

marXperts

marquadb

User Manual



marquadb

Quadrature Decoder for Incremental Encoders

User Manual

Version 1.1

June 2024

Written by Dr. Claudio Klein

Copyright 2024 marXperts GmbH

All rights reserved.

Should you have questions concerning the system or its use, please contact us via phone or email.

marXperts GmbH
Werkstr. 3
22844 Norderstedt / Germany
Tel.: +49 (40) 529 884 - 0
Fax: +49 (40) 529 884 - 20
info@marxperts.com
www.marxperts.com

Table of Contents

1. How to use this manual.....	4
2. Declarations.....	4
2.1 Europe.....	4
2.2 North America.....	4
2.3 Waste Electrical and Electronic Directive.....	4
3. Function.....	5
4. Hardware components.....	6
5. Inputs.....	7
6. Outputs.....	8
7. Communication.....	9
8. Commands.....	9

1. How to use this manual

Before you start to operate the **marquadb** box please read the User Manual and the Technical Documentation included in the documentation package carefully.

2. Declarations

2.1 Europe

The instrument complies with the EMC Directives 2014/30/EU, the Low Voltage Directive 2014/35/EU as well as the RoHS directive 3032/2012.

Compliance was demonstrated by conformance to the following specifications listed in the Official Journal of the European Communities:

- EN61326-1: 2018 (Electrical Safety)
- EN301 489-17: V3.1.1: 2017 (EMC for radio equipment and services)
- EN301 48901 V2.2.3: 2019 (EMC for radio equipment and services)
- EN300 328 V2.2.2: 2019 (Wideband transmission system in the 2.4 GHz band)
- EN6300: 2018 (RoHS)

2.2 North America

The instrument has been found to comply with the specifications for a class B digital device pursuant to Part 15 of the FCC rules and meets all requirements of the Canadian Interference Causing Equipment Standard ICES-003 for digital devices.

2.3 Waste Electrical and Electronic Directive

End-users may return the instruments to Marxperts GmbH for disposal without being charged for disposal.

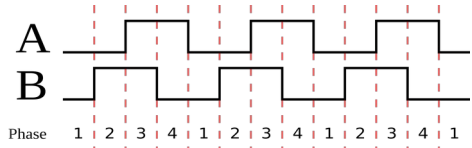
This offer is valid only under the following conditions:

- the unit has been sold to a company or institute within the EU
- the unit is currently owned by a company or institute within the EU
- the unit is complete and not contaminated

The instrument does not contain batteries. If not returned to the manufacturer, it is the owner's responsibility to follow the local rules for disposing electronical equipment.

3. Function

The **marquadb** box is a microcontroller that counts signals (“A quad B”) from incremental encoders. Incremental encoders are linear or rotary electromechanical devices that have 2 output signals, A und B, which issue pulses when the device is moved. Incremental encoders report position increments nearly instantaneously, which allows them to monitor the movements of high speed mechanisms in near real-time. While either A and B signal would show the progress of a movement, the phase shift between A and B allows to determine the direction of movement. In the figure show above, signal B is leading A, so the direction of movement is negative.



The **marquadb** box counts pulses from up to 3 sources independently, but not simultaneously. The counting works in either direction. The instrument will report the direction of the movement and the time elapsed to count pulses from which the speed of the movement can be derived. However, the actual function of the **marquadb** box is to trigger an action after reaching a given count of pulses. The box feeds a signal (TTL like) into one of the coaxial outputs. The level of the coaxial output is either HIGH or LOW and is as follows:

- LOW if the box is not counting
- HIGH if the box is counting
- switch to LOW if the number of pulses has been counted
- switch back to HIGH immediately or after a configurable delay
- LOW if the box stops counting





By default, LOW means 0.0 Volt and HIGH means 3.3 Volt. It is possible to reverse the levels if desired. The **marquadb** box is not a real-time instrument. The time to switch between LOW and HIGH is in the order of magnitude of 5 microseconds but it is possible to increase the duration of the output signal.

A typical use of the instrument is to provide trigger signals to any sort of hardware as a motor coupled to an encoder is moving. Trigger signals will be created after counting a given number of pulses. The instrument does not need to know about physical properties of the motor. It just counts A and B pulses of the incremental encoder.

Example: a motor giving 1000 encoder pulses per mm of movement should trigger a camera that shoots a photo after every movement of 1 mm. This requires a camera capable of receiving TTL-type trigger signals.

4. Hardware components

The device ships with the following components:

#	Description	
1	marquadb controller box	
2	USB-A to USB-B cable, 5m <ul style="list-style-type: none"> Connects the box with the PC 	
3	Optional: A/B signal cable as used for Attocube piezo motor controllers AMC100/NUM <ul style="list-style-type: none"> D-Sub9 at box, 37-pins at Attocube controller Connects the box with up to 3 motors 	
4	Optional: coaxial cable <ul style="list-style-type: none"> BNC at box, LEMO 00 (NIM/CAMAC) at other end Connects the box with a camera 	

5. Inputs



USB

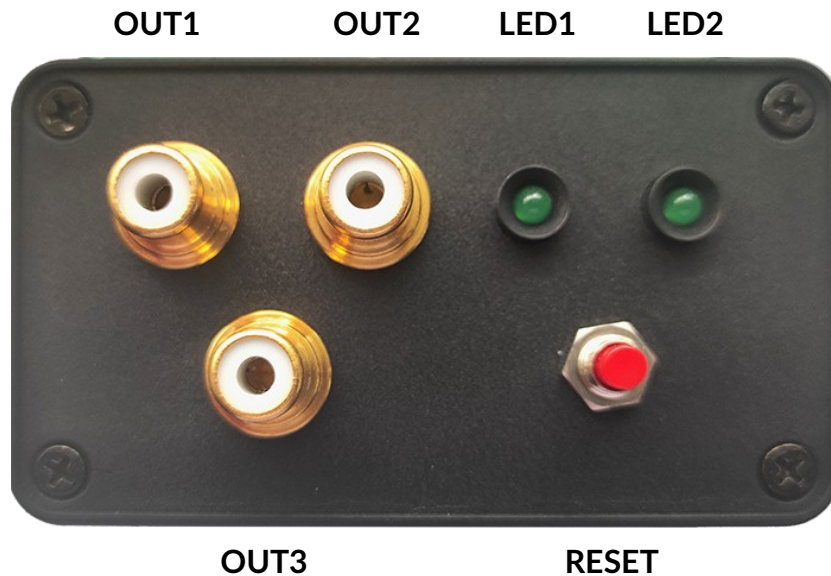
D-Sub9 (encoders)

The **marquadb** box features a USB-B connector at the back side as well as a D-Sub9 connector. The box has to be connected to a PC using the USB cable.

The A, B and ground lines from up to 3 incremental encoders are fed into the controller via a 9-pin connector. The pin assignments are shown in the table below.

Pin	Assignment	
1	Encoder 1: signal A	
2	Encoder 1: signal B	
3	Encoder 1: GND	
4	Encoder 2: signal A	
5	Encoder 2: signal B	
6	Encoder 2: GND	
7	Encoder 3: signal A	
8	Encoder 3: signal B	
9	Encoder 3: GND	

6. Outputs



The output signals are supplied to coaxial connectors that must connect the box (brass colored connector) with a target device, e.g. a camera. When the controller is idle, the output on the coaxial output is LOW (0.0 Volt). When the controller starts to count, the output signal is set HIGH (3.3 Volt). After reaching a given number of counts, the output signal drops to LOW. This signal can be used to trigger the read-out of a camera or some action in some other type of hardware. This operation will be repeated for a given number of times. The duration of the signal switching HIGH-LOW-HIGH is approx. 5 microseconds. It is possible to invert the signals (HIGH=0 V, LOW=3.3 V).

When the controller is counting signals, LED1 will be lit. Otherwise, when the controller is idle, LED1 is off. LED2 will operate similarly but will turn on only if the output signal is HIGH and otherwise be turned off. Since the switching time between HIGH and LOW is very short, both LEDs will normally appear to look the same. The settable delay time must be at least 100 milliseconds to see the difference.

The RESET button will reboot the controller which is an alternative to unplugging the USB cable. When booting up, LED1 flicker 5 times while LED2 is constantly lit. After the initialization sequence, both LED's will be turned off.

7. Communication

The **marquadb** controller must be controlled from the data collection PC via a USB connection (USB-B to USB-A). The controller provides a conventional serial interface that understands plain ASCII commands and that sends output to the serial interface as plain text strings.

It is therefore possible to operate the box "manually" or via an API. You may use a variety of programs that use serial connections, e.g. *PuTTY* on Windows or *minicom* on Linux. Please use the following serial connection settings:

- baudrate: 115200
- parity: None
- stopbits: 1
- bytesize: 8 bits
- flow-control: none

On Linux, you could thus a simple command like the following, making sure, that the device file has the proper permissions for the user to read from it and write to it:

```
minicom -D /dev/ttyACM0 -b 115200
```

On Linux OS, /dev/ttyACM0 would be a typical device name. On Windows, it would be rather **COMn** where **n** is a single digit.

Note: when implementing a communication API using the commands below, make sure to also read the text strings generated by the controller, even if you are not making use of them.

8. Commands

The controller understands the following commands (strings in brackets are optional).

- **counts N lines L channel C** - enter counting mode for N counts with L encoder lines (pulses) each on channel C (default: N=0, L=1000, C=1)
 - **N L [C]** - as above but without keyword "counts" and "lines" and with the option to supply channel 1 to 3
 - **init [T [L]]** - initialize with T lines as tolerance and L lines to start (default: T=1, L=1000)
 - **chan[ne] C** - count signals from channel C (1 to 3, default: 3)
 - **help** - shows usage
 - **set** - shows current values of settable parameters
 - **show** - shows progress of ongoing counting including time elapsed
 - **high** - sets the default signal level to HIGH (3.3 V)
-

- **low** – sets the default signal level to LOW (0 V)
- **led1|2 on|off** – turn LED1|2 on or off
- **out1|2|3 on|off** – turn OUT1|2|3 on (HIGH) or off (LOW)
- **tol[erance] T** - tolerance for counted signals for reaching target (default: T=1)
- **usec U** – time in microseconds to switch back the output level from LOW to HIGH after a count event (default: U = 0)
- **end | abort | stop** - end ongoing counting before reaching target
- **verbose [false>true]** - toggles verbosity. Use argument True or False

To start counting N events, it is sufficient to just enter N. After issuing the command, the counting starts and the output signal is set to HIGH (3.3 V). The parameter L is the number of lines (pulses) to count before generating a trigger signal on the corresponding output OUT1, OUT2 or OUT3. This process is repeated for N cycles.

The duration of the output signal, ie. the switch HIGH-LOW-HIGH, is governed by the CPU speed of the controller and is about 5 microseconds. The duration can be altered using command „usec U“ where U is the duration of the signal in microseconds and defaults to 0. If all N counts are completed, the output is set to LOW and the controller returns to the idle state.

While counting, LED1 and LED2 are turned on. If the counting mode is active, all further commands to count lines are ignored. It is not possible to count lines simultaneously on more than 1 channel.

Example:

To count 4 times 250 lines on channel 3, issue command „4 250 3“. You will get some feedback similar to:

```

Channel 3: counting 4 @ 250 lines +/- 1
#   1 @ 249      in 373.9 ms
#   2 @ 499      in 373.9 ms
#   3 @ 749      in 373.9 ms
#   4 @ 999      in 373.9 ms
Channel 3: counted 4 @ 250 lines
Total # lines: 999
Total time:      1495.6 ms
Time/count:      373.9 ms
    
```

As can be seen, the instrument returns the time elapsed and the total no. of counted lines. The total number of lines will be either positive or negative, indicating the direction of movement. The number of pulses to be counted, however, will always be given as positive number, regardless of the actual direction of movement.