Orbis™
Using advanced functions over SPI

Programming the Orbis encoder via the SPI interface allows you to:

- Set the position offset (zero position),
- set multiturn counter value (if available),
- perform the self-calibration function,
- store the current Orbis configuration parameters in a non-volatile memory,
- reset the Orbis configuration parameters to the factory settings.

Related products

Orbis true absolute rotary encoder

Programming is done by sending separate bytes to the encoder.

Each byte represents the first MOSI byte in the SPI frame. Each byte must be in a separate data frame (an NCS signal cycle transmits only one byte via MOSI). The command sequences for each command.

Each of the programming options must be started with specific unlocking sequence. This prevents unintended changes to the encoder configuration parameters. The sequences of the individual programming options are described below.

> The delay between the individual bytes sent during programming must be at least 1 ms.

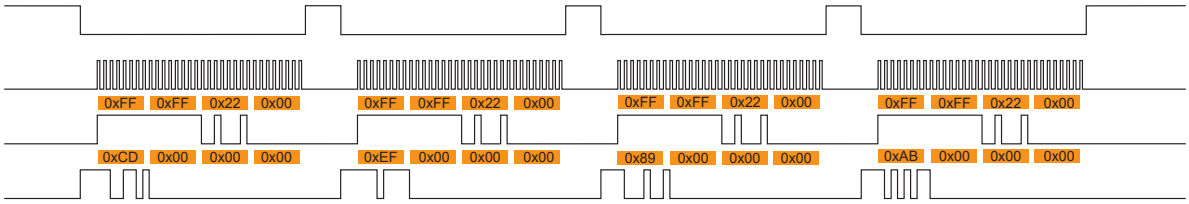
Unlocking sequence

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

Byte in sequence	Programming command byte	
	HEX	ASCII
B1	0xCD	/
B2	0xEF	/
B3	0x89	/
B4	0xAB	/

Whenever the sequence is interrupted by a wrong byte, the unlock stage is reset and must be started from the beginning.

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



In an SPI transaction, only one byte (command) must be sent to the encoder. In the above example, 3 additional bytes are transmitted because the application reads the encoder's position data at the same time as it programmes the encoder. The SPI transaction can be terminated at any time with the NCS signal high. If the encoder has been 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 not one of these, the encoder is "locked" again.

Programming command bytes

The programming command bytes vary depending on the desired programming function. The valid programming bytes are listed in the following table. Some of them perform the desired task immediately (e.g., resetting factory defaults), while others require additional data bytes (e.g., setting the position offset). Each task is are described in the following sections. When programming is complete, the programming option returns to the 'locked' phase.

Programming feature	Programming command byte		Additional data bytes required
	HEX	ASCII	
Setting zero offset	0x5A	'Z'	4
Multiturn counter setting	0x4D	'M'	4
Configuration parameters save	0x63	'c'	0
Reset to factory defaults	0x72	'r'	0
Triggering self-calibration	0x41	'A'	0

Setting zero offset

First transfer zero position offset into RAM. Second, send the command to store this value into non-volatile memory. Pos_offset parameter has max value of 16383.

Byte in sequence	Programming command byte		Purpose
	HEX	ASCII	
B1	0xCD	/	Unlock sequence
B2	0xEF	/	
B3	0x89	/	
B4	0xAB	/	
B5	0x5A	'Z'	Setting zero offset command
B6	0x00	/	Pos_offset
B7	0x00	/	
B8	HH (High byte)		
B9	LL (Low byte)		

Programming example: Encoder position offset setting to 5144 (0x1418) counts

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

Byte in sequence	Programming command byte		Purpose
	HEX	ASCII	
B1	0xCD	/	Unlock sequence
B2	0xEF	/	
B3	0x89	/	
B4	0xAB	/	
B5	0x5A	'Z'	Setting zero offset command
B6	0x00	/	Encoder position shifted for 5144 counts
B7	0x00	/	
B8	0x14	/	
B9	0x18	/	

To save the offset position to non-volatile memory, the encoder configuration parameters must be saved according to the section **Saving configuration parameters to non-volatile memory**.

Multiturn counter

Multiturn counter value can be between 0 and 65535.
Saving to non-volatile memory is performed on power-off.

Byte in sequence	Programming command byte		Purpose
	HEX	ASCII	
B1	0xCD	/	Unlock sequence
B2	0xEF	/	
B3	0x89	/	
B4	0xAB	/	
B5	0x4D	'M'	Multiturn counter setting command
B6	0x00	/	New multiturn counter value
B7	0x00	/	
B8	HH (High byte)		
B9	LL (Low byte)		

Saving configuration parameters to non-volatile memory

The bytes listed in the following table must be sent to the encoder in exact sequence.
After this sequence has been completed, the encoder is non-responsive for a few milliseconds.

Byte in sequence	Programming command byte		Purpose
	HEX	ASCII	
B1	0xCD	/	Unlock sequence
B2	0xEF	/	
B3	0x89	/	
B4	0xAB	/	
B5	0x63	'c'	Save configuration parameters command

Reset to factory defaults

The command resets the zero position offset to 0 and the self-calibration parameters to factory defaults.
The bytes listed in the following table must be sent to the encoder in exact sequence.
After this sequence is completed, the encoder is non-responsive for a few milliseconds.

Byte in sequence	Programming command byte		Purpose
	HEX	ASCII	
B1	0xCD	/	Unlock sequence
B2	0xEF	/	
B3	0x89	/	
B4	0xAB	/	
B5	0x72	'r'	Reset to factory defaults command

Triggering self-calibration

The self-calibration function eliminates the error caused by eccentricity, which accounts for much of the accuracy of the encoder and is caused by the eccentric mounting of the ring. This function eliminates the error of one sine wave per revolution. If the multiturn counter is used in the encoder and the speed is higher than ±300 RPM, it may have an incorrect value after self-calibration. In this case the multiturn error flag is set.

Requirements:

- Free mechanical rotation for one full revolution (360°)
- Good signal over the entire calibration angle
- Maximum time available is 10 seconds
- Direction is not important
- Maximum speed during self-calibration is 600 RPM
- Self-calibration must be started when there is no error present (green LED)
- LED should be visible to check self-calibration status

On encoders produced after October 2024, use function “i” to read results of the Self-calibration function (for more information refer to the document BRD01 available at **RLS Media Center**).

Byte in sequence	Programming command byte		Purpose
	HEX	ASCII	
B1	0xCD	/	Unlock sequence
B2	0xEF	/	
B3	0x89	/	
B4	0xAB	/	
B5	0x41	'A'	Triggering self-calibration command

Saving to non-volatile memory occurs automatically when calibration is sucessfull and LED flashes green rapidly. LED should be visible to check the status of self-calibration. Alternatively, on encoder produced after October 2024, use function “i” to read the status over the SPI interface.
Rotation speed and direction during self-calibration may be inconsistent. The only requirement is that the shaft makes at least one complete revolution within 10 seconds of sending the command.
On encoders produced before October 2024, there is no command to query the status whether the self-calibration has been completed successfully or not. Refer to the LED. Fast flashing green means that the process has been completed successfully.

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.

Document issues

Issue	Date	Page	Description
2	3. 1. 2022	-	New design
3	10. 10. 2024	5	Added function to read self-calibration status