Application note
MBD03_02
Issue 2, 26th June 2019

Programming AksIM-2 encoders with Asynchronous serial interface

Programming of the AksIM-2 encoder via UART interface allows to:

- set position offset,
- set multiturn counter (if available),
- change baud rate value,
- configure continuous response feature,
- start/stop continuous response feature,
- running the self-calibration function
- store current AksIM configuration parameters to a non-volatile memory,
- reset AksIM configuration parameters to the 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 possibility is described below.

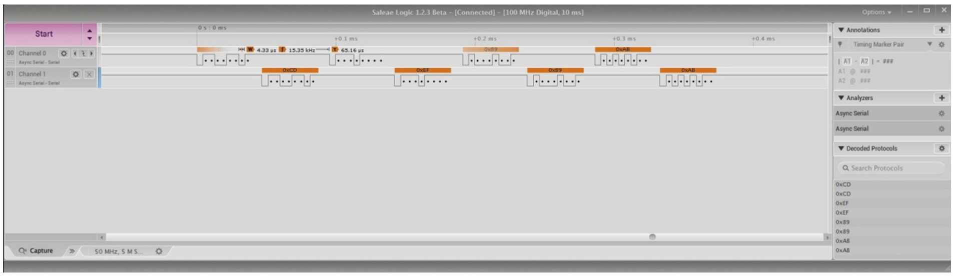
Unlocking sequence

To unlock the encoder and enable programming, 4 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 with wrong byte, the unlock stage is reset and waits for the new sequence.

Picture below shows the unlocking sequence captured by the logic analyzer.



Unlocking sequence should not be sent with maximum speed, but every echo byte should be read before sending next byte. Delay between each byte sent during programming must be a least 1 ms.
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.

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 executes 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
Partial arc for Self-calibration setting	'p'	(0x70)	2
Self-calibration start	'A'	(0x41)	0
Self-calibration status / result	'i'	(0x69)	0
Read error map *	'e'	(0x65)	0
Write error map *	'E'	(0x45)	1024
Write protection status	'w'	(0x77)	0
Write protect enable	'W'	(0x57)	0
Configuration parameters save	'c'	(0x63)	0
Reset to factory settings	'r'	(0x72)	0

* Contact RLS for details

Programming execution

Position offset setting

After sending a programming command byte ‘Z’, which enables position offset setting, 4 additional data bytes are required. These bytes contain the actual position offset data in encoder’s counts. When all 4 bytes are received, the new position offset value is set.
Table below shows, how these 4 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 or smaller than zero, value 0 is set as a new offset.
After changing zero position for a bigger value, acceleration error might appear.
After every setting of a new position offset, verify or adjust multiturn counter value (if present).

After position offset programming, the value is stored in a 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 5).

Default (factory) position offset is set to 0.
New position = Absolute position - Offset

Multiturn counter setting

After sending a programming command byte ‘M’, which enables multiturn counter setting, 4 additional data bytes are required.
These bytes contain the desired multiturn counter. When all 4 bytes are received, the new multiturn counter is set. Table below shows, how these 4 bytes are transformed into a 32-bit value, which will represent a new multiturn counter value.

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, 4 additional data bytes are required. These bytes contain the new required baud rate in bits per second. When all 4 bytes are received, the new baud rate is set. Table below shows, how these 4 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 5). 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 to program any baud rate in increments of 1 BAUD. Maximum supported baud rate is 1 Mbps.

Continuous-response setting

After sending a programming command byte ‘T’, which enables continuous-response setting, 4 additional data bytes are required. These bytes contain:

- the required period of continuous responds in microseconds;
- a command that should be continuously responded;
- a setting for automatic start of responding after power-up of the encoder.

Table below shows, how these 4 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	**Meaning**
b7 – b0	ASCII command for continuous responding (one from the basic command set, refer to AksIM-2 data sheet, document number MBD01.)
Byte B3	**Meaning**
b7 – b0	b15 – b8 of the 16-bit period value
Byte B4	**Meaning**
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)” in document MBD01 (page 22) could be programmed. If none of the valid commads is programmed, it is discarded and set back to default (ASCII ‘3’).

After continuous-response programming, the parameters are stored in a 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 5). 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 microsecond. The maximum programmed period could be 65535 microseconds. 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 microsecond can be applied.

See the “Continuous-response setting - Example” on page 6, which shows the appropriate continuous-response programming sequence.

> NOTE: When continuous response is active, any command sent to the encoder might result in echo byte being inserted into the position data packet. Recommended is to stop continuous transmission with command ‘P’ before sending any other command to the encoder.

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.

Self-calibration

Self-calibration of the AksIM-2 is suitable after assembly of the readhead. It improves the accuracy of the encoder, which depends on the installation precision. User must first unlock the programming option by sequence 0xCD 0xEF 0x89 0xAB, following the 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.

Before self-calibration procedure, the status should be read. Command 0x69 will return 2 bytes, first is echo byte and status byte follows which includes a two bit counter and two status bits. Counter is incremented after end of each self-calibration procedure. 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 val 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. Error bits show success or reasons for failure.

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

If mechanics do not allow rotation for 360° then calibration arc length can be reduced to minimum 180° with a command 0x70. This command must be followed by 2 bytes that contain required arc length (in degrees). New arc length must be set before executing self-calibration function. Performance of self calibration procedure is reduced if arc length is reduced.

Self-calibration status byte values

Bit	Meaning
b7	Reserved
b6	Calibration was already performed (error map was changed)
b5	No correction needed (mechanical installation is perfect)
b4	Arc parameter out of range.
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 : b0	Counter

Unlock sequence: 0xCD 0xEF 0x89 0xAB
Self-calibration Start command: 0x41
Self-calibration Status request: 0x69
Self-calibration arc length: 0x70

Parameter write protection

Write protection is used to lock the write access of any writable parameter. After locking the write access, the readhead cannot be programmed anymore. All settings or change possibilities are disabled. It also cannot be reset to the factory settings.
All parameters are still readable. Current status of write protection can be read with the command 0x77.

Unlock sequence: 0xCD 0xEF 0x89 0xAB
Write protect enable command: 0x57
Write protection current status: 0x77

Saving configuration parameters

After sending a programming command byte ‘c’, configuration parameters of the encoder are saved in a non-volatile memory.
These parameters include:

- current baud rate,
- position offset,
- continuous-response settings (period, command, auto-start enable command).

Unlock sequence: 0xCD 0xEF 0x89 0xAB
Save parameters to non-volatile memory command: 0x63

> NOTE: Saving parameters to non-volatile memory takes 80 ms. During this time encoder position is not calculated.
> In case encoder has multi-turn counter option, speed should not exceed ±300 RPM. Otherwise multi-turn counter value becomes invalid.

Reset of the encoder to the factory settings

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

Unlock sequence: 0xCD 0xEF 0x89 0xAB
Reset of the encoder to the factory settings: 0x72

> NOTE: Reseting encoder to factory settings takes 80 ms. During this time encoder position is not calculated.
> In case encoder has multi-turn counter option, speed should not exceed ±300 RPM. Otherwise multi-turn counter value becomes invalid.
>
> WARNING: After locking the write access, the encoder cannot be reset to the factory settings.

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 7.

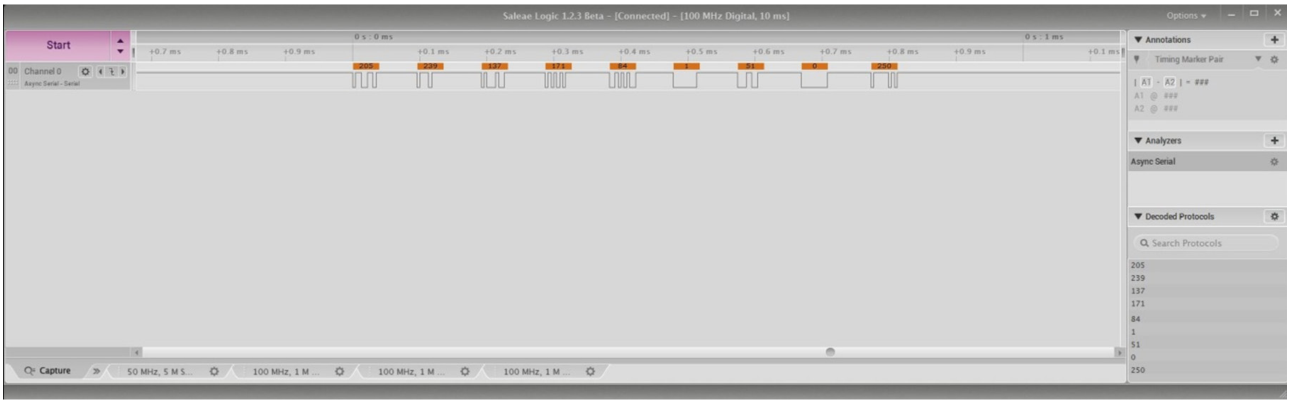
Continuous-response setting

Continuous-response configuration parameters:
- Period: 250 microseconds (4 kHz)
- Command: '3' (shortest possible, 3 bytes of position + E/W)
- Auto-start after power-on: Enabled

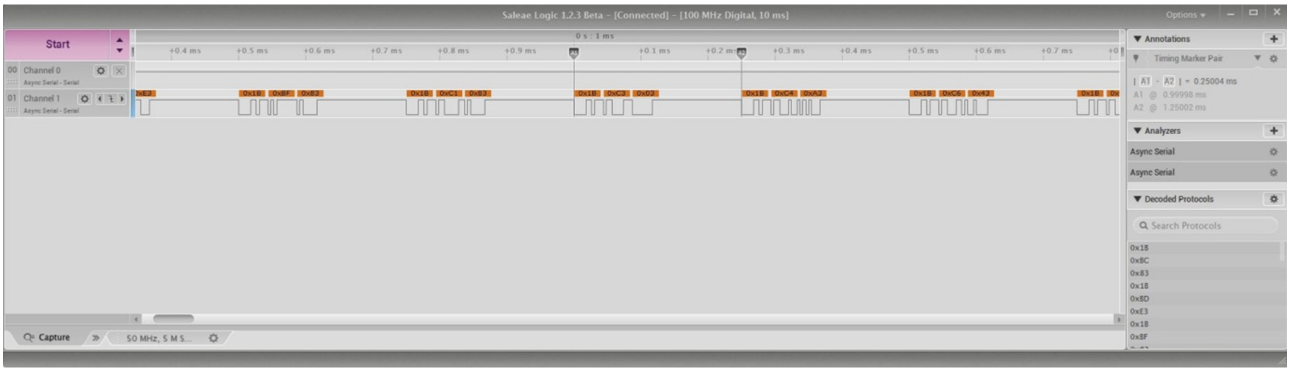
Bytes listed in table below should be sent to the encoder in exact order.

Byte in sequence	Value			Purpose
	DEC	HEX	ASCII	
B1	205	0xCD	/	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	/	

Picture 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" below. After next cycle of the power supply, the encoder will start to transfer data automatically with 250 µs period as shown in picture 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	/	Unlock sequence
B2	239	0xEF	/	
B3	137	0x89	/	
B4	171	0xAB	/	
B5	99	0x63	'c'	Save configuration parameters

Reset of configuration parameters to the factory settings

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	/	Unlock sequence
B2	239	0xEF	/	
B3	137	0x89	/	
B4	171	0xAB	/	
B5	114	0x72	'r'	Reset configuration parameters

Head office

RLS merilna tehnika d.o.o.
Poslovna cona Žeje pri Komendi
Pod vrbami 2
SI-1218 Komenda
Slovenia

T +386 1 5272100
F +386 1 5272129
E mail@rls.si
www.rls.si

Document issues

Issue	Date	Page	Corrections made
1	18. 12. 2017	-	New document
2	26. 6. 2019	2-5	New commands added