

AN64: How to run a linear motor with TMC4671

Document Revision V1.0 • 2021-June-25

The **TMC4671** hardware controller performs **Field Oriented Control (FOC)** for two-phase stepper motors and for three-phase permanent magnet synchronous motors (PMSM) and it supports linear motor control.

Why use TMC4671 for linear motor servo control? The TMC4671 provides hardware closed loop torque control, velocity control, and position control even for linear motors decoupling motor control from application.

Contents

1	Linear Motor and TMC4671	2
2	Linear Motor with Incremental Encoder Feedback	2
2.1	Linear Motor Configuration	2
2.2	Motor Type and Pole Pairs	2
2.3	Feedback Selection	3
2.4	Encoder Resolution	3
2.5	Encoder Initialisation	4
2.6	Linear Scaling Factor	5
3	Linear Motor Setup with TMCL-IDE and Wizard	5
3.1	TMC4671 Evaluation Board	5
3.2	Overview	6
3.3	Open TMC4671 Wizard	6
3.4	Wizard Selection	7
3.5	Settings	7
3.6	Open Loop	8
3.7	ADC config	8
3.8	ABN encoder	9
3.9	Encoder Test Drive (Torque Mode)	10
3.10	Save Configuration	11
3.11	Feedback Selection	11
3.12	Linear Scaling Factor	11
3.13	PI Tuning	12
3.14	Position Mode	12
4	Additional Ressources	13
5	Revision History	14



1 Linear Motor and TMC4671

When using the TMC4671 to control a linear motor the servo control modes torque control, velocity control, and position control for three-phase permanent magnet synchronous motors (PMSM) are applicable. To close the loop the TMC4671 provides an ADC engine, as well as multiple position feedback options which can be configured to the linear motor.

2 Linear Motor with Incremental Encoder Feedback

In this architecture a linear incremental encoder (ABN) provides the feedback for commutation/torque as well as velocity and position control.

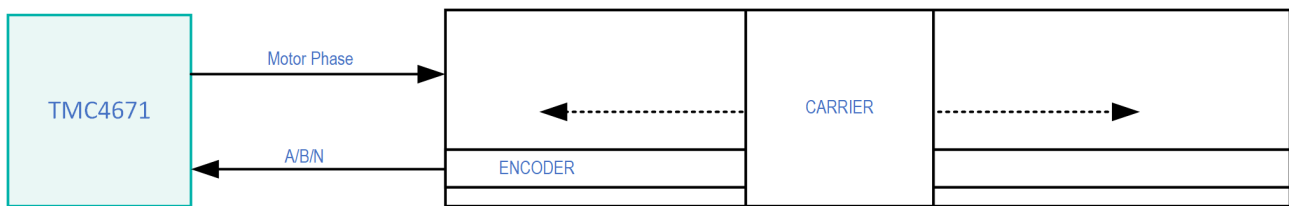


Figure 1: Linear Motor with ABN Encoder and TMC4671

2.1 Linear Motor Configuration

The table below gives an overview on the most important registers to configure the TMC4671 for a linear motor:

Address	Register Name	Function
0x1B _h	MOTOR_TYPE_N_POLE_PAIRS	set motor type and number of pole pairs.
0x50 _h	VELOCITY_SELECTION	selection of velocity signal source
0x51 _h	POSITION_SELECTION	selection of position signal source
0x52 _h	PHI_E_SELECTION	selection of electrical angle used for commutation by FOC
0x25 _h	ABN_DECODER_MODE	ABN encoder mode (direction, ABN pulse polarities and handling)
0x26 _h	ABN_DECODER_PPR	ABN encoder positions per revolution, PPR = $l_{pr} \cdot 4$

In the following a guideline for the configuration is given.

2.2 Motor Type and Pole Pairs

The TMC4671 supports several motor types such as one phase, two phase as well as three-phase motors. Depending on the selected motor in register MOTOR_TYPE_N_POLE_PAIRS (0x1B_h) the PWM output for the gatedrivers are selected accordingly. The pole pairs define the electrical periods per mechanical period for an rotating motor. For a linear motor the pole pair is set set to one. Thus the MOTOR_TYPE_N_POLE_PAIRS (0x1B) will be set as follows for a three-phase linear motor:

Action:



Address	Register Name	Value
0x1B _h	MOTOR_TYPE_N_POLE_PAIRS	N_POLE_PAIRS = 1 (number of pole pairs) MOTOR_TYPE = 3 (three phase BLDC)

Result:

The electrical revolution equals the mechanical revolution matching a linear motor. The FOC is using three-phase control scheme.

2.3 Feedback Selection

The feedback selection registers define the type of the position feedback (e.g. digital hall, analog hall, incremental encoder,...) For a linear motor with an ABN incremental encoder the TMC4671 is configured as follows:

Action:

Address	Register Name	Value
0x50 _h	VELOCITY_SELECTION	3 (phi_e_abn)
0x51 _h	POSITION_SELECTION	3 (phi_e_abn)
0x52 _h	PHI_E_SELECTION	3 (phi_e_abn)

Result:

Primary ABN encoder interface (pins 33, 34, 35) of the TMC4671 is used as feedback for torque, velocity and position mode.

2.4 Encoder Resolution

The encoder feedback selected in the Feedback Selection chapter 2.3 will be used for commutation. For correct commutation it is important to set the encoder feedback resolution (ABN_DECODER_PPR, 0x26) which is defined by pulses per electrical period. For a linear motor the electrical period covers exactly the magnet pole pair distance as depicted in figure 2:



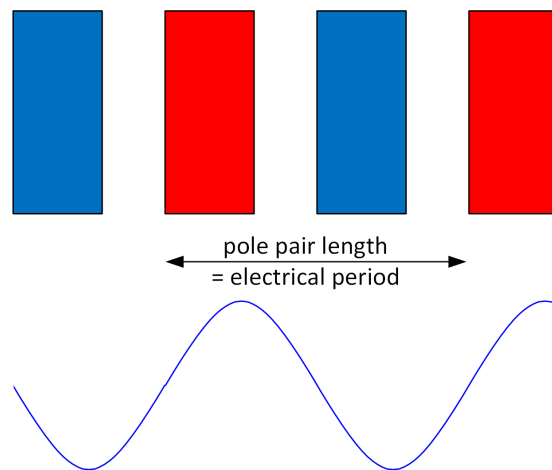


Figure 2: Pole pair and electrical period

The encoder resolution is determined by linear encoder resolution and magnetic pole pair length of the linear motor:

$$\text{ABN_DECODER_PPR} = \frac{\text{pole pair length}}{\text{linear encoder resolution}} \quad (1)$$

Example:

A linear motor with a pole pair length of 3cm and a linear encoder resolution of $1\ \mu\text{m}$ results in $\text{ABN_DECODER_PPR} (0 \times 26) = 30000$.

The TMC4671 supports encoder resolutions up to $2^{24} - 1$. The ABN decoder count is mapped to the mechanical angle PHI_M in the range of $0 \dots 65535$. As mentioned in section 2.2 the mechanical angle PHI_M is equal to the electrical angle due to the polepair count = 1.

2.5 Encoder Initialisation

A linear incremental encoder does not provide information on the absolute motor position within an electrical period. The purpose of the encoder initialization is to obtain a known motor position within its electrical period. For a linear motor the electrical period equals the pole pair length. To initialise an encoder the wizard in the TMCL-IDE provides several inbuilt functions.

For the linear motor with only incremental feedback the encoder initialization mode 0 is suitable. Principle is to align the motor into a known position by applying a defined magnetic field. For this initialization method the linear motor should be able to move one pole pair. The resulting motor position is then well defined as it is the beginning of the electrical period thus the commutation angle can be set to zero as shown in figure 3:

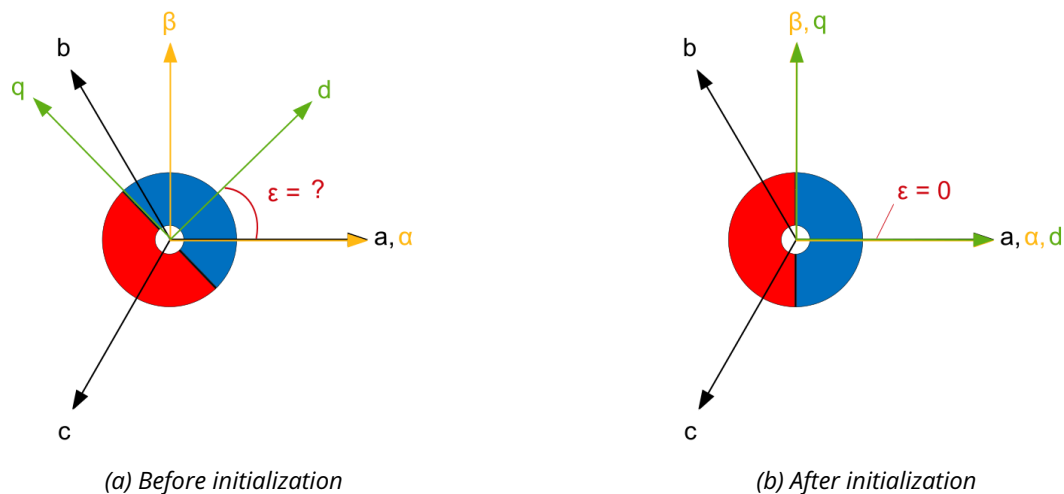


Figure 3: Encoder initialization mode 0 principle

The encoder function is implemented in the TMCL-IDE and in the [TMC4671 API](#) function `tmc4671_doEncoderInitializationMode0()`.

2.6 Linear Scaling Factor

To map from internal units to linear units (e.g. one micrometer) a linear scaling factor is introduced which translates between electrical period and linear position.

For a linear motor (as depicted in figure 1) the linear scaling factor is determined by the pole pair length in μm :

$$\text{linear scaling factor} = \text{pole pair length}[\mu\text{m}] \quad (2)$$

Linear scaling unit is $\frac{\mu\text{m}}{\text{mechanical revolution}}$ with the mechanical revolution being equal to one.

Example: A magnetic pole pair length of 3cm results in a linear scaling factor of $30000 \frac{\mu\text{m}}{\text{mechanical revolution}}$. With this scaling factor one position step will be equal to one micrometer.

The linear scaling factor is implemented in the [TMC-API](#) and available in the TMCL-IDE. It is not part of the TMC4671 registers.

3 Linear Motor Setup with TMCL-IDE and Wizard

3.1 TMC4671 Evaluation Board

The Trinamic evaluation boards are being used as an exemplary hardware platform for this application note. Following parts are used:

- [TMC4671-EVAL](#)
- [TMC6100-EVAL \(powerstage\)](#)
- [TMCL-IDE](#)



- Linear motor with an incremental linear encoder (ABN)
 - pole pair length = 3cm
 - linear encoder resolution = $1\mu\text{m}$

3.2 Overview

Configuration by TMCL-IDE wizard

1. Motor and pole pairs
2. Motor current ADC
3. Encoder (resolution and initialization)

Configuration with TMCL-IDE tools

1. Feedback Selection
2. Linear Scaling Factor
3. PI tuning
4. Position mode

3.3 Open TMC4671 Wizard

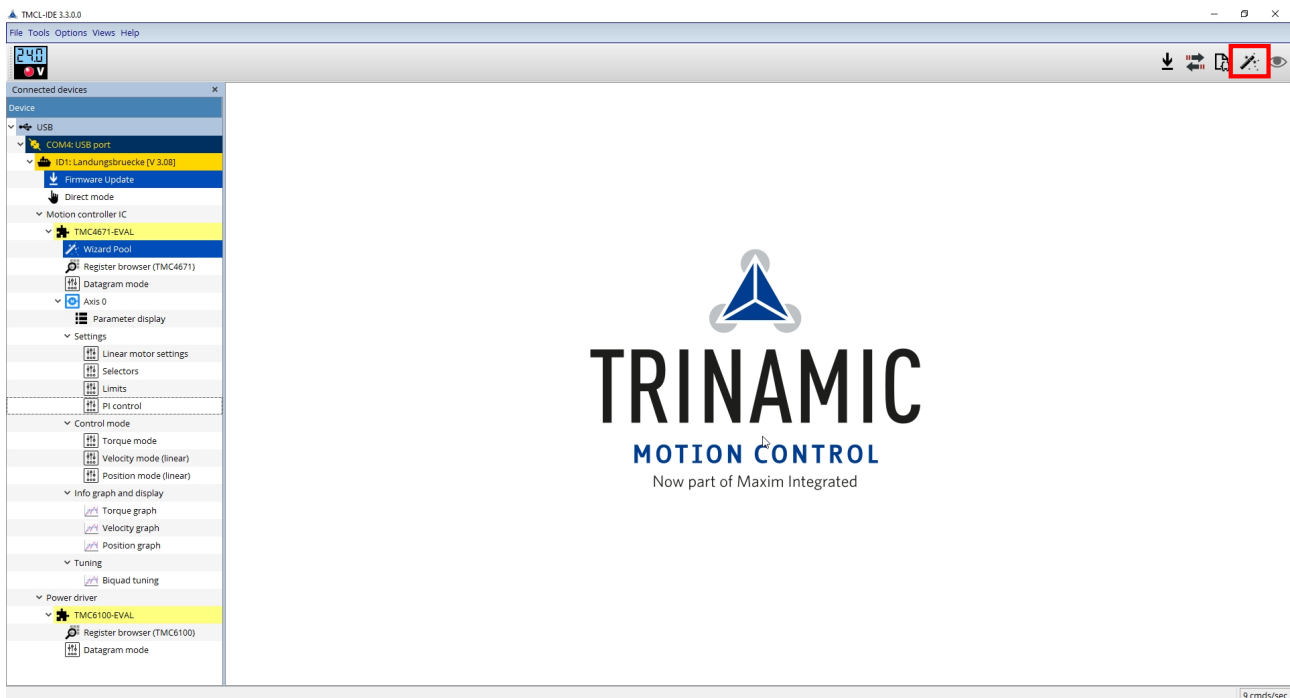


Figure 4: Selector settings



3.4 Wizard Selection

Introduction

Select your preferred wizard pages

The TMC467X family provides high performance hardware FOC controllers. For starting up the system several settings have to be made. The Weasel Wizard guides you through this configuration process and helps you to achieve best controller performance and fast commissioning. The wizard is structured in seven main steps which can be used dependent of your hardware setup. Please select the wizard pages you like to use:

Enable page	Description
<input checked="" type="checkbox"/> Introduction	This page
<input checked="" type="checkbox"/> General Settings	Bring up the PWM
<input checked="" type="checkbox"/> Open Loop	Let the Motor run for the first time with open loop control
<input checked="" type="checkbox"/> ADC Selection	Select ADC measurement
<input checked="" type="checkbox"/> ADC Configuration	Configure ADC Scaling Factors and Offsets
<input type="checkbox"/> Digital Hall Configuration	Configure Digital Hall Feedback systems (if available)
<input type="checkbox"/> Digital Hall Test Drive	Test Digital Hall Feedback systems (if available)
<input checked="" type="checkbox"/> ABN Encoder Configuration	Configure ABN(N) Encoder Interface for feedback of position and velocity
<input checked="" type="checkbox"/> ABN Encoder Test Drive	Test ABN(N) Encoder Interface for feedback of position and velocity
<input type="checkbox"/> Analog Encoder Configuration	Configure Analog Encoder like analog hall sensors or Sin/Cos encoders
<input type="checkbox"/> Analog Encoder Test Drive	Test Analog Encoder like analog hall sensors or Sin/Cos encoders
<input checked="" type="checkbox"/> Summary	Generate TMCL Scripts and save configurations

Please follow instructions carefully for quick start. Be careful with voltage references in open loop control mode and check motor currents with current probes if possible. Make sure that all supply voltages are applied correctly and that you can safely turn off your drive in case of failure. Therefore, you need to cut power supply or disable the power stage.

Special for Evaluation boards: if you are using a Trinamic power stage and evaluation board, you might be able to use standard values at some configuration steps.

3.5 Settings

Set number of pole pairs = 1 for the linear motor.

General configuration

Which basic parameters must be checked?

Before starting up the motor with open loop control, the PWM has to be adjusted to your power module/inverter. If you are using a standard evaluation board from Trinamic, you can set Default values with the following buttons. First select a universal Power Evaluation.

1. Select a Power driver: TMC6100-EVAL

2. Choose Motor_Type according to your motor and power stage. The TMC467X supports DC motors, two phase stepper motors (FOC2) and BLDC/PMSM motors (FOC3).
Motor type: Three phase BLDC

3. Set number of pole pairs according to your motor. If you don't know the number of pole pairs of your motor, we can determine the number of pole pairs in the next steps. You can also determine this value from your motor's nameplate data using the formula below. Stepper Motors usually have 30 pole pairs (1.8°).

$$p_{motor} = (60 * f_{nom} [Hz]) / n_{nom} [rpm]$$
 e.g. $n_{nom} = 3000$ rpm, $f_{nom} = 200$ Hz, $p = 4$

$$p_{motor} = 360 / (step_angle * 4)$$
 e.g. $step_angle = 1.8^\circ$, $p = 50$
 Pole pairs: 1

4. Set PWM_MAXCNT (0x18) to change switching frequency.
Calculation: $PWM_MAXCNT = 1/(f [Hz] * 10 ns) - 1$
 PWM frequency: 25 kHz

5. Set Brake Before Make (BBM) times according to your power stage. These values can be defined separately for high side and low side switches.
 BBM low side: 25 BBM high side: 25

6. Select the pwm chopper mode. For typical applications use "centered PWM for FOC" to enable the pwm and "PWM = OFF" for free running.
 PWM chopper mode: centered PWM for FOC



3.6 Open Loop

Run the motor in open loop for ADC and encoder configuration.

Wizard Pool COM4/USB/id1/Landungsbruecke/TMC4671-EVAL [M1] Weasel configurator wizard (TMC4671) (3/8)

Adr	Name	Value
0x52	PHLE_SELECTION	phle_openloop
	MODE_MOTION	uq_ud_ext
	MODE_RAMP	no velocity ramping
0x63	MODE_FF	disabled
	MODE_PID_SMPLE	0
	MODE_PID_TYPE	parallel PI
0x24	UD_EXT	13000
	UQ_EXT	
0x1F	OPEN_LOOP_PHI_DIRECTION	<input type="checkbox"/> Open loop phi direction
0x20	OPEN_LOOP_ACCELERATION	60
0x21	OPEN_LOOP_VELOCITY_TARGET	5
0x1B	N_POLE_PAIRS	1
	MOTOR_TYPE	Three phase BLDC
0x1C	PHLE_EXT	0

Export to TMC/PC host

Open Loop settings

Description

Running the motor with Open Loop Control allows further configuration. For Open Loop mode, the motor should be decoupled from any load or the load should be as low as possible.

- You can enable Open Loop mode by choosing the commutation angle source (PHLE_SELECT0N, 0x52) and the correct MOTION_MODE (0x63).
- Enter a target velocity in rpm for the open loop angle generator and a value for the acceleration in rpm/s. Typical values are 30 rpm (0.5 Hz) and 120 rpm/s.
- Now we are at the critical point. Watch the motor currents with the current probe measurement if available. Increase UD_EXT in small steps until the motor is running. If you increase UD_EXT too much, very high currents can be impressed into the motor and you might damage your power stage and/or your motor.

HINT: Some Motors start spinning only at very low target velocities and high currents. Try varying these parameters.

How to use the open loop mode?

To start the motor in open loop mode do the following:

- select the phle_open_loop mode in register 0x52 (def: 1)
- select uq_ud_ext mode without ramp in register 0x63 (def: 8)
- select the used down amplifier in register 0x24 (def: UD_EXT=1, UQ_EXT=0)
- set an acceleration in register 0x20 (def: 60 [rpm/s]) and a target velocity in register 0x21 (def: -30 [rpm])

You can also click "Set defaults" to set the default values and start the motor. Afterwards, you can use the control box to set new target velocities.

How to estimate the motor pole pair count?

For an estimation of the motor pole pair count of your motor, the motor must be rotated with a controlled commutation angle, a defined force, and no load.

- select a slow velocity in register 0x21 (e.g. -1 [rpm]) and start the motor in controlled mode with the actual open loop settings
- clear the estimated result at a motor position which is easy to remember
- read the estimated number of motor poles after exactly one revolution and update register 0x1B

Estimated motor pole pairs:

Intro: Settings: Open loop: **ADC config:** ABN encoder: Summary:

3.7 ADC config

Wizard Pool COM4/USB/id1/Landungsbruecke/TMC4671-EVAL [M1] Weasel configurator wizard (TMC4671) (4/8)

Adr	Name	Value
	ADC_I0_SELECT	ADCS0_I0_RAW (sigma delta ADC)
	ADC_I1_SELECT	ADCS0_I1_RAW (sigma delta ADC)
0x0A	ADC_I_UX_SELECT	IUX = ADC_I1
	ADC_I_V_SELECT	V = ADC_I2
	ADC_I_WY_SELECT	WY = ADC_I0
	cfg_dsmodulator_a	int. dsMOD
	mdck_polarity_a	<input type="checkbox"/>
	mdat_polarity_a	<input type="checkbox"/>
	sel_mdck_mdck_a	<input checked="" type="checkbox"/>
	blanking_a	0
0x04	cfg_dsmodulator_b	int. dsMOD
	mdck_polarity_b	<input type="checkbox"/>
	mdat_polarity_b	<input type="checkbox"/>
	sel_mdck_mdck_b	<input checked="" type="checkbox"/>
	blanking_b	0
0x05	dsADC_MCLK_A	536 870 912
0x06	dsADC_MCLK_B	0
0x07	dsADC_MDEC_A	334
	dsADC_MDEC_B	334

Export to TMC/PC host

ADC selection

How to select the correct ADC inputs?

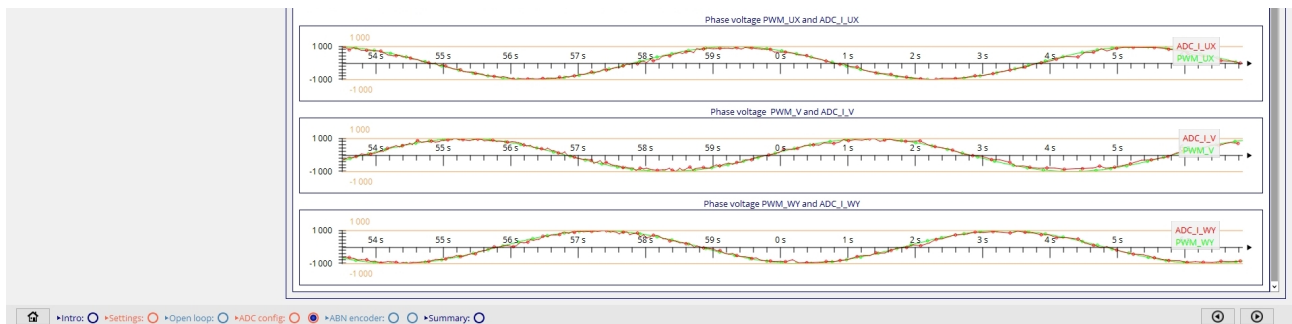
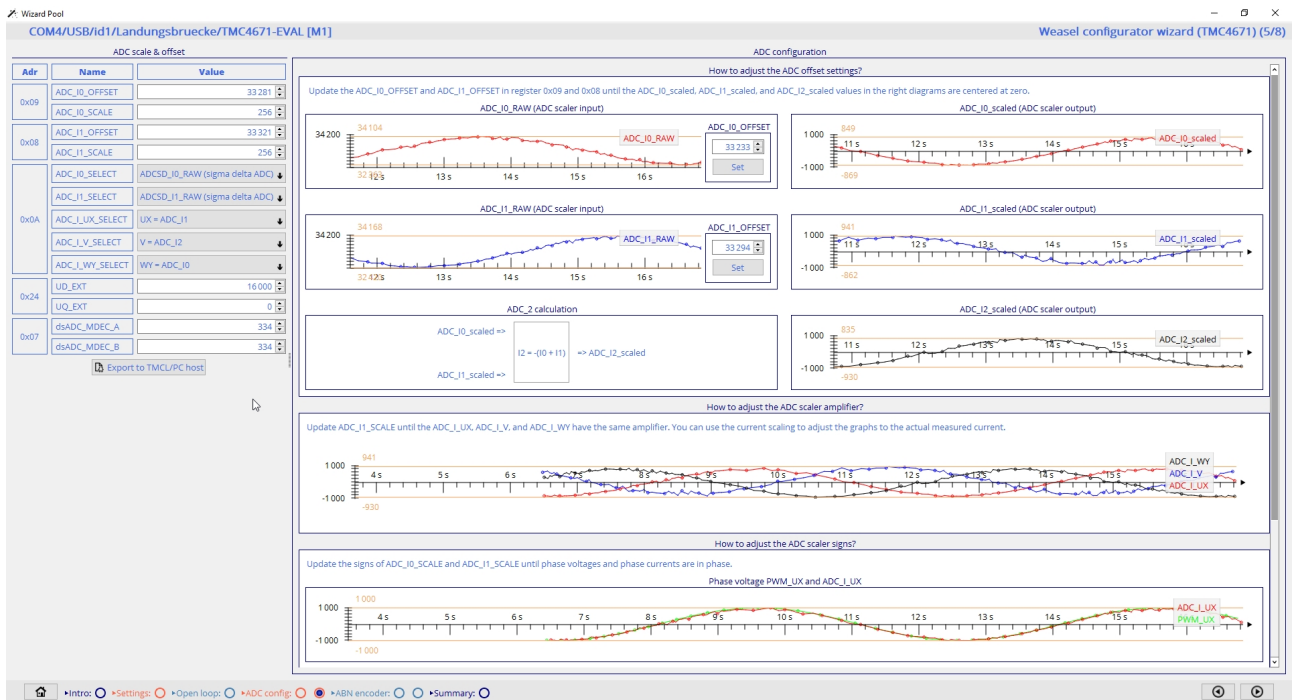
The TMC4671 has different ADC inputs for phase current measurement and you can also provide digitalized phase current values via SPI. All these input signals are provided to the internal ADC scaler component of the controller. You can configure the ADC Scaler to fit your hardware design. Choose the correct ADCs by matching your hardware design. The ADC signals should look like sinusoidal signals. For stepper motors they have a phase of 90° and for BLDC/PMSM motors they should have a phase of 120°.

Offsets and gains can be configured in the next steps of this wizard.

Turn motor slowly

Intro: Settings: Open loop: **ADC config:** ABN encoder: Summary:





3.8 ABN encoder

Init with offset estimation (Wizard) or init with offset estimation (Firmware).



Wizard Pool

COM4/USB/id1/Landungsbruecke/TMC4671-EVAL [M1] Weasel configurator wizard (TMC4671) (6/8)

Encoder configuration

Adr	Name	Value
0x25	apol	<input type="checkbox"/> Polarity of A pulse.
	bpol	<input type="checkbox"/> Polarity of B pulse.
	npol	<input type="checkbox"/> Polarity of N pulse.
	use_abn_abn_n	<input type="checkbox"/> N and A and B
	cm	<input type="checkbox"/> Clear count at N-pulse.
	direction	<input type="checkbox"/> Decoder count direction.
0x26	ABN_DECODER_PPR	30 000
0x27	ABN_DECODER_COUNT	25 713
0x28	ABN_DECODER_COUNT_N	25 719
0x29	ABN_DECODER_PHI_E_OFFSET	0
	ABN_DECODER_PHI_E_OFFSET	0
0x2A	ABN_DECODER_PHI_M	-9 331
	ABN_DECODER_PHI_E	-9 331
0x6B	PID_POSITION_ACTUAL	-674 668
	N_POLE_PAIRS	1
0x1B	MOTOR_TYPE	Three phase BLDC

[Export to TMC/PC host](#)

Encoder configuration

2) Direction check

Check whether the Open Loop angle and the encoder angle run in the same direction and update the direction bit if necessary:

The direction is different, please change the encoder direction bit in register 0x25. The direction is the same, the direction bit is set correct.

3) Encoder resolution

Update the encoder resolution in register 0x26. The estimated value can help you to find the correct value.

Estimated encoder steps: 29278 [Reset estimation](#)

4) Encoder N-Pulse setting

For correct positioning it is vital to have the correct impulse set to minimize the position error.

The N-pulse is short, the position can be correctly determined from both directions. The N-pulse is long resulting in a position error. To get a short impulse use A and B as N-pulse by setting the bit in register 0x25.

How to initialize the motor?

[Init with offset estimation \(Firmware\)](#) [Init with offset estimation \(Wizard\)](#) [Init with hall sensor support](#)

Intro: Settings: Open loop: ADC config: ABN encoder: Summary:

3.9 Encoder Test Drive (Torque Mode)

Wizard Pool

COM4/USB/id1/Landungsbruecke/TMC4671-EVAL [M1] Weasel configurator wizard (TMC4671) (7/8)

Encoder test drive

Adr	Name	Value
0x63	MODE_MOTION	torque_mode
	MODE_RAMP	no velocity ramping
	MODE_FF	disabled
	MODE_PID_SMP_L	0
	MODE_PID_TYPE	parallel PI
0x52	PHI_E_SELECTION	phi_e_abn
0x54	PID_FLUX_I	256
	PID_FLUX_P	256
0x56	PID_TORQUE_I	256
	PID_TORQUE_P	256
0x5E	PID_TORQUE_FLUX_LIMITS	1 000
0x64	PID_FLUX_TARGET	0
	PID_TORQUE_TARGET	0
0x50	VELOCITY_SELECTION	PHI_E_SELECTION
	VELOCITY_METER_SELECTION	default

[Export to TMC/PC host](#)

Encoder test drive (torque mode)

How to drive a motor with an encoder?

To start the motor with digital hall sensor feedback in torque mode do the following:

- select the torque mode in register 0x63 (def: 1)
- select phi_e_hall in register 0x52 to select digital hall sensor
- set P and I parameters for torque and flux control in register 0x54 and 0x56 (def: P=256, I=256)
- set the maximum current limit in register 0x5E (def: 1000)
- set a target current in register 0x64 (def: FLUX_TARGET=0, TORQUE_TARGET=500)

You can also click "Set defaults and start" to start with default values.

PID_TORQUE_TARGET: 500

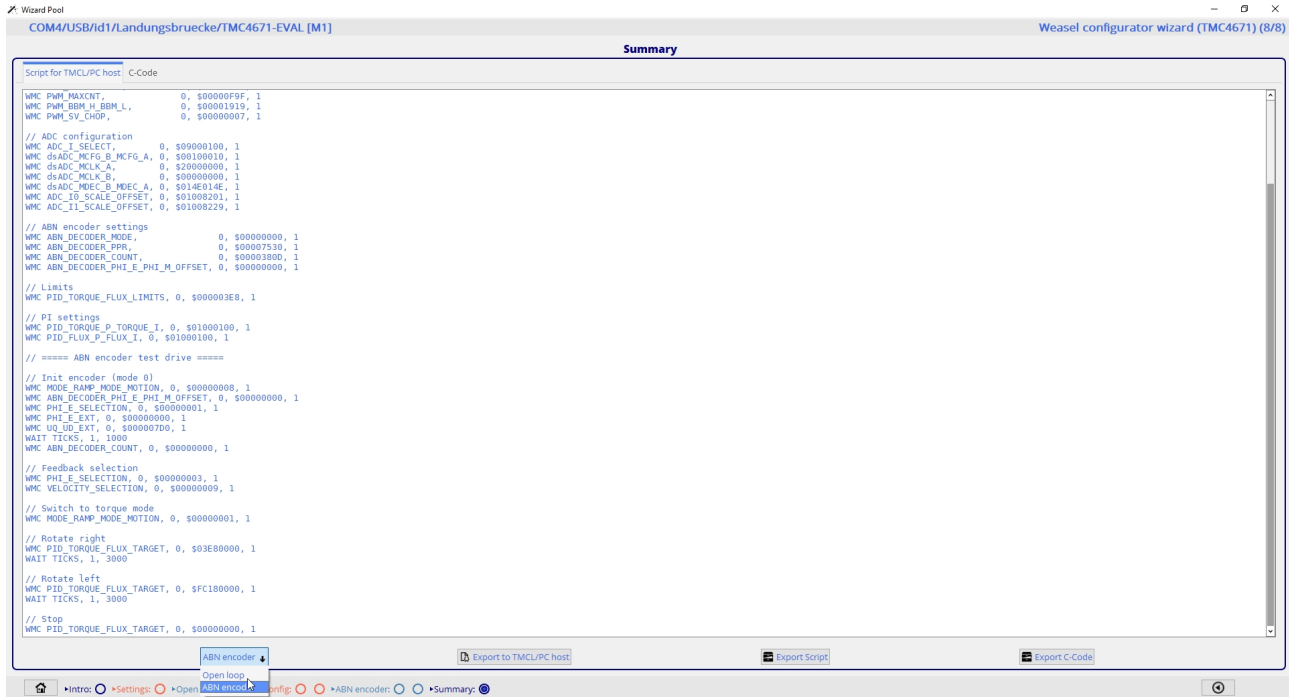
[Set defaults and start](#)

Intro: Settings: Open loop: ADC config: ABN encoder: Summary:



3.10 Save Configuration

On the summary page choose ABN encoder for a preconfigured script. This will include the encoder alignment (mode 0) Then save the script using the *Export Script* button.



3.11 Feedback Selection

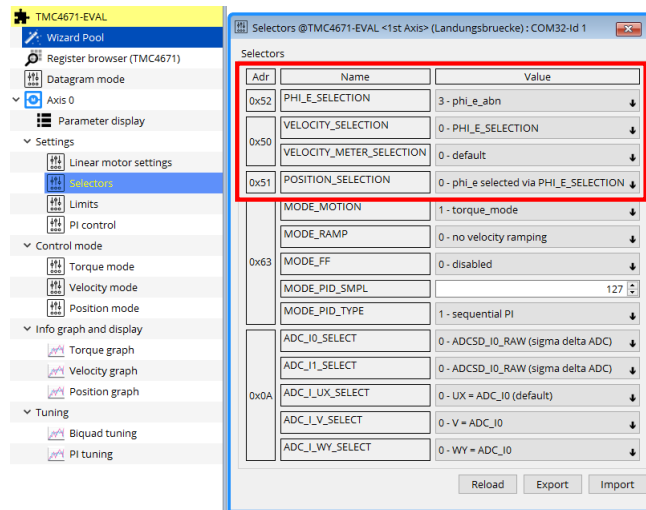


Figure 5: Selector settings

3.12 Linear Scaling Factor

Clear the position and enable linear scaling factor.



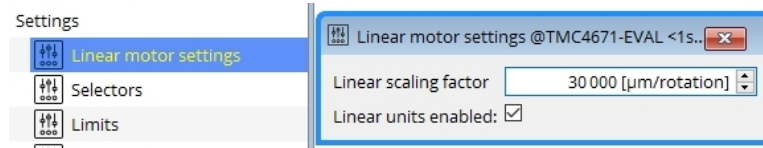


Figure 6: Linear motor settings

3.13 PI Tuning

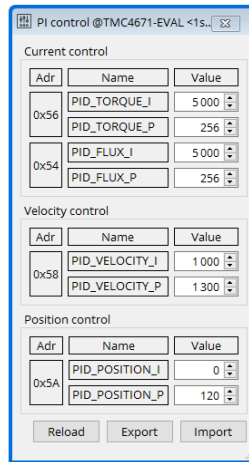


Figure 7: Example PI values after tuning

For PI tuning refer to the [PI tuning appnote](#).

3.14 Position Mode

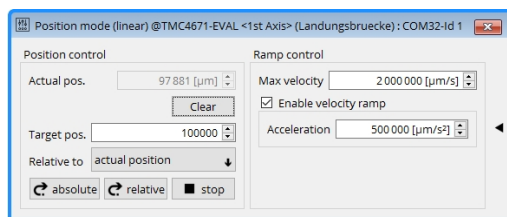


Figure 8: Position mode tool with linear units



Position mode test run:

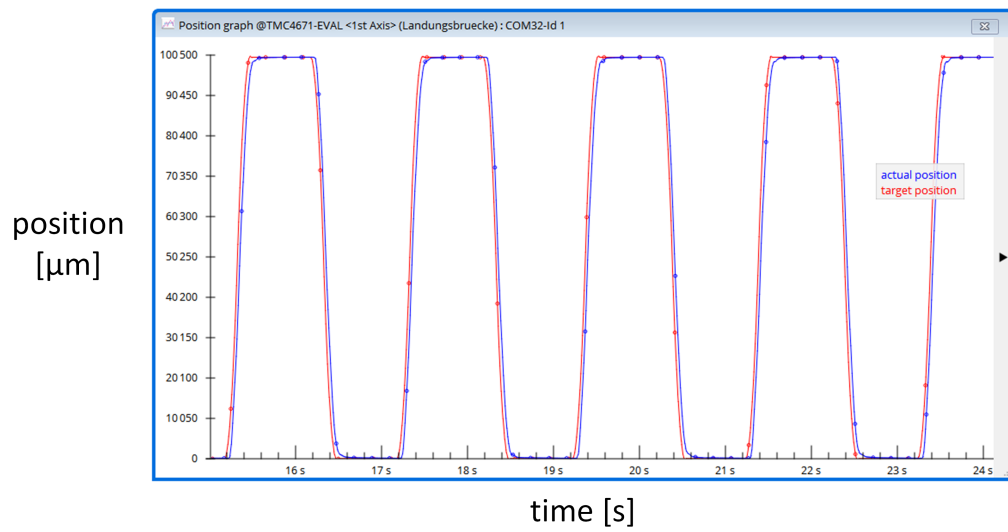


Figure 9: Position mode with target position of 100000 micrometer with linear units

4 Additional Resources

- [TM4671 product page](#)
- [TMC4671 PI tuning appnote](#)
- [Driving a linear stage with TMC4671](#)
- [TMC4671 API on github](#)
- [TMC4671 Python resources](#)



5 Revision History

Version	Date	Author	Description
V1.0	25.06.2021	JPX	Initial version

