

CANSim 4 – Modbus interface

Author: Jan Brabec

Date: 31. 5. 2021

Version: 1.0

About this document

This document is written to describe Modbus API in CANSim 4 device and how to use it in test automation. Last part of the text is dedicated to demo application in C# .NET where Modbus API is used.

Modbus

Modbus is a data communications protocol originally published by Modicon (now Schneider Electric) in 1979 for use with its programmable logic controllers (PLCs). Modbus has become a de facto standard communication protocol and is now a commonly available means of connecting industrial electronic devices.

Modbus is popular in industrial environments because it is openly published and royalty-free. It was developed for industrial applications, is relatively easy to deploy and maintain compared to other standards, and places few restrictions - other than the datagram (packet) size - on the format of the data to be transmitted.

The development and update of Modbus protocols have been managed by the Modbus Organization since April 2004, when Schneider Electric transferred rights to that organization. The Modbus Organization is an association of users and suppliers of Modbus-compliant devices that advocates for the continued use of the technology.

More information about Modbus protocol can be found on many places on the Internet for example in Wikipedia [1] or on Modbus Organization site [2]

CANSim 4 – Modbus implementation

In CANSim 4 device is Modbus protocol accessible on COM port via USB. It is necessary to have FTDI driver installed before first Modbus connection. Driver installation is described in CANSim 4 device user manual, which can be downloaded on CANSim 4 – Product site [3]. Modbus is implemented in CANSim device from 1.48 FW version, be sure that your device is fully updated. Modbus interface is working together with CANSim 4 Mode 6 – MEB only.

COM port parameters are following:

- Speed: 115200 Baud
- Parity: Odd
- Length: 8 bits

HEAD OFFICE
Digiteq Automotive s.r.o.
Novodvorská 994/138
142 21, Prague 4
Czech Republic

CONTACTS
+420 239 046 750
info@digiteqautomotive.com
www.digiteqautomotive.com

BILLING INFORMATION
Company ID. No. 26 46 60 23
VAT ID No.: CZ 26 46 60 23
registered by the Municipal Court in
Prague, Section C, Insert 84128

MANAGEMENT BOARD
Andreas Hauptvogel
Milan Klaus

BANK ACCOUNT
Commerzbank Aktiengesellschaft
Jugoslávská 1, 120 21 Prague 2
Account No.: 10634787/6200 (EUR)
IBAN: CZ42620000000010634787
SWIFT: COBACZPXXXX

- Stop bits: 1

Only RTU version of protocol is implemented. Modbus address can be any from 0 to 255 (CANSim device reads all of them), but it doesn't answer in broadcast address 0. We suppose to use address 1.

Addresses of registers in following text are the same addresses, which are transmitted inside protocol. Be careful, some implementations of Modbus protocol use register numbers one bigger.

Following parts of protocol are implemented:

04 - Read Input Registers

- 30000 Bitmap of dip switches settings
- 30001 Backlight knob position
- 30002 Steering knob position
- 30003 P-5 knob position
- 30004 P-6 knob position
- 30005 RPM knob position
- 30006 Speed knob position
- 30007 Voltage on input power connector
- 30008 Voltage on pin KL30 in DSUB connector
- 30018 Bitmap of switches position (KL15, ...)
- 30020 Device serial number (BCD coding)
- 30021 Device FW version

Notes:

- Knob positions (30001 – 30006) are 12 bit numbers where 0x0000 is 0% and 0x0FFF is 100% of value on specific knob.
- Voltages (30007 and 30008) are 12 bit RAW values from AD convertor.
- Reading of one register per command is implemented (number of registers must be 1).
- Structure of bitmap with switches position (30018) is following:

| Bit position | Signal |
|--------------|---------------------|
| 0 | Reverse |
| 1 | PDC |
| 2 | Day/Night Backlight |
| 4 | Sleep enable |
| 6 | KL-S |
| 7 | KL-15 |

03 - read holding registers / 06 - write single holding register

- 40000 Device configuration
- 40001 Backlight knob position
- 40002 Steering knob position
- 40003 P-5 knob position
- 40004 P-6 knob position
- 40005 RPM knob position
- 40006 Speed knob position
- 40020 Bitmap of switches position (KL15, ...)

Notes:

- Device configuration has the following structure:

| Bit position | Signal |
|--------------|--|
| 0 | 0 – device control panel enabled 1 – device remote control enabled |
| 1 | Start signalization – this bit is set to 1 after device restart. It is possible to erase this bit by writing 1 on this position. |

- Knob positions 40001 – 40006 have the same structure as 30001 – 30006 in previous command.
- Structure of bitmap with switches position 40020 is the same as 30018 in previous command.
- Command 03 - read holding registers can read only one register (number of registers must be 1)

01 - read coils / 05 - write single coil

- 15 KL-15 (writing 1 on this address sets 1 also on next coil KL-S)
- 16 KL-S (writing 0 on this address sets 0 also on previous coil KL-15)
- 17 PDC & Rev

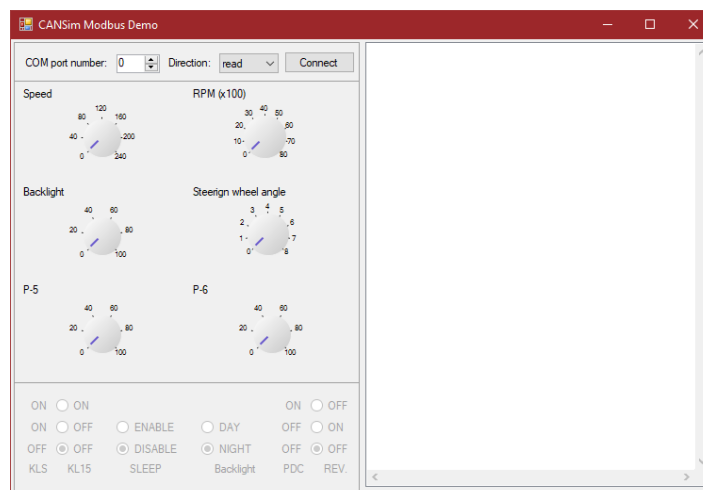
Notes:

- Command 01 - read coils can read only one coil (number of coils must be 1)
- This signals can be read and write together by previous command with address 40020.

CANSim Modbus Demo

Modbus protocol implementation can be found on many places on the Internet and many of them are freeware. We've tested two. First uModbus is Python implementation and it is available in pip (pip install uModbus) [3]. Second EasyModbus is .NET and Java implementation [4]. .NET EasyModbus we use in our Demo application.

CANSim Modbus Demo is application written in C# with Visual Studio 2019. It uses .NET Framework 4.8. Demo uses two 3th party components. First is EasyModbus [4] and second is Knob Control [5] to make knob control in application. Demo application can read position of knobs and switches from CANSim device or can control this values remotely from application. The application is distributed with source code for better understanding.



References

- [1] Wikipedia, „Modbus,“ [Online]. Available: <https://en.wikipedia.org/wiki/Modbus>.
- [2] Modbus Organization, [Online]. Available: <https://modbus.org>.
- [3] Digiteq Automotive s.r.o., „CANSim 4 - Product site,“ [Online]. Available: <https://cansim.digiteqautomotive.com>.
- [4] Advanced Climate Systems, „uModbus 1.0.4,“ [Online]. Available: <https://pypi.org/project/uModbus/>.
- [5] Rossmann Engineering, „EasyModbusTCP/UDP/RTU .NET,“ [Online]. Available: <https://sourceforge.net/projects/easymodbustcp/>.
- [6] F. Lacharme, „Knob Control,“ [Online]. Available: <https://github.com/fabricelacharme/KnobControl>.