

# SEI Absolute Encoder Communications Protocol

## USER GUIDE

Each encoder on the bus is assigned a unique hexadecimal address between 0 and E. The host computer is the master of the bus, while the encoders are slaves. Encoders cannot initiate communication, rather the host sends commands and the encoders respond. The first byte of a command always includes the address of the encoder to be selected. Address F is used to communicate to all devices on the bus at once. An encoder responds by activating the busy line if a valid command is received with the correct address. If the address is incorrect or the command is invalid (including framing errors), the encoder ignores the command. If the address is F, all encoders activate the busy line until the command is processed by everyone (wired-OR). If an encoder has the busy line active while processing a command, other encoders ignore all data.

In the case of a single-byte command, the host only sends one byte which contains the address and the command. The selected encoder activates the busy line, sends the response, and then releases the busy line. In the case of a multiple-byte command, the host first sends a byte which contains the address and an indication that a multiple-byte command is in process. The selected encoder activates the busy line to acknowledge the selection. The host sends the rest of the command and the busy line remains active until the particular command is completed. In the simplest single encoder configuration, if factory default modes don't need to be changed, only single-byte commands are needed. 16-bit and 32-bit numbers are transmitted with the most significant byte first (big-endian convention).

## SINGLE BYTE USER COMMANDS

### *Request Command*

7	6	5	4	3	2	1	0
cmd3	cmd2	cmd1	cmd0	addr3	addr2	addr1	addr0

### *Addr3-0*

Address of encoder to be selected (0 to E). Transmitting an address between 0 and E inclusive causes a single encoder to respond; address F selects any and all encoders on the bus, and when used in a command it will cause every encoder on the bus to respond to the transmitted command simultaneously (it does not cause the encoders to serialize their responses). If a command that returns data is sent by the host with an address of F, over an SEI network with multiple encoders, the data received by the host from the encoders will likely be garbled because all of the encoders will attempt to send the requested data at the same time. If the primary purpose of a command is to *transmit* parameters (such as to set resolution, see below), then the operation will succeed on multiple encoders when address F is used, but the multiple encoders will still attempt

to send back their checksum bytes simultaneously, causing the checksum to be garbled when the host receives it. This may or may not be a problem, depending on application (the application can choose to ignore the garbled checksum, and assume that the operation was successful).

Here are some examples involving the use of address F. In the first example, it makes no sense to read resolution from multiple encoders using address F; the encoders will all send data at slightly differing times, and it will appear to the host that only one encoder is responding (and responding with garbled data). On the other hand, it is possible to set resolution on all encoders on a network with address F; all the encoders will set their resolution to the requested value, but the checksum received by the host will likely be garbled because they will all send their responses at slightly different times.

In practical terms, the only commands that should be used with address F in a multi-encoder network are the strobe command, the check serial number command, the sleep command, and the wakeup command. If there is only one encoder on the bus, address F can be used for all operations.

cmd3-cmd0	Request Type
0000	(reserved for control code)
0001	position
0010	position + 1 byte status
0011	position + 2 bytes time + status
0100	strobe
0101	sleep
0110	wakeup
0111	reserved
1000	reserved
1001	reserved
1010	reserved
1011	reserved
1100	reserved
1101	reserved
1110	reserved
1111	(multiple byte command)

### ***Position***

The encoder sends its current position. The format is either 1 or 2 bytes in single-turn mode, 4 bytes in multi-turn mode, most significant byte first, (see "Change Mode Command" for details).

### ***Time***

The encoder sends the 2 byte value of its free running counter at the time of the position reading - most significant byte first. The hardware counter in encoders with version 3 firmware increments at 1.843MHz (1 count per 0.54 usec), while the counter in encoders with version 4 firmware increments at 7.373 MHz (0.136 usec per count).

## ***Strobe***

An SEI absolute encoder runs in one of two modes: free-running (also referred to as asynchronous mode) or strobe mode. The choice of which mode is active depends on bit “stb” in the Mode byte (the Mode byte bits are described later in this document under the “Change Mode” multi-byte command). When the “stb” mode bit is cleared to zero in the mode byte the encoder is free-running and the single byte strobe command (described in the table above, command nibble equal to 0100 binary) has no effect. In free-running mode the encoder continuously cycles through the computation of position; when queried the position value returned is the result from the last valid computation, and there is no correlation between the acquisition time of different encoders. The cycle time for the computation is 7 msec for firmware version 4 and up, while the cycle time for firmware version 3 is 4 msec. Thus an synchronous read of position in free-running mode will return a result that is up to one cycle time old (7 or 4 msec, depending on firmware version).

When the strobe mode bit “stb” in the user mode byte equals 1, the encoder operates in strobe mode: the encoder enters a waiting state, waiting for a single byte strobe command to be received before computing position. Thus, when multiple encoders are in strobe mode they are all waiting for the single byte strobe command, and they will all start computing position synchronously after all of them receive the single-byte strobe command (the strobe command should be sent with address F so that all the encoders are addressed simultaneously). The host must still wait at least one cycle time (7 or 4 msec) after issuing a strobe command in order to allow the encoder to finish the computation before attempting to read position (otherwise, the position read during the computation will be from a previous computation). Once the encoder finishes the computation it will go back to waiting for the next strobe command; during this period the host can read out position from multiple encoders in separate position queries to specific addresses, and the data that will be returned is all from the same instant that the last strobe command was issued to all the encoders.

Free-running mode is the default mode as sent from the factory, and in free-running mode the single-byte strobe command described in the table above has no effect. The user mode byte can be programmed to power up in strobe mode using the “Change Default Mode” multi-byte command described later in this document.

## ***Sleep***

The encoder goes to sleep (without responding) after the command is executed, and the current consumption drops below 2 mA. If there are multiple encoders on the bus, use address F to put them all asleep at once. Any activity on the data bus wakes up all encoders (use the wakeup command and address F to wake them synchronously).

## ***Wakeup***

This command wakes up all encoders on the bus; the encoders will not send any response to this command, however. Wait at least 5 msec after this command to send the next command.

## Status

The encoder sends 1 byte status as follows:

7	6	5	4	3	2	1	0
err3	err2	err1	err0	sum3	sum2	sum1	sum0

err3-err0	Meaning	Corresponding SEI Error Code (see source code for SEISrv32.c for more information)
0000	no error	n/A
0001	not enough light	28101
0010	too much light	28102
0011	misalignment or dust	28103
0100	misalignment or dust	28104
0101	misalignment or dust	28105
0110	hardware problem	28106
0111	fast mode error (V1.X)	28107
1000	multiturn pos. not initialized	28108

### Sum3-0

Exclusive OR of all 4-bit nibbles of the request byte received and data bytes returned by the encoder, excluding the status byte. This is a good check of data integrity, especially in the case of a bus with multiple encoders.

**Note:** In multi-turn mode, the error 1000 binary (0x8 in the upper hex nibble) is sent until a “set origin” or a “set position” command is received. This allows the host to know a reset or power loss has occurred and the multi-turn position is no longer valid.

## MULTIPLE BYTE COMMANDS

All multiple byte commands start with the request byte F0+addr; after receiving this byte, the addressed encoder will acknowledge by activating the busy line. After the acknowledgment, the encoder is ready to receive the rest of the command. Some commands may require the use of address F to select all encoders on the bus (like check serial number, baud rate change, etc...). When addressing multiple encoders, after the first busy received, wait 5 msec to send the rest of the command, to make sure they are all ready. When the command is successfully completed, the encoder sends a checksum byte, and then releases the busy line. If the command is invalid or failed, the encoder releases the busy line without sending a checksum. The checksum byte is the exclusive OR of the request byte, command bytes, and bytes returned by the encoder.

## ***Set Origin Command***

2 bytes: request byte, 01

Returns checksum if command is successful.

Sets the absolute 0 at the current position.

In single-turn mode, the new origin is stored in EEPROM, therefore, it will be effective after resets and power downs until a "Set Origin" or a "Set Absolute Position" command is received.

In multi-turn mode, the 32 bit counter is reset, but not stored in EEPROM. This is effective until a reset occurs or a "Set Origin" or a "Set Absolute Position" command is received.

## ***Set Absolute Position Command***

Single-turn: 4 bytes: request byte, 02, pos. MS byte, pos. LS byte

Multi-turn: 6 bytes: request byte, 02, 4 bytes pos. (MS to LS)

Returns checksum if command is successful.

Sets the given absolute position (at the current resolution) at the current position.

In single-turn mode, the new origin is stored in EEPROM, therefore it will be effective after resets and power downs, until a "Set Origin" or a "Set Absolute Position" command is received.

In multi-turn mode, the 32 bit counter is set, but not stored in EEPROM. This is effective until a reset occurs or a "Set Origin" or a "Set Absolute Position" command is received.

## ***Read Serial Number***

2 bytes: request byte, 03

Returns 5 bytes: 4 bytes serial number and checksum if command is successful.

## ***Check Serial Number***

10 bytes: request byte, 04, 4 bytes serial number, 4 bytes mask

Returns nothing.

The encoder does a logical AND of its serial number with the mask supplied; the result is compared to the serial number supplied. If they match, the busy line is held active until another byte is received. Otherwise the busy line is released. This command is used to determine if an encoder with a particular serial number is present on the bus.

## ***Fail Serial Number***

10 bytes: request byte, 05, 4 bytes serial number, 4 bytes mask

Returns nothing.

The encoder does a logical AND of its serial number with the mask supplied; the result is compared to the serial number supplied. If they don't match, the busy line is held active until another byte is received. If they match the busy line is released. This is useful to determine if an encoder, whose serial number is known, is the only one on the bus.

### ***Get Address***

6 bytes: request byte, 06, 4 bytes serial number

Returns 2 bytes: 1 byte address and checksum if command is successful, only if serial number matches.

The encoder compares its serial number with the one supplied; if they match, it returns its address (0 to E). Otherwise, it returns nothing.

### ***Assign Address***

7 bytes: request byte, 07, 4 bytes serial number, 1 byte address

Returns checksum if command is successful.

The encoder compares its serial number with the one supplied; if they match, it assigns itself the address supplied (must be between 0 and E). The new address is stored in EEPROM, therefore, it will be effective after resets and power downs.

### ***Read Factory Info***

2 bytes: request byte, 08

Returns 15 bytes: 2 bytes model number, 2 bytes version, 2 bytes configuration, 4 bytes serial number, month, day, 2 bytes year and checksum if command is successful.

### ***Read Resolution Command***

2 bytes: request byte, 09

Returns 3 bytes: resolution MS byte, resolution LS byte and checksum if command is successful.

A zero value means 16 bit resolution.

### ***Change Resolution Command***

4 bytes: request byte, 0A, resolution MS byte, resolution LS byte

Returns checksum if command is successful.

The resolution can be any number between 0 and FFFF, 0 is for full 16 bit position. However, the accuracy is only guaranteed to 12 bits. The new resolution is stored in EEPROM, therefore, it will be effective after resets and power downs.

### ***Read Mode Command***

2 bytes: request byte, 0B

Returns 2 bytes: mode and checksum if command is successful.

## Change Mode Command (temporary)

3 bytes: request byte, 0C, mode

Returns checksum if command is successful.

The mode is changed temporarily and will be effective until the encoder is reset, power down, or another mode change command is received. It is not stored in the EEPROM. Mode byte as follows:

7	6	5	4	3	2	1	0
reserved	/256	reserved	incr	size	multi	stb	rev

### **rev**

rev = 1, the position increases counter clockwise.

rev = 0, the position increases clockwise.

### **stb**

stb = 1, the encoder operates in strobe mode: it waits for a strobe request before computing the position; this mode is used to synchronize multiple encoders. After entering this mode, wait at least one computation cycle time before sending the first strobe command.

stb = 0, the encoder operates in asynchronous mode: the encoder cycles through computing position continuously, and sends the most current valid position when requested. The data can be from 0 to one cycle time old (version 3 firmware cycle time is 4 msec, and version 4 firmware cycle time is 7 msec).

### **multi**

multi = 1, multi-turn mode: a 32 bit counter keeps track of the position (it increases or decreases over multiple turns, i.e. 3 1/2 turns at a resolution of 100 would be 350). This counter is cleared at reset or power-up.

multi = 0, single-turn mode: position is between zero and the max resolution, according to the shaft angle.

**Note:** in older versions (V1.X), this bit indicated a fast mode (3 msec update rate) with a 9 bit accuracy.

Also, any other command besides position inquires can corrupt the multi-turn position.

### **size**

Only effective in single-turn mode:

size = 1: the encoder always sends the position in 2 bytes, even if the resolution is 256 decimal or less.

size = 0: the position is sent as 1 byte if the resolution is up to 256 decimal, or as 2 bytes if above 256 decimal.

In multi-turn mode, the position is always 4 bytes and this bit is ignored.

### ***incr***

Only effective in multi-turn mode:

*incr* = 1: the encoder sends the position change since the last request, as a 32 bit signed number.

*incr* = 0: the encoder sends the position as a 32 bit signed number.

### ***/256***

Only available for analog version.

Only effective in multi-turn mode:

*/256* = 1: the encoder position is divided by 256.

*/256* = 0: the encoder position is normal.

### ***Change Power Up Mode Command***

3 bytes: request byte, 0D, mode.

Returns checksum if command is successful.

Same as “**Change Mode Command**” described above, except the mode is stored in EEPROM, therefore it will be effective across resets and power cycles.

### ***Reset Command***

2 bytes: request byte, 0E

Returns checksum if command is successful.

After releasing the busy line the encoder does a software reset (the baud rate returns to 9600 after the checksum byte is sent). The encoder requires 35 msec after reset to be ready to receive new commands. The other ways to reset are: turning the power off, or sending a break condition (dataL low, dataH high) for at least 1 second. Note that the reset threshold of the power input is between 4.5 and 5.5 volts. The EEPROM parameters are not affected, but any temporary mode byte changes or baud rate change are lost.

## **USER DIAGNOSTIC COMMANDS**

### ***Loopback Mode***

2 bytes: request byte, 10

Returns no checksum.

After receiving this command the encoder returns every byte it receives. This is useful to test the SEI bus. The busy line remains active during this test. The test is aborted if a delay of 350msec or more occurs. (rev 1.01 firmware was 35 msec). The host must release the data lines after each byte sent, to allow the encoder to respond.

## **Off line Command**

2 bytes: request byte, 11

Returns checksum if command is successful.

After releasing the busy line the encoder does not respond to any command. To place it back on line it must receive a break condition (dataL low, dataH high) for at least 1 second or power down and up again. We do not recommend using this command. Not all firmware revisions support this command.

## **Change Baud Rate Command (temporary)**

3 bytes: request byte, 0F, baud rate byte

Returns checksum if command is successful.

The rate will be changed as follows and will be effective after checksum is sent until the encoder is reset or another baud rate change command is received. At reset, the baud rate always defaults to 9600. Note: If multiple encoders are on the same bus, use address F to set them all to the same baud rate.

<b>Baud Rate Byte</b>	<b>0x00</b>	<b>0x01</b>	<b>0x10</b>	<b>0x11</b>	<b>0x12</b>	<b>0x13</b>	<b>0x14</b>	<b>0x15</b>
Baud Rate	115200	57600	38400	19200	9600	4800	2400	1200

### **ABOUT US DIGITAL**

With over a million off-the-shelf configurations, plus any number of custom product offerings, US Digital® has delivered quality in motion since 1980. US Digital offers stellar service, delivering motion control solutions best suited to each unique need. Automated systems, continuous improvement protocols and stringent testing ensures we bring quality to every product manufactured.

Located in Vancouver, Washington, our vertically integrated facility and personalized service provides customers with industry lead times significantly shorter than the competition, offering same-day fulfillment on most orders.

**Visit [usdigital.com](https://usdigital.com) or contact:**

[info@usdigital.com](mailto:info@usdigital.com) | **Worldwide:** 360.260.2468 | **US Toll Free:** 800.736.0194