Programming AksIM-2 encoders
with Asynchronous serial interface

Programming the AksIM-2 encoder via the 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,
- perform the self-calibration function
- save current AksIM configuration parameters to non-volatile memory,
- reset AksIM configuration parameters to factory defaults.

Related product

AksIM-2 off-axis absolute magnetic encoder

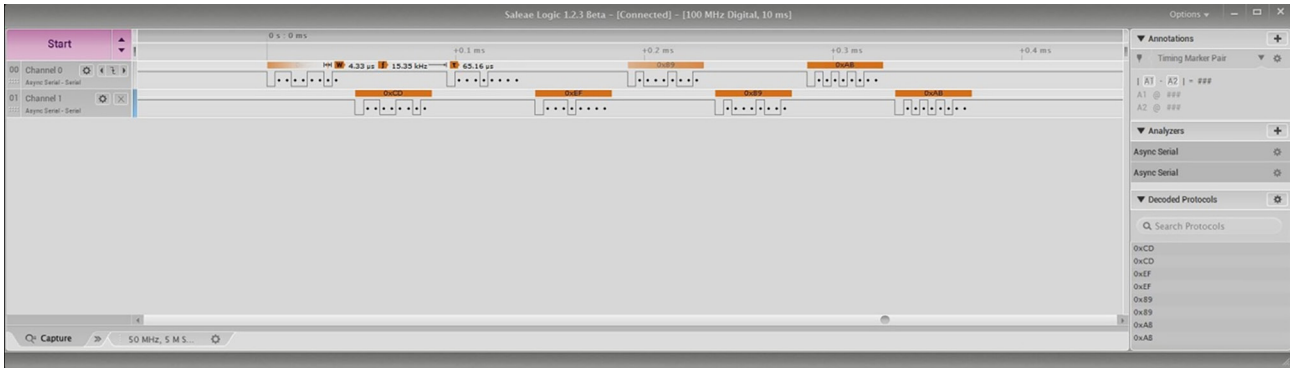
Unlocking sequence

Programming is done by sending separate bytes to the encoder.
Each of the programming options must be started with the special unlock sequence.
This prevents unintentional changes on configuration parameters of the encoder.
The procedure for each programming option is described below.

To unlock the encoder and enable programming, 4 exact bytes must be sent in specified order as listed in the following table.

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.
Screen capture below shows the unlocking sequence captured by the logic analyzer.



Unlocking sequence, commands and data should not be sent at maximum speed, but each echo byte should be read before the next byte is sent.
The delay between each byte sent during programming must be at 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 these, the encoder will be »locked« again.

Programming command bytes

Programming command bytes vary depending on the desired programming feature. The valid programming bytes are listed in the following table. Some of these perform the desired task immediately (e.g. resetting factory defaults), while others require additional data bytes (e.g. setting a new baud rate). Specific tasks are described in the following sections. Each programming command must begin with the Unlock sequence. When the command is executed, the encoder always returns to the locked state.

Programming feature	Unlock required	Programming command byte (ASCII)		Additional data bytes required	Data bytes returned **
Position offset setting	Y	'Z'	(0x5A)	4	1
Multiturn counter setting	Y	'M'	(0x4D)	4	1
Baud rate setting	Y	'B'	(0x42)	4	1
Continuous-response setting	Y	'T'	(0x54)	4	1
Continuous-response start	Y	'S'	(0x53)	0	1
Continuous-response stop	Y	'P'	(0x50)	0	1
Partial arc for Self-calibration setting	Y	'p'	(0x70)	2	1
Self-calibration start	Y	'A'	(0x41)	0	1
Self-calibration status / result	N	'i'	(0x69)	0	1+7
Read error map *	Y	'e'	(0x65)	0	1+1024
Write error map *	Y	'E'	(0x45)	1024	1
Write protection state	N	'w'	(0x77)	0	1
Activate write protection	Y	'W'	(0x57)	0	1
Configuration parameters save	Y	'c'	(0x63)	0	1
Reset to factory settings	Y	'r'	(0x72)	0	1

* For more information, please **contact RLS**.
** First byte is always echo of the command.

Programming execution

Position offset setting

After sending a programming command byte 'Z', which allows the position offset to be set, 4 more data bytes are required. These bytes contain the actual position offset data in encoder count values. When all 4 bytes are received, the new position offset value is set. The following table shows how these 4 bytes are converted into a 32-bit value representing 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 less than zero, the value 0 is set as the new offset. An acceleration error may occur after changing the zero position to a larger value.
After each setting of a new position offset, check or correct the multiturn counter value (if present).

After programming the position offset, the value is stored in a RAM and used immediately. For permanent storage in a non-volatile memory, another programming operation must be performed to save configuration parameters (see chapter **Saving configuration parameters**).

By default (factory setting), the position offset is set to 0.
New position = Absolute position – Offset

Multiturn counter setting

After sending a programming command byte 'M', which allows the multiturn counter to be set, 4 more data bytes are required. These bytes contain the desired multiturn counter. When all 4 bytes are received, the new multiturn counter is set. The following table shows how these 4 bytes are converted into a 32-bit value that represents 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 chapter **Saving configuration parameters**). 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 allows the continuous-response to be set, 4 more data bytes are required. These bytes contain:

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

The following table shows how these 4 bytes are converted into separate configuration parameters.

Byte		Meaning
B1	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
B2	b7 – b0	ASCII command for continuous responding (one from the basic command set, refer to AksIM-2 data sheet **MBD01**.)
B3	b7 – b0	b15 – b8 of the 16-bit period value
B4	b7 – b0	b7 – b0 of the 16-bit period value

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

Default command for continuous responses is ASCII '3' (short response). Any other command from the table "Command (ASCII)" in the **MBD01** document can be programmed. If none of the valid commads are programmed, it will be discarded and reset back to the default (ASCII '3').

After continuous-response programming, the parameters are stored in a RAM. For permanent storage in a non-volatile memory, another programming procedure must be performed to save configuration parameters (see chapter **Saving configuration parameters**). After that, the programmed values become valid after the first subsequent power cycle of the encoder.

The resolution of the period for continuous-response is 1 microsecond. The maximum programmable period can be 65535 microseconds. The shortest possible period depends on the baud rate used and the command selected. If the programmed period is too short and the previous frame has not yet been transmitted, the transmission of the next frame will be delayed by the programmed period. If the user prefers the shortest possible period regardless of the baud rate and command settings, a period setting of 1 microsecond can be set.

See programming example for **Continuous-response setting** which shows the appropriate continuous-response programming sequence.

> 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', the continuous-response feature is activated immediately. The encoder starts data transmission 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 mounting the readhead. It improves the accuracy of the encoder, which depends on the mounting precision. The user must first unlock the programming option with the sequence 0xCD 0xEF 0x89 0xAB, then use the 0x41 command to start the self-calibration procedure. During the procedure, communication via the UART interface is not possible; the encoder does not respond to incoming commands. The first command received during this period is queued and processed at the end of the calibration cycle. Completion of the process is indicated by rapid flashing of LED for 3 seconds. If the self-calibration was successful, LED flashes green, otherwise it flashes red. Afterwards the UART interface is active again.

Before the self-calibration process, the status should be read. The 0x69 command returns 2 bytes, first the echo byte and then the status byte, which contains a two-bit counter and two status bits. The counter is incremented at the end of each self-calibration process. The controller must remember the current self-calibration counter (bits 1:0). After sending the self-calibration command, LED must be observed for completion. If LED is not visible, the readhead should be polled via UART interface until communication with the readhead is re-established or wait 10 seconds, which is the longest possible time for completion. The self-calibration status value should then be read again. When the self-calibration counter has increased by 1 (compared to the previously read value), the self-calibration function has been completed. If the self-calibration was successful, both status bits (b3, b2) are zero. Error bits indicate the success or the reasons for the error.

Speed and direction of rotation during the self-calibration are not important and may be inconsistent. The only requirement is that the shaft makes at least one complete revolution within 10 seconds of sending the command.

If the mechanics do not allow 360° rotation, the length of the calibration arc can be reduced to at least 180° with the 0x70 command. This command must be followed by 2 bytes containing the desired arc length (in degrees). The new arc length must be set before the self-calibration function is executed. The performance of the self-calibration procedure is reduced if the arc length is reduced.

Self-calibration status
Command 'i' returns status of self-calibration. Returned data consists of:

Address	Type	Range	Units	Meaning / usage
B1	U8		bit	Status - see table below
B2..3	U16	0 – 500	µm	Ring eccentricity shift from rotation axis centre
B4..5	U16	0 – 360	degrees	Ring eccentricity angle (phase)
B6..7	S16	-500 – 500	µm	Readhead radial shift (positive = outside)

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

Self-calibration status byte values
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

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

Reseting encoder to factory defaults 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.

After locking the write access, the encoder cannot be reset to the factory defaults.

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.

	Value			
Byte in sequence	**DEC**	**HEX**	**ASCII**	**Purpose**
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**.

Continuous-response setting

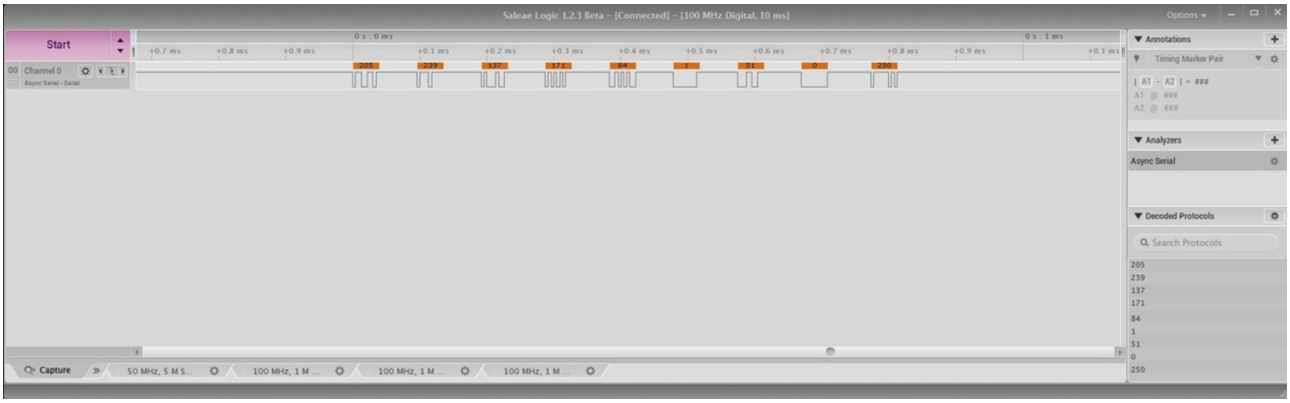
Continuous-response configuration parameters:

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

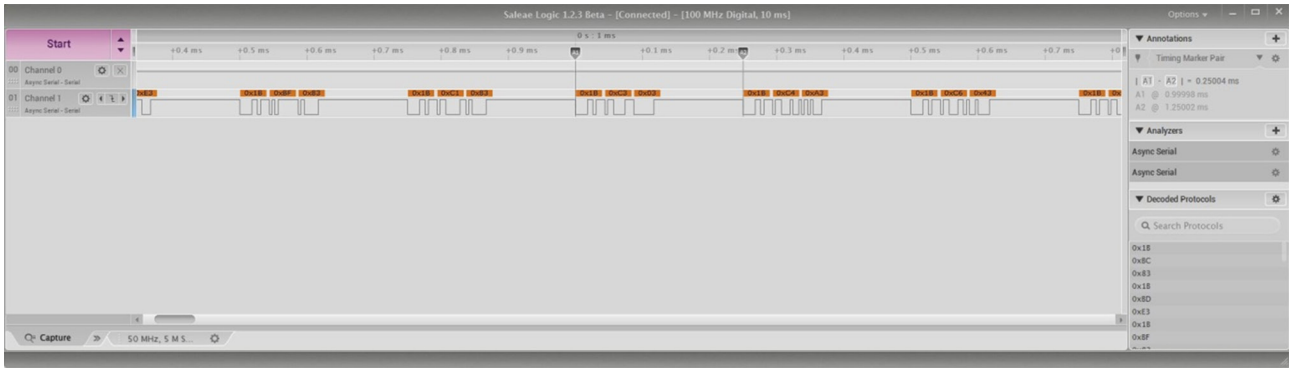
Bytes listed in table below should be sent to the encoder in exact order.

	Value			
Byte in sequence	**DEC**	**HEX**	**ASCII**	**Purpose**
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	/	

Screen capture 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 on the following screen capture.



Saving configuration parameters in a non-volatile memory

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

	Value			
Byte in sequence	**DEC**	**HEX**	**ASCII**	**Purpose**
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.

	Value			
Byte in sequence	**DEC**	**HEX**	**ASCII**	**Purpose**
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

Global support

Visit our **website** to contact your nearest sales representative.

A RENISHAW associate company