

MCST3601 – stallGuard2™

Introduction

Among the multitude of useful features offered by the MCST3601 stepper motor driver board, the “stallGuard2™” function is one that is worth knowing and using. When operated in open loop, it is not possible to know if a stepper motor lost steps during the operational cycle of the application. The stallGuard2™ feature is able to detect if the motor stalls and allows reacting to this situation. This is particularly useful when for example unexpected events might lead to load increase on the motor shaft or if the application’s end of travel needs to be detected without reference switch. The present Application Note explains how to configure the stallGuard2™ function.

StallGuard2 operating principle

The stallGuard2™ function measures the difference between the energy going into the motor and the energy used for moving the mechanics. Using the back-EMF of the motor, it evaluates the synchronism between the rotor and the electronic commutation. If the load is high and the rotor starts to be delayed with the electronic commutation, stallGuard2™ will be able to detect it and allows to react in different ways, depending on user needs:

- Interrupt generation
- Motor stop
- Error counter increment, flag activation, etc.

The Back-EMF effect, from which the stall is detected, is highly dependent on the motors intrinsic properties and varies with operational conditions, like motor speed, temperature, etc. For this reason, stallGuard2™ feature has to be finely tuned for a specific operational condition.

StallGuard2 parametrization

Despite the empirical approach to use for tuning the stallGuard2™ feature, the following procedure will help you to identify faster the optimal parameters.

Please use the following procedure to configure stallGuard2™:

1. Set the motor in nominal application conditions. Nominal speed, current, temperature and load conditions shall be met.
2. Tune stallGuard2™ parameters by setting the detection threshold and increasing progressively the load, up to the stall detection value. Several iterations may be necessary.
3. Set the Stop on stall value. This number corresponds to the speed above which the stallGuard2™ detection is activated. Perform several starts/stops of the motor to validate this parameter.
4. Check and validate the defined settings in the complete operational cycle defined by the application.

The listed operations are fully detailed in the following points.

1. Motor nominal rotation setup

Before being able to setup stallGuard2™, motor rotation has to be initiated in the nominal conditions of the application:

- Connect MCST3601 board to the computer through USB cable
- Connect the stepper motor to the MCST3601 board
- Start TMCL-IDE. Verify that the board is detected
- Setup motor parameters, either in direct mode or using shortcode
 - o Direct mode (for advanced users):
 - Open the Direct mode dialog box
 - Setup motors parameters
 - o Shortcode mode (download from faulhaber.com):
 - Open the TMCL creator dialog box
 - From TMCL creator, open an example code of the required motor
 - Assemble the code , Download the code 
 - Verify that the jumpers are mounted in accordance with the code comments
- Connect the main power supply to the MCST3601 board
- Open the velocity mode and coolStep dialog boxes
- Setup the nominal velocity (values can be taken from shortcode)
- Click on “play” in the required direction to start rotation
- Check that motor is rotating as desired

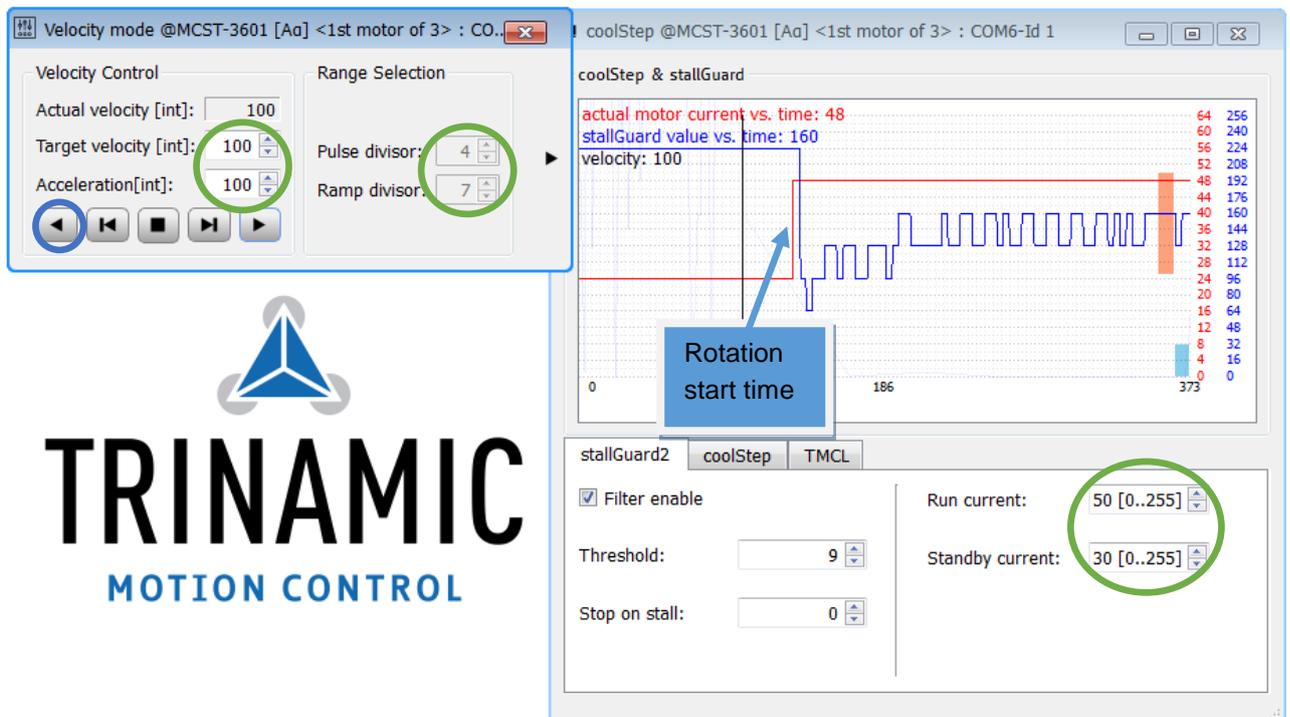
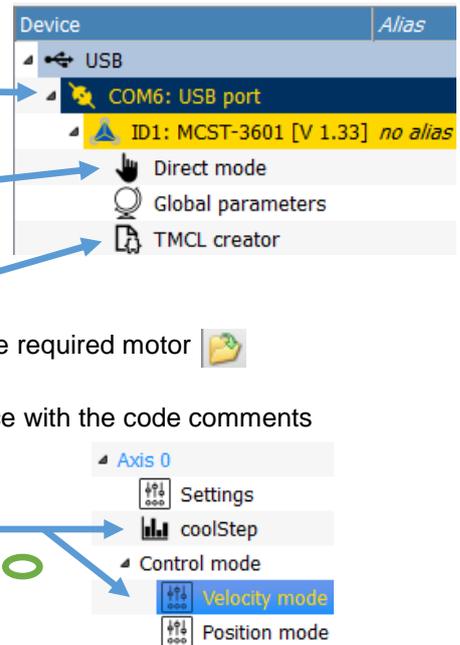


Figure 1: Motor start in the nominal application conditions

When the rotation is launched, the current shown by the red line on the graph rises to its “Run current” value. stallGuard2™ value, represented by the blue line on the graph, will stay to zero or reach any random value depending on the value that is set through the “Threshold” field (default value = 0).

2. stallGuard2™ parameters tuning

The optimal stallGuard2™ parameters have to be defined with the motor rotating in the conditions of the application.

While the motor is running, the following operation have to be performed:

- Increase the Threshold parameter in order to see the blue line (stallGuard2™ value) at a level higher than zero but lower than the graph saturation level 
- It is recommended to enable the filter, in order to get a mean measurement (more stable) 
- Increase the load applied on motor shaft, up to a level that makes the stallGuard2™ value reach zero:
 - o If zero value cannot be reached, even when the motor stalls: decrease parameter “Threshold” 
 - o If zero value is too easily reached: increase parameter “Threshold” 
 - o After Threshold modification, check the stallGuard2™ value when increasing the load again

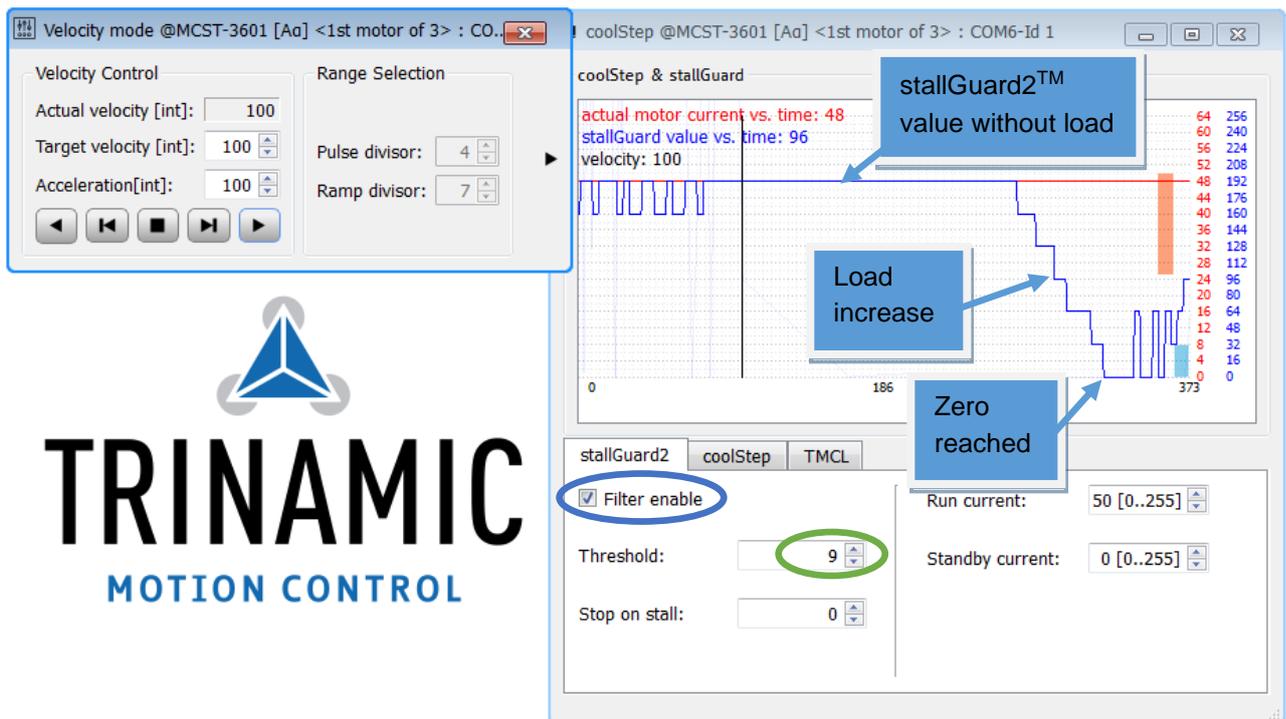


Figure 2: StallGuard2 parameters tuning

If the stallGuard2™ Threshold is correctly set, the threshold value is situated at a value above zero but can drop to zero if a defined load is applied to the motor shaft. This load must be close to the value that makes the motor stall.

3. Stop on stall value definition

The stallGuard2™ parameters are optimized for a limited range of motor working conditions (defined by speed, current, temperature and load). It means that during acceleration, the motor is not at the speed conditions for which the stallGuard2™ has been optimized. Stall could be therefore abusively detected by stallGuard2™ during acceleration if the “Stop on stall” parameter did not exist.

“Stop on stall” parameter defines the speed above which stallGuard2™ is activated and motor can be stopped in case of stall detection. It usually has to be set to a level that is slightly below the application nominal speed. Respecting this, stallGuard2™ is activated only when nominal speed is almost reached.

This parameter can be set through the “Stop on stall” field  and uses the same speed unit than the “Target velocity”  parameter.

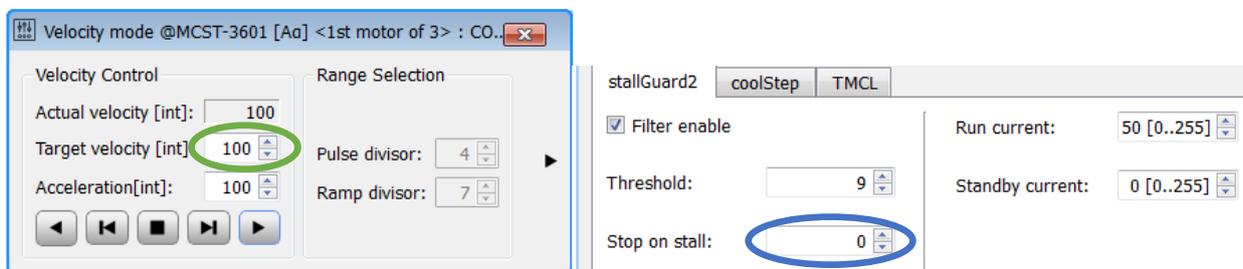


Figure 3: Stop on stall parameter definition

Setting the “Stop on stall” parameter to zero will completely disable the automatic motor stop function in case of stall detection. If this value is set to zero and a stall is detected, motor will continue to run.

4. Configuration verification and validation

After the complete StallGuard parametrization and tuning, shorts tests can be done to verify and validate that StallGuard function is working as desired. The following cases can be simulated for the validation:

- **Motor rotation at nominal load:**
Apply nominal load on motor shaft. Try several starts/stops of the motor and check that stallGuard2™ is not activated (motor is not stopped). Vary speed, current, acceleration, load, temperature (if possible) of the motor inside the nominal working range. Check that no stall is abusively detected in all the situations.
- **Stall detection case:**
Apply a load that should make the motor stall. Check that stallGuard2™ has detected the stall situation and stopped the motor. Vary speed, current, temperature (if possible) and check that in every situation the stall is detected by stallGuard2™ and the motor is stopped.

stallGuard2™ function usage

The stallGuard2™ functions can be used through several ways, according to the user preferences and needs. When the StallGuard parameters are defined, they can be introduced in user code and the stall detection process can be managed in multiple manners. Some of them are presented hereafter.

Include stallGuard2™ Parameters in processor code:

When the stallGuard2™ is fully configured and validated through the dedicated interface windows, the parameters can be integrated in the processor code, using the “Copy to TMCL creator” button from TMCL tab, as illustrated by the following screenshot:

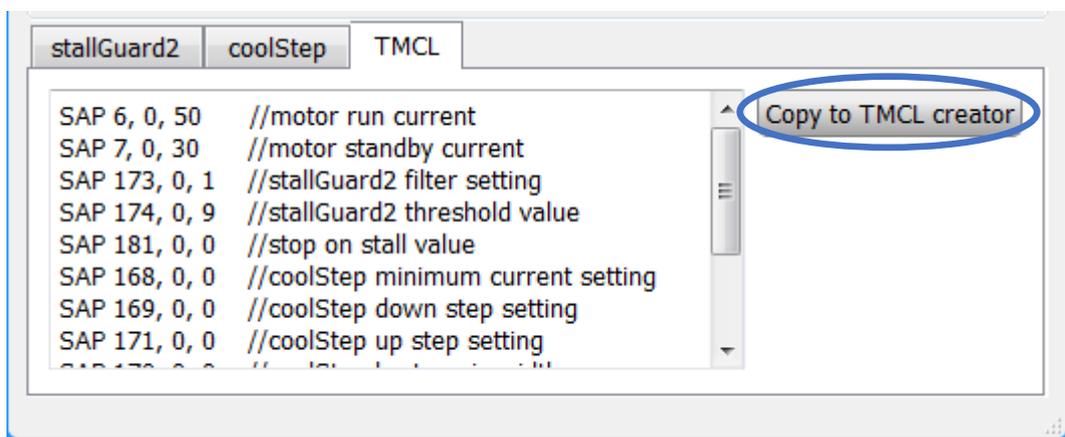


Figure 4: Copy StallGuard parameters to processor code

This setup code can be added at the beginning of the user program, with all the others microprocessor parameters definitions.

stallGuard™ Interrupt management:

If necessary, some dedicated code can be added by user for the stallGuard2™ interrupt management, if the application requests a special operation to be executed in case of stall detection.

To activate interrupts, the following functions have to be used:

```
VECT 15, StallGuardIrq //define the interrupt vector (15 for StallGuard)
EI 15 //enable this interrupt
EI 255 //globally switch on interrupt processing
```

The interrupt routine will contain the user code:

```
//Here is the interrupt handling routine
StallGuardIrq:
    //write user code here
RETI //end of interrupt
```

The interrupt routine can contain every functions that the user could need, including for example counter increment, output activation, motor operation, etc.

stallGuard™ status reading:

The axis parameter registers include a field containing the error flags. Among the latter figures the stallGuard2™ status flag:

stallGuard2™ status : 1 → threshold reached

This parameter can be accessed by the following command:

```
GAP 208, 0 //get driver error flags (bit 0 = StallGuard status)
```

stallGuard2™ combined with CoolStep

The StallGuard2 function brings another advantage when combined with the coolStep™ function. The latter allows the current to be dynamically adjusted according to the motor torque needs. The function is fully described in the application note AN017.

Legal notices

Copyrights. All rights reserved. No part of this Application Note may be copied, reproduced, saved in an information system, altered or processed in any way without the express prior written consent of Dr. Fritz Faulhaber & Co. KG.

Industrial property rights. In publishing the Application Note Dr. Fritz Faulhaber & Co. KG does not expressly or implicitly grant any rights in industrial property rights on which the applications and functions of the Application Note described are directly or indirectly based nor does it transfer rights of use in such industrial property rights.

No part of contract; non-binding character of the Application Note. Unless otherwise stated the Application Note is not a constituent part of contracts concluded by Dr. Fritz Faulhaber & Co. KG. The Application Note is a non-binding description of a possible application. In particular Dr. Fritz Faulhaber & Co. KG does not guarantee and makes no representation that the processes and functions illustrated in the Application Note can always be executed and implemented as described and that they can be used in other contexts and environments with the same result without additional tests or modifications.

No liability. Owing to the non-binding character of the Application Note Dr. Fritz Faulhaber & Co. KG will not accept any liability for losses arising in connection with it.

Amendments to the Application Note. Dr. Fritz Faulhaber & Co. KG reserves the right to amend Application Notes. The current version of this Application Note may be obtained from Dr. Fritz Faulhaber & Co. KG by calling +49 7031 638 385 or sending an e-mail to mcsupport@faulhaber.de.