

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *WinSystems PCM-518* at our website: ***Click HERE***

INSTRUCTION MANUAL

Sensoray Model 518
PC104 Sensor Coprocessor

Revised June 9, 1997

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table of Contents

Chapter 1: Basics . . 1

Limited Warranty . . 1

Special Handling Instructions . . 1

Chapter 2: Introduction . . 2

Functional Description . . 2

Specifications . . 3, 4

Chapter 3: Hardware Configuration . . 5

Interrupts . . 5

Port Mapping . . 5, 6

Hardware Filters. . 7

Chapter 4: Programming . . 8

Programming Model . . 8

Status Register . . 8

Control Register . . 9

Sample Low-level Drivers . . 10

Commands . . 12

Declare Channel Sensor . . 13

Set Filter Time Constant . . 14

Read Board Temperature . . 15

Read Channel Data . . 16

Read Data From All Channels . . 17

Set Channel Alarm Limits . . 18

Set Open Sensor Data Values . . 19

Read Alarm Flags . . 20

Set Gage Zero . . 21

Set Gage Span . . 22

Tare Gage . . 23

Read Gage Calibration . . 24

Set Gage Calibration . . 25

High Speed Mode . . 26

Low Power Standby . . 27

Read Firmware Version . . 28

Read Product I.D. . . 29

Set Coefficients . . 30

Calibrate 518 . . 32

Sensor Tables . . 33

Back Compatibility . . 34

Chapter 5: Sensor Connections . . 35

RTDs and Resistors . . 35

Thermocouple and DC Voltage . . 36

Thermistor/Custom Resistive Sensors . . 37

Strain and Pressure Gages . . 38

4 to 20 mA Current Loops . . 38

Chapter 6: Sensor Specifics . . 39

User-defined Resistive Sensors . . 39

Application Example . . 39

Thermocouples . . 40

Verifying Thermocouple Calibration . . 41

Strain and Pressure Gages . . 42

Measurement Resolution . . 42

Calibration . . 43

Chapter 7: Theory of Operation . . 45

Software . . 45

Scanner . . 45

Post-processor . . 45

Command Processor . . 46

Hardware . . 46

Reference Section . . 46

Measurement Section . . 46

Excitation Section . . 47

Timing . . 47

Scan Rate . . 47

Communication Latency . . 48

Processor Speed . . 49

Chapter 8: Command Summary . . 51

Chapter 9: Schematics . . 55

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter

1

Basics

Limited Warranty

Sensoray Company, Incorporated (Sensoray) warrants the model 518 hardware to be free from defects in material and workmanship and perform to applicable published Sensoray specifications for two years from the date of shipment to purchaser. Sensoray will, at its option, repair or replace equipment that proves to be defective during the warranty period. This warranty includes parts and labor.

The warranty provided herein does not cover equipment subjected to abuse, misuse, accident, alteration, neglect, or unauthorized repair or installation. Sensoray shall have the right of final determination as to the existence and cause of defect.

As for items repaired or replaced under warranty, the warranty shall continue in effect for the remainder of the original warranty period, or for ninety days following date of shipment by Sensoray of the repaired or replaced part, whichever period is longer.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. Sensoray will pay the shipping costs of returning to the owner parts which are covered by warranty.

Sensoray believes that the information in this manual is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, Sensoray reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult Sensoray if errors are suspected. In no event shall Sensoray be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, SENSORAY MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF SENSORAY SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. SENSORAY WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

Special Handling Instructions

The Model 518 circuit board contains CMOS circuitry that is sensitive to Electostatic Discharge (ESD). Special care should be taken in handling, transporting, and installing the 518 to prevent ESD damage to the board. In particular:

1. Do not remove the 518 from its protective antistatic bag until you are ready to configure the board for installation.
2. Handle the 518 only at grounded, ESD protected stations.
3. Remove power from the PC104 bus before installing or removing the 518 circuit board

All brand, product, and company names are trademarks or registered trademarks of their respective owners.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter

2

Introduction

Functional Description

The 518 interfaces eight process sensors directly to the PC104 bus. Each of the eight sensor channels may be independently configured (via software) to accept thermocouples, RTD's, strain gages, thermistors, resistors, 4 to 20 milliamp current loops, or DC voltage inputs. Each sensor channel makes use of differential inputs to allow for wide common mode variation.

Pulsed sensor excitation is provided for thermistors, resistors, RTD's, and strain gages. Pulsed excitation minimizes sensor self-heating effects and helps to reduce board power consumption. Excitation signals are electrically separated from channel differential inputs so that three- or four-wire sensor connections may be implemented.

Cold junction compensation is provided for thermocouples, and all inputs are protected from high differential and common mode voltages.

The 518's onboard microprocessor continuously scans the eight channels. As each channel is scanned, the sensor signal is digitized, linearized, and converted to engineering units appropriate for the sensor type. The most recent sensor data from each channel is immediately accessible to the host processor.

Alarm limits are individually programmable (via software) for each channel. A status flag is set when any channel limit violation occurs, eliminating the need for the host processor to poll sensor data for limit violation detection.

Communication with the host processor takes place over two contiguous I/O ports which may be mapped anywhere in the system I/O address space. Communication and scanning tasks execute in an interrupt-driven multitasking local environment to minimize communication latency while maximizing channel scan throughput.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Specifications

Table 1: General Specifications

Parameter	Specification
Input power	+/-12VDC @35mA, typical +5VDC @ 150mA, typical
Operating temperature	-25C to 85C
CMRR	80dB, minimum @ CMV < 5V
A/D converter	16-bit integrating
Measurement time per channel	16.67 or 8.0 milliseconds
Total time slot per channel	22 or 13 milliseconds
Strain/Pressure gage excitation	10VDC, pulsed
Resistive sensor excitation	1.3mADC, pulsed (0 to 400 ohms) 5VDC in series w/4K (other ranges)
Input protection	Up to 70VAC CMV < 2 seconds
Data format	16-bit 2’s complement
PC104 Interface	8-bit module, stackthrough bus

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 2: Sensor Specifications

Sensor Type	Range	Resolution	Accuracy
Thermocouple B C E J K N T S R	 0C to 1820C 0C to 1820C -270C to 990C -210C to 760C -270C to 1360C -270C to 1347C -270C to 400C 0C to 1760C 0C to 1760C	0.1C	0.2C
Thermistor Omega 44006 or 44031	-55C to 145C	0.01C	0.05C
RTD 10 ohm copper 100 ohm platinum 385 or392 120 ohm nickel T/R 2	 0C to 119C -200C to 800C -200Cto409.5875C -100F to 482F	 0.1C 0.05C 0.0125C 0.1F	 0.6C 0.2C 0.2C 0.3F
Strain/Pressure (4-wire bridge) 120 ohm	-100 to +100 mV (10V excitation)	5uV	30uV
DC Voltage Range 1 Range 2 Range 3	 -5 to +5V -500 to +500mV -100 to +100mV	 200uV 20uV 5uV	 400uV 30uV 30uV
Current Loop Range 1	 4 to 20mA	 0.01%	 0.02%
Resistance Range 1 Range 2 Range 3	 0 to 400 ohm 0 to 3 Kohm 0 to 600 Kohm	 0.02 ohm 0.125 ohm 31 ohm	 0.04 ohm 0.25 ohm 130 ohm

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter

3

Hardware Configuration

The coprocessor board requires installation of shorting shunts to select hardware options. Hardware configuration must be completed before the coprocessor can be programmed.:

Table 3: default configuration from factory

Option	Default
Base address	2B0 hex
PC104 Interrupts	Disabled
Channel Filters	Disabled

Interrupts

The 518 may be configured to interrupt the host processor in response to various alarm and interprocessor communication events. Enabling and disabling of these individual interrupt sources is accomplished by means of an onboard interrupt control register. All of these interrupt sources are routed to a single PC104 bus interrupt request line.

An option shunt must be installed to select the appropriate interrupt level if you will be using interrupts in conjunction with 518 operation. Any interrupt level from IRQ2 to IRQ7 may be selected. Choose an interrupt level that does not conflict with other devices in your system, then install a shunt at that position on the 518 board.

It is not necessary to install any IRQ shunts if you will not be using interrupts.

Port Mapping

The 518 may be mapped to any four-byte address block within the range 000 to 3FF hex. Although the board occupies a four-byte block of I/O space, it uses only the first two address locations in the block.

To avoid addressing conflicts, you must not map the 518 into any address range occupied by other devices. Similar to many other I/O cards, the 518 does not decode the full PC104 16-bit I/O address -- only the low ten address bits are decoded. As a consequence, “images” of the 518 will appear throughout the 16-bit address range at intervals of 400H bytes. You should ensure that these images do not conflict with other devices.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Option shunts E1 through E8 are used to select the 518 base address. Shunts are factory set to locate the board at base address 2B0H. This address should not conflict with any standard I/O address assignments. If you require a different base address, use the following table to determine the correct shunt programming for your target base address.

Table 4: I/O address jumpers

Nibble Value	High Byte		Low Address Byte					
	Low Nibble		High Nibble				Low Nibble	
	E8	E7	E6	E5	E4	E3	E2	E1
0	Install	Install	Install	Install	Install	Install	Install	Install
1	Install	Remove	Install	Install	Install	Remove		
2	Remove	Install	Install	Install	Remove	Install		
3	Remove	Remove	Install	Install	Remove	Remove		
4			Install	Remove	Install	Install	Install	Remove
5			Install	Remove	Install	Remove		
6			Install	Remove	Remove	Install		
7			Install	Remove	Remove	Remove		
8			Remove	Install	Install	Install	Remove	Install
9			Remove	Install	Install	Remove		
A			Remove	Install	Remove	Install		
B			Remove	Install	Remove	Remove		
C			Remove	Remove	Install	Install	Remove	Remove
D			Remove	Remove	Install	Remove		
E			Remove	Remove	Remove	Install		
F			Remove	Remove	Remove	Remove		

Example: select 390 hex as the 518's base port address. In this case, the high address byte is "3". From the above table it can be seen that shunts E8 and E7 should be removed. The low address byte, "90", consists of high nibble "9" and low nibble "0". From the above table, the high nibble value "9" requires shunts E5 and E4 to be installed and shunts E6 and E3 to be removed, while low nibble "0" requires shunts E2 and E1 to be installed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Hardware Filters

Each channel is provided with a hardware filter which may be used to reduce thermocouple and aid in detecting open sensor conditions. This feature is in addition to the software filter algorithm.

Option shunts E9 through E24 are used to enable the hardware filters. A pair of shunts is assigned to each channel. To enable a channel's filter, you must install both shunts. To disable a channel's filter, you must remove both shunts. You should enable the filter only for channels that are connected to thermocouples.The following table shows shunt assignments for all channels:

Table 5:

CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
E9	E10	E11	E12	E13	E14	E15	E16
E17	E18	E19	E20	E21	E22	E23	E24

For example, install shunts E12 and E20 to enable the filter for channel 3.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter

4 Programming

Programming of the 518 is achieved via a set of built-in commands. Commands are sent from the PC104 master (aka “host”) to the 518. Some commands cause responses from the 518; these responses are sent from 518 to the host. This chapter explains the mechanism behind this bidirectional communication and describes the coprocessor command set in detail.

Programming Model

The coprocessor occupies four consecutive addresses in the host’s I/O space, of which only the lower two addresses are actually used. Both of these addresses may be written to and read from, but each address has a distinctly different function for read and write operations.

The base port (518 board low port address) forms the data path between 518 and host processors. Commands are sent to the 518 and command responses are passed back to the host through this port.

The data port (base address) consists of two hardware registers: *command* and *data*. When the host sends a command to the 518, it is really storing a byte in the command register. When the host reads a command response from the 518, it is reading a byte from the data register. Because the command register is write-only and the data register read-only, they are able to share the same I/O address.

The control port (base address + 1) consists of the read-only status register and write-only control register. As the name implies, the status register supplies coprocessor status information to the host. The control register is used to invoke 518 soft resets and to enable and disable host interrupts.

Table 5: Programming model

Port Address	Read	Write
Base + 0: Data	Data Register	Command Register
Base + 1: Control	Status Register	Control Register

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Status Register

The status register provides the host with information used for coprocessor status monitoring and communication handshake control. When the host reads the status register, a byte of the following form is returned:

Table 6: Status Register

D7	D6	D5	D4	D3	D2	D1	D0
CRMT	DAV	ALARM	FAULT	X	X	X	X

Status register bits have the following meanings:

Table 7: Status Register bit functions

Bit	Function
CRMT	(Command Register eMpTy) indicates the 518 is ready to accept a command byte into its command register.
DAV	(Data AVailable) indicates that the 518 data register contains a data byte ready for access by the host.
ALARM	Indicates one or more programmed channel limits were exceeded.
FAULT	Indicates board reset in progress or board fault detected. In normal operation this bit goes active for approximately one-half second following a board reset. The FAULT bit reflects the state of the red LED fault indicator.

Before writing a byte to the command register the host must test the CRMT bit. When CRMT contains a logic 1, the command register is ready to accept a new command byte. The host should write to the command register only when CRMT contains a logic 1.

Similarly, the DAV status bit must be tested before reading a byte from the data register. When DAV contains a logic 1, a new byte is available in the data register for reading by the host.

Although these handshake rules are simple, failure to observe them will almost certainly result in communication errors.

IMPORTANT NOTE: The CRMT, DAV, and ALARM status bits are not valid when the FAULT bit is active. After resetting the 518 board (by means of either soft or hard reset), the host processor should not attempt to handshake to or from the 518 until the FAULT bit changes to the inactive (logic 0) state.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Control Register

The control register provides the means for host management of interrupts and coprocessor soft reset.

Table 8: Control Register

D7	D6	D5	D4	D3	D2	D1	D0
SET/$\overline{\text{CLR}}$	0	0	INT/$\overline{\text{RST}}$	0	ICMD	IDAT	IALARM

Control register bits have the following functions:

Table 9: Control Register bit functions

Name	Function
INT/$\overline{\text{RST}}$	Specifies the control function to be performed. When set to logic zero, the 518 board is reset and all other control register bits are ignored. When set to logic one, the other control register bits behave as described below.
SET/$\overline{\text{CLR}}$	Specifies whether selected interrupts are to be enabled or disabled. When set to logic one, all selected interrupts are enabled. When set to logic zero, selected interrupts are disabled.
ICMD	Selects CRMT interrupt. When set to logic one, the CRMT interrupt is enabled or disabled as determined by the state of the SET/$\overline{\text{CLR}}$ bit. When set to logic zero, the CRMT interrupt enable is unchanged. While enabled, the host will be interrupted whenever the status register CRMT bit is asserted.
IDAT	Selects DAV interrupt. When set to logic one, the DAV interrupt is enabled or disabled as determined by the state of the SET/$\overline{\text{CLR}}$ bit. When set to logic zero, the DAV interrupt enable is unchanged. While enabled, the host will be interrupted whenever the status register DAV bit is asserted.
IALARM	Selects ALARM interrupt. When set to logic one, the ALARM interrupt is enabled or disabled as determined by the state of the SET/$\overline{\text{CLR}}$ bit. When set to logic zero, the ALARM interrupt enable is unchanged. While enabled, the host will be interrupted whenever the status register ALARM bit is asserted.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Sample Low-level Drivers

We recommend that you incorporate procedures in your application software that will conceal the handshake protocol from higher host software levels. This will serve to simplify your programming requirements as well as reduce software maintenance costs.

This section contains sample software drivers that implement the required handshake protocol. In addition, the QuickBasic routines outlined below are referenced throughout this chapter in programming examples. Feel free to plagiarize and adapt these routines for your application.

```
; ********** 80x86 ASSEMBLY LANGUAGE DRIVERS **********
; This procedure sends a command byte to the 518.
; Entry:    DX points to 518 base port address.
;           AL contains command byte to send to 518.
XMIT:     PUSH AX             ; SAVE COMMAND BYTE TO BE SENT TO 518
          INC DX              ; POINT DX TO 518 STATUS PORT
XLOOP:    IN AL,DX            ; READ STATUS PORT
          TEST AL,80H         ; AND MASK "COMMAND REGISTER EMPTY" BIT
          JE XLOOP            ; LOOP BACK UNTIL REGISTER IS EMPTY
          DEC DX              ; POINT DX TO 518 COMMAND/DATA PORT
          POP AX              ; RESTORE COMMAND BYTE
          OUT DX,AL           ; AND WRITE IT INTO 518 COMMAND REGISTER
          RET                 ; EXIT COMMAND REGISTER DRIVER ;

; This procedure reads a data byte from the 518.
; Entry:    DX points to 518 base port address.
; Exit:     AL contains data byte read from 518.
RCV:      INC DX              ; POINT DX TO 518 STATUS PORT
RLOOP:    IN AL,DX            ; READ STATUS PORT
          TEST AL,40H         ; AND MASK "DATA AVAILABLE" BIT
          JE RLOOP            ; LOOP BACK UNTIL DATA IS AVAILABLE
          DEC DX              ; POINT DX TO 518 COMMAND/DATA PORT
          IN AL,DX            ; READ DATA FROM 518 DATA REGISTER
          RET                 ; EXIT DATA REGISTER DRIVER
```

```
' ********** QUICKBASIC DRIVERS **********
' This subprogram handshakes a byte to the 518 command register:
SUB Send518Byte (BasePort As Integer, CommandByte As Integer)
        DO : LOOP WHILE (INP(BasePort + 1) AND &H80) = 0        'wait for CRMT
        OUT (BasePort, CommandByte)                             'send the byte
END SUB

' This function handshakes a byte from the 518 data register:
FUNCTION Read518Byte% (BasePort As Integer)
        DO : LOOP WHILE (INP(BasePort + 1) AND &H40) = 0        'wait for DAV
        Read518Byte = INP(BasePort)                             'read the byte
END FUNCTION
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

You may find the following routines useful if you are writing a QuickBasic application. The first procedure reads a 16-bit value from the coprocessor, adjusting the sign if necessary. The second procedure sends a 16-bit value to the coprocessor. Note that these two routines make calls to the QuickBasic procedures listed previously.

```
'This function reads a 16-bit integer value from the 518.
FUNCTION Read518Word% (BasePort%)
        Lval& = Read518Byte%(BasePort%)                 'read high data byte from 518
        Lval& = Lval& * 256 + Read518Byte%(BasePort%)   'read & concatenate low byte
        IF Lval& > 32767 THEN Lval& = Lval& - 65536&    'adjust sign, if necessary
        Read518Word% = Lval&                            'exit with value
END FUNCTION

'This subprogram writes a 16-bit integer value to the 518.
SUB Send518Word (BasePort%, Value%)
        CALL Send518Byte (BasePort%, (Value% \ 256) AND &HFF)'send high byte to 518
        CALL Send518Byte (BasePort%, Value% AND &HFF)        'send low byte to 518
END SUB
```

```
/******************************** C DRIVERS ****************************/
/* ADSENDBYTE handshakes a command byte to the 518 command register */
void adSendByte ( short base_port, short cmd_byte )
{
        while ( ( inp( base_port + 1 ) & 128 ) == 0 ) ;      /* wait for CRMT */
        outp ( base_port, cmd_byte ) ;                       /* send the byte */
}
/* ADREADBYTE handshakes a data byte from the 518 data register */
short adReadByte ( short base_port )
{
        while ( ( inp ( base_port + 1 ) & 64 ) == 0 ) ;      /* wait for DAV */
        return ( inp ( base_port ) ) ;                       /* read the byte */
}
/* ADSENDWORD handshakes a 16-bit value to the 518 */
void adSendByte ( short base_port, short cmd_word )
{
        adSendByte ( base_port, cmd_word >> 8 ) ;      /* send high byte */
        adSendByte ( base_port, cmd_word ) ;           /* send low byte */
}
/* ADREADWORD handshakes a data byte from the 518 */
short adReadWord ( short base_port )
{
        short hiByte = adReadByte ( base_port ) << 8 ;  /* read high byte */
        return ( hiByte | adReadByte ( base_port ) ) ;  /* read & concatenate low byte */
}
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Commands

The 518 is accessed through a simple yet powerful built-in command set. Commands vary in length depending on the size of encapsulated data. In some cases invocation of a command will cause the coprocessor to return data to the host. In such cases the host must read the command response before issuing another command.

Table 10: Format of first command byte

D7	D6	D5	D4	D3	D2	D1	D0
OPCODE				CHANNEL			

Each command consists of at least one byte. The first byte of each command typically adheres to the above format. The opcode -- always present in the first command byte -- specifies the function to be performed. The channel number is required only in commands that address a particular channel.

These conventions are used in describing commands and responses:

1. The symbolic name *CHAN* represents a coprocessor channel number. Valid channel numbers range from 0 through 7, inclusive.
2. A *byte* is represented as a number or expression contained by parenthesis. For example, (16 + CHAN) represents a byte having a value equal to 16 plus a channel number.
3. A *byte string* is represented as an ordered sequence of bytes separated by commas. The sequence is defined from left to right. For example, the byte string (95+CHAN),(5),(0) contains three bytes.
4. A *command* consists of a byte string which is passed from host to 518 in the proper sequence.
5. A command *response* consists of a byte string which is passed from 518 to host in the proper sequence.

The remainder of this chapter discusses the coprocessor command set. Each command function is described along with the associated command and response byte strings. Sample program segments are listed to illustrate command usage. The program segments are all written in QuickBasic but can be easily converted to the programming language of your choice.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Declare Channel Sensor

This command declares the type of sensor connected to a channel. Typically, this command is executed once for each channel during the coprocessor initialization sequence (i.e., after a reset). Each execution of this command specifies the sensor type for a single channel, therefore eight invocations must occur to declare all eight channel sensors.

A two-byte sequence forms the command string. The first command byte contains the *Define Channel Sensor* opcode and desired channel number. The second command byte contains a "sensor definition code". Each sensor type is assigned a unique 8-bit sensor definition code. Refer to the *Sensor Tables* section for a complete list of sensor codes.

Undeclared channel sensor types default to the 5 volt input range. Any attempt to declare a "reserved for future use" sensor type will also result in the default sensor type. Declaring a sensor type as "Disabled" inhibits scanning of the corresponding channel, resulting in increased throughput for all remaining active channels.

COMMAND: **(16 + CHAN), (SENSOR DEFINITION CODE)**
RESPONSE: **NONE**

EXAMPLE:

```
' Configure channel 2 for interface to type K thermocouple.

Channel% = 2 Opcode% = 16        'DEFINE CHANNEL SENSOR command opcode
SensorCode% = 3                  'K thermocouple sensor definition code (from table)

CALL Send518Byte (BasePort%, Channel% + Opcode%)
CALL Send518Byte (BasePort%, SensorCode%)
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Set Filter Time Constant

This command establishes a software single-pole low-pass filter for the specified channel. The second command byte specifies a filter factor **F** which may have any value from 0 to 255, inclusive. The approximate relationship between F and filter percentage **P** is:

$$P = \frac{F}{2.55}$$

All filter constants default to 0 after a reset. This effectively disables the software filters, thereby maximizing the frequency response of all channels. A non-zero factor In electrically noisy environments may help to reduce measurement noise.

Rather than computing the theoretical value of filter constants, we recommend that you experiment with different filter constants to find the best response-time/noise-immunity tradeoff point. Some applications require knowledge of the filter time constant. This function describes the approximate relationship between time constant and filter factor:

$$t = \left(\frac{-N}{45 \ln\left(\frac{F}{256}\right)} \right)$$

where: t is the time constant in seconds
F is the filter factor in the range 0 to 255
N is the number of active channels.

COMMAND: **(96 + CHAN), (FILTER CONSTANT)**

RESPONSE: **NONE**

EXAMPLE:

```
' Set up channel 4 with a 12.5% low-pass filter. |

Channel% = 4
Opcode% = 96                                        'command opcode
FilterPercent% = 12.5                               'filter percentage
FilterConstant% = FilterPercent% * 2.55             'compute filter constant

CALL Send518Byte (BasePort%, Channel% + Opcode%)    'send channel & opcode to 518
CALL Send518Byte (BasePort%, FilterConstant%)       'send filter constant to 518
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Read Board Temperature

This command returns the temperature at the optional 7409TC termination board. Useful for monitoring the temperature of thermocouple reference junctions, this function is also a good debugging aid during 518 installation. The returned temperature value is scaled to 0.10 degrees C/bit.

If no 7409TC is connected to the 518, the *Read Board Temperature* command will return a meaningless value.

COMMAND: (64)

RESPONSE: (BOARD TEMP MSB), (BOARD TEMP LSB)

EXAMPLE:

```
' Read and display the 7409TC temperature in degrees F:

CALL Send518Byte (BasePort%, 64)          'send opcode to 518
Temp! = Read518Word (BasePort%) / 10!     'read response & convert to degrees C

DegreesF! = 1.8 * Temp! + 32.0            'convert to degrees F
PRINT DegreesF!                           'display temperature on CRT
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Read Channel Data

This command returns the most recently acquired sensor data from a channel. The returned integer value is scaled as a function of the channel sensor type. Regardless of sensor type, the returned value is always represented in 16-bit two's complement form. Refer to the *Sensor Tables* section for a listing of sensor scaling factors.

COMMAND: **(CHAN)**
RESPONSE: **(HIGH DATA), (LOW DATA)**

EXAMPLE

```
' Channel 6 is configured for +/- 5 volt input signal measurement.
' Read and display the voltage measured at channel 6.

Channel% = 6
Scalar! = 0.0005                          'from Sensor Tables, scalar = 500uV

CALL Send518byte (BasePort%, Channel%)    'send command to 518
ChanData% = Read518word (BasePort%)       'read scaled data from 518

V! = ChanData% * Scalar!                  'convert scaled data to volts
PRINT "Channel 6 data: "; V!; "Volts"     'display voltage
```

EXAMPLE:

```
' Channel 4 is configured for a type K thermocouple.
' Read and display the temperature measured at channel 4.

Scalar! = 0.17

CALL Send518byte (BasePort%, 4)
Chan4temp! = Scalar! * Read518word (BasePort%)

PRINT "Channel 4 data: "; Chan4temp!; "degrees Centigrade"
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Read Data From All Channels

This command returns the most recent sensor data from all eight channels, starting with channel 0. Data is returned from all eight channels, including disabled channels.

Sensor data is scaled as a function of the declared sensor type connected to each channel. Refer to the *Sensor Data* section for a list of sensor data scalars. In the case of disabled channels, the returned value is meaningless.

COMMAND: **(88)**

RESPONSE: **(CH0 MSB), (CH0 LSB),**
(CH1 MSB), (CH1 LSB),
(CH2 MSB), (CH2 LSB),
(CH3 MSB), (CH3 LSB),
(CH4 MSB), (CH4 LSB),
(CH5 MSB), (CH5 LSB),
(CH6 MSB), (CH6 LSB),
(CH7 MSB), (CH7 LSB)

EXAMPLE:

```
‘ Read sensor data from all eight 518 channels into a linear array.

SUB Read8Channels (BasePort%, DataArray%())

    CALL Send518byte (BasePort%, 88)‘send command to 518

    FOR Channel% = 0 TO 7          ‘read scaled data from 518 into array
        DataArray%(Channel%) = Read518word (BasePort%)
    NEXT Channel%

END SUB
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Set Channel Alarm Limits

This command declares upper and lower alarm limits for a channel. An alarm will ‘sound’ if channel sensor data strays outside of the specified limits. The alarm sounds by setting the ALARM bit in the status register and optionally interrupting the host.

Alarm limits assume default values when the coprocessor is reset. All low limits default to -32768, and all high limits default to 32767. These values effectively disable the alarm function.

To disable the lower limit, specify a lower limit of -32768. Similarly, declare an upper limit of 32767 to disable the upper limit.

When an alarm “sounds”, the violated limit is reset to the default value. This prevents the alarm from sounding again after the host has acknowledged the alarm. The host must reprogram the violated limit to re-arm it for monitoring by the coprocessor.

COMMAND: **(32 + CHAN),**
(HIGH LIMIT MSB), (HIGH LIMIT LSB),
(LOW LIMIT MSB), (LOW LIMIT LSB)

RESPONSE: **NONE**

EXAMPLE:

```
‘ Channel 7 is monitoring a K thermocouple connected to a plastic |
‘ extruder. The extruder temperature must fall between 400 and 450 |
‘ degrees C for proper operation. This code segment programs the |
‘ channel 7 alarm limits so that the 518 alarm flag will be raised |
‘ if the temperature strays outside operating limits. |

Chan% = 7
Opcode% = 32                                    ‘command opcode
Scalar! = 0.17                                  ‘K thermocouple data scalar (from table)
HiLim% = 450.0 / Scalar!                        ‘compute high alarm limit value
LoLim% = 400.0 / Scalar!                        ‘compute low alarm limit value

CALL Send518byte (BasePort%, Chan% + Opcode%) ‘send channel & opcode to 518
CALL Send518word (BasePort%, HiLim%)          ‘send high limit to 518
CALL Send518word (BasePort%, LoLim%)          ‘send low limit to 518
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Set Open Sensor Data Values

This command establishes open sensor data values for all channels. The second command byte contains eight flags, one for each channel. Flags are mapped to channels as follows: bit 7 (most significant) to channel 7, bit 6 to channel 6, etc.

If a flag is set (logic 1) and the corresponding sensor is open, the value 32767 will replace the sensor data. On the other hand, if a flag is reset and the sensor is open, the value -32768 will replace the sensor data.

This function is useful for triggering alarms when open sensors are detected, and for forcing the desired system response to fault conditions in closed-loop process control applications.

COMMAND: (80), (OPEN SENSOR DATA FLAGS)
RESPONSE: NONE

EXAMPLE:

```
‘ A proportional controller uses a J thermocouple on channel 1 to monitor |
‘ furnace temperature. If the thermocouple fails, the 518 should indicate |
‘ a high temperature on channel 1 to ensure that the controller turns off |
‘ the furnace heating element. This code segment sets up channel 1 to |
‘ fail high (all other channels are set up to fail low). |

CALL Send518byte (80)       ‘send command to 518
CALL Send518byte (&H02)     ‘send flags to 518: (x,x,x,x,x,x,1,x)
```

EXAMPLE:

```
‘ Prompt the operator for the desired open-sensor conditions for all |
‘ eight sensor channels, then send the required command to the 518. |

Flags% = 0                                                    ‘initialize open-sensor flags

FOR Channel% = 0 TO 7
      INPUT “Enter channel”; Channel%; “flag (0 or 1):”, F%   ‘get flag from operator
      Flags% = Flags% + F% * 2 ^ Channel%                     ‘set flag if 1 was entered
NEXT Channel%

CALL Send518byte (80)                                         ‘send opcode to 518
CALL Send518byte (Flags%)                                     ‘send flags to 518
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Read Alarm Flags

This command returns the status of all channel alarms in two response bytes. The first response byte contains the status of all high alarms while the second byte contains the status of all low alarms. Bit 7 (most significant) through bit 0 (least significant) of each alarm status byte corresponds to channel 7 through 0, respectively. A status bit containing logic one indicates a violated alarm limit while a logic zero indicates no limit violation.

In addition to returning alarm flag data, this command also resets all channel alarm flags as well as the coprocessor status register ALARM bit.

COMMAND: **(48)**

RESPONSE: **(HIGH ALARM LIMIT FLAGS), (LOW ALARM LIMIT FLAGS)**

EXAMPLE:

```
' Read alarm flags from 518 and print messages for any alarming channels. |

CALL Send518byte (48)                          'send command to 518

HiFlags% = Read518byte (BasePort%)             'read high alarm flags from 518
LoFlags% = Read518byte (BasePort%)             'read low alarm flags from 518

FOR Channel% = 0 TO 7                          'print alarm messages
      Mask% = 2 ^ Channel%
      IF HiFlags% AND Mask% THEN PRINT "Channel "; Channel%; "high alarm"
      IF LoFlags% AND Mask% THEN PRINT "Channel "; Channel%; "low alarm"
NEXT Channel%
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Set Gage Zero

This command sets the zero-load value on a strain gage channel. In effect, the *Set Gage Zero* command tells the 518 that there is presently no load on the channel load cell. Typically, this command is issued just prior to a *Set Gage Span* command. Together, the *Set Gage Zero* and *Set Gage Span* commands may be used to calibrate a strain gage channel.

COMMAND: **(176 + CHAN)**
RESPONSE: **NONE**

See the *Strain Gages* section for more details on strain gage commands.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Set Gage Span

This command sets the effective gain of a strain gage. Many gage installations simulate gage loads by multiplexing a simulated zero and full-scale load signal onto the gage channel. To utilize this command, switch the zero-load signal onto the channel and issue the *Set Gage Zero* command. Next, switch the full-load signal onto the channel and issue the *Set Gage Span* command.

COMMAND: **(208 + CHAN), (DATA MSB), (DATA LSB)**

RESPONSE: **NONE**

EXAMPLE:

```
' Calibrate the load cell connected to channel 1. Full-load is
' 40,000 pounds, and the 518 output units are to be scaled to 10
' lbs/bit. At full-load, therefore, 518 output will be 4,000.

Channel% = 1
ZeroCmd% = 176                          'SET GAGE ZERO command opcode
SpanCmd% = 208                          'SET GAGE SPAN command opcode
MaxLoad% = 4000                         'coprocessor counts at full load

PRINT "Apply zero load to gage and press any key when ready .. "
DO : WHILE INKEY$ = "" 'wait for key press

CALL Send518byte (BasePort%, ZeroCmd% + Channel%)'send ZERO command to 518

PRINT "Now apply 40,000 pound load to gage and press any key when ready .. "
DO : WHILE INKEY$ = ""                  'wait for key press

CALL Send518byte (BasePort%, SpanCmd% + Channel%)'send SPAN command to 518
CALL Send518word (BasePort%, MaxLoad%)
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Tare Gage

This command will tare the strain/pressure gage connected to a single channel. The *Tare Gage* command may be used to compensate for the weight of an empty container prior to measuring the container when it holds some substance. Before invoking this command, the gage channel should be calibrated. See the *Set Gage Zero* and *Set Gage Span* commands for details.

COMMAND: **(112 + CHAN)**

RESPONSE: **NONE**

EXAMPLE:

```
‘ An empty concrete truck is to be tared prior to loading. The truck scale |
‘ is monitored by load cell connected to channel 7. Channel 7 has been |
‘ previously calibrated using the SET GAGE ZERO and SET GAGE SPAN commands. |

Channel% = 7
TareCmd% = 112

CALL Send518byte (BasePort%, TareCmd% + Channel%)
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Read Gage Calibration

This command returns the internal slope and offset from a strain gage channel. The response string is six bytes long. The first four response bytes contain the gage slope, encoded in the floating point format used internally by the coprocessor. The last two response bytes contain the lumped gage offset, represented as a 2's complement integer. This lumped gage offset consists of the present tare offset combined with bridge and coprocessor circuit offsets.

In order for the response string to have valid meaning, the strain gage must already have been calibrated. This prior calibration may be accomplished in either of two ways: using the *Set Gage Zero* and *Set Gage Span* commands, or using the *Set Gage Calibration* command.

COMMAND: **(128 + CHAN)**

RESPONSE: **(S0), (S1), (S2), (S3), (S4), (S5)**

EXAMPLE:

```
' This subprogram reads a channels gage parameters into a linear array. |
' The channel must be configured for a strain gage sensor, and should be |
' calibrated before calling this procedure. |

SUB ReadGageParameters (BasePort%, Channel%, ParmArray%())

    CALL Send518byte (BasePort%, 128 + Channel%)      'send opcode to 518

    FOR Parm% = 0 TO 5                                'read 6 parameter bytes from 518
        ParmArray%(Parm%) = Read518byte (BasePort%)
    NEXT Parm%

END SUB
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Set Gage Calibration

This command transfers strain gage slope and offset parameters to the coprocessor for a specific strain gage channel. Typically, these parameters will have been previously obtained via execution of the *Read Gage Calibration* command.

Note: it is necessary to implement a time delay between invocation of a *Set Gage Calibration* and *Read Gage Calibration* command for the same channel. The time interval between execution of these two commands should be at least 500 milliseconds to ensure proper operation of the *Read Gage Calibration* command.

COMMAND: **(144 + CHAN), (S0), (S1), (S2), (S3), (S4), (S5)**

RESPONSE: **NONE**

EXAMPLE:

```
‘ This subprogram copies gage parameters from a linear array back
‘ onto a 518 coprocessor. Assume that the ReadGageParameters
‘ procedure (previous example) was used to load the array.

SUB WriteGageParameters (BasePort%, Channel%, ParmArray%())

        CALL Send518byte (BasePort%, 144 + Channel%)

        FOR Parm% = 0 TO 5
                CALL Send518byte (BasePort%, ParmArray%(Parm%))
        NEXT Parm%

END SUB
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

High Speed Mode

This command increases the channel scan rate to allow higher throughput. The default channel processing time is decreased from 22 milliseconds/channel to 13 milliseconds/ channel. The increased scan rate reduces accuracy by approximately fifty percent and will cause some boards to exhibit slightly increased noise levels.

A board reset (hard or soft) will always restore the board to the default scan rate. A board reset is the only way to resume the default scan rate after invoking the *High Speed Mode* command.

COMMAND: **(240), (8), (0)**
RESPONSE: **NONE**

EXAMPLE:

```
‘ This code fragment switches the 518 to High Speed Mode.

CALL Send518byte (BasePort%, 240)
CALL Send518byte (BasePort%, 8)
CALL Send518byte (BasePort%, 0)
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Low Power Standby

This command places the sensor coprocessor in a low-power standby mode. Power consumption is reduced by halting the logic clock, thereby minimizing current draw in the coprocessor logic circuitry section. The analog section remains fully powered to eliminate warmup noise when full power resumes. See the *Theory of Operation* for a discussion of warmup noise.

COMMAND: (67)
RESPONSE: NONE

EXAMPLE:

```
CALL Send518byte (67)        ' Switch the 518 to Low Power Standby mode
```

After releasing the 518 from the low power standby mode, the host should wait at least 600 milliseconds before communicating again with the coprocessor. This delay will give the 518 time to self-calibrate and acquire good sensor data before any data requests from the host. To release the coprocessor from standby, invoke the *Release Standby* command as follows:

COMMAND: (66)
RESPONSE: NONE

EXAMPLE:

```
CALL Send518byte (67)        ‘ Release the 518 from Low Power Standby mode
CALL delay(600)              ‘ don’t talk to the 518 for at least 600 milliseconds
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Read Firmware Version

This command returns the coprocessor firmware version number times 100. The firmware version number is printed on the EPROM device plugged into the coprocessor board.

COMMAND: **(240), (5), (0)**

RESPONSE: **(Version_MSB), (Version_LSB)**

EXAMPLE:

```
' This code fragment reads and prints the coprocessor firmware version.

CALL Send518byte (BasePort%, 240)
CALL Send518byte (BasePort%, 5)
CALL Send518byte (BasePort%, 0)

Version% = Read518word (BasePort%)

PRINT USING "518 firmware version = ##.##", Version% / 100!
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Read Product I.D.

This command returns an identifier specific to the addressed Sensoray product. The product identifier specifies the Sensoray product model number; the value *518* will be returned by a model 518 sensor coprocessor.

COMMAND: **(240), (4), (0)**

RESPONSE: **(ProductID_MSB), (ProductID_LSB)**

EXAMPLE:

```
‘ This code fragment reads and prints the coprocessor product identifier.

CALL Send518byte (BasePort%, 240)
CALL Send518byte (BasePort%, 4)
CALL Send518byte (BasePort%, 0)

ID% = Read518word (BasePort%)

PRINT USING “This product is a Sensoray Model ####”, ID%
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Set Coefficients

This command defines the linearization coefficients for a channel previously declared as a USER-DEFINED RESISTIVE SENSOR. Three coefficients are specified. The three values belong to a polynomial of the form:

$$f(R) = aR^2 + bR + c$$

Each coefficient is represented by four bytes coded in the coprocessor's internal floating point format. The first three bytes constitute the mantissa, while the fourth byte represents the exponent. See chapter 6 for further details on user-defined resistive sensors.

COMMAND: **(192 + CHAN),**
(A_M0),(A_M1),(A_M2),(A_EXP),
(B_M0),(B_M1),(B_M2),(B_EXP),
(C_M0),(C_M1),(C_M2),(C_EXP)

RESPONSE: **NONE**

The following code segment will convert a real value to the coprocessor's internal floating point format and handshake the resulting four bytes onto the coprocessor. This procedure is used in the example that follows. You may use this procedure as is or adapt it to whatever programming environment you may be using.

```
SUB Send518Real (Number!)

    IF Number! <> 0 THEN
        Exponent! = INT(LOG(2! * ABS(Number!)) / LOG(2!) + .0000001)
        Mantissa& = ABS(Number!) / (2 ^ Exponent!) * 256 ^ 3
        IF Number! > 0 THEN Mantissa& = Mantissa& AND &H7FFFFF
        MantByte0& = Mantissa& AND &HFF&
        MantByte1& = (Mantissa& AND &HFF00&) \ 256
        MantByte2& = (Mantissa& AND &HFF0000) \ 256 ^ 2
    ELSE
        MantByte0& = 0
        MantByte1& = 0
        MantByte2& = 0
        Exponent = -128
    END IF

    CALL Send518Byte(MantByte0)
    CALL Send518Byte(MantByte1)
    CALL Send518Byte(MantByte2)
    CALL Send518Byte(128 + Exponent)

END SUB
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

EXAMPLE:

```
‘ Channel 2 is connected to a linear slide potentiometer. It is
‘ necessary to measure the potentiometer’s position as a function
‘ of resistance. A good approximation to the potentiometer’s
‘ transfer function is determined to be:
‘
‘              f(R) = 0 * (R ^ 2) + 0.01 * (R) + 0. |
‘
‘ The following code segment will set up channel 2 (assumed to have
‘ previously been declared as a “User-defined Resistive Sensor”) for
‘ linearizing the potentiometer’s measured resistance value:

CALL Send518Byte (192 + 2)      ‘ send command & opcode to 518
CALL Send518Real (0)            ‘ send A coefficient to 518
CALL Send518Real (.01)          ‘ send B coefficient to 518
CALL Send518Real (.1)           ‘ send C coefficient to 518
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Calibrate 518

This command calibrates coprocessor internal standards. All measurements made by the coprocessor are referenced to these standards. To perform a board calibration, you must supply two reference voltages (5 volts and 500 millivolts) and a reference resistance (380 ohms).

References must be calibrated in the following order: 5 volts, 500 mV, 380 ohms.

COMMAND: **(224 + CHAN), (CAL_CODE), (CAL_DATA_MSB), (CAL_DATA_LSB)**

RESPONSE: **(DON'T_CARE)**

Table 11: select codes for the Calibrate command

CAL_CODE	FUNCTION
0	5 volt range
1	500 millivolt range
2	400 ohm range

Example: This code fragment will assist you in calibrating the 518.

```
FOR calCode% = 0 TO 2

    SELECT CASE calCode%
        CASE 0: refName$ = "5 volt":   scalar% = 5000: sensorCode% = 0
        CASE 1: refName$ = "500 mV": scalar% = 50: sensorCode% = 13
        CASE 2: refName$ = "380 ohm": scalar% = 40: sensorCode% = 9
    END SELECT

    INPUT "What channel is "; refName$; " reference connected to"; chan%

    CALL Send518Byte (16 + chan%)                 'declare channel sensor type
    CALL Send518Byte (sensorCode%)

    INPUT "What is exact value of "; refName$; " reference"; actualValue!

    CALL Send518Byte (224 + chan%)                'issue Calibrate command
    CALL Send518Byte (calCode%)                   'issue Calibrate command
    CALL Send518Byte (actualValue% * scalar%)     'issue Calibrate command

    junk% = Read518Byte%                          'wait for 518 to calibrate

    OUT BasePort% + 1, 0                          'reset 518 board

NEXT calCode%
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Sensor Tables

Table 12: standard sensor definition codes and scale factors

Sensor Type	Code	Data Scaling/Bit
Voltage		
+/- 100 mV f.s.	17H	5 microvolts
+/- 500 mV f.s.	16H	20 microvolts
+/- 5V	15H	200 microvolts
Resistance		
400 ohm f.s.	0AH	0.02 ohm
3 Kohm f.s.	14H	0.125 ohm
600 Kohm f.s.	20H	31 ohm
Custom resistive sensor	0CH	defined by programmer
Thermocouple		
B	24H	0.10 C
C	23H	0.10 C
E	01H	0.10 C
J	1BH	0.10 C
K	1CH	0.10 C
N	22H	0.10 C
T	1DH	0.10 C
S	1EH	0.10 C
R	1FH	0.10 C
RTD		
10 ohm copper	2CH	0.10 C
100 ohm platinum, 385	18H	0.05 C
100 ohm platinum, 392	19H	0.05 C
100 ohm platinum, 385	2AH	0.0125 C
100 ohm platinum, 392	2BH	0.0125 C
120 ohm nickel, T/R 2	21H	0.1 F
Thermistor		
Omega 44031, 10 Kohm	1AH	0.01 C
Current Loop		
4 - 20 mA	11H	0.01% (4mA=0%, 20mA=100%)
Pressure/Strain Gage		
>120 ohm full bridge	0FH	defined by programmer
Disabled Channel	13H	not applicable

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Back Compatibility

In addition to the sensor definition codes listed above, several additional codes exist to maintain compatibility with the earlier Sensoray products. Use of these codes is not recommended for new designs.

Table 13: back-compatible sensor codes

Sensor Type	Code	Data Scaling/Bit
Voltage 0 to +5V f.s. 0 to +1.65V f.s. 0 to +80mV f.s.	 00H 0EH 0DH	 500 microvolts 100 microvolts 10 microvolts
RTD 385 392	 07H 08H	 0.1 C 0.1 C
Thermocouple J K T S R	 02H 03H 04H 05H 06H	 0.11 C 0.17 C 0.15 C 0.60 C 0.50 C
Thermistor Omega 44031, 10 Kohm	 0BH	 0.02 C

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter

5

Sensor Connections

All sensors are connected to the 518 by means of connector P1. Optionally, sensors may be connected to the Sensoray model 7409TC screw termination board which in turn connects to 518 connector P1. Sensors may be electrically connected and disconnected from the coprocessor or 7409TC without removing the 518 or power from the PC104. The following discussions assume the presence of a 7409TC termination board.

Each sensor has as many as five connections to a channel circuit. Two connections are used for all sensor types: V+ and V-. These two signals are the positive and negative differential voltage sense inputs.

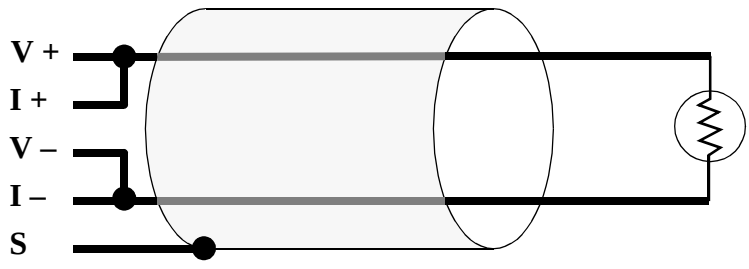
In addition to the sense terminals, all resistive sensors require connection to the I+ and I- terminals. The I+ and I- terminals provide positive and negative excitation for resistive sensors. The fifth lead — labeled *S* for shield —connects to a the cable shield (if present). The following sections describe how to connect sensors to a 7409TC for proper operation.

RTDs and Resistors

RTDs and resistors can be connected to the 7409TC in three possible configurations: two-wire, three-wire, and four-wire. In the following discussions, *RTD* is used interchangeably with *resistor*.

The simplest configuration is the two-wire circuit. As the name implies, this configuration requires only two wires. The V+ and I+ terminals are shorted together, and V- and I- are shorted together at the 7409TC. This scheme conserves wire, but introduces error as a result of the voltage difference between the V/I junctions and RTD. There are two possible solutions to this problem: use short wires, or use more than two wires.

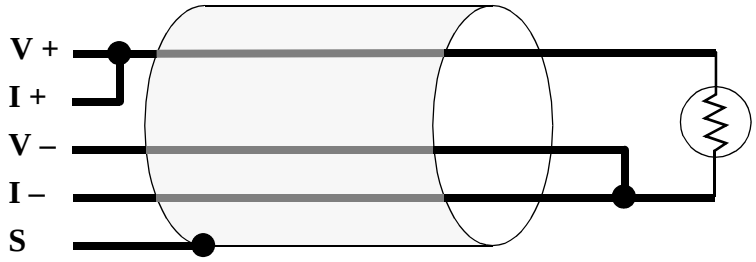
Two-wire connection.



By using three wires, it is possible to reduce RTD cable loss errors by 50 percent. Instead of shorting V- and I- together at the 7409TC, separate wires can be run from each of these terminals out to the RTD. The wires are then shorted together at the RTD. In this situation, the high impedance V- terminal is able to detect the voltage at the RTD before any voltage loss can occur.

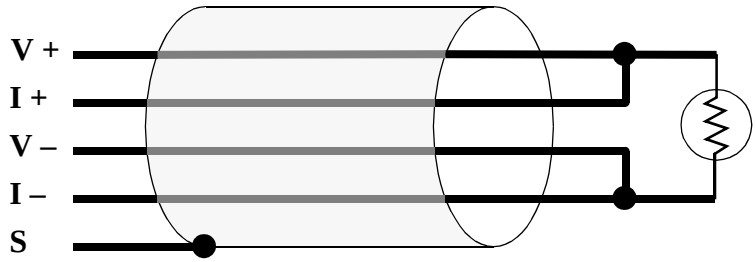
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Three-wire connection



A three-wire RTD still has half the cable loss of the equivalent two-wire RTD. To eliminate cable losses entirely, the four-wire circuit must be used. Like the three-wire circuit, separate wires are run from the V- and I- terminals to the RTD. In addition, separate wires are run from V+ and I+ to the RTD, where they are shorted together. This eliminates cable loss effects from the V+ line, as well as the V- line.

Four-wire connection

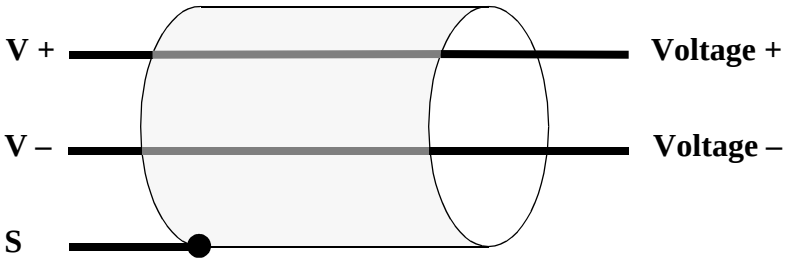


Sensoray recommends the four-wire hookup for all installations. Regardless of the chosen hookup scheme, you should always use shielded cable. The cable shield must be connected only to the S terminal on the 7409TC.

Thermocouple and DC Voltage

Thermocouples and DC voltages are connected directly to the V+ and V- terminals. Thermocouples and DC voltage sources should not be connected to the I+ or I- terminals.

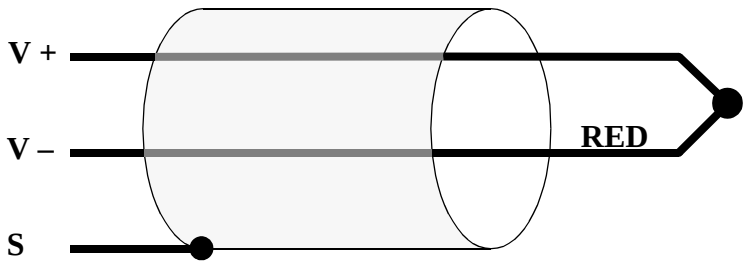
DC voltage connection



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Thermocouple wires are color coded to indicate polarity. The positive thermocouple wire should always be connected to the V+ terminal, and the negative thermocouple wire should always be connected to the V- terminal. Please note that ***the red thermocouple wire is always negative***, by convention.

Thermocouple connection



Electrical noise is often present on thermocouple and voltage signals. It is not within the scope of this manual to discuss all treatments of such noise, however, a few simple techniques are available which will solve most noise problems. The following rules apply to both thermocouple and DC voltage channels:

1. Look at the sensor signal with an oscilloscope and DVM to see if you have the signal you think you have. “Noise” can often be the result of an incorrect sensor installation.
2. Install the 7409TC hardware filter jumper on the offending channel.
3. Ground the “hot” end of the thermocouple. Sometimes the thermocouple is not referenced to the PC104 return. Noise induced into the thermocouple wire from the environment can elevate common mode voltage beyond the coprocessor input range. Make sure that the ground is referenced to the PC104 bus ground. In the case of a DC voltage source, try grounding either the V+ or V- signal (but not both) to limit the common-mode voltage.
4. Invoke the channel software filter.

Thermistor/Custom Resistive Sensors

In the following discussion, *thermistor* is used interchangeably with *custom resistive sensor*.

Thermistors have much higher resistance than RTDs over most of their operating range. As a result, the RTD two-wire configuration is recommended if your sensor will be operating only at higher resistance values. If your thermistor will be operating at lower resistance values, you should consider implementing a three- or four-wire hookup to prevent cable-losses.

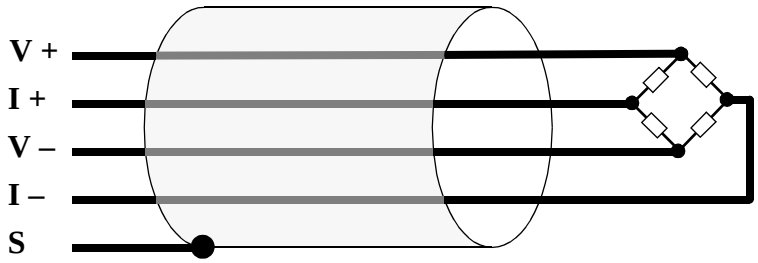
Thermistors should be connected using shielded cable. The shield should be connected only to the 7409TC S terminal.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Strain and Pressure Gages

In a typical strain/pressure gage, four wires connect the gage to the measurement system. Two wires supply gage excitation, while the other two wires carry the gage output signal. If you are using a six-wire gage, you must connect the gage's excitation source and sense leads together at the 7409TC I+ and I- terminals.

Strain/pressure gage connection



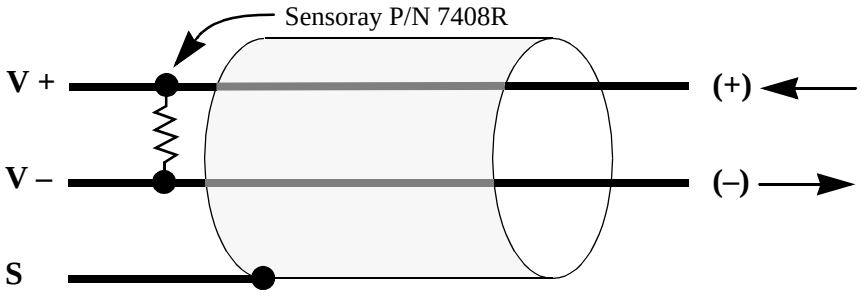
Strain and pressure gages should use shielded cable. The shield should be connected only to the channel S terminal on the 7409TC.

4 to 20 mA Current Loops

The coprocessor supports 4-20 mA current loop inputs on any channel, however, a 250-ohm 0.01% resistor must be installed as shown in the following diagram. These resistors are available as an option for the 518: order Sensoray part number 7408R.

Make sure that any current loop sensors are at the *grounded* end of the loop so that the common-mode voltage does not exceed 5 volts.

4-20 mA current loop connection



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter

6

Sensor Specifics

User-defined Resistive Sensors

User-defined resistive sensors may be interfaced to the coprocessor subject to these constraints:

1. Sensor resistance does not exceed 300K ohms.
2. A quadratic linearization polynomial is sufficiently accurate.

This is the procedure for interfacing a user-defined resistive sensor:

1. Fit the sensor data to a polynomial of the form $f(R) = aR^2 + bR + c$, where: R is the resistance of the sensor in ohms, $f(R)$ is the corresponding sensor data value in the desired 518 integer output units, and a, b, and c are constant coefficients derived from the curve fit.
2. Install commands in the host processor's 518 initialization string to declare the channel as a 'user-defined resistive sensor' and to define the channel sensor linearization coefficients. See the *Define Channel Sensor* and *Set Coefficients* commands for information.

Application Example

A non-linear resistive position transducer is to be interfaced to the 518 on channel 5. Position data is to be acquired from the sensor and displayed in centimeters with resolution of one millimeter. The following data have been determined empirically:

Table 14: measured position transducer data

RESISTANCE (ohms)	0	50	100	150
POSITION (mm)	-7.2	1673	4303	7883

A curve is fit to to the transducer, resulting in the linearization function:

$$P = 0.19R^2 + 24.1R - 7.2$$

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This code fragment will declare the channel 5 sensor type to be a User-defined Resistive Sensor and specify the linearization coefficients. Note that the “Send518Real” procedure is described in the *Set Coefficients* command section.

```
CALL Send518Byte (16 + 5)          ‘ declare channel 5 as User-defined
CALL Send518Byte (&HC)             ‘       resistive sensor

CALL Send518Byte (192 + 5)         ‘ declare channel 5 linearization coefficients:
CALL Send518Real (.19)             ‘       A
CALL Send518Real (24.1)            ‘       B
CALL Send518Real (7.2)             ‘       C
```

After defining the channel 5 sensor type and coefficients, invoking the *Read Sensor Data* command for channel 5 will return a value with data scaled to 1 mm/bit. The following fragment will fetch channel 5 data from the coprocessor and display in the desired format:

```
Send518byte (BasePort%, 5)         ‘ send READ DATA command for channel 5
ChanData% = Read518word%(BasePort%)  ‘ read channel 5 data from 518
PRINT USING “###.#”; ChanData% / 10.0  ‘ display in centimeter form
```

Thermocouples

A thermocouple typically consists of a wire pair whose wires are made of two dissimilar metals. At one end of the pair, the two wires are electrically shorted together (by soldering, welding, etc.). This “end” of the thermocouple— commonly referred to as the “hot-junction”— is thermally attached to the point to be measured. The other end — the “cold-junction” — is connected to a measuring device (in this case, a 7409TC).

Thermocouples generate an open-circuit voltage that is proportional to the temperature gradient between hot- and cold-junctions (Re: the Seebeck effect). Since this voltage is a function of the temperature difference between junctions, it is necessary to know the temperature at the cold-junction in order to accurately determine the temperature at the hot-junction.

An integrated circuit temperature transducer resides on the Sensoray 7409TC termination board. This transducer, because of its close physical proximity to the thermocouple terminations, represents an accurate measurement of the thermocouple cold-junctions. Periodically, the 518’s CPU measures the I.C. transducer signal and computes the corresponding thermocouple voltage, known as the “cold-junction compensation voltage”.

When a thermocouple channel is measured, the 518’s CPU adds the thermocouple voltage to the cold-junction compensation voltage to form the “corrected thermocouple voltage”. This is the voltage one would measure if the cold- junction were maintained at zero degrees C. Finally, the corrected thermocouple voltage is converted into the appropriate temperature units via a non-linear mapping function.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Verifying Thermocouple Calibration

Thermocouple applications sometimes require periodic validation of thermocouple calibration. Two methods are commonly used: ice bath and millivolt calibrator.

Ice Bath Method

The ice-point or boiling-point method has the advantage of requiring simple and inexpensive equipment: a beaker full of ice water or boiling water. The thermocouple hot-junction is immersed into a stirred ice bath (ice cubes and water) or into boiling water. Time is given to allow the thermocouple to reach thermal equilibrium. The operator then verifies that 518 output data is consistent with the water temperature (0 degrees C for an ice bath, 100 degrees C for boiling water).

Certain precautions must be taken when executing this procedure. Water can be absorbed into the thermocouple insulation if the thermocouple hot end is inserted directly into the bath. Should the water form a conductive path, the effective hot-junction will move away from the intended location. This relocated hot-junction is unlikely to have the same temperature as the bath, resulting in an incorrect temperature indication. To avoid this problem, a thermowell should be used to keep the thermocouple from making direct contact with the water.

A simple and inexpensive thermowell can be constructed from a piece of quarter-inch diameter copper tubing. Cut a length of tube that is too long to be completely submerged in the bath. Crimp one end of the tube with a hammer to make it water-tight. Insert the thermocouple hot-junction all the way into the open end of the tube. Now insert the crimped end of the thermowell into the water bath.

Millivolt Calibrator Method

The millivolt calibrator is typically more accurate than a water bath if the thermocouple operating point is much higher than 100 degrees Centigrade. This procedure is straight-forward and accurate provided the user remembers this important fact: thermocouple voltage is generated across the length of the thermocouple wire, NOT at the hot-junction. Since the 518 compensates for the cold-junction being other than zero degrees C, error will result if a temperature independent voltage (corresponding to a corrected thermocouple voltage) is used to verify thermocouple calibration.

Here is a valid way to check thermocouple calibration using a millivolt calibrator:

1. Measure the temperature of the 7409TC termination board with a calibrated low thermal mass temperature probe and meter.
2. Use the NBS millivolt-temperature look-up tables to determine the thermocouple voltage corresponding to the 7409TC temperature. This is the thermocouple correction voltage $\mathbf{V_c}$.
3. Decide on a temperature $\mathbf{T_{sim}}$ to be simulated by the millivolt calibrator.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4. Use the NBS tables to obtain the thermocouple voltage $\mathbf{V_{sim}}$ corresponding to T_{sim}. Note that V_{sim} is the corrected thermocouple voltage with the hot-junction at T_{sim}.
5. "Uncorrect" V_{sim} by subtracting off the correction voltage V_c. The resulting voltage is $\mathbf{V_{test}}$.
6. Set up a calibrated millivolt calibrator to output V_{test}. Apply V_{test} to the 518 channel under test. Verify that the 518 temperature output value is equal to T_{sim}.

Strain and Pressure Gages

A strain gage is often assembled into a device known as a load cell. In operation, a stress is applied to the load cell, which in turn stresses the gage. Load cells vary widely in shape, size, and material, depending on the application. Even so, all load cells serve the same basic purpose: apply stress to the strain gage as an approximately linear function of the load.

Once a strain gage is mounted to a load cell, the entire assembly takes on a sensitivity specification, typically rated in PSI units. This rating is defined in terms of the ratio of gage output voltage to input voltage at full load. For example, a gage might be rated at "3mV/V". This means that the gage produces three millivolts out for every one volt of input excitation at a full load condition.

Measurement Resolution

Strain gage excitation is supplied by the 518 coprocessor in the form of a strobed DC voltage. This voltage varies slightly from one coprocessor board to another, but is on the order of 10 volts in amplitude.

Knowing the bridge input voltage $\mathbf{V_{in}}$ and gage sensitivity specification **K**, the maximum gage output voltage $\mathbf{V_{out}}$ may be computed:

$$V_{out} = V_{in} \times K$$

Any 518 channel configured for a strain/pressure gage is measured with 5 microvolt resolution. To compute the strain gage resolution **N**, divide the gage full load output voltage by 5 microvolts (or multiply by 1/5uV):

$$N = V_{out} \times 200000 = V_{in} \times K \times 200000$$

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Thus, the strain gage output is resolved by the coprocessor into one part in N. It follows that the gage resolution is:

$$Resolution = \frac{Load_{max}}{V_{in} \times K \times 200000}$$

As an example, suppose a gage is rated at 3 mV/V at 100 PSI full scale. The 518 coprocessor would be able to resolve a load to:

$$N = \frac{100}{10V \times 0.003V \times 200000} = 0.017 pounds$$

In order to successfully measure a gage with the coprocessor:

1. The gage input impedance as viewed by the coprocessor excitation source must not be less than 120 ohms. If the input impedance is less than 120 ohms, a resistor must be wired in series with the gage input terminals to reduce excitation circuit loading.
2. The gage output voltage must fall between -500 mV and +500 mV under all load conditions (including any offset caused by bridge imbalance). If the gage output exceeds these range limits, the maximum output can be reduced by decreasing the gage excitation: connect a resistor in series with the gage input terminals.

If a resistor is wired in series with the gage excitation terminals, it should be a metal film type with low temperature coefficient. Keep in mind that a series resistor changes the excitation voltage as seen by the gage. The above equations for gage resolution still apply, but the excitation value must be adjusted to compensate for the series resistance.

Calibration

Strain gages are calibrated using a two-level calibration hierarchy. At the top level the gage zero and span values are established. This calibration function is performed once at start-up time to define gage characteristics and at periodic intervals to compensate for the drift of gage parameters with time, mechanical changes of the load cell, and temperature. At the second level the load is tared prior to each load measurement to compensate for empty container weight.

Zero and Span Adjustment

Default gage parameters are assumed by the 518 when a channel sensor is declared to be a strain gage. These default values are very unlikely to be correct for the installed gage. As a result, the gage zero and span must be explicitly defined for proper gage operation. Also, gage parameters change over time, drift with temperature, and are influenced by mechanical hysteresis effects within the load cell. To compensate for these variations we recom-

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

mend that the zero and span be recalibrated occasionally during operation, even after the initial calibration. Note that the coprocessor does not 'remember' the gage calibration when it is reset or powered down, so an initial gage calibration is always required.

The coprocessor permits gage calibration by either of two methods: load simulation, or calibration data transfer. The load simulation procedure requires either physical or simulated loading of the gage, while the calibration transfer method needs only data transfer between host processor and coprocessor.

Many gage installations achieve zero and span calibration with the aid of gage simulation signals. Each gage produces three signals that are multiplexed onto a single 518 input channel. One signal is the load signal (L), representing the real load. The other two are simulated load signals, one for a zero load (Z) and the other a full load (F) condition. In the following discussion, the symbolic names L, Z, and F are used to refererence either physical or simulated loads (whichever you are using to calibrate your gage with).

During normal operation, the L signal is applied to the 518 input channel. When it is time to calibrate the gage zero and span, the following procedure is used:

1. The Z signal is connected to the input channel, and the host sends a *Set Gage Zero* command to the 518. This establishes the gage zero load condition.
2. The F signal is connected to the input channel, and the host sends a *Set Gage Span* command to the 518. The corresponding full load count is downloaded to the 518 as part of this command to establish the gage full load condition.

Once the above procedure has been completed, the gage calibration may be read from the coprocessor via the *Read Gage Calibration* command. The six bytes returned by this command may be saved for future use.

The next time the gage is to be calibrated, either method may be used (even if the coprocessor has just been reset or powered up). If gage calibration via the *Set Gage Zero* and *Set Gage Span* commands is a problem in your application, you may opt to calibrate the gage using the *Set Gage Calibration* command.

Taring

In many gaging applications it is necessary to measure the weight of material within a container. This is done by first weighing the empty container to determine it's *tare* weight.

The 518 is designed to tare loads and automatically deduct the tare weight from all subsequent output data to the host processor. Assuming that a gage has been calibrated for zero and span, a load may be tared as follows:

1. Position the empty container onto the load cell assembly.
2. Issue a *Tare Gage* command from to the 518 coprocessor.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter

7

Theory of Operation

Software

Execution of the internal coprocessor software is managed by a resident multi-tasking kernel. The kernel is responsible for scheduling and executing concurrent foreground tasks, processing intertask communication requests, and servicing real-time communication interrupts from the system bus interface.

The following paragraphs describe the inner-workings of tasks that are controlled by the kernel.

Scanner

This task performs the fundamental data acquisition function central to the coprocessor's purpose.

The scanner begins a data conversion by selecting the next channel to be digitized. The analog front-end multiplexers are switched to the desired channel, and the pulsed excitation source is activated if the channel is connected to a resistive sensor type. The scanner then goes to sleep while the sensor signal settles out.

When the sensor signal stabilizes, the scanner initiates an A/D conversion. Other tasks are allowed to run while the conversion is taking place.

Upon completion of the conversion, the A/D data is sent to the post-processor task.

Post-processor

The post-processor is responsible for refining uncorrected A/D data into finished form.

This task begins by receiving uncorrected channel data from the scanner task. The data is corrected for circuit offset and gain errors, then linearized and converted to engineering units appropriate for the declared sensor type. The resulting value is passed through a single-pole low-pass filter.

Next, the filtered data is checked for alarm limit violations. If a limit violation is detected , the violated channel's limits are both reset to their default values, the channel high or low alarm status flag is asserted, and the status register alarm bit is set.

Finally, the processed channel data is stored in RAM for host processor access.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Command Processor

The command processor fetches and executes commands from the host processor. Top priority is given to this task in order to minimize communication latency between the host and coprocessor.

This task is awakened when a command is received from the host. The command processor decodes the opcode embedded in the first byte of the command string, and control is passed to the appropriate command processing procedure.

Hardware

Although the coprocessor analog circuitry is proprietary to Sensoray, an overview of the circuitry is provided here to give you some insight into the inner-workings of the board.

Coprocessor analog circuitry is partitioned into three sections: reference, measurement and excitation. The reference section contains standards that are used to enhance the accuracy of all sensor measurements. The measurement section includes analog switches, various amplifiers and the A/D converter. The excitation section is responsible for stimulating resistive sensors.

Reference Section

The reference section consists of a set of reference standards that are used to enhance the stability and accuracy of all sensor measurements. These standards are very stable over time and temperature. The exact values of the reference standards are stored in EEPROM on the coprocessor board when the board is calibrated.

Reference standards are measured periodically by the onboard CPU. These measurements are interleaved with external sensor measurements. One reference standard is measured per approximately every twenty external sensor measurements. Each digitized reference standard value is stored in onboard RAM and used to null offset and gain errors in the external sensor measurements.

A phenomenon known as *warm-up noise* occurs when the 518 is warming up from cold start or when subjected to a thermal transient. When the coprocessor is exposed to changing ambient temperatures, sensor data will appear somewhat noisy. This is caused by circuit gain and offset characteristics changing faster than reference standards are measured. Warm-up noise will subside when the 518 reaches thermal stability.

Measurement Section

The measurement section is responsible for selecting and digitizing the external sensor input signals, as well as the internal reference standards.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The measurement section starts a digitization by selecting the channel to be measured. The selected channel is switched through an eight-channel differential analog multiplexer. Reference standards are switched through dedicated analog switches so as to not rob input channels from the external sensor channels.

Next, the selected channel is passed through a programmable gain amplifier. Amplifier gain is under control of the coprocessor's onboard CPU. The onboard CPU sets amplifier gain level based on the declared sensor type in the case of external sensor channels, or the reference type in the case of internal reference standard channels.

Finally, the amplified signal is applied to the input of an integrating A/D converter.

Channel selection, amplifier gain, and A/D functions are all under control of the coprocessor's CPU. None of there circuits are directly accessible to the host.

Excitation Section

The excitation section supplies one of three pulsed DC signals to resistive sensors, depending on the declared sensor type. The pulse duration is the sum of the sensor settling time plus the A/D conversion time -- approximately 22 milliseconds total,or 13 milliseconds in the High Speed Mode."

In the case of strain and pressure gages, a fixed 10 volt excitation is applied. This voltage is current limited so that shorting the excitation signals together or to ground will not damage coprocessor circuitry.

In the case of RTD's and the 400 ohm resistance range, a fixed current of approximately 1.2 milliamps is forced through the sensor.

Finally, a fixed 5 volts is applied in series with a reference resistor to sensors measured on all other resistance ranges.

Timing

Three timing parameters are important from a system integration viewpoint: channel scan rate, communication latency, and host processor speed. These timing elements are considered below.

Scan Rate

Channel scan rate is defined as the number of conversions per channel per second. The scan rate is influenced primarily by the number of active channels in the scan loop. A secondary influence is the frequency of communication between host and 518 coprocessor.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The basic channel processing time -- the *scan time* -- is approximately 22 milliseconds at the default scan rate, or 13 milliseconds at the High Speed Mode scan rate. This is the amount of time required for the 518 to configure its analog section, allow the front end to settle and digitize the input signal. Note that the software linearization and filtering functions don't require any additional time as they execute concurrently with digitization.

Clearly, the scan time will decrease as the number of active channels decrease. Channels can be removed from the scan loop by disabling them with the *define channel sensor* command in conjunction with the *disabled channel* sensor definition code. The update rate for each channel is approximated by this function:

$$\text{samples per second} \approx \frac{1}{(N\text{umber of active channels}) \times (C\text{hannel processing time})}$$

Another important parameter is the refresh time. The refresh time is the time spacing between updates of any particular channel's measured data. This parameter may be critical in applications that require guaranteed new data every time channel data is read. In such applications, it is necessary to know the worst-case refresh time -- the minimum time between successive reads from any channel while guaranteeing fresh data. The worst-case refresh time is given by this function:

$$\text{worst-case refresh time} = [(\text{number of active channels}) + 1] \times (\text{channel scan time})$$

All communication functions are interrupt-driven in the 518's local environment. As a result, both communication latency and overhead are kept to a minimum. Even so, time spent communicating with the host is frequently time stolen from the scanning function. Keep in mind that communication traffic between host and 518 will tend to reduce the scan rate and increase the worst-case refresh time.

Communication Latency

Communication latency is defined here as communication delays between host and 518. Perhaps the most important measure of latency is the amount of time elapsed from a host request for sensor data to the acquisition of that data. This is called the a*cquisition latency*.

Acquisition latency can be viewed as having two parts: *command response delay* is the time from request to first response byte, and *data queue delay* is the time between response bytes.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Two commands return sensor data to the host processor, *read all channels* and *read channel data*. The worst-case command response delay is the same for both commands: 70 microseconds. The worst-case data queue delay is 20 microseconds.

The *read channel data* command consumes a maximum of 90 microseconds. This is computed as follows: 70 microseconds from command issue to availiability of the first response byte, plus 20 microseconds from the time the host reads the first byte to availiability of the second response byte. This analysis ignores any host processor timing overhead.

The *read all channels* command consumes 370 microseconds as follows: 70 microsecond command response delay plus 15 more data bytes with 20 microsecond data queue delay each.

As it turns out, acquisition latency actually takes somewhat longer than the theoretical maximums discussed here. Additional time is consumed by the host processor itself, either by polling the 518's handshake status port or by interrupt overhead in the case of interrupt-driven host processors.

Processor Speed

Host processor speed is constrained by the bus interface circuitry on the sensor coprocessor board. The bus interface is designed to work with hosts operating at up to 20 MHz with no wait states. If you have a faster processor or a heavily loaded backplane, you may need to insert wait states into your host's access cycles to ensure reliable operation. Many popular high-speed processors automatically introduce wait states during I/O operations.

At the other end of the speed spectrum, there is no bottom limit on host processor speed except for the following: read and write strobes to the 518 must not exceed 5 microseconds in duration. Communication handshake integrity cannot be guaranteed if host read/write access strobes exceed 5 microseconds.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter

8 *Command Summary*

READ CHANNEL DATA - read linearized sensor data from one channel command

command: (CHAN)

response: (HIGH DATA),(LOW DATA)

DEFINE CHANNEL SENSOR - declare channel sensor type command

command: (16 + CHAN),(SENSOR DEFINITION CODE)

response: NONE

DEFINE CHANNEL ALARM LIMITS - declare normal data limits for one channel

command: (32 + CHAN),
(HILIMIT MSB), (HILIMIT LSB),
(LOLIMIT MSB), (LOLIMIT LSB)

response: NONE

READ ALARM FLAGS - read channel alarm status flags and turn off board alarm

command: (48)

response: (HIGH ALARM LIMIT FLAGS), (LOW ALARM LIMIT FLAGS)

READ BOARD TEMPERATURE - read 7409TC reference temperature

command: (64)

response: (BOARD TEMP MSB), (BOARD TEMP LSB)

RELEASE LOW POWER STANDBY - exit Low Power Standby mode

command: (66)

response: NONE

LOW POWER STANDBY - enter Low Power Standby mode

command: (67)

response: NONE

SET OPEN SENSOR DATA VALUES - establish data values for open sensor condition

command: (80), (OPEN SENSOR DATA FLAGS)

response: NONE

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

READ DATA FROM ALL CHANNELS

command: (88)

response: (CH0 MSB), (CH0 LSB),
(CH1 MSB), (CH1 LSB),
(CH2 MSB), (CH2 LSB),
(CH3 MSB), (CH3 LSB),
(CH4 MSB), (CH4 LSB),
(CH5 MSB), (CH5 LSB),
(CH6 MSB), (CH6 LSB),
(CH7 MSB), (CH7 LSB)

SET FILTER TIME CONSTANT - specify filter factor for one channel

command: (96 + CHAN), (FILTER CONSTANT)

response: NONE

TARE GAGE - zero gage data value with present load

command: (112 + CHAN)

response: NONE

READ GAGE CALIBRATION - fetch gage calibration data from one channel

command: (128 + CHAN)

response: (S0), (S1), (S2), (S3), (S4), (S5)

SET GAGE CALIBRATION - download gage calibration data to one channel

command: (144 + CHAN),
(S0), (S1), (S2), (S3), (S4), (S5)

response: NONE

SET GAGE ZERO - set gage zero-load condition

command: (176 + CHAN)

response: NONE

SET COEFFICIENTS - define custom resistive sensor linearization coefficients

command: (192 + CHAN),
(A_M0), (A_M1), (A_M2), (A_EXP),
(B_M0), (B_M1), (B_M2), (B_EXP),
(C_M0), (C_M1), (C_M2), (C_EXP)

response: NONE

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SET GAGE SPAN - set gage full-load value

command: (208 + CHAN),
(MSB DATA), (LSB DATA)

response: NONE

CALIBRATE 518

command: (224 + CHAN),
(CAL_CODE),
(CAL_DATA_MSB), (CAL_DATA_LSB)

response: (DON'T_CARE)

READ PRODUCT I.D. - read Sensoray product identifier

command: (240), (4), (0)

response: (ProductID_MSB), (ProductID_LSB)

READ FIRMWARE VERSION - return coprocessor firmware version number times 100

command: (240), (5), (0)

response: (Version_MSB), (Version_LSB)

HIGH SPEED MODE - enter high speed mode

command: (240), (8), (0)

response: NONE

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP