing loop 1 gain (20), controller 4 (single-point write) | 04 | 06 | 00 | 00 | 00 | 14 | 89 | 90 |
| 5. Writing digital output 30 (on), controller2 (single coil write) | 02 | 05 | 03 | A8 | FF | 00 | 0D | AD |

Sample Packet for Host Transmission, a Multipoint Write

Writing TI loops 3 (100) and 4 (150), controller 10. The data must be written to sequential locations.

| Slave Address in Hex | Function in Hex | Address High in Hex | Address Low in Hex | Number of Registers High in Hex | Number of Registers Low in Hex | Byte Count in Hex | Data in Hex | CRC High in Hex | CRC Low in Hex |
|----------------------|-----------------|---------------------|--------------------|---------------------------------|--------------------------------|-------------------|-------------|-----------------|----------------|
| 0A | 10 | 00 | 86 | 00 | 02 | 04 | 00 64 00 96 | 9F | 70 |

NOTE: The string is longer for multiple write; checking the BYTE COUNT can help in determining if a command timeout is valid.

Sample Packet for Slave Transmission, a multipoint write.

| Slave Address in Hex | Function in Hex | Address High in Hex | Address Low in Hex | Number of Registers High in Hex | Number of Registers Low in Hex | CRC High in Hex | CRC Low in Hex |
|----------------------|-----------------|---------------------|--------------------|---------------------------------|--------------------------------|-----------------|----------------|
| 0A | 10 | 00 | 86 | 00 | 02 | A1 | 5A |

Modbus® RTU Data Table Summary

Each addressable register holds two bytes of data. Each parameter value requires only one register to store any of these types of data. The data type for each parameter is indicated in the tables on the following pages.

| Data Type and Symbol | Data Size |
|----------------------|-----------|
| Unsigned char (UC) | 1 byte |
| Signed char (SC) | 1 byte |
| Unsigned int (UI) | 2 bytes |
| Signed int (SI) | 2 bytes |
| Bit | 1 bit |

Because each loop is individually configurable, the number of instances of many parameters depends on the number of loops in the controller. Therefore, the number of registers for these parameters is listed in the tables on the following pages in terms of the number of loops in the controller.

The storage requirements for some parameters depend on the number of digital inputs or digital outputs to the controller (MAX_DIGIN and MAX_DIGOUT). The storage of ramp-soak profile parameters depends on the number of profiles (MAX_RSP), the number of segments per profile (MAX_SEG), the number of triggers per segment (MAX_TRIG), and the number of events per segment (MAX_EVENT).

The table below shows the values for each of these factors. Use them to calculate the number of registers for each parameter.

| | |
|--|----|
| MAX_CH: | |
| CLS204 (4 loops + 1 pulse loop) | 5 |
| CLS208 (8 loops + 1 pulse loop) | 9 |
| CLS216 / MLS316 / CAS200 (16 loops + 1 pulse loop) | 17 |
| MLS332 (32 loops + 1 pulse loop) | 33 |
| MAX_DIGIN | 8 |
| MAX_DIGOUT | 35 |
| MAX_RSP | 17 |
| MAX_SEG | 20 |
| MAX_TRIG | 2 |
| MAX_EVENT | 4 |

NOTE: Data table parameters 46 to 60, 100 and 103 are ramp-soak parameters. They are only used in controllers with the ramp-soak option. Parameters 81 to 95 are enhanced features and are only available in controllers with the enhanced features option.

Ordering of Heat and Cool Channel Parameters

For parameters that have both heat and cool settings, the heat values are stored in the first registers and the cool values are stored in the registers starting at the listed address plus MAX_CH.

Ordering of Ramp-Soak Profile Parameters

Ramp-soak profile parameters are ordered first by profile, then by segment where applicable. So, for example, the first 8 registers of the Ready Events parameter are the ready segment event states for the first profile (profile A), the next 8 registers are for profile B, and so on. In the case of the segment triggers, the first register contains the first trigger setting for the first segment of profile A, the second register contains the settings for the second trigger for the first segment of profile A, the third register contains the settings for the first trigger for the second segment of profile A, and so on.

Relative and Absolute Modbus® Addresses

In the tables on the following pages, absolute addresses are in decimal and relative addresses are in hexadecimal. Absolute addresses include the type of register. Refer to the absolute address to determine which function to use to read or write values (see the Function Codes on page 26).

Relative addresses indicate the register offset from the first register of the particular type. For example, the first register for the Derivative parameter is at 40067, which is offset 66 (x42) registers from the beginning of the holding registers at 40001.

Modbus® RTU Protocol Data Table

| Number | Description | Absolute Address | Relative Address in Hex | Type | Number of Registers |
|--------|-----------------------------|------------------|-------------------------|------|---------------------|
| 0 | Proportional Band/Gain | 40001 | 0000 | UC | MAX_CH * 2 |
| 1 | Derivative Term | 40067 | 0042 | UC | MAX_CH * 2 |
| 2 | Integral Term | 40133 | 0084 | UI | MAX_CH * 2 |
| 3 | Input Type | 40199 | 00C6 | UC | MAX_CH |
| 4 | Output Type | 40265 | 0108 | UC | MAX_CH * 2 |
| 5 | Setpoint | 40331 | 014A | SI | MAX_CH |
| 6 | Process Variable | 40364 | 016B | SI | MAX_CH |
| 7 | Output Filter | 40397 | 018C | UC | MAX_CH * 2 |
| 8 | Output Value | 40463 | 01CE | UI | MAX_CH * 2 |
| 9 | High Process Alarm Setpoint | 40529 | 0210 | SI | MAX_CH |
| 10 | Low Process Alarm Setpoint | 40562 | 0231 | SI | MAX_CH |
| 11 | Deviation Alarm Band Value | 40595 | 0252 | UC | MAX_CH |
| 12 | Alarm Deadband | 40628 | 0273 | UC | MAX_CH |
| 13 | Alarm_Status | 40661 | 0294 | UI | MAX_CH |
| 14 | Not used | 40694 | 02B5 | | 33 |
| 15 | Ambient Sensor Readings | 40727 | 02D6 | SI | 2 |
| 16 | Pulse Sample Time | 40729 | 02D8 | UC | 1 |
| 17 | High Process Variable | 40730 | 02D9 | SI | MAX_CH |
| 18 | Low Process Variable | 40763 | 02FA | SI | MAX_CH |
| 19 | Precision | 40796 | 031B | SC | MAX_CH |
| 20 | Cycle Time | 40829 | 033C | UC | MAX_CH |
| 21 | Zero Calibration | 40895 | 037E | UI | 2 |
| 22 | Full-scale Calibration | 40896 | 037F | UI | 2 |
| 23 | Job Select Dig Inputs | 40897 | 0380 | UC | 1 |
| 24 | Job Sel Dig Ins Active | 40898 | 0381 | UC | 1 |

| Number | Description | Absolute Address | Relative Address in Hex | Type | Number of Registers |
|--------|------------------------------------|------------------|-------------------------|------|-------------------------------|
| 25 | Digital Inputs | 10899 | 0382 | Bit | MAX_DIGIN |
| 26 | Digital Outputs | 00907 | 038A | Bit | MAX_DIGOUT |
| 27 | Not used | 40942 | 03AD | UC | 1 |
| 28 | Override Digital Input | 40943 | 03AE | UC | 1 |
| 29 | Override Polarity | 40944 | 03AF | UC | 1 |
| 30 | System Status | 40945 | 03B0 | UC | 4 |
| 31 | System Command Register | 40949 | 03B4 | UC | 1 |
| 32 | Data Changed Register | 40950 | 03B5 | UC | 1 |
| 33 | Input Units | 40951 | 03B6 | UC | MAX_CH * 3 |
| 34 | EPROM Version Code | 41050 | 0419 | UC | 1 |
| 35 | Options Register | 41062 | 0425 | UC | 1 |
| 36 | Process Power Digital Input | 41063 | 0426 | UC | 1 |
| 37 | High Reading | 41064 | 0427 | SI | MAX_CH |
| 38 | Low Reading | 41097 | 0448 | SI | MAX_CH |
| 39 | Heat/Cool Spread | 41130 | 0469 | UC | MAX_CH |
| 40 | Startup Alarm Delay | 41163 | 048A | UC | 1 |
| 41 | High Process Alarm Output Number | 41164 | 048B | UC | MAX_CH |
| 42 | Low Process Alarm Output Number | 41197 | 04AC | UC | MAX_CH |
| 43 | High Deviation Alarm Output Number | 41230 | 04CD | UC | MAX_CH |
| 44 | Low Deviation Alarm Output Number | 41263 | 04EE | UC | MAX_CH |
| 45 | Not used | 41296 | 050F | | 1 |
| 46 | Channel Profile and Status | 41297 | 0510 | UC | MAX_CH |
| 47 | Current Segment | 41330 | 0531 | UC | MAX_CH |
| 48 | Segment Time Remaining | 41363 | 0552 | UI | MAX_CH |
| 49 | Current Cycle Number | 41924 | 0783 | UI | MAX_CH |
| 50 | Tolerance Alarm Time | 41957 | 07A4 | UI | MAX_RSP |
| 51 | Last Segment | 41990 | 07C5 | UC | MAX_RSP |
| 52 | Number of Cycles | 42023 | 07E6 | UC | MAX_RSP |
| 53 | Ready Setpoint | 42056 | 0807 | SI | MAX_RSP |
| 54 | Ready Event States | 42089 | 0828 | UC | MAX_RSP * MAX_DIGOUT |
| 55 | Segment Setpoint | 42174 | 087D | SI | MAX_RSP * MAX_SEG |
| 56 | Triggers and Trigger States | 42834 | 0B11 | UC | MAX_RSP * MAX_SEG * MAX_TRIG |
| 57 | Segment Events and Event States | 44154 | 1039 | UC | MAX_RSP * MAX_SEG * MAX_EVENT |
| 58 | Segment Time | 46794 | 1A89 | UI | MAX_RSP * MAX_SEG |
| 59 | Tolerance | 47454 | 1D1D | SI | MAX_RSP * MAX_SEG |
| 60 | Ramp/Soak Flags | 48114 | 1FB1 | UC | MAX_CH |
| 61 | Output Limit | 48147 | 1FD2 | SI | MAX_CH * 2 |
| 62 | Output Limit Time | 48213 | 2014 | SI | MAX_CH * 2 |
| 63 | Alarm_Control | 48279 | 2056 | UI | MAX_CH |

| Number | Description | Absolute Address | Relative Address in Hex | Type | Number of Registers |
|--------|-------------------------------------|------------------|-------------------------|------|---------------------|
| 64 | Alarm_Acknowledge | 48312 | 2077 | UI | MAX_CH |
| 65 | Alarm_Mask | 48345 | 2098 | UI | MAX_CH |
| 66 | Alarm_Enable | 48378 | 20B9 | UI | MAX_CH |
| 67 | Output Override Percentage | 48411 | 20DA | SI | MAX_CH * 2 |
| 68 | AIM Fail Output | 48477 | 211C | UC | 1 |
| 69 | Output Linearity Curve | 48478 | 211D | UC | MAX_CH |
| 70 | SDAC Mode | 48544 | 215F | UC | MAX_CH * 2 |
| 71 | SDAC Low Value | 48610 | 21A1 | SI | MAX_CH * 2 |
| 72 | SDAC High Value | 48676 | 21E3 | SI | MAX_CH * 2 |
| 73 | Save Setup to Job | 48742 | 2225 | UC | 1 |
| 74 | Input Filter | 48743 | 2226 | UC | MAX_CH |
| 75 | Loop Alarm Delay | 48776 | 2247 | UI | MAX_CH |
| 76 | Not used | 48809 | 2268 | | 1 |
| 77 | Loop Names | 48810 | 2269 | UI | MAX_CH * 2 |
| 78 | T/C Failure Detection Flags | 48876 | 22AB | UC | MAX_CH |
| 78 | Channel Name (CAS200) | 48876 | 22AB | UC | MAX_CH * 8 |
| 79 | Restore PID Digital Input | 48909 | 22CC | UC | MAX_CH |
| 80 | Manufacturing Test | 48942 | 22ED | UI | 1 |
| 80 | Manufacturing Test (CAS200) | 49014 | 2235 | UI | 1 |
| 81 | PV Retransmit Primary Loop Number | 48943 | 22EE | UC | MAX_CH * 2 |
| 82 | PV Retransmit Maximum Input | 49009 | 2330 | SI | MAX_CH * 2 |
| 83 | PV Retransmit Maximum Output | 49075 | 2372 | UC | MAX_CH * 2 |
| 84 | PV Retransmit Minimum Input | 49141 | 23B4 | SI | MAX_CH * 2 |
| 85 | PV Retransmit Minimum Output | 49207 | 23F6 | UC | MAX_CH * 2 |
| 86 | Cascade Primary Loop Number | 49273 | 2438 | UC | MAX_CH |
| 87 | Cascade Base Setpoint | 49306 | 2459 | SI | MAX_CH |
| 88 | Cascade Minimum Setpoint | 49339 | 247A | SI | MAX_CH |
| 89 | Cascade Maximum Setpoint | 49372 | 249B | SI | MAX_CH |
| 90 | Cascade Heat/Cool Span | 49405 | 24BC | SI | MAX_CH * 2 |
| 91 | Ratio Control Master Loop Number | 49471 | 24FE | UC | MAX_CH |
| 92 | Ratio Control Minimum Setpoint | 49504 | 251F | SI | MAX_CH |
| 93 | Ratio Control Maximum Setpoint | 49537 | 2540 | SI | MAX_CH |
| 94 | Ratio Control Control Ratio | 49570 | 2561 | UI | MAX_CH |
| 95 | Ratio Control Setpoint Differential | 49603 | 2582 | SI | MAX_CH |
| 96 | Loop Status | 49636 | 25A3 | UC | MAX_CH |
| 97 | Output Type/Disable | 49669 | 25C4 | UC | MAX_CH * 2 |
| 98 | Output Reverse/Direct | 49735 | 2506 | UC | MAX_CH * 2 |
| 99 | Controller Type | 49801 | 2648 | UC | 1 |
| 100 | Ramp/Soak Profile Number | 49802 | 2649 | UC | MAX_CH |

| Number | Description | Absolute Address | Relative Address in Hex | Type | Number of Registers |
|--------|----------------------------|------------------|-------------------------|------|---------------------|
| 101 | Controller Address | 49835 | 266A | UC | 1 |
| 102 | Baud Rate | 49836 | 266B | UC | 1 |
| 103 | Ready Events (Modbus® RTU) | 49837 | 266C | UC | MAX_RSP * 8 |

3

Chapter 3: Controller Parameter Descriptions

This section provides specific details for each data table parameter including data type, variable range, and default values where applicable.

The Controller Menus section on the next page shows all of the menus for CLS200 controllers. (Controller features and menus vary; not all of the menus shown here apply to each controller.) This is for reference only, to help you find applicable controller parameters to test your software.

CAUTION: The controller’s parameters are all read/write, and the controller does not check the content of data written to it. It is possible to write to any parameter, even though it may not be meaningful to do so.

Some of the controller’s functions are not listed as data table parameters in this chapter. If a parameter is not listed in this chapter, one of the following situations applies:

- The parameter is set only in the controller’s menus; it is not set in host software. For example, jobs are loaded through the controller’s front panel only.
- The function may be a bit set in a byte in host software. For example, the panel lock feature does not have its own parameter; it is set in the System Command register.

Correlating Menu Items with Parameters

There is not a one-to-one correspondence between parameters found on the controller menus and the data table items. Some parameters that appear separately on the controller’s display are combined when stored in the data table and when read or written via serial communications. The following tables lay out the correspondence between menu items and the data table.

In the following tables, “RS” in the Product(s) column indicates that the parameter is found in controllers equipped within the Ramp and Soak option firmware. Similarly, “EF” indicates that the parameter is found in controllers equipped with the Enhanced Features option.

| Menu/Parameter | Product(s) | Parameter(s) |
|----------------------------|------------|--------------|
| Single-Loop Display | | |
| Setpoint | All Units | 5 |
| Process Variable | All Units | 6 |
| Heat Output Percent | Not CAS200 | 8 |
| Cool Output Percent | Not CAS200 | 8 |
| Control Mode | Not CAS200 | 4 or 96 |

| Menu/Parameter | Product(s) | Parameter(s) |
|---------------------------------|-------------|------------------|
| Process/Deviation/Sensor Alarms | All Units | 13, 64 |
| System Alarms | All Units | 13, 15 |
| Assign R/S Profile | RS | 46 |
| Ramp Soak Profile | RS | 100 |
| Current Segment | RS | 47 |
| Time Remaining | RS | 48 |
| Cycle Number | RS | 49 |
| Set Mode | RS | 46 |
| Reset | RS | 46 |
| Global Menu | | |
| Load Setup From Job | All Units | Front Panel Only |
| Job Save Number | All Units | 73 |
| Job Select Dig Inputs | All Units | 23 |
| Job Sel Dig Ins Active | All Units | 24 |
| Output Override Dig Input | Not CAS200 | Front Panel Only |
| Override Dig In Active | Not CAS200 | Front Panel Only |
| Startup Alarm Delay | All Units | 40 |
| Ramp/Soak Time Base | RS | 31 |
| Keyboard Lock Status | All Units | 31 |
| Power Up Output Status | Not CAS200 | 31 |
| Process Power Digin | Not CAS200 | Front Panel Only |
| Controller Address | Not CAS200 | 101 |
| Communications Baud Rate | Not CAS200 | 102 |
| Communications Protocol | All Units | Front Panel Only |
| Communications Err Check | All Units | Front Panel Only |
| AC Line Freq | All Units | 31 |
| Dig Out Polarity on Alarm | All Units | 31 |
| Firmware Info | All Units | 34, 35, 99 |
| Input Menu | | |
| Input Type | All Units | 3 |
| Loop Name | Not CAS200 | 77 |
| Channel Name | CAS200 Only | 78 |
| Input Units | All Units | 33 |
| Input Reading Offset | All Units | 17, 18 |
| Reversed T/C Detect | Not CAS200 | 78 |
| Input Pulse Sample Time | All Units | 16 |
| Display Format | All Units | 19 |
| Input Scaling Hi PV | All Units | 17 |
| Input Scaling Hi Rdg | All Units | 37 |
| Input Scaling Lo PV | All Units | 18 |
| Input Scaling Lo Rdg | All Units | 38 |
| Input Filter | All Units | 74 |
| Control Params Menu | | |
| Heat Control PB | Not CAS200 | 0 |

| Menu/Parameter | Product(s) | Parameter(s) |
|-----------------------------|------------|--------------|
| Heat Control TI | Not CAS200 | 2 |
| Heat Control TD | Not CAS200 | 1 |
| Heat Control Filter | Not CAS200 | 7 |
| Cool Control PB | Not CAS200 | 0 |
| Cool Control TI | Not CAS200 | 2 |
| Cool Control TD | Not CAS200 | 1 |
| Cool Control Filter | Not CAS200 | 7 |
| Spread | Not CAS200 | 39 |
| Restore PID Digital Input | Not CAS200 | 79 |
| Outputs Menu | | |
| Heat Control Output | Not CAS200 | 4 |
| Heat Output Type | Not CAS200 | 4, 97 |
| Heat Output Cycle Time (TP) | Not CAS200 | 20 |
| SDAC Mode | Not CAS200 | 70 |
| SDAC Lo Value | Not CAS200 | 71 |
| SDAC Hi Value | Not CAS200 | 72 |
| Heat Output Action | Not CAS200 | 4, 98 |
| Heat Output Limit | Not CAS200 | 61 |
| Heat Output Limit Time | Not CAS200 | 62 |
| Sensor Fail Ht Output | Not CAS200 | 67 |
| Heat T/C Brk Out Avg | Not CAS200 | 78 |
| Heat Output | Not CAS200 | 69 |
| Cool Control Output | Not CAS200 | 4 |
| Cool Output Type | Not CAS200 | 4, 97 |
| Cool Output Cycle Time (TP) | Not CAS200 | 20 |
| SDAC Mode | Not CAS200 | 70 |
| SDAC Lo Value | Not CAS200 | 71 |
| SDAC Hi Value | Not CAS200 | 72 |
| Cool Output Action | Not CAS200 | 4, 98 |
| Cool Output Limit | Not CAS200 | 61 |
| Cool Output Limit Time | Not CAS200 | 62 |
| Sensor Fail CI Output | Not CAS200 | 67 |
| Cool T/C Brk Out Avg | Not CAS200 | 78 |
| Cool Output | Not CAS200 | 69 |
| Alarms Menu | | |
| Hi Proc Alarm Setpt | All Units | 9 |
| Hi Proc Alarm Type | All Units | 63, 65 |
| Hi Proc Alarm Output | All Units | 41 |
| Dev Alarm Value | All Units | 11 |
| Hi Dev Alarm Type | All Units | 63, 65 |
| Hi Dev Alarm Output | All Units | 43 |
| Lo Dev Alarm Type | All Units | 63, 65 |
| Lo Dev Alarm Output | All Units | 44 |
| Lo Proc Alarm Setpt | All Units | 10 |

| Menu/Parameter | Product(s) | Parameter(s) |
|-----------------------------|------------|------------------|
| Lo Proc Alarm Type | All Units | 63, 65 |
| Lo Proc Alarm Output | All Units | 42 |
| Alarm Deadband | All Units | 12 |
| Alarm Delay | All Units | 40, 75 |
| Manual I/O Test Menu | | |
| Digital Inputs | All Units | 25 |
| Test Digital Output | All Units | 26 |
| Digital Output Number | All Units | 26 |
| Keypad Test | All Units | Front Panel only |

Additional menus in Ramp and Soak and Enhanced Features options

| Menu/Parameter | Product(s) | Parameter(s) |
|----------------------------------|------------|----------------------------------|
| Setup Loop PV Retransmit | EF and RS | |
| Heat Output Retrans PV | EF and RS | 81 |
| PV Retransmit Minimum Input | EF and RS | 84 |
| PV Retransmit Minimum Output | EF and RS | 85 |
| PV Retransmit Maximum Input | EF and RS | 82 |
| PV Retransmit Maximum Output | EF and RS | 83 |
| Cool Output Retrans PV | EF and RS | 81 |
| PV Retransmit Minimum Input | EF and RS | 84 |
| PV Retransmit Minimum Output | EF and RS | 85 |
| PV Retransmit Maximum Input | EF and RS | 82 |
| PV Retransmit Maximum Output | EF and RS | 83 |
| Setup Loop Cascade | EF | |
| Cascade Primary Loop Number | EF | 86 |
| Cascade Base Setpoint | EF | 87 |
| Cascade Minimum Setpoint | EF | 88 |
| Cascade Maximum Setpoint | EF | 89 |
| Cascade Heat Span | EF | 90 |
| Cascade Cool Span | EF | 90 |
| Setup Loop Ratio Control | EF | |
| Ratio Control Master Loop Number | EF | 91 |
| Ratio Control Minimum Setpoint | EF | 92 |
| Ratio Control Maximum Setpoint | EF | 93 |
| Ratio Control Ctrl Ratio | EF | 94 |
| Ratio Control SP Diff | EF | 95 |
| Setup Ramp/Soak Profile | RS | |
| Edit Ramp & Soak Profile | RS | Front Panel only |
| Copy Setup From Profile | RS | Front Panel only |
| Out-of-Tolerance Alarm Time | RS | 50 |
| Ready Segment Setpoint | RS | 53 |
| Ready Segment Edit Events | RS | Front Panel only |
| Ready Event Output | RS | 54 (ANA or AB) and 103 (Modbus®) |

| Menu/Parameter | Product(s) | Parameter(s) |
|------------------------------|------------|------------------|
| External Reset Input Number | RS | Front Panel only |
| Edit Segment Number | RS | Front Panel only |
| Segment ## Seg Time | RS | 58 |
| Segment ## Seg Setpt | RS | 55 |
| Segment ## Edit Seg Events | RS | Front Panel only |
| Seg ## Event # Output | RS | 57 |
| Seg ## Ev# DO## Active State | RS | 57 |
| Segment ## Edit Seg Trggrs | RS | Front Panel only |
| Seg ## Trig # Input NR | RS | 56 |
| Seg ## Tr# DI## ACTIVE STATE | RS | 56 |
| SEG ## TR# DI## TRIG | RS | 56 |
| SEGMENT ## SEG TOLERANCE | RS | 59 |
| SEGMENT ## LAST SEGMENT | RS | 51 |
| REPEAT CYCLES | RS | 52 |

Parameters (by number)

Proportional Band/Gain (0)

The CLS200 controller lets users modify the Proportional Band (PB), but they internally represent the PB as a Gain value.

- Range: 1 to 255
- Heat/Cool: 35
- Pulse: 20

Users edit the Proportional Band, but the controller uses a Gain value internally. This equation illustrates the relationship between the PB and Gain:

$$\text{Proportional Band} = \frac{(\text{High Range Value}) - (\text{Low Range Value})}{\text{Gain}}$$

For example, on a J-type thermocouple the high range is 1400 and low range is –350, so a gain of 35 gives a PB of 50 degrees. (The input type ranges are listed in the Input Type section and in the *User's Guide* for your controller.)

Derivative Term (1)

This parameter contains the derivative term for PID output calculations. (The derivative term is also known as the TD or Rate.)

- Range: 0 to 255 seconds
- EX PROM: 125
- Pulse Loop: 0
- All Others: 0

Integral Term (2)

This parameter contains the integral term for PID output calculations. (The integral term is also known as the Reset or TI.)

- Range: 0 to 6000 seconds per repeat. (Setting the TI to 0 seconds turns off the integral action.)
- STD Heat: 180
- STD Cool: 60
- Pulse: 0

Input Type (3)

This parameter specifies the input type.

- Range: 0 to 19 defined values
- Default: 1 (J-type thermocouple)

The following input types are currently defined:

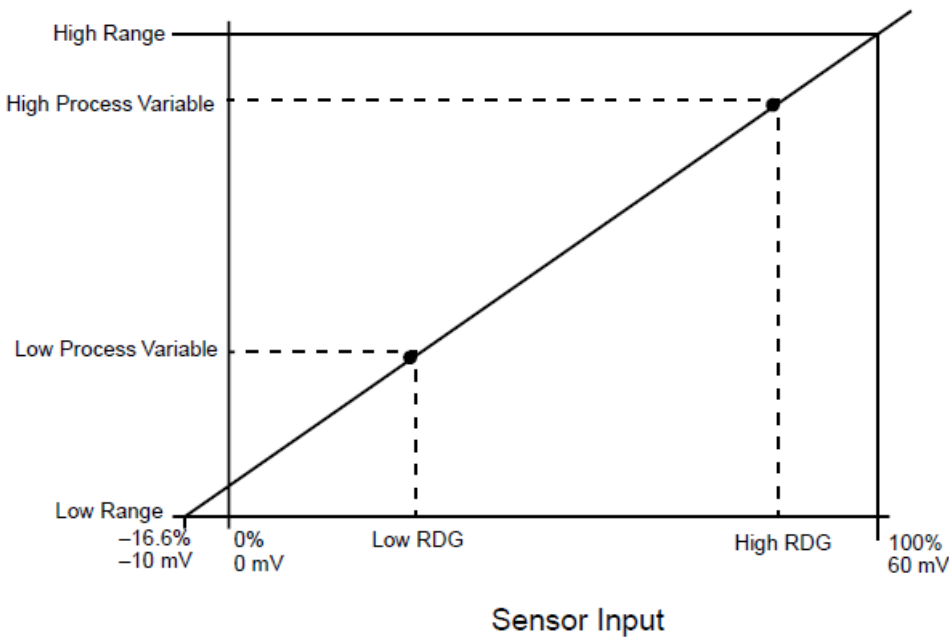
| Decimal Number | Description | Hex Value | Range |
|----------------|--|-----------|----------------------------------|
| 0 | Linear | 00 | –10 to 60 mV (scalable) |
| 1 | J-type thermocouple | 01 | –350 to 1400°F –212 to 760°C |
| 2 | K-type thermocouple | 02 | –450 to 2500°F –268 to 1371°C |
| 3 | T-type thermocouple | 03 | –450 to 750°F –268 to 399°C |
| 4 | S-type thermocouple | 04 | 0 to 3200°F –18 to 1760°C |
| 5 | R-type thermocouple | 05 | 0 to 3210°F –18 to 1766°C |
| 6 | B-type thermocouple | 06 | 150 to 3200°F 660 to 1760°C |
| 7 | Pulse Input | 07 | 0 to 2000 Hz |
| 8 | RTD1 (high resolution) (Not available in CLS216 and CAS200) | 08 | –148 to 527°F –100 to 275°C |
| 9 | RTD2 (low resolution) (Not available in CLS216 and CAS200) | 09 | –184 to 1544°F –120 to 840°C |
| 10 | Skip Channel | 0A | N/A |
| 11 to 13 | Reserved for 8LS carbon potential | N/A | N/A |
| 14 | N/A | N/A | N/A |
| 15 | N/A | N/A | N/A |
| 16 | N/A | N/A | N/A |
| 17 | N/A | N/A | N/A |
| 18 | N/A | N/A | N/A |
| 19 | Motor speed | 13 | –10 to 60 mV (scalable) |
| 20 | E thermocouple | 14 | –328 to 1448°F –200 to 787°C |

The input type determines the ranges for these other parameters:

- Process variable
- Setpoint
- Proportional band
- High process alarm
- Low process alarm
- Deviation band alarm
- Heat/cool spread
- Alarm deadband

Ranges for all input parameters are determined as follows:

- For thermocouple and RTD inputs, the high and low ranges are fixed at the values shown in the previous table
- For pulse and linear inputs, the high and low ranges are determined by the values entered for input scaling, as shown in the graph below



Output Type (4)

This parameter defines the output type of a given output pin.

- Default: Heat outputs default to manual control, enabled, reverse action, time proportioning. Cool outputs default to disabled, direct action, time proportioning
- Range: A 1-byte value determined as shown in the tables below

| Bit | Bit set to 0 | Bit set to 1 |
|---------|------------------------------|---------------------------|
| 0 and 1 | See below | |
| 2 | Loop is in automatic control | Loop is in manual control |
| 3 | Autotune mode off | Autotune mode on |
| 4 | Output disabled | Output enabled |

| Bit | Bit set to 0 | Bit set to 1 |
|-----|--|--|
| 5 | Spare | Spare |
| 6 | Output not set to Serial DAC (see table below) | Output set to Serial DAC (see table below) |
| 7 | Output set to Reverse action | Output set to Direct action |

Bits 0 and 1 work together to determine some of the output's characteristics:

| Bit 6 Setting | Bit 1 Setting | Bit 0 Setting | Result |
|---------------|---------------|---------------|--------------------|
| 0 | 0 | 0 | Time Proportioning |
| 0 | 0 | 1 | DZC |
| 0 | 1 | 0 | Analog (only 8LS) |
| 0 | 1 | 1 | On/Off |
| 1 | 0 | 0 | SDAC |
| 1 | 0 | 1 | 3P DZC |

NOTE: Bit 2 in the heat output-type byte determines the loop's control status (Automatic or Manual).
The controller ignores Bit 2 in the cool output-type byte.

Setpoint (5)

This parameter contains the process setpoint, expressed in engineering units. (The setpoint is affected by the Precision parameter.)

- The setpoint's range depends on the loop's input type
- The default setpoint for a J-type thermocouple is 250 (25°F)

Process Variable (6)

The process variable contains the compensated input measurement, expressed in engineering units. (This parameter is affected by the Precision parameter.)

- The process variable's range depends on the loop's input type (described in the Input Type section).

Output Filter (7)

The adjustable output filter dampens the control output response. (Setting the number of scans to 0 disables the filter.)

- Range: 0 to 255 scans
- Default: 3 scans

Output Value (8)

This parameter contains the output value, based on a full-scale value of 32700 equals 100%. You can write to the output value at any time, but a write command is only meaningful for loops set to Manual control.

- Range: 0 to 32700 (0 to 100%)
- Default: 0

High Process Alarm Setpoint (9)

The high process alarm setpoint is the absolute high process variable limit, expressed in engineering units. (This parameter is affected by the Precision parameter.)

- Range: –999 to 2500 (legal range determined by input type)
- Default: 10000 (1000°F) for a J-type thermocouple

Low Process Alarm Setpoint (10)

The low process alarm setpoint is the absolute low process variable limit, expressed in engineering units. (This parameter is affected by the Precision parameter.)

- Range: –999 to 2500 (legal range determined by input type)
- Default: 0

Deviation Alarm Band Value (11)

The deviation alarm band value indicates the amount of deviation from the setpoint before an alarm is issued. The amount of deviation is expressed in engineering units. (This parameter is affected by the Precision parameter when the input type is linear but not when a thermocouple type is selected.)

- Range: 0 to 255
- Default: 5

Alarm Deadband (12)

The alarm deadband prevents the alarm output from fluctuating rapidly when the input is near the alarm setpoint. (This parameter is affected by the Precision parameter when the input type is linear but not when a thermocouple type is selected.)

- Range: 0 to 255
- Default: 2

NOTE: Process alarms, deviation alarms, and failed sensor alarms are set individually for each loop. They are controlled by bit settings in a number of alarm variables.

This section uses the following expressions interchangeably:

- Bit set to 1 = True = Set
- Bit set to 0 = False = Cleared

Users can set loop alarms to warn them of high and low process variables and high and low deviation from the setpoint. Users can also set a loop alarm deadband value that prevents alarm “chattering” in process and deviation alarms. (There are also failed sensor alarms for some input types; users cannot configure the failed sensor alarms.)

All of these alarms are individually indicated. They can also be individually enabled, disabled, and acknowledged. The host software can observe and set process alarm and deviation alarm values.

NOTE: Alarms in CLS200 controllers depend on these five 16-bit variables, all unsigned integers: Alarm_Status, Alarm_Mask, Alarm_Enable, Alarm_Control, Alarm_Acknowledge.

Each of these variables is responsible for a different alarm behavior, attribute, or condition. Each of

the controller's alarms has 1 bit in these variables; the table below shows the bit map for each variable. The bits are treated as 16 separate Boolean (TRUE/FALSE) variables.

Alarm_Status (13)

This parameter provides the current status of all the alarms for a loop, except for special ramp-soak alarms. When an alarm occurs, the controller sets the appropriate bit in Alarm_Status. When the alarm clears, the controller clears the bit for that alarm.

When using the Anafaze protocol, when an Alarm_Status bit has changed, and there are no higher-priority status codes to return, the controller will return an Ex (see the STS section in Chapter 1), in the high nibble of the next communications status byte it sends to the host software. The software should upload the Alarm_Status integers or words for all loops to determine which alarms have changed and in which loops.

Host software can read this parameter, but should not write to it.

NOTE: An alarm clears when its alarm condition is no longer present. If an alarm clears, and it has not already been acknowledged, it remains an unacknowledged alarm until the operator acknowledges it (by pressing the controller's Alarm Ack key or through software).

CAUTION: Do not use host software to alter the status of the Alarm_Status bits.

When Alarm_Status Trips

An alarm trips only if the Alarm_Mask bit, Alarm_Enable bit, and the alarm condition are all TRUE. For example, the high process alarm trips if the Alarm_Mask bit and Alarm_Enable bit are all true, and the process exceeds the high process alarm setpoint value.

When Alarm_Status Clears

An alarm clears when the condition that caused it clears, or when the Alarm_Mask bit is set to FALSE. For example, the high process alarm clears if the process goes below the high process alarm setpoint.

| Bit | Alarm Name |
|-----|---|
| 0* | Spare |
| 1 | Spare |
| 2 | Low deviation |
| 3 | High deviation |
| 4 | Low process |
| 5 | High process |
| 6 | T/C Reversed |
| 7 | T/C Short |
| 8 | T/C break (or open) |
| 9 | RTD open (not in CLS216 or CAS200) |
| 10 | RTD short (not in CLS216 or CAS200) |
| 11 | N/A |
| 12 | Ambient Warning (version 3.4 and later) |
| 13 | Ambient Cal Error |

| Bit | Alarm Name |
|-----|----------------------|
| 14 | Full-scale Cal Error |
| 15 | Offset Cal Error |

*Least significant bit

Ambient Sensor Readings (15)

This parameter returns the value of the system ambient sensor in degrees Fahrenheit to a tenth of a degree. (The ambient sensor is used for ambient temperature compensation for thermocouples.) The CLS200 controller has only one ambient sensor. However, the block size has been allocated to allow a maximum of six ambient sensors per system.

Pulse Sample Time (16)

This parameter is the sample period in seconds for the pulse counter input.

- Range: 1 to 20 seconds
- Default: 1 second

High Process Variable (17)

This parameter is one of four points used to scale inputs, expressed in engineering units. (This parameter is affected by the Precision parameter.)

- Range: –9999 to 30000
- Default: 14000 (1400F) for J-type thermocouple

Low Process Variable (18)

This parameter is one of four points used to scale inputs, expressed in engineering units. (This parameter is affected by the Precision parameter.)

- Range: –9999 to 30000
- Default: –3500 (–350°F) for J-type thermocouple

NOTE: Whenever the Input Type or Units are changed, the High Process Variable and Low Process Variable parameters are set to the default values for the Input Type. see Input Type (3) on page 46 for a list of these values.

NOTE: Use the High Process Variable and Low Process Variable for an offset on all types of inputs. Add or Subtract the offset from both the High and Low Process Variables full-scale values. For example, for a J-type thermocouple, the full range is 14000 (1400°F) to –3500 (–350°F) with an offset of +5 degrees set the high Process Variable to 14050 and the low Process Variable to –3450, moving the full range scale up 5 degrees.

Precision (19)

The precision determines the number of decimal places in the associated parameters.

The value stored in this parameter affects several other parameters.

- Range: –1 to +4
- Default: –1 for a J-type thermocouple

The controller does not store any floating-point values. Instead, it uses the Precision parameter to simulate floating-point calculations for these other parameters:

- Process variable
- Setpoint
- Proportional band
- High process alarm
- Low process alarm
- Deviation band alarm
- Heat/cool spread
- Alarm deadband
- Ready setpoint
- Segment setpoint
- Tolerance value

When the controller sends values for these parameters to host software, it sends them as integers. To convert the numbers to a decimal, the host software must do the following:

1. Take the number from the controller.
2. Divide it by $10^{|p|}$ (where $|p|$ is the absolute value of the precision setting. So for example, if the precision is set to -1 , divide values read via communications by 10, and if the precision is 2, divide by 100).
3. If the original precision value (from the table below) was a negative number, round the number to the nearest integer. Otherwise, display the number.

Exception

For the deviation band alarm, heat/cool spread, and alarm deadband parameters, follow the procedure above, but if the precision value from the table is negative, display the raw number read from the controller.

Display Format Menu

When users edit the controller's Display Format menu for a linear input, they are really editing the precision for that input.

Available precision values

| Precision Value | Display Format | Sample Raw Value Read via Communications | Scaled Display Value |
|-----------------|--------------------|--|----------------------|
| -1 | -999 to 3000 | 2556 | 257 |
| 0 | -9999 to 30000 | 2556 | 2556 |
| 1 | -999.9 to 3000.0 | 2556 | 255.6 |
| 2 | -99.99 to 300.00 | 2556 | 25.56 |
| 3 | -9.999 to 30.000 | 2556 | 2.556 |
| 4 | -0.9999 to +3.0000 | 2556 | 0.2556 |

Cycle Time (20)

This parameter contains the time proportioning time base, expressed in seconds.

- Range: 0 to 255 seconds
- Heat: 10
- Cool: 3

Zero Calibration (21)

This parameter returns the controller's zero calibration counts.

Full-scale Calibration (22)

This parameter returns the controller's full-scale calibration counts

Job Select Dig Inputs (23)

This parameter selects how many digital inputs to use to select jobs and how many jobs the controller stores in memory.

- 0 = No inputs used, no jobs
- 1 = 1 input used (2 jobs)
- 2 = 2 inputs used (4 jobs)
- 3 = 3 inputs used (8 jobs)

Job Sel Dig Ins Active (24)

This parameter selects for what signal level the job select digital inputs are considered active (true).

- 0 = active (true) when low
- 1 = active (true) when high

Digital Inputs (25)

This parameter returns the state of the controller's digital inputs. If the input is an open circuit (high) the corresponding value returns a 1. If the input is connected to common or low a 0 is returned.

Anafaze/AB protocol returns all 8 bits in 1 byte

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---------|---------|---------|---------|---------|---------|---------|---------|
| Input 8 | Input 7 | Input 6 | Input 5 | Input 4 | Input 3 | Input 2 | Input 1 |

The Modbus® RTU protocol stores the states of the inputs in eight individually addressable discrete input registers. Use Modbus® function x02 to read the states of the digital inputs.

Digital Outputs (26)

This parameter contains the state of the controller's digital outputs. If the output is off (an open circuit), the value is 0. If the output is on, the value is 1. To turn on an output, set the corresponding bit to 1.

The Anafaze/AB protocol stores the states for up to eight outputs in 1 byte. So, a total of 5 bytes are required to store the states of the 35 digital outputs in a controller. See the table below to decode the bits.

- Default: 0 (off).

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|----------|-------|-------|-------|-------|-------|-------|-------|
| Byte 0 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| Byte 1 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 |
| Byte 2 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 |
| Byte 3 | 32 | 31 | 30 | 29 | 28 | 27 | 26 | 25 |
| Byte 4 | Not used | | | | | 35 | 34 | 33 |

The Modbus® RTU protocol stores the states of the outputs in 35 individually addressable coil registers.

Use Modbus® Function x01 to read the state

When setting through Modbus® use Function x05 to set the output on or off. The value to send for ON is FF00. To turn OFF send 0000.

Override Digital Input (28)

This parameter enables the output override feature and selects a digital input to trigger it. When the feature is enabled and the specified input is activated, the controller sets all loops to manual mode at the heat and cool outputs at the levels specified by the Output Override parameters.

- Range: 0 to 8 (Disabled = 0, Enabled = 1 to 8, indicating the selected digital input)
- Default: 0

Override Polarity (29)

Specify whether a low or high signal activates the output override feature.

- Range: 0 or 1 (Low = 0, High = 1)
- Default: 0

System Status (30)

The system status command consists of internal registers that flag hardware and software exceptions. The system status command takes 4 bytes. (None of the controllers use the last 2 bytes; they are reserved.)

First two system status bytes (more than 1 bit can be set at a time)

| Bit | 0 | 1 |
|-------|---------------------------------|----------------------------------|
| 0 | Battery OK | Dead battery |
| 1 | Init Start OK | Bad Init Start Bit |
| 2 | AIM OK | AIM comm failure |
| 3 | No Ambient Error | Ambient Error |
| 4 | No Ambient Warning | Ambient Warning |
| 5 | Zero Calibration OK | Bad zero calibration |
| 6 | Full-scale calibration value OK | Bad full-scale calibration value |
| 7 | Reserved | |
| 8 | Alarms delay Off | Alarms delay On |
| 9-13 | Reserved | |
| 14-15 | See next table | |

Partial bit map of system status byte 2 (the CLS200 and CAS200 use bits 14 and 15 only)

| Bit 15 Setting | Bit 14 Setting | Result |
|----------------|----------------|-------------------------|
| 0 | 0 | Controller has 4 loops |
| 0 | 1 | Controller has 8 loops |
| 1 | 0 | Controller has 16 loops |

System Command Register (31)

The System Command Register is a register of mode and configuration flags for the controller's processor.

- Default: 0

| Bit | 0 | 1 |
|-----|---|---|
| 0 | Use default output data on startup. | Use memory data on startup. |
| 1 | Operator keys enabled. | Operator keys locked. |
| 2 | Unit set to 60 Hz. | Unit set to 50 Hz. |
| 3 | Ramp-soak time base in hours and minutes. | Ramp-soak time base in minutes and seconds. |
| 4 | Alarm digital outputs active Low. | Alarm digital outputs active High |
| 5 | Manufacturing Test | |
| 6 | Parameter reset bit | |
| 7 | Reserved | |

CAUTION: Switching to Manufacturing Test mode may cause loss of data when used in normal operation. Use only when these tests are absolutely necessary.

Data Changed Register (32)

The data changed register acts as a First-In-First-Out (FIFO) to point to parameters that have changed internally. The host software must query this register after receiving a "Data Changed" status flag to determine which data has changed. (For more information about "Data Changed" status flags, see the Status Byte section in Chapter 1.)

- Range: 0 to 255.

The general program flow goes like this:

1. If there is anything in the controller's internal Data Changed Stack, the controller returns a "Data Changed" status flag. (See the Status Byte section in Chapter 1 for an explanation of the "Data Changed" status flag.)
2. The host receives a "Data Changed" flag in the communications packet.
3. The host reads the Data Changed Register for the command number of the parameter that changed and then uploads that data block.
4. The controller notes that the Data Changed Register has been read and waits for an Acknowledge from the host. When the Acknowledge has been received, the controller checks its own Data Changed Stack. If there are still data blocks that have been changed, but not uploaded, the controller puts the next command number in the Data Changed Register and continues to return a "Data Changed" status in communications packets. Otherwise, the Data Changed Register is cleared and no more Data Changed status flags are returned. (For more help on this topic, see the Status Byte section in Chapter 1.)

Input Units (33)

The controllers show the process variable expressed in engineering units. This parameter consists of three-character strings that provide text representation of the engineering units for each loop. (The default input units for a J-type thermocouple—the default input type— are in °F.)

For thermocouple and RTD inputs, the input units are preset; the third character of the loop's string indicates whether the reading is in degrees Celsius (if the second and third characters are “°C”), or in degrees Fahrenheit (if the second and third characters are “°F”). For linear and pulse inputs, users can select three characters to display.

Valid characters for input units

| Character | Decimal Value | Hex Value |
|-------------------|---------------|-----------|
| (Space) | 32 | 20 |
| # | 35 | 23 |
| ° (degree symbol) | 223 | DF |
| % | 37 | 25 |
| / | 47 | 2F |
| A to Z | 65 to 90 | 41 to 5A |
| 0 to 9 | 48 to 57 | 30 to 39 |

EPROM Version Code (34)

The firmware code for the EPROM version consists of 3 bytes. The first one contains the controller/unit model. The second one contains the EPROM major revision, and the third one contains the EPROM minor revision.

Model codes (other codes are reserved for older controllers.)

| Code | Model |
|------|--------|
| 9 | MLS300 |
| 10 | CL200 |
| 12 | CAS200 |

CAUTION: Do not write to the EPROM version parameter.

Options Register (35)

The options register is register of flags denoting firmware options in the controller.

Defined options

| Bit | 0 | 1 |
|-----|------------------------------------|-------------------------|
| 0 | Reserved | |
| 1 | Cascade option not present | Cascade-enhanced option |
| 2 | Reserved | |
| 3 | Real time clock option not present | Real time clock option |
| 4 | Reserved | |
| 5 | Ramp-soak not present | Ramp-soak |
| 6 | Math package not present | Math package |
| 7 | Reserved | |

Process Power Digital Input (36)

Enable the thermocouple short detection feature by selecting a digital input.

- Range: 0 to 8 (Disabled = 0, Enabled = 1 to 8 indicating the selected digital input)
- Default: 0

High Reading (37)

This parameter contains one of four points used to scale inputs. For thermocouple and RTD inputs, the high reading is expressed in tenths of a degree Celsius or Fahrenheit. For linear inputs, the high reading value is expressed in tenths of a percent of full-scale. For pulse inputs, the high reading value is expressed in Hertz.

- Range: -999 to 9999
- Default: 14000 (1400°F) for a J-type thermocouple

Low Reading (38)

This parameter contains one of four points used to scale inputs. For thermocouple and RTD inputs, the low reading is expressed in tenths of a degree Celsius or Fahrenheit. For linear inputs, the low reading value is expressed in tenths of a percent of full-scale. For pulse inputs, the low reading value is expressed in Hertz.

- Range: -999 to 9999
- Default: -3500 (-350°F) for a J-type thermocouple

NOTE: Change the High Reading and Low Reading settings only when the Input Type is set to Linear or Pulse.

Heat/Cool Spread (39)

This parameter describes a deadband about the setpoint for heat/cool loops. The heat/cool spread is in units of the input variable. This parameter is affected by the Precision parameter. (This parameter is affected by the Precision parameter when the input type is linear but not when a thermocouple type is selected.)

- Range: 0 to 255
- Default: 5

Startup Alarm Delay (40)

This parameter designates a delay time for process and deviation alarms on power-up. The controller does not report process and deviation alarms for the specified number of minutes after the controller powers up.

The startup alarm delay is a global alarm parameter; it applies to all process and deviation alarms for every loop.

The startup alarm delay does not apply to failed-sensor alarms and AIM failure alarms.

- Range: 0 to 255 minutes
- Default: 0

High Process Alarm Output Number (41)

This parameter assigns the output number to which the high process alarm output is directed.

- Range: 0 to 34
- Default: 0 (no high process alarm output)

Low Process Alarm Output Number (42)

This parameter assigns the output number to which the low process alarm output is directed.

- Range: 0 to 34
- Default: 0 (no low process alarm output)

High Deviation Alarm Output Number (43)

This parameter assigns the output number to which the high deviation alarm output is directed.

- Range: 0 to 34
- Default: 0 (no high deviation alarm output)

Low Deviation Alarm Output Number (44)

This parameter assigns the digital output number to which the low deviation alarm output is directed.

- Range: 0 to 34
- Default: 0 (no low deviation alarm output)

Channel Profile and Status (46)

In this byte, bits 0 to 4 hold the ramp-soak profile number for the loop. Bits 5 to 7 hold the profile's status (Ready, Running, Hold, Trigger Wait, or Out of Tolerance).

- Range: See tables below.
- Default: 0.
- This is a profile setting

Profile number and reference letter in bits 0 to 4

| Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Profile Number | Reference Letter |
|-------|-------|-------|-------|-------|----------------|------------------|
| 0 | 0 | 0 | 0 | 0 | 0 | A |
| 0 | 0 | 0 | 0 | 1 | 1 | B |
| 0 | 0 | 0 | 1 | 0 | 2 | C |
| 0 | 0 | 0 | 1 | 1 | 3 | D |
| 0 | 0 | 1 | 0 | 0 | 4 | E |
| 0 | 0 | 1 | 0 | 1 | 5 | F |
| 0 | 0 | 1 | 1 | 0 | 6 | G |
| 0 | 0 | 1 | 1 | 1 | 7 | H |
| 0 | 1 | 0 | 0 | 0 | 8 | I |
| 0 | 1 | 0 | 0 | 1 | 9 | J |
| 0 | 1 | 0 | 1 | 0 | 10 | K |
| 0 | 1 | 0 | 1 | 1 | 11 | L |
| 0 | 1 | 1 | 0 | 0 | 12 | M |

| Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Profile Number | Reference Letter |
|-------|-------|-------|-------|-------|----------------|------------------|
| 0 | 1 | 1 | 0 | 1 | 13 | N |
| 0 | 1 | 1 | 1 | 0 | 14 | O |
| 0 | 1 | 1 | 1 | 1 | 15 | P |
| 1 | 0 | 0 | 0 | 0 | 16 | Q |

Profile status in bits 5, 6, and 7

| Bit 7 | Bit 6 | Bit 5 | Status |
|-------|-------|-------|-----------------------------------|
| 0 | 0 | 0 | Profile in Ready state "S" |
| 0 | 0 | 1 | Profile is running "R" |
| 0 | 1 | 0 | Profile is holding "H" |
| 0 | 1 | 1 | Profile in trigger wait state "W" |
| 1 | 0 | 0 | Profile out of tolerance "O" |
| 1 | 1 | 1 | No profile assigned |

Current Segment (47)

This parameter returns the segment number that is currently executing. The controller's front panel displays (current segment parameter value+1).

If the user has not assigned a ramp-soak profile to the loop, this parameter is undefined.

If the loop is in Ready state, this parameter has a value of -1. In Ready state, the controller's front panel displays segment 0.

- Range: -1 to 19
- Default: 0
- This is a profile setting

Segment Time Remaining (48)

If the time base is in minutes and seconds, this parameter holds the total remaining seconds, up to 999 minutes and 59 seconds. If the time base is in hours and minutes, this parameter holds the total remaining minutes, up to 999 hours, 59 minutes.

- Range: 0 to 59999
- Default: 0
- This is a profile setting

Current Cycle Number (49)

This parameter returns the number of the current cycle. (Command 52 Number of Cycles, returns the total number of cycles to execute.)

- Range: 0 to 9999
- Default: 0
- This is a profile setting

Tolerance Alarm Time (50)

The value of this parameter decrements once each time unit (each minute if the time base is set to hours and minutes, or each second if the time base is set to minutes and seconds), while the profile is out of tolerance. When a ramp-soak segment is out of tolerance for longer than the tolerance alarm time, the controller goes into tolerance alarm and the tolerance timer resets.

- Range: 0 to 59999
- Default: 0
- This is a profile setting

Last Segment (51)

This parameter denotes the last segment in the profile.

- Range: 0 to 19
- 0 = Seg 1, 1 = Seg 2, 2 = Seg 3...and so on to...19 = Seg 20.
- Default: 19
- This is a profile setting

Number of Cycles (52)

This parameter represents the total number of times to repeat the current profile. Users can set 1 to 99 repeat cycle profiles or they can set the profile to cycle continuously.

- Range: 0 (continuous), 1 to 99
- Default: 1
- This is a profile setting

Ready Setpoint (53)

This parameter represents the ready segment's setpoint. The first segment begins ramping from the ready segment setpoint. When the profile ends, the process returns to this setpoint. (This value is affected by precision.)

- Range: Low Process Variable to high Process Variable. (–999 to 9999, legal value determined by input type.)
- Default: 0
- This is a profile setting

| If you send this... | and assign to a loop with precision... | ...you get this... |
|---------------------|--|--------------------|
| 3000 | -1 | 300 |
| 3000 | 0 | 300 |
| 3000 | 1 | 300.0 |
| 3000 | 2 | 30.00 |
| 3000 | 3 | 3.000 |
| 3000 | 4 | 0.3000 |

Because the profile is created in advance of assigning it to a loop, the controller stores the profile setpoints and tolerance as integers and scales them once the profile is assigned to a loop. For historic

reasons, loops with a precision of 0 are treated as if the precision was -1 when a profile is assigned. In that case (whether the Input type is RTD2 or Linear with precision 0) the setpoints and tolerances are divided by 10 when the profile is assigned to the loop. See the table for how to adjust the number you send or read through communications to this ramp and soak profile parameter.

Ready Event States (54)

This parameter describes the ready segment's output state for all outputs that are not used for control or for the SDAC clock. When the loop goes to the ready state, these outputs assume the state specified by this parameter.

- Range: 0 (off) or 1 (on)
- Default: 0 (off)
- This is a profile setting.

In the Anafaze/AB protocol, the states are stored as bits in 5 bytes.

Bits representing outputs 1 to 34

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0* |
|--------|----------|-------|-------|-------|-------|-------|-------|--------|
| Byte 0 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| Byte 1 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 |
| Byte 2 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 |
| Byte 3 | 32 | 31 | 30 | 29 | 28 | 27 | 26 | 25 |
| Byte 4 | Reserved | | | | | | 34 | 33 |

*Bit 0 is the least significant bit.

When accessed via the Modbus® RTU protocol, this parameter contains the ready segment event outputs for the first 10 profiles (A to J) only. For access to all the ready segment events, see parameter 103.

Segment Setpoint (55)

This parameter represents the setpoint reached at the end of the segment. (This value is affected by precision)

- Range: Low Process Variable to High Process Variable. (–999 to 9999, legal value determined by input type.)
- Default: 0
- This is a profile setting

| If you send this... | and assign to a loop with precision... | ...you get this... |
|---------------------|--|--------------------|
| 3000 | -1 | 300 |
| 3000 | 0 | 300 |
| 3000 | 1 | 300.0 |
| 3000 | 2 | 30.00 |
| 3000 | 3 | 3.000 |
| 3000 | 4 | 0.3000 |

Because the profile is created in advance of assigning it to a loop, the controller stores the profile setpoints and tolerance as integers and scales them once the profile is assigned to a loop. For historic

reasons, loops with a precision of 0 are treated as if the precision was -1 when a profile is assigned. In that case (whether the Input type is RTD2 or Linear with precision 0) the setpoints and tolerances are divided by 10 when the profile is assigned to the loop. See the table for how to adjust the number you send or read through communications to this ramp and soak profile parameter.

Triggers and Trigger States (56)

This byte holds the trigger input number, its active state, and its latch status. Triggers are saved in memory as indicated below.

Seg(1) Trig(1), Seg(1) Trig(2), Seg(2) Trig(1), Seg(2) Trig(2), etc.

- Range: 0 (no trigger), inputs 1 to 8
- Default: 0 (no trigger assigned)
- This is a profile setting

| Bit | Bit set to 0 | Bit set to 1 |
|--------|--|---|
| 0 to 3 | See below | |
| 4 to 5 | Reserved | |
| 6 | Unlatched trigger (trigger must remain true throughout the segment). | Latched trigger (trigger must be true at beginning of segment). |
| 7 | Active Off | Active On |

Bits 0 to 3 determine the input number for the trigger

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Input Number |
|-------|-------|-------|-------|--------------|
| 0 | 0 | 0 | 0 | No trigger |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 2 |
| 0 | 0 | 1 | 1 | 3 |
| 0 | 1 | 0 | 0 | 4 |
| 0 | 1 | 0 | 1 | 5 |
| 0 | 1 | 1 | 0 | 6 |
| 0 | 1 | 1 | 1 | 7 |
| 1 | 0 | 0 | 0 | 8 |

Segment Events and Event States (57)

This parameter holds the event output number and event state for a profile segment. Users can designate up to four events per segment; each event takes 1 byte. Events are saved in memory as indicated below.

Seg(1) Event(1), Seg(1) Event(2), Seg(1) Event(3), Seg(1) Event(4), Seg(2) Event(1), etc.

- Range: Outputs 1 to 34 (except outputs used for control or for the SDAC clock)
- Default: 0 (no event assigned)
- This is a profile setting

Bit map of the event bytes:

| Bit | Bit set to 0 | Bit set to 1 |
|--------|---|--------------------|
| 0 to 5 | Determines the output number for an event | |
| 6 | Reserved | |
| 7 | Event is active Off | Event is active On |

Bits 0 to 5 determine the output number for an event

| Result | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|------------------------------------|-------|-------|-------|-------|-------|-------|
| No events | 0 | 0 | 0 | 0 | 0 | 0 |
| Digital output 1 is an event | 0 | 0 | 0 | 0 | 0 | 1 |
| Digital output 2 is an event | 0 | 0 | 0 | 0 | 1 | 0 |
| Digital output 3 is an event | 0 | 0 | 0 | 0 | 1 | 1 |
| Digital outputs 4 to 33 are events | ... | ... | ... | ... | ... | ... |
| Digital output 34 is an event | 1 | 0 | 0 | 0 | 1 | 0 |

Segment Time (58)

This parameter represents the duration of a segment, in the time units selected elsewhere (hours and minutes or minutes and seconds).

- Range: 0 to 59999
- Default: 0
- This is a profile setting

Tolerance (59)

The tolerance parameter represents an user-defined allowable deviation from setpoint. If the process goes above a positive tolerance value or below a negative tolerance value, it is considered out of tolerance. See also Tolerance Alarm Time (50). This value is also affected by precision.

- Range: -99 to 99
- Default: 0
- This is a profile setting

Ramp/Soak Flags (60)

This parameter is a byte register for each loop that has a ramp-soak profile assigned to it.

Bits in the ramp/soak flag byte:

| Bit | Bit set to 0 | Bit set to 1 |
|--------|---|---|
| 0 | The segment is within the tolerance time period. | The segment is out of tolerance time period. |
| 1* | The segment does not require alarm acknowledgment. (Segment may be out of tolerance but is not in tolerance alarm.) | The user must acknowledge alarm. (Segment may be within tolerance.) |
| 2 | Reserved | |
| 3 | The profile is in trigger wait state | The profile is not in trigger wait state (remote hold) |
| 4 to 7 | Reserved | |

*The controller toggles a bit in the data-changed register to notify high level software that the alarm status has changed.

Output Limit (61)

This parameter sets a limit on the output power percentage on either heat or cool outputs for any loop. The output limit works with the output limit time. While the limit is in effect, the output level will never exceed the specified limit.

- Range: 0 to 32700 (0 to 100%). Setting the limit to 100% (32700) disables it
- Default: 32700 (100%), or disabled

Output Limit Time (62)

This parameter describes the time span that the output limit is in effect.

- Range: 0 to 999 seconds. Setting the output limit time to 0 makes the output limit continuous; setting the output limit time from 1 to 999 seconds gives a range from 1 second to about 16 minutes
- Default: 0 seconds (continuous output limit)

The output limit only affects loops in automatic control (AUTO). The time-out period is restarted whenever:

- A loop switches from manual to automatic control
- The controller restarts

Alarm_Control (63)

Setting a bit for an alarm in the Alarm_Control variable makes it a control alarm; clearing the bit makes it a standard alarm. This table explains the difference between standard alarms and control alarms. See Alarm_Status on page 40 for alarm bits.

| Function | Description |
|----------|--|
| Alarm | When an alarm occurs: <ul style="list-style-type: none">• The controller's display changes to the loop that's in alarm.• An alarm message flashes on the display. The alarm's digital output activates.• The global alarm output activates.• The operator must press the Alarm Ack key to stop the flashing message and deactivate the global alarm before the controller will accept other input from the keypad. |
| Control | When the process is beyond the alarm set point or deviation band: <ul style="list-style-type: none">• The alarm digital output activates on alarm and deactivates when the loop goes out of alarm.• The global alarm output does not activate.• Users do not have to acknowledge control alarms by pressing the alarm Ack key. |

Alarm_Acknowledge (64)

When an alarm occurs and it is not a control alarm, the corresponding Alarm_Acknowledge bit is set. Clearing the bit acknowledges the alarm. See Alarm_Status on page 40 for alarm bits.

Alarm_Mask (65)

When users turn alarms on or off from the front panel keypad, they are really setting or clearing the Alarm_Mask variable. (When an alarm is turned on, its Alarm_Mask bit is set.)

Setting Alarm_Mask to TRUE does not trip or clear alarms; instead, it lets the alarm checking routine check for them. The controller does not set or clear Alarm_Mask by itself; users edit this bit through the front panel keypad or in host software.

See Alarm_Status on page 40 for alarm bits.

Alarm_Enable (66)

The Alarm_Enable variable is like the Alarm_Mask variable, except it is a temporary mask. An alarm will not trip until its Alarm_Enable bit becomes true and the following conditions are also met:

- The alarm condition occurs
- The Alarm_Mask bit is set

This variable is currently used for deviation and process alarms. Process alarms are automatically enabled at startup. If a deviation alarm is turned on (its Alarm_Mask bit is set), then the alarm is disabled until the process variable comes within the deviation range:

$$(PV \leq SP + DEV - DB)$$

to

$$(PV \geq SP - DEV + DB)$$

If the process variable is already in this range, the Alarm_Enable bit is immediately set.

See Alarm_Status on page 40 for alarm bits.

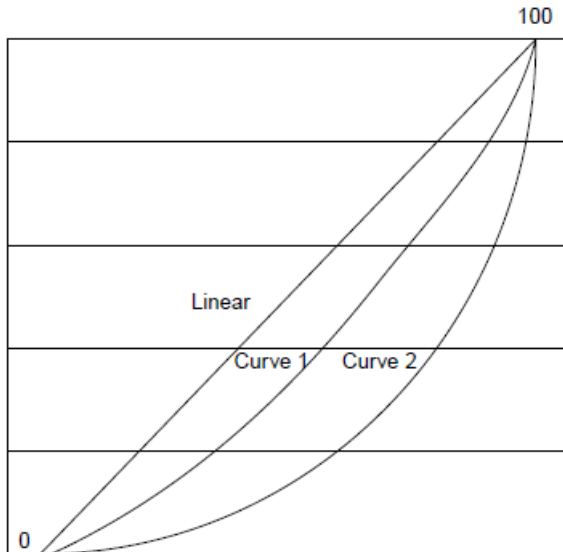
Output Override Percentage (67)

When a sensor failure occurs and the loop is in Automatic mode, the loop switches to Manual mode at the output override percentage. If users have configured an output override digital input, they set every loop to the output override percentage when they change the polarity of the output override's digital input to the active state.

- Range: 0 to 32700 (0 to 100%)
- Default: 0

Output Linearity Curve (69)

This parameter lets users set heat and cool outputs to one of two nonlinear output curves, or to Linear (no curve).



- Range: Users can select 0 for no curve (linear), 1 for a slight curve, and 2 for a more pronounced curve
- Default: 0 (no curve)

SDAC Mode (70)

This parameter toggles the SDAC between current and voltage output.

- Range: 0 to 1, where 0 = voltage and 1 = current
- Default: 0

SDAC Low Value (71)

This parameter sets a low range value for the output device. This value must be less than the SDAC high value.

- Range: 0 to 999 for voltage outputs and 0 to 1999 for milliamp outputs
- Default: 000 (0.00 volts) or 400 (4.00 mA)

SDAC High Value (72)

This parameter sets a high range value for the output device. This value must be greater than the SDAC low value.

- Range: 1 to 1000 for voltage outputs and 1 to 2000 for current outputs. (The controller displays this value with a fixed decimal place, so the user sees a range of 1 to 10.00 for voltage inputs and 1 to 20.00 for current outputs.)
- Default: 1000 (10.00 volts) or 2000 (20.00 mA).

Save Setup to Job (73)

This parameter saves the current setup to one of eight jobs. It is used as a command; it is not used as data. When this parameter is set to a non-zero value, the current job is saved to that job number. The

parameter is then immediately reset to zero. The job is only saved once, when the controller receives this command. To resave the job, resend the command.

- Range: 1 to 8
- Default: 0 (no job)

Input Filter (74)

The adjustable input filter dampens the analog input response.

- Range: 0 to 255 scans (setting the number of scans to 0 disables the filter.)
- Default: 3 Scans

Loop Alarm Delay (75)

Delays process. deviation and failed sensor alarms for a loop until the alarm condition has been continuously present for longer than the specified alarm delay time.

- Range: 0 to 255
- Default: 0 seconds (no loop alarm delay)

Loop Names (77)

This parameter assigns a two-character name to each loop in CLS200 and MLS300 controllers.

- Range: 0 to 9, A to Z, °, /, %
- Default: The loop's number

T/C Failure Detection Flags (78)

This parameter determines the thermocouple failure detection scheme for each loop in a CLS200 controller. The following bits enable (bit set) or disable (bit cleared) the action.

| Bit | Bit set to 0 | Bit set to 1 |
|--------|--|---|
| 0 | Reversed thermocouple detection disabled. | Reversed thermocouple detection enabled. |
| 1 | Heat thermocouple break output averaging disabled. | Heat thermocouple break output averaging enabled. |
| 2 | Cool thermocouple break output averaging disabled. | Cool thermocouple break output averaging enabled. |
| 3 to 7 | Reserved | |

- Default: all bits cleared (i.e. all disabled)

Channel Name (78)

This parameter assigns an eight-character name to each channel in the CAS200. For the CLS200 or MLS300 see Loop Names (77).

Restore PID Digital Input (79)

This parameter specifies the digital input to restore the PID control from MAN mode to AUTO mode for each loop. When a thermocouple break occurs and the loop is in AUTO mode, the controller sets the loop to MAN mode. If the digital input is other than 0 and the input is low, the loop goes back to AUTO mode when the thermocouple-break condition clears.

- Range: 0 (none) to 8
- Default: 0 (none)

Manufacturing Test (80)

This parameter is used for manufacturing to perform specific tests on each bit.

| Bit | Bit set to 0 | Bit set to 1 |
|---------|--|---|
| 0 | Watchdog timer test disabled. | Watchdog timer test enabled. |
| 1 | Halt mux scanning for calibration test disabled. | Halt mux scanning for calibration test enabled. |
| 2 to 15 | Reserved | |

CAUTION: This command should only be used when it is necessary to perform the above tests. Use of this command in normal operation can result in loss of data.

PV Retransmit Primary Loop Number (81)

This parameter specifies the primary loop number for obtaining the process variable (PV) from the current loop or another loop. Setting this to 0 disables PV retransmit for the loop.

- Range: 0 (none) to MAX_CH
- Default: 0 (none)

PV Retransmit Maximum Input (82)

This parameter specifies the maximum Process Variable input allowed for PV retransmit calculation. This value must not be lower than the minimum input (command 84).

- Range: -999 to 9999 (depending on precision in primary loop)
- Default: 14000 (1400°F) for a J-type thermocouple

PV Retransmit Maximum Output (83)

This parameter specifies the maximum output (%) allowed for Process Variable retransmit calculation. Once specified, the output will not exceed this percentage. This number must not be lower than the minimum output (command 85).

- Range: 0 to 100
- Default: 100%

PV Retransmit Minimum Input (84)

This parameter specifies the minimum PV input allowed for PV retransmit calculation. This number must not be higher than the maximum input (command 82).

- Range: -999 to 9999 (depending on precision in primary loop)
- Default: -3500 (-350°F) for J-type thermocouple

PV Retransmit Minimum Output (85)

This parameter specifies the minimum output (%) allowed for PV retransmit calculation. Once set, the output will not drop below this percentage. This number must not be higher than the maximum output (command 83).

- Range: 0 to 100
- Default: 0%

Cascade Primary Loop Number (86)

This parameter specifies the cascade primary loop number to obtain the output values. This number cannot be the same as the current loop. Setting this to 0 disables the cascade feature for this loop.

- Range: 0 (none) to MAX_CH
- Default: 0 (none)

Cascade Base Setpoint (87)

This parameter specifies the setpoint used as an offset for each loop.

- Range: -999 to 9999 (depending on the precision in the primary loop)
- Default: 250 (25°F) for a J-type thermocouple

Cascade Minimum Setpoint (88)

This parameter specifies the minimum allowable setpoint in the cascade calculation. The resulting setpoint will not go lower than this. This number must not be higher than the maximum setpoint (command 89).

- Range: -999 to 9999 (depending on the precision in the primary loop)
- Default: 250 (25°F) for a J-type thermocouple

Cascade Maximum Setpoint (89)

This parameter specifies the maximum allowable setpoint in the cascade calculation. The resulting setpoint will not go higher than this. This number must not be lower than the minimum setpoint (command 88).

- Range: -999 to 9999 (depending on the precision in the primary loop)
- Default: 250 (25°F) for a J-type thermocouple

Cascade Heat/Cool Span (90)

This parameter is multiplied by the heat and cool outputs of the primary loop in the cascade calculation.

- Range: -9999 to 9999
- Default: 0

Ratio Control Master Loop Number (91)

This parameter specifies the ratio control master loop number to obtain the process variable. This number cannot be the same as the current loop. Setting this to 0 disables the ratio control feature for this loop.

- Range: 0 (none) to MAX_CH

- Default: 0 (none)

Ratio Control Minimum Setpoint (92)

This parameter specifies the minimum allowable setpoint in the ratio control calculation. The resulting setpoint will not go lower than this. This number must not be higher than the maximum setpoint (command 89).

- Range: –999 to 9999 (depending on the precision in the master loop)
- Default: 250 (25°F) for a J-type thermocouple

Ratio Control Maximum Setpoint (93)

This parameter specifies the maximum allowable setpoint in the ratio control calculation. The resulting setpoint will not go higher than this. This number must not be lower than the minimum setpoint (command 88).

- Range: –999 to 9999 (depending on the precision in the master loop)
- Default: 250 (25°F) for a J-type thermocouple

Ratio Control Control Ratio (94)

This parameter is multiplied by the master loop process variable in the ratio control calculation. These values are multiplied by 10 to retain 0.1 ratio precision.

- Range: 1 to 9999 (control ratio 0.1 to 999.9)
- Default: 10 (control ratio 1.0)

Ratio Control Setpoint Differential (95)

This parameter specifies the setpoint used as an offset for each loop.

- Range: –999 to 9999 (depending on the precision in the master loop)
- Default: 0

Loop Status (96)

This parameter specifies a character status indicating if the loop is in manual or automatic mode. Autotuning and ramp-soak status are also included. The characters are identical to that shown on the controller's bar display. This is an expansion of commands 4 and 46.

- Range: 'A' to 'Z' (uppercase letters), status are currently defined as:
65 = A = Automatic
77 = M = Manual 84 = T = Tuning
83 = S = Ramp/Soak Ready state (Start) 82 = R = Ramp/Soak Running
72 = H = Ramp/Soak Holding
87 = W = Ramp/Soak trigger Wait state 79 = O = Ramp/Soak Out of Tolerance
- Default: 77 (Manual)

Output Type/Disable (97)

This parameter specifies the output type (if not disabled) for the heat and cool outputs. Setting this to 255 indicates the output is disabled. Any other value indicates an enabled output. This is an expansion of command 4.

- Range: 0 to 255, output types are currently defined as:

0 = Time Proportioning
1 = DZC
2 = Reserved
3 = On/Off
4 = SDAC
5 = 3P DZC
255 = Disabled

- Default: 0 for heat outputs, 255 for cool outputs

Output Reverse/Direct (98)

This parameter specifies the heat and cool output control action as either Reverse or Direct. This is an expansion of command 4.

- Range: 0 or 1
0 = Reverse
1 = Direct
- Default: 0 for heat outputs, 1 for cool outputs

Controller Type (99)

This parameter specifies the controller type. This is an expansion of command 30.

- Range: 0 to 3:
0 = Controller has 4 loops
1 = Controller has 8 loops
2 = Controller has 16 loops
3 = Controller has 32 loops

Ramp/Soak Profile Number (100)

This parameter specifies the assigned profile number for each loop. Setting this to 255 indicates no profile is assigned for that loop. This is an expansion of command 46.

- Range: 0 to 255
- Default: 255 (no profile assigned)

Reference letters are assigned as follows: 0 = A, 1 = B, 2 = C, etc.

Controller Address (101)

This parameter stores the controller's network address. Changes to this parameter are effective the next time the controller powers up.

- Range: 1 to 247
- Default: 1

In the Anafaze/AB protocol, the controller responds to the address stored in this parameter plus 7. For example, if this parameter is set to 1 in a controller, that controller responds to messages with a DST byte value of 8 (1 plus 7).

Baud Rate (102)

This parameter stores the baud rate at which communications will operate. Changes to this parameter are effective the next time the controller powers up.

- Range: 0 to 2, with these rates available:
0 = 9600
1 = 2400
2 = 19200
- Default: 19200

Ready Events (103)

This parameter is accessible using the Modbus® RTU protocol only. It describes the ready segment's output states for all outputs that are not used for control or for the SDAC clock. When the loop goes to the ready state, these outputs assume the state specified by this parameter.

- Range: 0 (off) or 1 (on).
- Default: 0 (off).

Five consecutive Modbus® registers hold the ready events for each ramp and soak profile. The table below indicates in which bit each ready state event is stored for a profile. The events correspond directly to the controller's outputs. For example, output 20 is set by setting bit 3 in register 2. Note that eight registers are allocated for each profile.

| Rgstr. | Bits 8 to 15 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0* |
|--------|--------------|-------|-------|-------|-------|-------|-------|-------|--------|
| 0 | Reserved | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 1 | Reserved | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 |
| 2 | Reserved | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 |
| 3 | Reserved | 32 | 31 | 30 | 29 | 28 | 27 | 26 | 25 |
| 4 | Reserved | Rsvd. | Rsvd. | Rsvd. | Rsvd. | Rsvd. | Rsvd. | 34 | 33 |
| 5 | Reserved | Rsvd. | Rsvd. | Rsvd. | Rsvd. | Rsvd. | Rsvd. | Rsvd. | Rsvd. |
| 6 | Reserved | Rsvd. | Rsvd. | Rsvd. | Rsvd. | Rsvd. | Rsvd. | Rsvd. | Rsvd. |
| 7 | Reserved | Rsvd. | Rsvd. | Rsvd. | Rsvd. | Rsvd. | Rsvd. | Rsvd. | Rsvd. |

*Bit 0 is the least significant bit.

Glossary

ACK (Acknowledge) A control code that signals that a syntactically correct message packet has been received.

Address See **Data table address** and **Device address**.

ASCII American Standard Code for Information Interchange. A code that assigns numeric values to characters.

Baud rate The rate at which data is transmitted over a line, measured in bits per second.

BCC Block Check Character. A method of error checking for packets.

| | |
|----------------------------------|---|
| Binary numbers | Numbers in base 2, where each digit has one or two distinct values (0 or 1). |
| Bit | An abbreviation for binary digit. (A binary digit can have one of two distinct values, 0 or 1). A group of four bits makes up a nibble. A group of eight bits makes up a byte. |
| Burst error | An erroneous series of 1's or 0's in the communications bit stream. |
| Byte | A group of eight bits. |
| Cleared, bit | When a bit is "cleared," it is set to False or 0. When a bit is "set," it is set to True or 1. |
| CLS200 | Compact Loop System. A Watlow PID controller with 4, 8 or 16 loops. |
| CRC | Cyclic Redundancy Check. A method of error checking for packets. |
| Data table address | The logical address where data is located. |
| Deadband, alarm | The range through which a process variable must travel from the alarm setpoint toward the process setpoint before an alarm clears. |
| Derivative control action | A control action in which the output value is proportional to the rate of change of the error between the process variable and setpoint. |
| Deviation alarm | The deviation alarm value is an absolute value that is always relative to the setpoint. The process is in Alarm in the following cases: Above (setpoint + deviation value) Below (setpoint – deviation value) |
| Device address | The communications address assigned to the controller or host software. |
| DLE | Data Line Escape. An escape code that signals that a control code follows it. |
| Double-bit error | An error in which two bits in a packet are incorrect. |
| Duplex, full | Communications in which each node can simultaneously transmit and receive. |
| ENQ | Enquiry. A control code that the software sends to the controller, asking it to resend the last ACK or NAK. |
| ETX | End Text. A control code that signals the end of a transmission. |
| Gain | The parameter that users can alter through host software to set the |

proportional control action. The gain is a unitless number that is inversely proportional to the proportional band. (See also **Proportional Band**)

Hexadecimal numbers

Numbers in base 16 where each digit has one of 16 possible values (0 to F).

Integral control action

Control action in which the output is proportional to the time integral of the difference between the setpoint and process variable. (In other words, the rate of change of the output is proportional to the input.)

Job A set of operating conditions (setpoints, alarms, PID constants, etc.).

Linear input CLS200 controllers have an optional linear input type that allows you to scale raw input readings to the engineering units of your process.

LSB Least Significant Byte. For all two-byte data types, the LSB is transmitted before the MSB (Most Significant Byte) in the packet.

MSB Most Significant Byte. For all two-byte data types, the MSB is transmitted after the LSB (Less Significant Byte) in the packet.

NAK Not Acknowledged. A control code that signals that a syntactically incorrect message packet has been received.

Nibble A group of four bits or half a byte.

Packet A packet consists of a sequence of bytes in a specific format; it can be as large as 256 bytes of data. Information is exchanged between host software and the controller in packets.

Parity Error detection coding for serial communications. CLS200 controllers use no parity.

Precision parameter

Indicates the decimal format for controller parameters such as process variable, setpoint, etc.

Proportional Band (PB)

The parameter that users alter from the front panel to set the proportional control action, expressed in engineering units of the process variable.

Proportional control action Control action with a continuous, linear relationship between the output value and the difference between the setpoint and the process variable.

Protocol A set of rules for the timing and format of messages in the network.

Pulse input An input for CLS200 controllers that measures a digital pulse signal.

| | |
|-----------------------------|---|
| Recipe | See Job . |
| SDAC | Serial Digital Analog Converter. A high-precision digital-to-analog converter for control outputs. |
| Serial communication | Communication between a computer and another device (controller, for example). CLS200 controllers supports two hardware networks: EIA/TIA-232 for single device at short distances, and EIA/TIA-485 for multiple devices at longer distances. |
| Set, bit | See Cleared, bit. |
| Single-bit error | An error in which only one bit is incorrect. |
| Stop bit | The last part of a character that assures that the next start element is recognized. |
| STX | Start Text. A control code that signals the beginning of a transmission. |
| TI | Time of Integral, or Reset. The parameter that users alter to set the integral control action, expressed in seconds per repeat. |
| TD | Derivative term, or Rate. The parameter that users alter to set the derivative control action, expressed in seconds. |