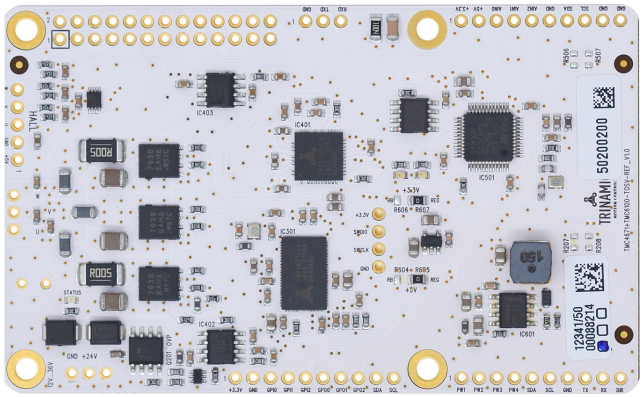
TMC4671+TMC6100-REF-TOSV HW & FW Manual

Hardware Version V1.00 | Document Revision V1.00 • 2020-June-17

TMC4671+TMC6100-REF-TOSV is an open source reference design for ventilators / respirator systems. It is a BDLC servo driver for voltages from +12V to +36V and motor rms phase currents up to 6A. It offers a wide range of connectivity with its SBC connector in Raspberry Pi style, the addon board header for a standard or custom pressure sensor addon board, and general purpose I/O. Communication is via standard TTL UART or RS485 (when using the standard addon board).



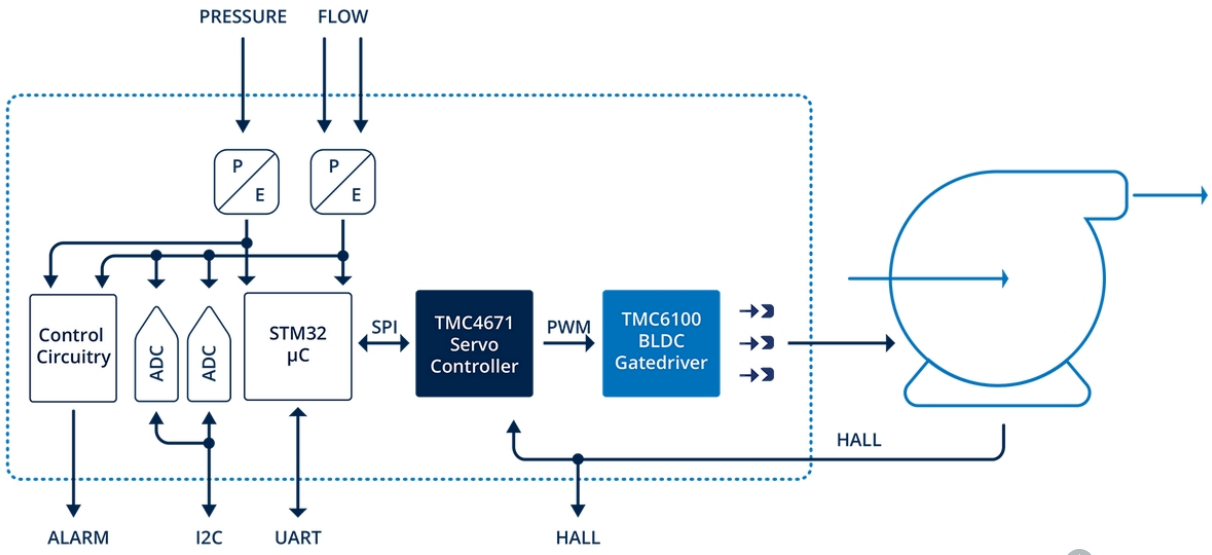
Features

- Single axis BLDC servo driver for up to 6A rms / phase
- Supply Voltage 12V..36V
- Integrated overvoltage protection
- HALL sensor interface
- Onboard temperature sensor
- Digital I/O and analog inputs
- Addon board connector for standard or custom sensor interface board
- Raspberry Pi connector for external SBC

Applications

- Ventilators
- Respirator
- Laboratory automation
- Controlled air-flow applications
- Other BLDC applications

Simplified Block Diagram



Read entire documentation.

TRINAMIC
MOTION CONTROL
Now part of Maxim Integrated

Contents

1 Module Features

TMC4671+TMC6100-REF-TOSV is an open source reference design for ventilators / respirator systems. It is a BDLC servo driver for voltages from +12V to +36V and motor rms phase currents up to 6A. It offers a wide range of connectivity with its SBC connector in Raspberry Pi style, the addon board header for a standard or custom pressure sensor addon board, and general purpose I/O. Communication can be done via standard TTL UART or RS485 (when using the standard addon board).

- Supply Voltage +12V to +36V DC
- Motor phase currents up to 6A RMS
- HALL sensor interface with +5V sensor supply
- Supply rail monitoring with overvoltage protection using external brake resistor
- Digital UART interface for communication and control with an SBC or other host
- SBC I/O connector in Raspberry Pi style to connect a single board computer (SBC) for higher level application control and HMI / display control
- General digital I/O header with GPI, GPO, I2C, and +3.3V
- Addon board header for a custom pressure sensor addon board
- Pressure sensor addon board reference design TOSV-SENSOR-REF
- TMCL-based firmware for configuration and permanent parameter storage

1.1 Open Source

This is an Open Source project! The following data is available as Open Source for download and own use on GitHub:

https://github.com/trinamic/TrinamicOpenSourceVentilator-TOSV.

- Module design, layout, and manufacturing data for the TMC4671+TMC6100-REF-TOSV
- Module design, layout, and manufacturing data for the TOSV-SENSOR-REF
- Complete firmware sources and Eclips CDT based project files for the TMC4671+TMC6100-REF-TOSV
- User Interface project for a Rasbberry Pi plus display for an example ventilator application
- Link to Grabcad for an example of a full housing with display, Raspberry Pi, TMC4671+TMC6100-REF-TOSV, and tube connections.

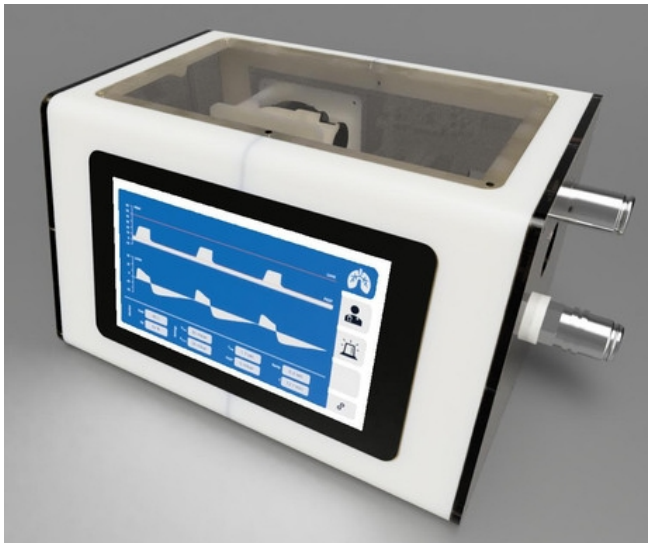


Figure 1: Fully featured ventilator design example with TMC4671+TMC6100-REF-TOSV

2 Order Codes

Order Code	Description	Size (LxW)
TMC4671+TMC6100-TOSV-REF	Open Source Ventilator Reference Design with TMC4671+TMC6100 chipset, MCU, SBC I/O header in Raspberry PI style, I/O header, and pressure sensor addon board header	92x56 (mm)
TOSV-SENSOR-REF	Open source sensor addon board reference design for TMC4671+TMC6100-TOSV-REF with pressure sensors, hysteresis comparators, RS485 transceiver, and IIC ADC	34x56 (mm)
TMC4671+TMC6100-TOSV-REF-KIT	TMC4671+TMC6100-TOSV-REF plus TOSV-SENSOR-REF, Raspberry Pi, display and blower (preliminary)	92x56 (mm)

Table 1: Order codes

3 Mechanical Information

The size of TMC4671+TMC6100-REF-TOSV is approximately 92mm x 56mm.
There are four M3 mounting holes for mounting the TMC4671+TMC6100-REF-TOSV into a case or for mounting it to an SBC or to fixate the pressure sensor addon board (highlighted in green).
The maximum component height without mounted headers is approximately. 3mm.

A 3D file and more information on the mechanical parameters can be derived from the original CAD and manufacturing data available as Open Source on GitHub:

https://github.com/trinamic/TOSV-HardwareReferenceDesign.

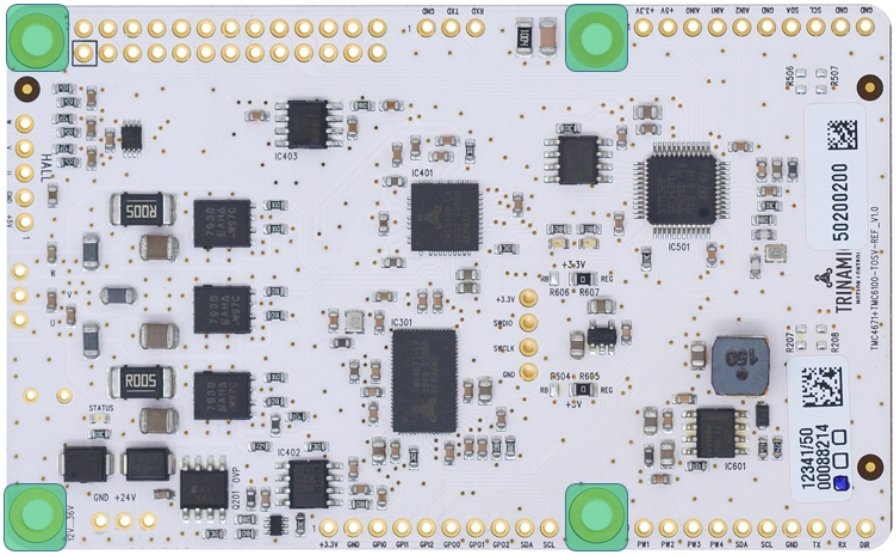


Figure 2: TMC4671+TMC6100-REF-TOSV top view mechanical dimensions

4 Connectors and LEDs

Connector types, pitch, and more information on the I/O signals and pinout can be derived directly from the original/latest CAD and manufacturing data available as Open Source on GitHub:

https://github.com/trinamic/TOSV-HardwareReferenceDesign.

4.1 High Power and Supply Connectors

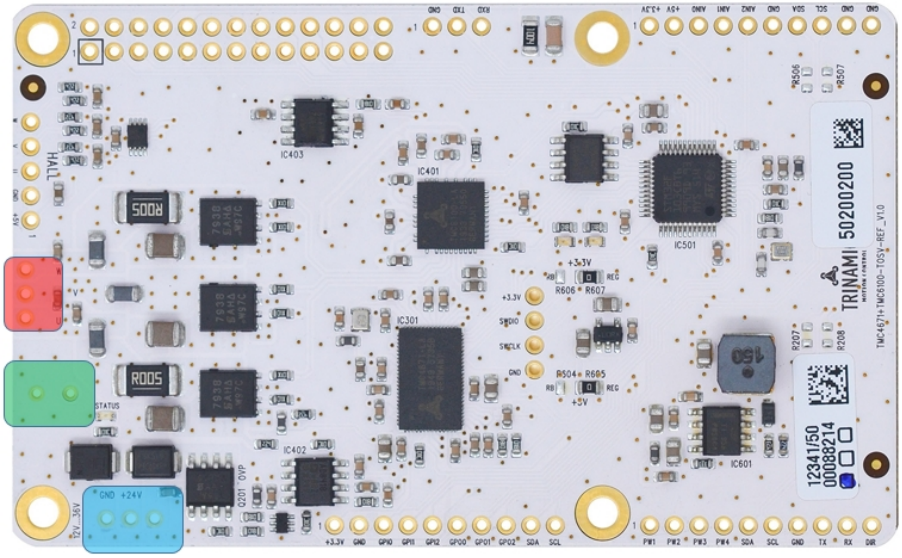


Figure 3: TMC4671+TMC6100-REF-TOSV power connectors

- Red: BLDC/3-phase motor connector
- Blue: Power supply input and overvoltage protection output connector
- Green: Optional placeholder for mounting an electrolyte capacitor to stabilize the supply rail and reduce current ripple

4.2 I/O Connectors and Interfacing

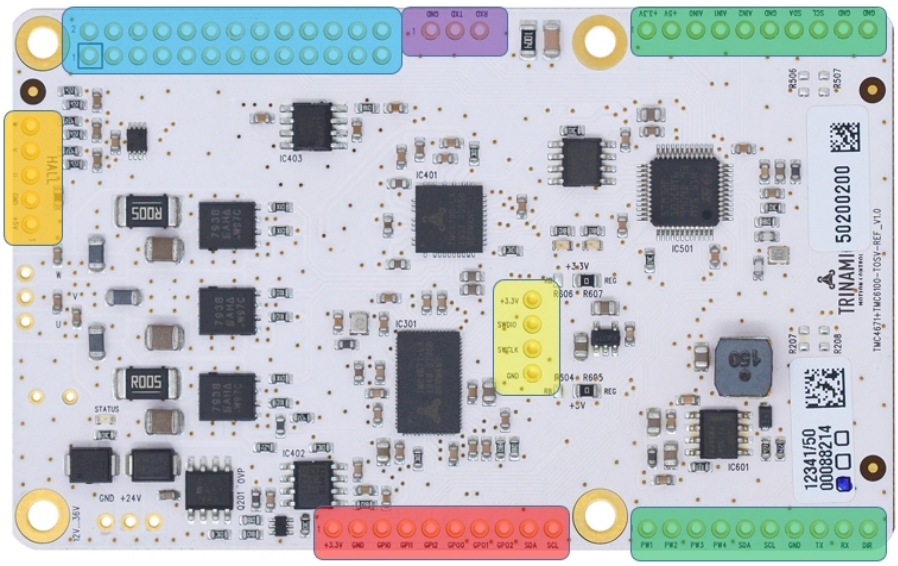


Figure 4: TMC4671+TMC6100-REF-TOSV I/O connectors

- Red: General purpose I/O header with I2C and +3.3V output.
- Blue: SBC connector in Raspberry Pi format and pinout.
- Green: Pressure sensor addon board connector with I2C, RS485 control lines, analog inputs, and digital inputs as well as +5V and 3.3V supply.
- Orange: Digital HALL signal input for motor position feedback with +5V HALL sensor supply.
- Purple: +3.3V TTL UART connector for communication with MCU.
- Yellow: Serial Wire Debug Interface to MCU for programming and debugging

5 TOSV-SENSOR-REF – Pressure Sensor Addon Board

The TOSV-SENSOR-REF is the pressure sensor addon board for the TMC4671+TMC6100-REF-TOSV. It is a small sized PCB that fits onto the addon board headers (see previous section) of the TMC4671+TMC6100-REF-TOSV. Connector types, pitch, and more information on the I/O signals and pinout can be derived directly from the original/latest CAD and manufacturing data of this addon board available as Open Source on GitHub:

https://github.com/trinamic/TOSV-AddonHardwareReferenceDesign.

The TOSV-SENSOR-REF includes:

- 2x analog single ended pressure sensors,
- 1x I2C differential pressure sensor for flow measurement,
- 1x RS485 transceiver to extend the connectivity of the UART interface of TMC4671+TMC6100-REF-TOSV,
- 1x I2C ADC connected to the SBC header connector that monitors the analog pressure sensors' outputs,
- and 4x digital outputs that are connected to 4 hysteresis comparators with adjustable levels (via potentiometers) to allow direct and quick indication of under-/overpressure on the two analog pressure sensors. These 4 signals are also routed to the SBC header connector.

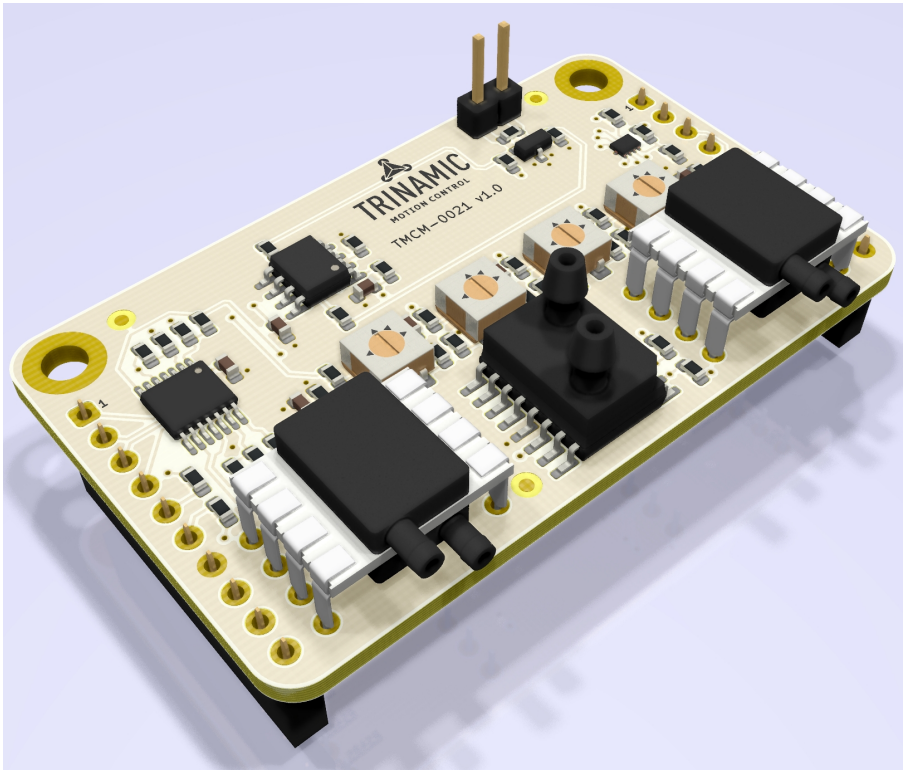


Figure 5: TMC4671+TMC6100-REF-TOSV pressure sensor addon board

6 Functional Description

6.1 TMCL Protocol

6.1.1 Binary Request Format

The TMCL protocol bases on a simple request/reply principle. The request is also called command, as it contains the command to be executed.

Every command has a mnemonic and a binary representation. When commands are sent from a host to a module, the binary format has to be used. Every command consists of a one-byte command field, a one-byte type field, a one-byte motor/bank field and a four-byte value field. So the binary representation of a command always has seven bytes. When a command is to be sent, it has to be enclosed by an address byte at the beginning and a checksum byte at the end. Thus the complete request consists of nine bytes.

The binary command format is as follows:

TMCL Command Format	
Bytes	Meaning
1	Module address
1	Command number
1	Type number
1	Motor or Bank number
4	Value (MSB first!)
1	Checksum

Table 2: TMCL Command Format

The checksum is calculated by adding up all bytes (including the module address byte) using 8-bit addition as shown in this C code example:

```
unsigned char i, Checksum;
unsigned char Command[9];

//Set the Command array to the desired command
Checksum = Command[0];
for(i=1; i<8; i++)
      Checksum+=Command[i];

   Command[8]=Checksum; //insert checksum as last byte of the command
//Now, send it to the module
```

6.1.2 Binary Reply Format

Every time a command has been sent to a module, the module sends a reply. The reply is also 9 byte long and formated is as follows:

TMCL Reply Format	
Bytes	Meaning
1	Reply address
1	Module address
1	Status (e.g. 100 means no error)
1	Command number
4	Value (MSB first!)
1	Checksum

Table 3: TMCL Reply Format

The reply contains a status code. The status code can have one of the following values:

TMCL Status Codes	
Code	Meaning
100	Successfully executed, no error
1	Wrong checksum
2	Invalid command
3	Wrong type
4	Invalid value
5	Configuration EEPROM locked
6	Command not available

Table 4: TMCL Status Codes

6.2 Open Source Firmware Project

6.2.1 Introduction

Trinamic set up an open source project to provide a starting point for software development. The project is hosted on GitHub:

https://github.com/trinamic/TOSV-Firmware.

The firmware's task is to generate the breathing cycles in a TOSV device. Modes and settings can be changed over a serial UART communication via the TMCL protocol. Also the internal states and measured sensor values can be read over UART via the TMCL protocol.

The following sections will give more details on the firmware.

6.2.2 Boot Loader

The software consists of a boot loader and the actual firmware. The TMC4671+TMC6100-REF-TOSV comes shipped with boot loader and firmware preprogrammed. With the help of the boot loader, the firmware

can be updated by the user over the TMCL protocol. Please note that the boot loader is only available as binary image. The image can be found in the git repository under the *bootloader* directory. When the boot loader is active both red and green state LEDs are flashing.

6.2.3 Software Functions

The Firmware supports the TMCL protocol described in section 6.1. The default baudrate is 115200 baud. The TMCL interpreter on the module will interpret received commands, read inputs and write outputs or whatever is necessary according to the specified command. And as soon as this step has been done, the module will send a reply back over the interface to the sender of the command. No new command should be send until the reply of the last command was received.

Changed settings can be stored in a non-volatile memory via the STAP (6.2.4.3) command.

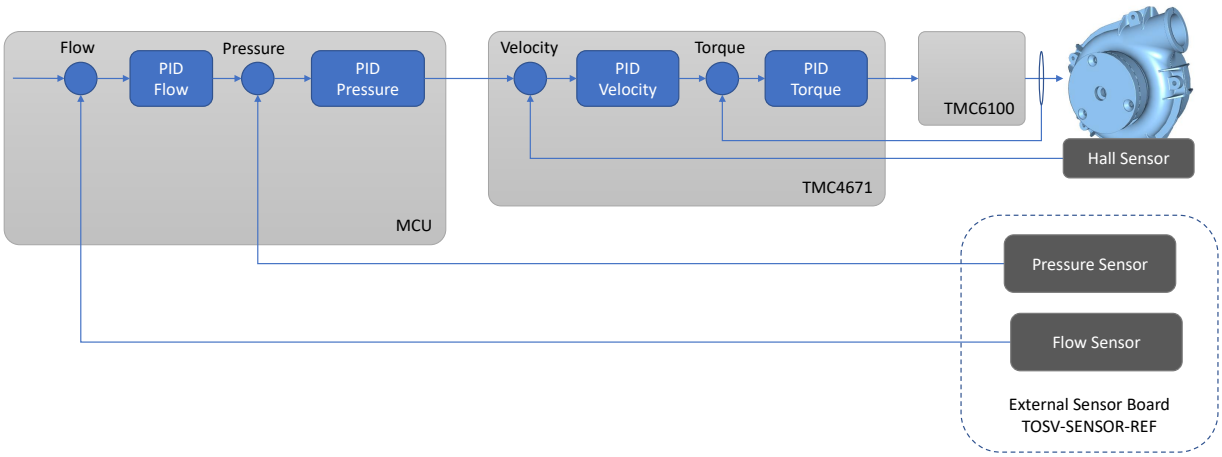


Figure 6: TMC4671+TMC6100-REF-TOSV Firmware Control Loops

Figure 6 shows the ventilation control loops. In pressure based ventilation mode the pressure setpoint is updated depending on inhalation and exhalation phase. In order to start the ventilation control the TOSV enable parameter 100 must be set (see section 6.2.5). Notice that the ventilation control needs a setup where the motor runs a blower and at least the pressure sensor gives feedback of a system that is filled with the blowers air.

The TMC4671+TMC6100-REF-TOSV measures the actual supply voltage for monitoring. If the brake chopper is enabled and the actual supply voltage is higher than the brake chopper voltage limit, the brake chopper output will be switched on. If the actual supply voltage is lower than the (voltage limit - hysteresis), the brake chopper output will be switched off.

6.2.4 Supported TMCL Commands

This sections gives a short overview of the available TMCL commands.

Overview of all TMCL Commands			
Command	Number	Parameter	Description
SAP	5	<type>, <motor number>, <value>	Set axis parameter
GAP	6	<type>, <motor number>	Get axis parameter
STAP	7	<type>, <motor number>, <value>	Store axis parameter
RSAP	8	<type>, <motor number>	Restore axis parameter

Table 5: Overview of all TMCL Commands

The module specific commands are explained in more detail on the following pages. They are listed according to their command number.

6.2.4.1 SAP (Set Axis Parameter)

With this command most of the parameters of the module can be specified. The settings will be stored in SRAM and therefore are volatile. Thus, information will be lost after power off. For a table with parameters and values which can be used together with this command please refer to section 6.2.5.

Internal function: The specified value is written to the axis parameter specified by the type number. **Related commands:** GAP
Mnemonic: SAP <type number>, <axis>, <value>

Binary representation:

Binary Representation			
Instruction	Type	Motor/Bank	Value
5	see chapter 6.2.5	0	<value>

Example Set brake chopper voltage limit to 24.1V. (*Mnemonic:* SAP 96, 0, 500)

Binary Form of SAP 96, 0, 241	
Field	Value
Target address	01_h
Instruction number	05_h
Type	60_h
Motor/Bank	00_h
Value (Byte 3)	00_h
Value (Byte 2)	00_h
Value (Byte 1)	00_h
Value (Byte 0)	$F1_h$
Checksum	57_h

6.2.4.2 GAP (Get Axis Parameter)

Most parameters of the TMC4671+TMC6100-REF-TOSV can be adjusted using e.g. the SAP command. With the GAP parameter they can be read out. For a table with type numbers and values that can be used together with this command please refer to section 6.2.5.

Internal function: The specified value gets copied to the accumulator.
Related commands: SAP
Mnemonic: GAP <type number>, <axis>

Binary representation:

Binary Representation			
Instruction	Type	Motor/Bank	Value
6	see chapter 6.2.5	0	<value>

Example Get the actual brake chopper voltage limit. (*Mnemonic:* GAP 96, 0)

Binary Form of GAP 96, 0	
Field	Value
Target address	01_h
Instruction number	06_h
Type	60_h
Motor/Bank	00_h
Value (Byte 3)	00_h
Value (Byte 2)	00_h
Value (Byte 1)	00_h
Value (Byte 0)	00_h
Checksum	67_h

6.2.4.3 STAP (Store Axis Parameter)

This command is used to store TMCL axis parameters permanently in the EEPROM of the module. This command is mainly needed to store the default configuration of the module. For a table with type numbers and values which can be used together with this command please refer to section 6.2.5.

Internal function: The axis parameter specified by the type and bank number will be stored in the EEPROM.

Related commands: SAP, GAP, RSAP.
Mnemonic: STAP <type number>, <bank>

Binary representation:

Binary Representation			
Instruction	Type	Motor/Bank	Value
7	see chapter 6.2.5	0	0 (don't care)

Example Store axis parameter 96. (*Mnemonic:* STAP 96, 0)

Binary Form of STAP 96, 0	
Field	Value
Target address	01_h
Instruction number	07_h
Type	60_h
Motor/Bank	00_h
Value (Byte 3)	00_h
Value (Byte 2)	00_h
Value (Byte 1)	00_h
Value (Byte 0)	00_h
Checksum	68_h

6.2.4.4 RSAP (Restore Axis Parameter)

With this command the content of an axis parameter can be restored from the EEPROM. By default, all axis parameters are automatically restored after power up. An axis parameter that has been changed before can be reset to the stored value by this instruction. For a table with type numbers and values which can be used together with this command please refer to section 6.2.5.

Internal function: The axis parameter specified by the type and bank number will be restored from the EEPROM.

Related commands: SAP, GAP, STAP
Mnemonic: RSAP <parameter number>, <bank>

Binary representation:

Binary Representation			
Instruction	Type	Motor/Bank	Value
8	see chapter 6.2.5	0	0 (don't care)

Example Restore axis parameter 96. (*Mnemonic:* RSAP 96, 0)

Binary Form of RSAP 96, 0	
Field	Value
Target address	01_h
Instruction number	08_h
Type	60_h
Motor/Bank	00_h
Value (Byte 3)	00_h
Value (Byte 2)	00_h
Value (Byte 1)	00_h
Value (Byte 0)	00_h
Checksum	69_h

6.2.5 Axis Parameters

Axis parameters are accessed with the GAP and SAP command. The TMC4671+TMC6100-REF-TOSV supports the parameter listed in table 6 below.

Like shown in figure 7 the axis parameters are used for accessing:

- the TOSV control and settings (internal parameter),
- the register values of the TMC4761 and TMC6100 chip and
- the measured values of pressure sensor and flow sensor.

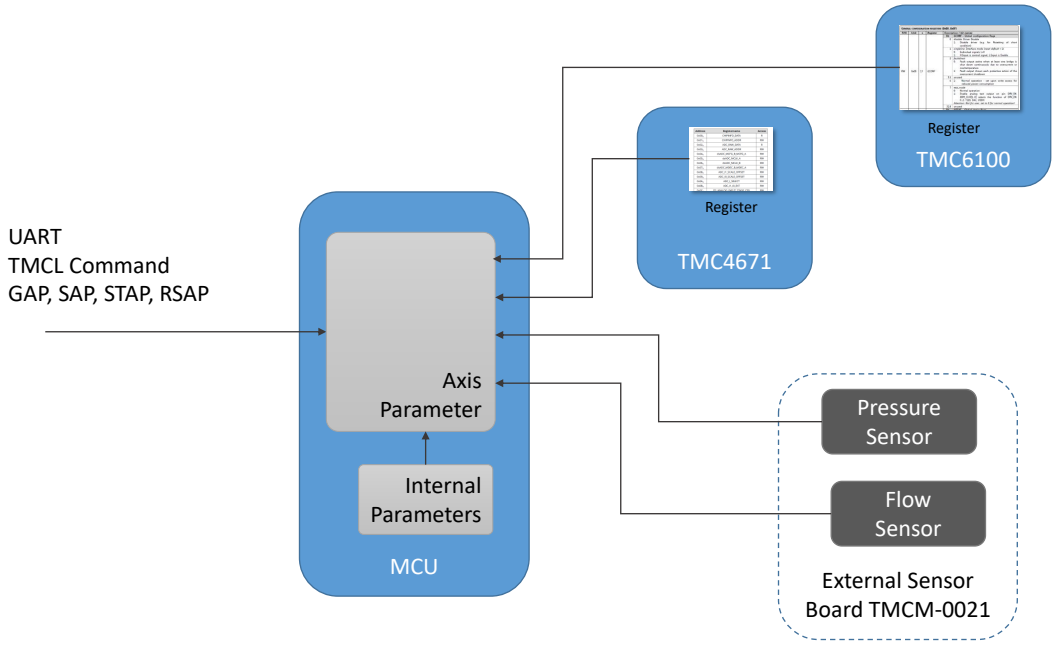


Figure 7: TMC4671+TMC6100-REF-TOSV Axis Parameter Mapping

Axis 0 Parameters of the TMC4671+TMC6100-REF-TOSV Module					
Number	Axis Parameter	Description	Range [Units]	Default	Access
0	status flags	Actual status flags.	0 ... 0	0	R
1	supply voltage	The actual supply voltage.	0 ... 1000 [0.1V]	240	R
2	driver temperature	The actual temperature of the motor driver.	−20 ... 150 [°C]	0	R
3	adc_I0_raw	Raw adc measurement of the phase_A shunt	0 ... 65535	32767	R
4	adc_I1_raw	Raw adc measurement of the phase_B shunt	0 ... 65535	32767	R

Number	Axis Parameter	Description	Range [Units]	Default	Access
5	adc_I0	Calculated current measurement for phase_A shunt and used offset	−32768 ... 32767	0	R
6	adc_I1	Calculated current measurement for phase_B shunt and used offset	−32768 ... 32767	0	R
7	adc_I2	Calculated current of phase_C from phase_A and phase_B measurements	−32768 ... 32767	0	R
8	adc_I0_offset	Manually set/get the dual-shunt phase_A offset.	0 ... 65535	32767	RW
9	adc_I1_offset	Manually set/get the dual-shunt phase_B offset.	0 ... 65535	32767	RW
10	motor pole pairs	Number of motor poles.	1 ... 255	4	RW
11	max current	Max. allowed absolute motor current. *This value can be temporarily exceeded marginal due to the operation of the current regulator.	0 ... 6000 [mA]	3000	RW
12	open loop current	Motor current for controlled commutation. This parameter is used in commutation mode 1.	0 ... 6000 [mA]	1000	RW
13	motor direction		0 ... 1	0	RWE
14	motor type	Select your motor type. 3 - Three phase BLDC	3 ... 3	0	R
15	commutation mode	Select a commutation mode that fits best to your motor's sensors. 0 - disabled 1 - open loop 2 - digital hall	0 ... 2	0	RW
16	motor PWM frequency	Sets the frequency of the motor PWM.	25000 ... 100000 [Hz]	25000	RWE
20	target current	Get desired target current or set target current to activate current regulation mode. (+= turn motor in right direction; -= turn motor in left direction)	−6000 ... 6000 [mA]	0	RW
21	actual current	The actual motor current.	−2147483648 ... 2147483647 [mA]	0	R
24	target velocity	The desired target velocity.	−200000 ... 200000 [rpm]	0	RW
25	ramp velocity	The actual velocity of the velocity ramp used for pressure and velocity mode.	−200000 ... 200000 [rpm]	0	R

Number	Axis Parameter	Description	Range [Units]	Default	Access
26	actual velocity	The actual velocity of the motor.	−2147483648 ...2147483647 [rpm]	0	R
27	max velocity	Max. absolute velocity for velocity and positioning mode.	0 ...200000 [rpm]	4000	RW
28	enable velocity ramp	An activated ramp allows a defined acceleration for velocity and position mode. 0 - Deactivate velocity ramp generator. 1 - Activate velocity ramp generator.	0 ...1	1	RWE
29	acceleration	Acceleration parameter for ROL, ROR velocity ramp.	0 ...100000 [rpm/s]	2000	RW
31	target pressure	The desired target pressure.	0 ...100000 []	0	RW
32	ramp pressure	The actual value of the pressure ramp used for pressure mode.	0 ...100000 []	0	RW
33	actual pressure	The actual pressure.	−2147483648 ...2147483647 []	0	R
34	max pressure	Max. absolute pressure.	0 ...2147483647 []	0	RWE
35	torque P	P parameter for current PI regulator	0 ...32767	0	RWE
36	torque I	I parameter for current PI regulator	0 ...32767	0	RWE
37	velocity P	P parameter for velocity PI regulator	0 ...32767	0	RWE
38	velocity I	I parameter for velocity PI regulator	0 ...32767	0	RWE
39	pressure P	P parameter for pressure PI regulator	0 ...32767	0	RWE
40	pressure I	I parameter for pressure PI regulator	0 ...32767	0	RWE
41	torque PI error sum	Sum of errors of current PI regulator.	−2147483648 ...2147483647	0	R
42	flux PI error sum	Sum of errors of flux PI regulator.	−2147483648 ...2147483647	0	R
43	velocity PI error sum	Sum of errors of velocity PI regulator.	−2147483648 ...2147483647	0	R
44	pressure PI error sum	Sum of errors of pressure PI regulator.	−2147483648 ...2147483647	0	R
45	volume PI error sum	Sum of errors of volume PI regulator.	−2147483648 ...2147483647	0	R
47	open loop commutation angle	Actual controlled angle value.	−32768 ...32767	0	R

Number	Axis Parameter	Description	Range [Units]	Default	Access
48	digital hall commutation angle	Actual digital hall angle value.	−32768 ...32767	0	R
49	max negativ current	Max. allowed negativ motor current.	0 ...6000 [mA]	800	RWE
50	hall polarity	Hall sensor polarity. 0 - standard 1 - inverted	0 ...1	0	RWE
51	hall direction	Hall sensor direction. 0 - standard 1 - inverted	0 ...1	0	RWE
52	hall interpolation	Hall sensor interpolation. 0 - off 1 - on	0 ...1	0	RWE
53	hall phi_e offset	Offset for electrical angle hall_phi_e of hall sensor.	−32768 ...32767	0	RWE
54	hall inputs	Raw hall sensor inputs.	0 ...7	0	R
56	volume P	P parameter for volume PI regulator	0 ...32767	0	RWE
57	volume I	I parameter for volume PI regulator	0 ...32767	0	RWE
95	enable brake chopper	Enable brake chopper functionality. 0 - Deactivate brake chopper. 1 - Activate brake chopper.	0 ...1	0	RWE
96	brake chopper voltage limit	If the brake chopper is enabled and supply voltage exceeds this value, the brake chopper output will be activated.	60 ...300 [0.1V]	260	RWE
97	brake chopper hysteresis	An activated brake chopper will be disabled if the actual supply voltage is lower than (limit voltage-hysteresis).	0 ...50 [0.1V]	5	RWE
99	tosv mode	Switch between pressure controlled mode and voluem controlled mode	0 ...1	0	RW
100	enable tosv	Enable/disable TOSV functionality.	0 ...1	0	RW
101	tosv state	Actual state of TOSV state machine.	0 ...5	0	R
102	tosv timer	Actual timer value of TOSV state machine.	0 ...65535 [ms]	0	R
103	tosv startup time	Startup time.	0 ...65535 [ms]	1000	RWE
104	tosv inhalation rise time	Inhalation rise time.	0 ...65535 [ms]	500	RWE
105	tosv inhalation pause time	Inhalation pause time.	0 ...65535 [ms]	1000	RWE

Number	Axis Parameter	Description	Range [Units]	Default	Access
106	tosv exhalation fall time	Exhalation fall time.	0 ... 65535 [ms]	500	RWE
107	tosv exhalation pause time	Exhalation pause time.	0 ... 65535 [ms]	1500	RWE
108	tosv LIMIT pressure	Target pressure LIMIT.	0 ... 70000	5000	RWE
109	tosv PEEP pressure	PEEP pressure.	0 ... 70000	1500	RWE
110	tosv actual flow	Actual flow sensor value.	−2147483648 ... 2147483647	0	R
111	tosv flow zeroing	Zeroing the flow offset.	0 ... 1	0	W
112	tosv target volume	Target volume.	−2147483648 ... 2147483647	0	RW
113	tosv actual volume	Actual Volume calculated from flow	−2147483648 ... 2147483647	0	R
114	tosv max volume	Target Volume limit	0 ... 70000	150	RWE
120	tosv asb enable	ASB feature - early state machine reset on spontaneous flow	0 ... 1	0	RWE
121	tosv asb flow threshold	ASB threshold	−2147483648 ... 2147483647	2000	RWE
122	tosv asb volume condition	Condtion of the volume to trigger ASB	0 ... 100	20	RWE
130	tosv flow reinit	Re-Initialization of the Flow-Sensor.	0 ... 1	0	W
240	debug value 0	Free used debugging value.	−2147483648 ... 2147483647	0	RW
241	debug value 1	Free used debugging value.	−2147483648 ... 2147483647	0	RW
242	debug value 2	Free used debugging value.	−2147483648 ... 2147483647	0	RW
243	debug value 3	Free used debugging value.	−2147483648 ... 2147483647	0	RW
244	debug value 4	Free used debugging value.	−2147483648 ... 2147483647	0	RW
245	debug value 5	Free used debugging value.	−2147483648 ... 2147483647	0	RW
246	debug value 6	Free used debugging value.	−2147483648 ... 2147483647	0	RW

Number	Axis Parameter	Description	Range [Units]	Default	Access
247	debug value 7	Free used debugging value.	−2147483648 ...2147483647	0	RW
248	debug value 8	Free used debugging value.	−2147483648 ...2147483647	0	RW
249	debug value 9	Free used debugging value.	−2147483648 ...2147483647	0	RW
250	Main loops	Main loops per second.	0 ...4294967295 [1/s]	0	R
251	Velocity loops	Velocity loops per second.	0 ...4294967295 [1/s]	0	R
252	Communication loops	Communication loops per second.	0 ...4294967295 [1/s]	0	R
255	enable driver	Enables the motor driver (enabled by default) 0 - driver disabled 1 - driver enabled	0 ...1	1	RW

Table 6: All TMC4671+TMC6100-REF-TOSV Axis 0 Parameters

The access abbreviations means R for readonly, RW for read/write, and RWE for read/write/storeable in EEPROM for direct use after next reboot.

6.2.6 Using the TMCL-IDE with the Board

During development the TMCL-IDE can be very helpful. All axis parameter can be read, written and live plotted. Especially to find the right motor parameters this helps a lot. In addition, the firmware can be updated.

In order to use the TMCL-IDE to connect to the board, an UART to USB cable is required. The UART side of the cable must be connected to the TTL UART connector of the board, see 4.

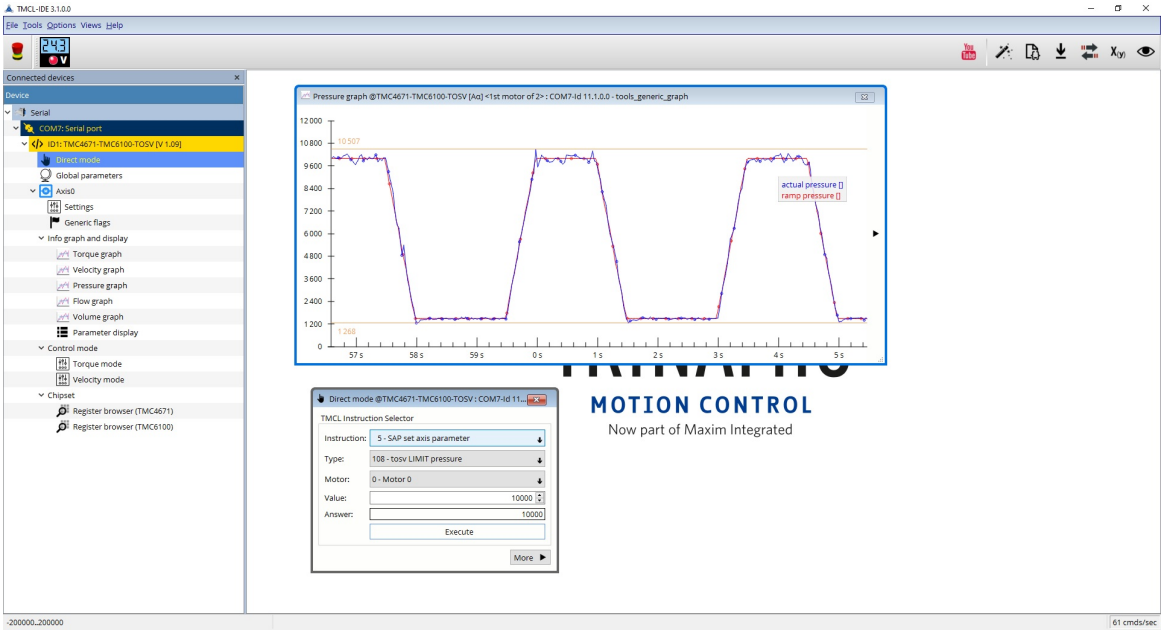


Figure 8: TMC4671+TMC6100-REF-TOSV TMCL-IDE

6.3 Open Source Python based User Interface Project

Trinamic set up an open source project for a possible user interface. The project is also hosted on GitHub:

https://github.com/trinamic/TOSV-UserInterface

The user interface software is supposed to be run on a Raspberry Pi, therefor the UART of the MCU is connected to the boards SBC connecter. Thus, if the TOSV board is stacked onto a Raspberry Pi, the Pi's UART is connected to the TOSV's UART.
The user interface software can also be run on other platforms that run Python. For example one can run the software on a Windows PC and connect to the board via USB-to-UART cable.

6.4 Writing your own Firmware

You are free to write your own firmware for the TMC4671+TMC6100-REF-TOSV . Flashing and debugging of the MCU can be done via the Serial Wire Debug Interface (SWD). Figure 4 shows the location of the SWD connector.

7 Operational Ratings and Characteristics

7.1 Absolute Maximum Ratings

Parameter	Min	Max	Unit
Supply voltage	+12	+38	V
Working temperature	-10	+40	° C

NOTICE

Never Exceed the absolute maximum ratings! Stresses above those listed under "'Absolute Maximum Ratings"' may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

7.2 Electrical Characteristics (Ambient Temperature 25° C)

Parameter	Symbol	Min	Typ	Max	Unit
Supply voltage	VDD	+12		+36	V
Max motor phase current (RMS)	$I_{phase,RMS}$			6	A

Table 8: Electrical Characteristics

7.3 Other Requirements

Specifications	Description or Value
Cooling	Free air or heat sink mounted with isolating gap pad
Working environment	Avoid dust, water, oil mist and corrosive gases, no condensation, no frosting

Table 9: Other Requirements and Characteristics

8 Figures Index

9 Tables Index

10 Supplemental Directives

10.1 Producer Information

10.2 Copyright

10.3 Trademark Designations and Symbols

Trademark designations and symbols used in this documentation indicate that a product or feature is owned and registered as trademark and/or patent either by TRINAMIC or by other manufacturers, whose products are used or referred to in combination with TRINAMIC's products and TRINAMIC's product documentation.

This HW & FW Manual is a non-commercial publication that seeks to provide concise scientific and technical user information to the target user. Thus, trademark designations and symbols are only entered in the Short Spec of this document that introduces the product at a quick glance. The trademark designation /symbol is also entered when the product or feature name occurs for the first time in the document. All trademarks and brand names used are property of their respective owners.

10.4 Target User

The documentation provided here, is for programmers and engineers only, who are equipped with the necessary skills and have been trained to work with this type of product.

The Target User knows how to responsibly make use of this product without causing harm to himself or others, and without causing damage to systems or devices, in which the user incorporates the product.

10.5 Disclaimer: Life Support Systems

TRINAMIC Motion Control GmbH & Co. KG does not authorize or warrant any of its products for use in life support systems, without the specific written consent of TRINAMIC Motion Control GmbH & Co. KG.

Life support systems are equipment intended to support or sustain life, and whose failure to perform, when properly used in accordance with instructions provided, can be reasonably expected to result in personal injury or death.

Information given in this document is believed to be accurate and reliable. However, no responsibility is assumed for the consequences of its use nor for any infringement of patents or other rights of third parties which may result from its use. Specifications are subject to change without notice.

10.6 Disclaimer: Intended Use

The data specified in this user manual is intended solely for the purpose of product description. No representations or warranties, either express or implied, of merchantability, fitness for a particular purpose

or of any other nature are made hereunder with respect to information/specification or the products to which information refers and no guarantee with respect to compliance to the intended use is given.

In particular, this also applies to the stated possible applications or areas of applications of the product. TRINAMIC products are not designed for and must not be used in connection with any applications where the failure of such products would reasonably be expected to result in significant personal injury or death (safety-Critical Applications) without TRINAMIC's specific written consent.

TRINAMIC products are not designed nor intended for use in military or aerospace applications or environments or in automotive applications unless specifically designated for such use by TRINAMIC. TRINAMIC conveys no patent, copyright, mask work right or other trade mark right to this product. TRINAMIC assumes no liability for any patent and/or other trade mark rights of a third party resulting from processing or handling of the product and/or any other use of the product.

10.7 Collateral Documents & Tools

This product documentation is related and/or associated with additional tool kits, firmware and other items, as provided on the product page at: www.trinamic.com.

11 Revision History

11.1 Hardware Revision

Version	Date	Author	Description
1.00	05.04.2020	HH	Launch release.

Table 10: Hardware Revision

11.2 Firmware Revision

Version	Date	Author	Description
1.00	10.04.2020	ED/BP	Launch release.

Table 11: Firmware Revision

11.3 Document Revision

Version	Date	Author	Description
1.00	10.06.2020	SK/ED/BP	Launch release.

Table 12: Document Revision