Programming Orbis encoders with Asynchronous serial interface

Programming of the Orbis encoder via UART interface allows to:
- set position offset,
- set multiturn counter (if available),
- set new baud rate value,
- configure continuous response feature,
- start/stop continuous response feature,
- run the self-calibration function,
- store current Orbis configuration parameters to a non-volatile memory,
- reset Orbis configuration parameters to factory settings.

Programming is executed by sending separate bytes to the encoder.

Each of the programming possibilities must be started with the special unlocking sequence. This prevents unintentional changes on configuration parameters of the encoder. The procedure of each programming posibility is described below.

NOTE: Delay between each byte sent during programming must be a least 1 ms.

Unlocking sequence

To unlock the encoder and enable programming, four exact bytes must be sent in specified order. They are listed in table below:

Byte in sequence	Byte value (HEX)
B1	0xCD
B2	0xEF
B3	0x89
B4	0xAB

Whenever the sequence is interrupted by a wrong byte, the unlock stage is reset and waits for the new sequence.

Image below shows the unlocking sequence captured by the logic analyzer.
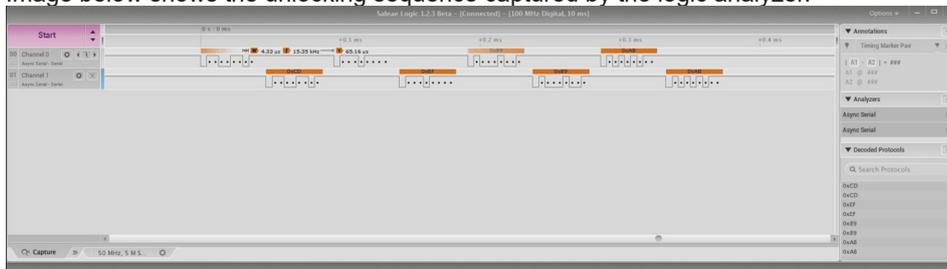
When the encoder is successfully unlocked, the fifth byte in a complete sequence must be one of the valid programming command bytes described below. If the fifth byte is none of them, the encoder becomes »locked« again.

A **RENISHAW** associate company

Programming command bytes

Programming command bytes vary depending on the desired programming feature. The valid programming bytes are listed in table below. Some of them execute the required task immediately (i.e. reset factory settings), while others need additional data bytes (i.e. setting new baud rate). Specific tasks are described in the following paragraphs. Always, when programming is completely executed, the programming possibility returns back to the »locked« stage.

Programming feature	Programming command byte (ASCII)		Additional data bytes required
Position offset setting	'Z'	(0x5A)	4
Multiturn counter setting	'M'	(0x4D)	4
Baud rate setting	'B'	(0x42)	4
Continuous-response setting	'T'	(0x54)	4
Continuous-response start	'S'	(0x53)	0
Continuous-response stop	'P'	(0x50)	0
Configuration parameters save	'c'	(0x63)	0
Configuration parameters reset	'r'	(0x72)	0

Programming execution

Position offset setting

After sending a programming command byte 'Z' which enables position offset setting four additional data bytes are required. These bytes contain the actual position offset data in encoder's counts. When all four bytes are received, the new position offset value is set. Table below shows, how these four bytes are transformed into a 32-bit value, which will present a position offset.

Byte in sequence	Bits in 32-bit position offset
B1	b31 – b24
B2	b23 – b16
B3	b15 – b8
B4	b7 – b0

If the applied position offset is larger than the actual encoder resolution, programming will be discarded. The position offset will not change.

After position offset programming, the value is stored in RAM and used immediately. For permanent storing in a non-volatile memory, another programming procedure for saving configuration parameters must be executed (see "Saving configuration parameters" on page 4).

Default (factory) position offset is set to 0.

Multiturn counter setting

After sending a programming command byte 'M', which enables multiturn counter setting, four additional data bytes are required. These bytes contain the desired multiturn counter. When all four bytes are received, the new multiturn counter is set. Table below shows, how these four bytes are transformed into a 32-bit value, which will present a multiturn counter.

Byte in sequence	Bits in 32-bit multiturn counter
B1	b31 – b24
B2	b23 – b16
B3	b15 – b8
B4	b7 – b0

Only lower 16 bits are used to preset multiturn counter. B1 and B2 must be zero.

Baud rate setting

After sending a programming command byte 'B', which enables baud rate setting, four additional data bytes are required. These bytes contain the new required baud rate in bits per second. When all four bytes are received, the new baud rate is set.
Table below shows, how these four bytes are transformed into a 32-bit value, which will present a new baud rate.

Byte in sequence	Bits in 32-bit baud rate value
B1	b31 – b24
B2	b23 – b16
B3	b15 – b8
B4	b7 – b0

After baud rate programming sequence is complete, the value is stored in RAM and used instantly. Further communication with the encoder is not possible with the previous baud rate. Master must reconfigure itself to the new baud rate and test the communication with the encoder. In case of proper communication, the new baud rate must be stored in a non-volatile memory by programming procedure for saving configuration parameters (see "Saving configuration parameters" on page 4). In that case, the new baud rate will be applied after each subsequent cycle of the power supply. If new configuration is not stored into non-volatile memory, previous setting will be restored on a power cycle.

Baud rate programming allows programming of any baud rate in increments of 1 BAUD.

Continuous-response setting

After sending a programming command byte 'T', which enables continuous-response setting, four additional data bytes are required. These bytes contain:
- required period of continuous responds in microseconds,
- command that should be continuously responded,
- setting for automatic start of responding after power-on of the encoder.

Table below shows how these four bytes are transformed into separate configuration parameters.

Byte B1	Meaning
b7 – b1	Not used
b0	1 to enable automatic start after power-on of the encoder 0 to disable automatic start after power-on of the encoder
Byte B2	
b7 – b0	ASCII command for continuous responding (one from the basic command set, refer to Orbis data sheet, document number MBD01.)
Byte B3	
b7 – b0	b15 – b8 of the 16-bit period value
Byte B4	
b7 – b0	b7 – b0 of the 16-bit period value

Automatic start of the continuous-responding after power-on of the encoder is disabled by default.

Default command for continuous responding is ASCII '3' (short response). Any other command from table "Command(ASCII)" could be programmed. If none of the valid commands is programmed, it is discarded and set back to default (ASCII '3').

After continuous-response programming, the parameters are stored in RAM. For permanent storing in a non-volatile memory, another programming procedure for saving configuration parameters must be executed (see "Saving configuration parameters" on page 4). After that, the programmed values become valid after first subsequent power cycle of the encoder.

Resolution of the period for the continuous-responding is 1 µs. The maximum programmed period could be 65535 µs. The shortest possible period depends on used baud rate and selected command. In case of programmed period being too short and previous frame was not transmitted yet, the transmission of the next frame will be delayed for the programmed period. If user prefers the shortest possible period regardless to the baud rate and command settings, a period setting of 1 µs can be applied.

See programming example on "Continuous-response setting" (page 5) which shows the appropriate continuous-response programming sequence.

A RENISHAW associate company

Continuous-response start

After sending a programming command byte 'S', continuous-response feature is immediately enabled. The encoder starts transmitting data according to the selected continuous-response period and command.

Continuous-response stop

After sending a programming command byte 'P', continuous-response feature is immediately disabled.

Saving configuration parameters

After sending a programming command byte 'c', configuration parameters of the encoder are immediately saved in a non-volatile memory. These parameters include:
- current baud rate,
- position offset,
- continuous-response settings (period, command, auto-start enable command).

Reset of the encoder to the factory settings

After sending a programming command byte 'r', configuration parameters of the encoder are immediately reset to the factory values.

Self-calibration

Self-calibration of Orbis encoders is suitable after readhead assembly. It improves the encoder accuracy which is dependent on the installation precision. User must first unlock the programming option by sequence 0xCD 0xEF 0x89 0xAB, following the SelfCalStart command 0x41 for starting self-calibration procedure. During the procedure, communication over UART interface is not possible; encoder will not respond to any incoming commands. First command received during this period will be put into queue and processed at the end of calibration cycle.

Procedure completion is indicated by fast LED blinking for 3 seconds. If self-calibration was successful, LED blinks green, otherwise it blinks red. After that, UART interface is active again. The self-calibration status byte can be read with 0x69 command. It will return an echo byte first, followed by a status byte, which includes a two bit counter and two status bits. Counter is incremented after end of each self-calibration procedure. Error bits show success or reasons for failure.

Before self-calibration procedure, the status should be read. Command 0x69 will return 2 bytes, first is echo byte and status byte follows. Controller must remember current self-calibration counter (bits 1:0). After the command for self-calibration is sent, LED has to be observed for the procedure completion. If LED is not visible, the readhead should be polled via UART interface until communication with the readhead is established again or waiting for 10 seconds, which is the longest possible time for completion. After that, self-calibration status register should be read again. If self-calibration counter has increased by 1 (compared to the value read before), the self-calibration function was completed. If self-calibration was successful, both of status bits (b3, b2) are zero.

Rotational speed and direction during the self-calibration is not important and could be uneven. The only requirement is that the shaft makes at least one complete turn during 10 second period after the command was sent.

Self-calibration status byte values

Bit	Meaning
b7 - b4	Reserved
b3	Calculated parameters out of range. Mechanical installation is not inside tolerances.
b2	Timeout. Encoder ring did not make a complete turn during 10 seconds.
b1	Counter bit 1
b0	Counter bit 0

Unlock sequence: 0xCD 0xEF 0x89 0xAB
Self-calibration Start command: 0x41
Self-calibration Status request: 0x69

Programming examples

Encoder position offset setting to 5144 (0x1418) counts

Bytes listed in the table should be sent to the encoder in exact order with at least 1 ms of delay between bytes.

Byte in sequence	Value			Purpose
	DEC	HEX	ASCII	
B1	205	0xCD	/	Unlock sequence
B2	239	0xEF	/	
B3	137	0x89	/	
B4	171	0xAB	/	
B5	90	0x5A	'Z'	Offset position setting enable
B6	0	0x00	/	Additional data bytes
B7	0	0x00	/	
B8	20	0x14	/	
B9	24	0x18	/	

To store offset position in a non-volatile memory, encoder's configuration parameters must be saved in accordance to "Saving configuration parameters in a non-volatile memory" on page 6.

Continuous-response setting

Continuous-response configuration parameters:
- period: 250 µs (4 kHz)
- command: '3' (shortest possible, 3 bytes of position + E/W)
- auto-start after power-on: Enabled

Bytes listed intable below should be sent to the encoder in exact order.

Byte in sequence	Value			Purpose
	DEC	HEX	ASCII	
B1	205	0xCD	/	Unlock sequence
B2	239	0xEF	/	
B3	137	0x89	/	
B4	171	0xAB	/	
B5	84	0x54	'T'	Continuous-response setting enable
B6	1	0x01	/	Auto-start enable
B7	51	0x33	'3'	Command for short response
B8	0	0x00	/	Period (250 us)
B9	250	0xFA	/	

Image below shows the described continuous-response programming sequence captured by the logic analyzer.
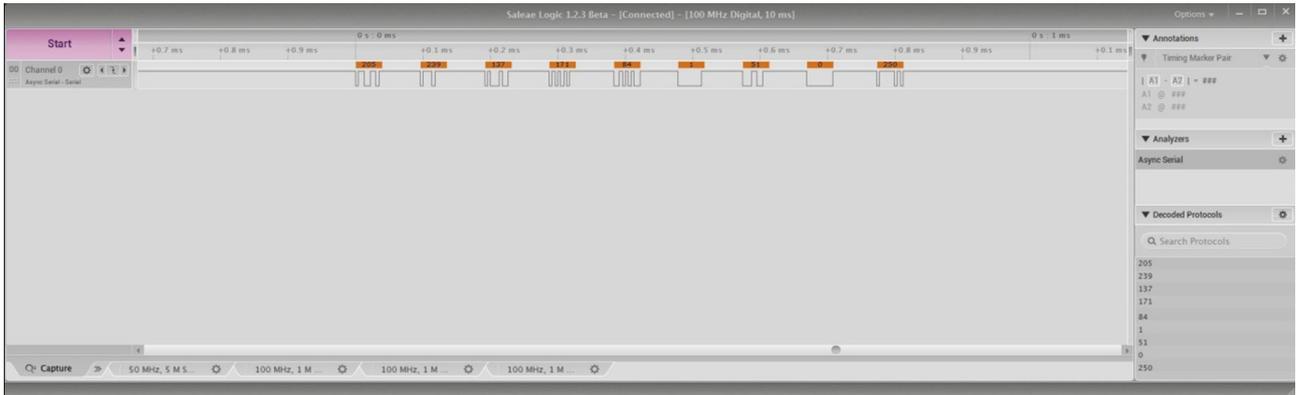
To store continuous-response settings in a non-volatile memory, encoder's configuration parameters must be saved in accordance to "Saving configuration parameters in a non-volatile memory" described below. After the next cycle of power supply, the encoder will start to transfer data automatically with 250 µs period as shown in image below.
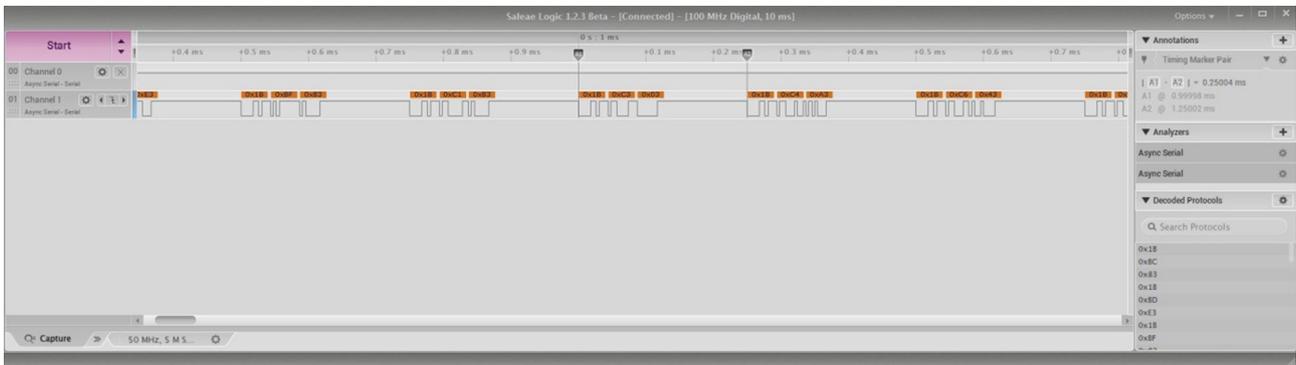
Saving configuration parameters in a non-volatile memory

Bytes listed in table below should be sent to the encoder in exact order.

Byte in sequence	Value			Purpose
	DEC	HEX	ASCII	
B1	205	0xCD	/	
B2	239	0xEF	/	
B3	137	0x89	/	Unlock sequence
B4	171	0xAB	/	
B5	99	0x63	'c'	Save configuration parameters

Reset of configuration parameters to the factory settings

Bytes listed in table below hould be sent to the encoder in exact order.

Byte in sequence	Value			Purpose
	DEC	HEX	ASCII	
B1	205	0xCD	/	
B2	239	0xEF	/	
B3	137	0x89	/	Unlock sequence
B4	171	0xAB	/	
B5	114	0x72	'r'	Reset configuration parameters

Document issues

Issue	Date	Page	Corrections made
1	22. 8. 2018	-	New document

A RENISHAW associate company