ScienceWorkshop 500 Interface

Serial Protocol and Programmer's Guide

The following is a specification of the protocol and practices for controlling a SW500 interface. The SW500 is a versatile acquisition device with multiple input channels and sampling modes, whose data interface conforms to RS-232. The SW500 can communicate with any Macintosh, IBM-PC, or other computer equipped with a standard RS-232 interface. All commands to the SW500 are 1 byte opcodes with 0 or more operand bytes. All bytes are specified Most-Significant Bit-left (bit 7) and Least Significant Bit-right (bit 0). All signed integers are in two's-compliment format.

When the SW500 first powers up, it executes boot code in permanent read-only memory. In this ROM mode, it can accept only identification, download, and execute commands (see section 2). Once initialized with downloadable software, it then executes the downloaded code from RAM and can accept its full command suite.

The SW500 can also collect data when not connected to a computer. The host computer can place the SW500 into a special logging state where the front panel button can then be used to control logging in the field. Data can later be retrieved from the SW500 by the host computer.

1.0 Serial Communication Parameters

RS-232 Data Signals Used

Tx	
Rx	
CTS	(from Host to interface) Interface suspends data transfer when CTS is raised by host.

RS-232 Communication Parameters

Baud Rate:	19,200 baud
Data Bits:	8
Stop Bits:	1
Parity:	none
Handshaking:	hardware/CTS for data transfer from interface to host only

2.0 Initialization Commands

Identify *(RAM or ROM mode)*
Command: **0 x 01**
Operand(s): **none**
Return Data: **“SW500i__” + 4-Byte Version Code (if in ROM mode)**
“SW500i__” + 4-Byte Version Code + “RAM” (if in RAM mode)

Identify the interface model number and version. The model number must be the constant 8-byte string indicated above (where “_” indicates an ASCII space character). The version code may vary with the revision of the unit, but will be of the form “1.0_”. If the interface is already initialized and running in RAM mode, then the *Identify* command will also return three flag bytes after the version code. These bytes must be “RAM”. If the “RAM” flag bytes are not returned, then the interface is running in ROM mode and requires download and execute commands.

Download *(ROM mode only)*
Command: **0 x 02**
Operand(s): **Download Record**
Return Data: **Checksum Byte**

Download one code/data record to be loaded into RAM on the interface. The format of a download record mirrors the Motorola S-Record format:

Bytes	Description
1	Number of bytes (address,code/data,checksum) to follow (255 maximum).
3	Initial address for code/data bytes (unsigned integer).
n	Code/Data bytes
1	Checksum byte

The download records for the SW500 reside in the file “SW500RAM.S28”, available from PASCO scientific (www.pasco.com). Each field in the download record is the binary equivalent of the hex values in an ASCII Motorola S-Record. (Only records beginning with the characters “S2” need to be converted to binary and downloaded.) One download command is issued for each “S2” S-Record in a download module. The SW500 computes a checksum on the address and code/data bytes of the record as they are received. The resulting checksum byte is returned to acknowledge receipt of the record. Any difference between the returned checksum byte and the downloaded checksum byte indicates that the record should be re-sent or that an unreliable serial connection exists. An *Execute* command may only be issued once all “S2” records from SW500RAM.S28 have been downloaded and acknowledged. The *Download* command will be ignored when the interface is running in RAM mode.

Execute *(ROM mode only)*
Command: **0X03**
Operand(s): **none**
Return Data: **Acknowledge Bytes**

Execute the downloaded RAM code in the interface. (This places the interface into RAM mode.) Appropriate *Download* commands must have been issued prior to this command. The *Execute* command will be ignored when the interface is running in RAM mode.

Reset *(RAM mode only)*
Command: **0x4**
Operand(s): **none**
Return Data: **none**

Reset the entire interface to power-on state. All download code, buffered data, and mode settings are cleared. After this command, the interface will be running in ROM mode.

3.0 Sampling Setup Commands (RAM mode only)

Input Select

Command: **0X11**

Operands: **Operand Byte 1**

bit 0	Analog Channel A
bit 1	Analog Channel A w/gain 10X
bit 2	Analog Channel B
bit 3	Analog Channel B w/gain 10X
bit 4	Analog Channel C
bit 5-7	Unused

Operand Byte 2

bit 0	Unused
bit 1	Unused
bit 2	Digital Channel 1 Events
bit 3	Digital Channel 2 Events
bit 4	Digital Channel 1 Counts
bit 5	Digital Channel 2 Counts
bit 6	Unused
bit 7	Motion Timer Flag

Return Data: **none**

Select analog and digital channels to be sampled and buffered during continuous sampling. Each selected channel in operand byte 1 will be sampled at the periodic rate specified by the *Rate Select* command. Digital input state change events will be reported for each channel selected in operand byte 2. Digital state change event counters will be accumulated for each selected channel in operand byte 2, and reported at the periodic rate. If the motion timer flag is set, then ping/echo timing will be performed using the two digital channels. (No selections for bit 2-5 are necessary when bit 7 is set.)

Rate Select

Command: **0x12**

Operands:

Operand Bytes 1-4	sample period **(unsigned long µsec)**
Operand Bytes 5-8	clock period **(unsigned long sample periods)**
Operand Bytes 9-10	ping period **(unsigned short 100 µsec ticks)**
Operand Byte 11	small buffer flag **(1 = small, 0=large)**

Return Data: **none**

Define the time base for continuous sampling and the periodic sample rate. The sample period has two components: sample period and clock period. When digital events, counting, or motion sensing are selected, the sample period must be set to 100 µsec. The clock period is then used to control how often analog samples and digital counters are sampled (in sample periods). Analog and digital counter samples will be generated every “clock period” sample periods. When no digital events, counting, or motion sensing are selected, then the sample period is set to the period for analog sampling and the clock period is set to 1. If the clock period is 0 then no periodic samples are generated (digital timing or motion sensing only). The minimum sample period and clock period are determined by the number and type of channels selected. Allow 50 µsec for each selected analog channel. Sampling digital channels periodically requires no additional time, but allow 20 µsec in the case where no analog channels are selected. The small buffer flag causes the interface to use a smaller than maximum buffer to yield a faster turn-around for high-speed data snapshots (useful for oscilloscope-like features).

Trigger Select

Command: **0x13**

Operands: **Operand Byte 1** channel **(unsigned byte)**

1 Trigger on first selected analog channel
2 Trigger on second selected analog channel
3 Trigger on third selected analog channel
4 Trigger on digital channel 1
5 Trigger on digital channel 2

Operand Byte 2 slope **(unsigned byte)**

0 Trigger on falling analog signal or low digital state
1 Trigger on rising analog signal or high digital state

Operand Bytes 3-4 level **(signed short)**

Analog trigger level representing +/- 10V (+/-1V with gain) (12 bits)

Return Data: **none**

Define a high-speed trigger condition. After each *Start Sampling* command, the SW500 will monitor the selected channel for the trigger condition. No data will be buffered or transmitted until this trigger condition is met. While waiting for a trigger condition, the *Check Sample State* command can be used to verify that the SW500 is waiting for the trigger condition. To convert from a floating-point voltage to the raw analog trigger level value, multiply voltage * 32767 and divide by 10.

Rotary Motion Resolution

Command: **0x14**

Operands: **Operand Byte 1** Number of Pulses

Return Data: **none**

Sets the resolution for a rotary motion sensor by enabling both digital channels as outputs, and sending one or two pulses. One pulse represents the low-resolution setting (360 counts per full rotation), and two pulses represent the high-resolution setting (1440 counts per full rotation). Both channels are initially high, and then pulsed low. The pulse width is ~5 µsec. The digital channel outputs are disabled after this command. To read data from a rotary motion sensor, select counters for both digital channels. One channel indicates a clockwise rotation, and the other a counter-clockwise rotation. Subtract one channel's counts from the other's to determine the net rotational change since the last sample. Accumulate the changes to determine the angular position.

Log State Store

Command: **0x15**

Operands: **Operand Byte 1-2** Number of State Bytes **(unsigned short)**

Return Data: **Checksum (signed byte)**

Stores log mode state data from the host in a specific RAM area for later retrieval. The format of this data is application-specific. The maximum number of bytes that can be stored is 3072. The returned checksum is the least significant byte of the one's compliment of the sum (signed long) of all data bytes.

Log State Retrieve

Command: **0x16**

Operands: **none**

Return Data: **Number of Bytes (unsigned short)**
State Bytes

Returns log mode state data to the host from a specific RAM area. The maximum number of bytes that can be retrieved is 3072. The number and order of bytes retrieved are exactly as stored with the last *Log State Store* command.

4.0 Sampling Control Commands

Start Sampling

Command: 0x21

Operands: none

Return Data: sample data records (see format below)

Start continuous sampling based on the current sampling configuration. Buffer is cleared first. While sampling progresses, no other command which returns data should be issued; as the returned data would be interspersed with data bytes form the buffer. Sampling produces a number of different types of data records. These are variable-length records whose type is indicated by the upper nibble of the first data byte. The following are descriptions of the record types:

Clocked Sample Record

Generated at the clocked sample period defined in the *Rate Select* command.

0x1d	(d= digital channel states 1,2. Bit 0=ch1, Bit 1=ch2)
Analog Channel A Sample	(2 bytes, signed, representing +/- 10v or +/- 1V w/gain)
Analog Channel B Sample	(2 bytes, signed, representing +/- 10v or +/- 1V w/gain)
Analog Channel C Sample	(2 bytes, signed, representing +/- 10v)
Digital Channel 1 Count	(2 bytes, unsigned, representing number of falling edges)
Digital Channel 2 Count	(2 bytes, unsigned, representing number of falling edges)

The digital states are always included in an analog sample record. If a particular analog input channel is not selected, then the bytes for that sample are omitted, and the order remains the same. If a particular digital channel is not selected for counting, then the bytes for that sample are omitted, and the order remains the same. When a digital counter value is reported, then the counter is cleared. Thus the count is a count of falling edges detected in the last clock period.
To convert from raw analog samples to voltage, multiply by 10 and divide by 32767.

Digital Event Record

Generated each time a change in the state of any event-selected digital I/O channel is detected.

0x2d	(d=digital channel states 1,2. Bit 0=ch1, Bit 1=ch2)
time of event	(4 bytes, unsigned: sample periods since last *Start Sampling*)

Pause Record

0x40

Generated each time a pause command is received. This record flags where pause commands are received in the data stream.

Motion Timer Record

Generated at the period defined in the *Rate Select* command when the motion timer is selected.

0x50	
timer value	(2 bytes, unsigned: µsec between pin and echo)
time of echo	(4-bytes, unsigned: sample periods since last *Start Sampling*)

Sample State Record

Generated in response to a *Check Sample State* command, or to report a buffer event such as the buffer becoming full, the trigger event being reached, or a data run ending.

0x6s (s=state flags)

bit 0=Interface has detected the trigger condition.
bit 1=Interface buffer has become full, and sampling has stopped automatically
bit 2=Interface is waiting for a trigger condition to begin sampling
bit 3=Interface has ended a recorded data run.

Trigger Time Offset Record

Generated when a trigger condition is reached. This will be the first record sent from the interface prior to the pre-trigger data. The time stamp offset in this record should be used to adjust all other time stamps that follow (in Digital Event and Motion Timer records) in a given recorded run. This will stamp the pre-trigger data with negative times, with time = 0 being the trigger point.

0xF0

time stamp offset (4-bytes, unsigned:sample periods since last *Start Sampling*)

Stop Sampling

Command: 0x22

Operands: none

Return Data: none

Stops all sampling and digital pinging. Sampling options are preserved. Buffer contents will be preserved and transferred. Note that sampling can stop automatically if the buffer becomes full, in which case a Sample State Record will appear at the end of the sample data.

Pause Sampling

Command: 0x23

Operands: none

Return Data: none

Pauses all sampling and digital pinging. The buffer is cleared. A pause event record is added to the buffer. Has no effect if not started sampling. Clocked samples, event reporting, motion sensor pinging, and sample time accumulation are suspended.

Resume Sampling

Command: 0x24

Operands: none

Return Data: none

Resumes sampling and digital pinging after a pause. Data is written to the buffer following the pause record. This command should only be used after sampling and pausing.

Reset Buffer

Command: 0x25

Operands: none

Return Data: none

Sets the buffer to an empty state.

Check Buffer State

Command: 0x26

Operands: none

Return Data: number of data bytes in buffer (2 bytes unsigned)
Number of logged data runs in buffer (1 byte unsigned)

Returns the number of data bytes and number of logged runs in the buffer. Data bytes may be left in the buffer after sampling has been stopped. They are not transferred to the host unless the host issues a *Read Buffer* command after stopping sampling. Multiple data runs are accumulated in the buffer by log commands issued by the user via the front panel button on the interface.

One-Shot Sample

Command: **0x27**
Operands: **none**
Return Data: **1 Clocked Sample Record and (optionally) 1 Motion Timer Record.**

Performs a single sample of each selected analog channel and digital counter. Also performs one ping-echo timing of a motion sensor if selected. The selections in the last *Input Select* command are used for one-shot sampling. If no analog channels or digital counters are selected, then a one-byte clocked sample record is generated containing only the digital states. If the motion timer is not selected, then no motion timer record is generated. See the *Start Sampling* command for a description of the Clocked Sample Record and Motion Timer Record.

Write Digital

Command: **0x28**
Operands: **Operand Byte 1**

bit 0-3	Unused
bit 4	Digital Channel 1 Data
bit 5	Digital Channel 2 Data
bit 6	Digital Channel 1 Output Enable (0=enable)
bit 7	Digital Channel 2 Output Enable (0=enable)

Return Data: **none**

Set the digital channel data bits and output enabling state. The data bits will be stored, but not output if the output enable bit for a channel is set to 1. If output is disabled for a channel, the the channel is an input.

Read Buffer

Command: **0x29**
Operands: **none**
Return Data: **sample data records (see format in *Start Sampling*)**

Causes any data bytes in the buffer to be transmitted to the host and immediately cleared from the interface. Data bytes may remain in the interface buffer after a *Stop Sampling* command is issued. The *Check Buffer State* command reports how many data bytes are available. Data bytes will also be in the buffer when the interface has been used for logging away from the computer. In this later case, the *Read Buffer Block* command can provide a more reliable transfer of a large amount of logged data.

Check Sample State

Command: **0x2A**
Operands: **none**
Return Data: **sample state record (see format in *Start Sampling*)**

Causes a sample state record to be inserted into the data stream. This command is useful for checking the interface state during prolonged waits for digital events, or very slow-clocked samples.

Enable Data Logging

Command: 0x2B
Operands: none
Return Data: none

Causes the interface to enter low-power sleep mode in preparation for disconnection of power and data cables, and eventual data logging. In this mode, the front-panel button is enabled so that the user may start and stop data logging recording to create multiple data runs. Data transmission to the host is disabled.

Disable Data Logging

Command: 0x2C
Operands: none
Return Data: none

Causes the interface to wake up from low-power sleep mode after data logging. The front-panel button is disabled. The *Check Buffer State* command can be used to determine if (and how much) log data is in the interface. The *Read Buffer* or *Read Buffer Block* command will cause the interface to transmit the buffer data.

Read Buffer Block

Command: 0x2D
Operands: Operand Bytes 1-2 Start Position **(unsigned)**
Operand Bytes 3-4 End Position **(unsigned)**
Return Data: sample data record bytes + 1 byte checksum

Causes a range of data bytes from the buffer (specified by the start and end position) to be transmitted to the host. The data remains in the buffer on the interface. A start position of 0 indicates the first byte in the buffer. The *Check Buffer State* command is used to determine the maximum end position. To read the entire buffer, request blocks with successive start/end positions (512 byte blocks are recommended). Note that a data record may be split between successive block reads. Each block will be followed by a single checksum byte (an unsigned byte which is the sum of all unsigned data bytes). If a computed checksum does not match the transmitted checksum, then a block may be re-requested any number of times. Once all data blocks have been successfully read, the buffer can be cleared with a *Reset Buffer* command.

Command Sequence Examples

Detecting and Downloading

- Send Identify command
- Wait for all returned data bytes
- Verify the interface model number and that the interface is in ROM mode
  (if already in RAM mode, then downloading is not necessary)
- Read lines from the file SW500RAM.S28, convert to binary and send a Download command for each line
- After each Download command, wait for a checksum byte and verify it
- After all download commands, send an Execute command
- Send Identify command again, wait for returned data bytes and verify RAM mode

Continuous Sampling

- Send Trigger Select command (to set or clear a trigger)
- Send Input Select command
- Send Rate Select command (use large buffer)
- Set Rotary Motion Resolution (if using an RMS)
- Send Start Sampling command
- Periodically read and decode sample data records from the serial port
  (Note that a sample data record may be split between two serial port reads on the host computer. Save a partial record from the end of one serial port read and add it to the beginning of the next serial port read. The expected length of each record type can be computed based on the selected channels.)
- Send Stop Sampling command

Burst (Scope) Sampling

- Send Trigger Select command (to set or clear a trigger)
- Send Input Select command (select analog channels only)
- Send Rate Select command
  (clock period should = 1, use small buffer)
- Send Start Sampling command.
  For higher sample rates (10 KHz and higher) it is likely that the interface will fill the small buffer and automatically stop sampling (indicated by a Sample State record at the end of the data).
- Read and decode sample data from the serial port until either a Sample State record indicates that sampling stopped automatically, or the desired number of samples has been read.
- If sampling did not stop automatically, send a Stop Sampling command
- Repeat all of the above steps for additional scope traces.