UM2529
User manual

STSW-L99SM81V GUI User Guide

Introduction

This document describes the STSW-L99SM81V Graphical User Interface (GUI) dedicated to set and control EVAL-L99SM81VQ and EVAL-L99SM81VY.

These evaluation boards are designed for Automotive Stepper Motor driver application.

The STSW-L99SM81V has been developed by using C++ and it works with a motherboard based on SPC560B microcontroller programmed with dedicated firmware that drives the L99SM81 assembled in the daughter board.

Contents

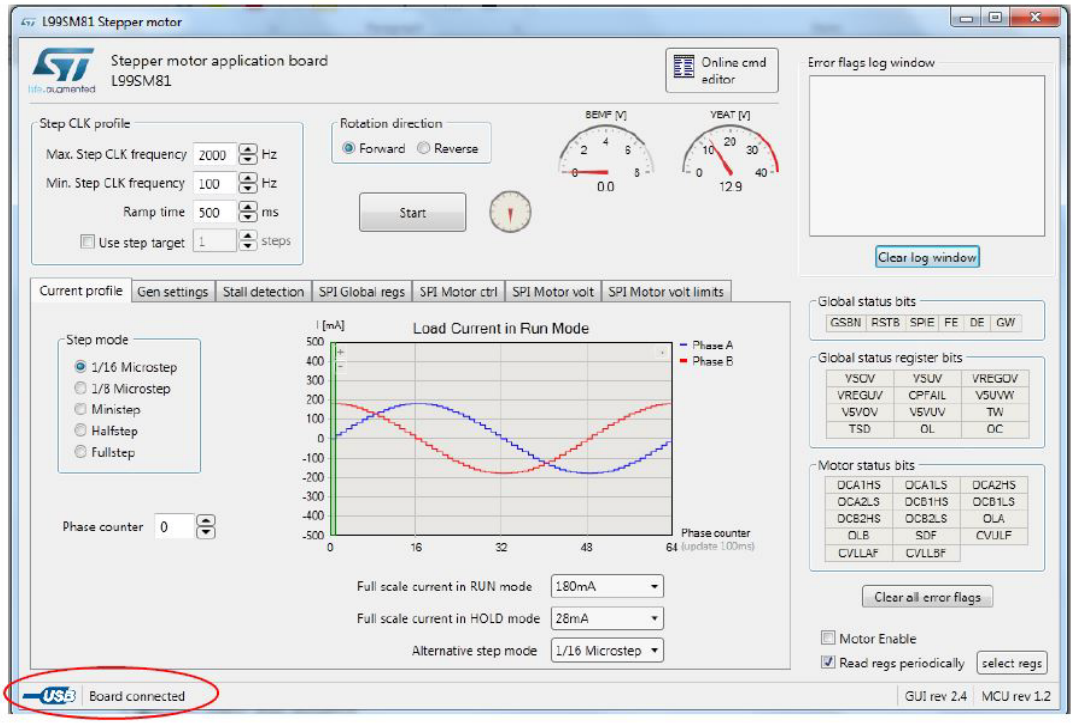
List of figures

1 STSW-L99SM81V GUI

1.1 Main GUI window

To run the GUI it is needed to have installed at least.NET framework 4.5 on the PC. Current installed.NET framework can be checked on the Control Panel → All Control Panel Items → Programs and Features.

Figure 1. Main GUI window

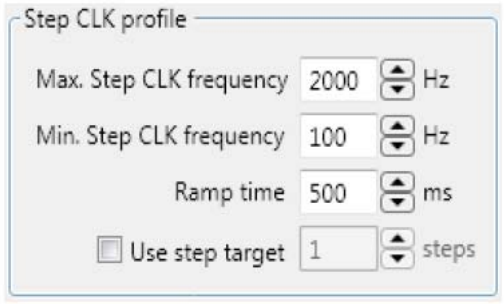


When the board is connected and the USB driver is installed the USB icon will turn visible.

Note: The debug window with communication interface messages will be shown by double clicking on the USB icon.

1.1.1 Step CLK profile

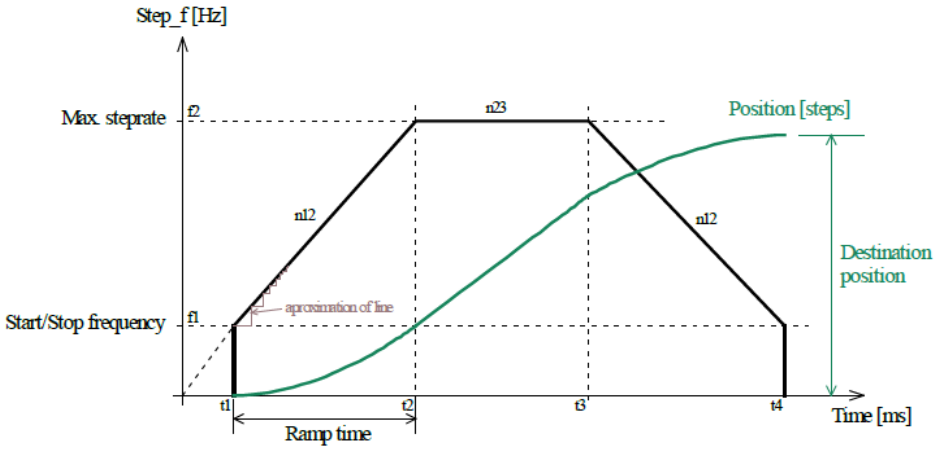
Figure 2. Step CLK profile



The “Min. Step CLK frequency“ is the step clock frequency at motor start up. The “Max. Step CLK frequency“ is the target step clock frequency at steady state. The “Ramp time” is the time needed to reach the target speed.

There is also the possibility to set a specific number of steps with the option “Use step target”. When this feature is enabled through the corresponding check box, the motor will stop once the programmed number of target steps is reached.

Figure 3. Step CLK profile graph



1.1.2 Current profile

In this window it is possible to set Step mode and current amplitude in RUN and HOLD mode.

The Phase counter value is accessible only when motor is stopped and CTRL1 pin is configured as OFF (MX1=0).

Figure 4. Current profile for MX1=1

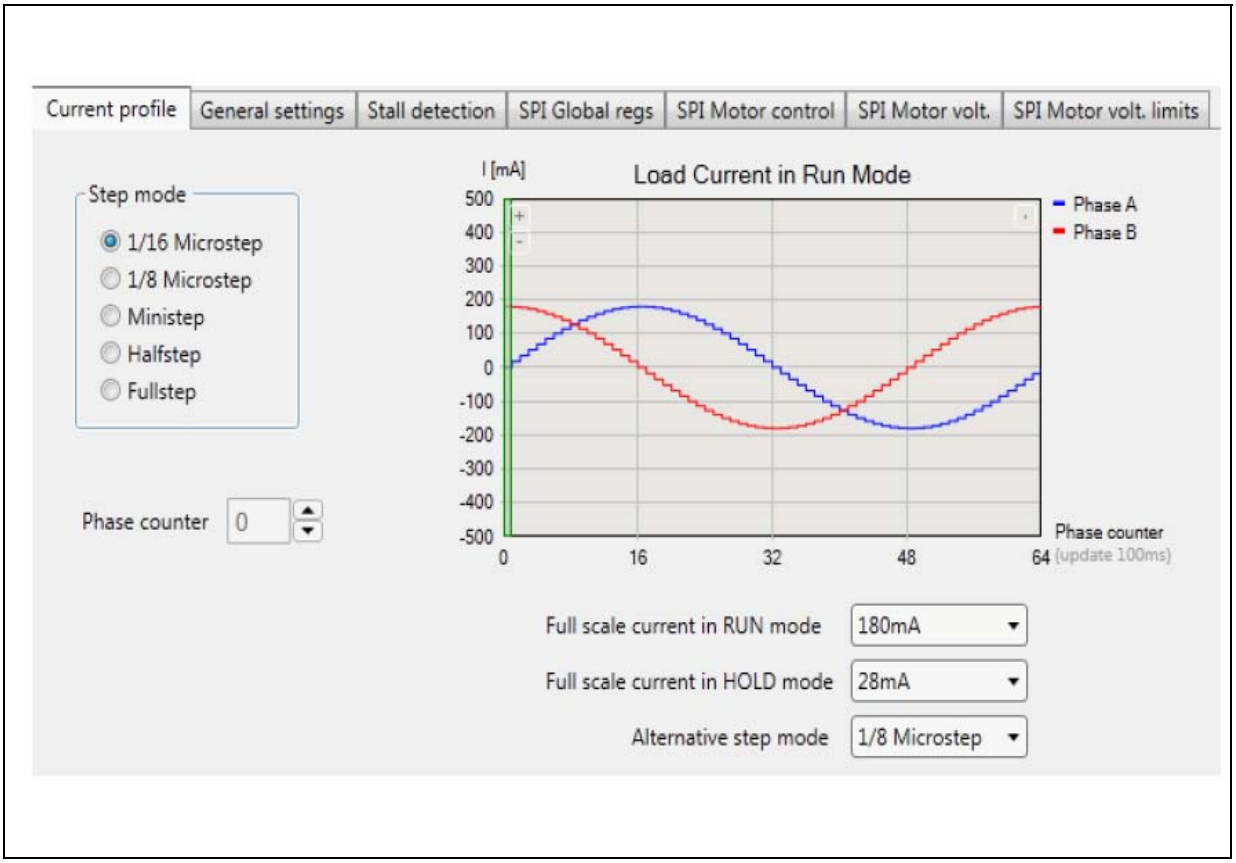


Figure 5. Current profile for MX1=0

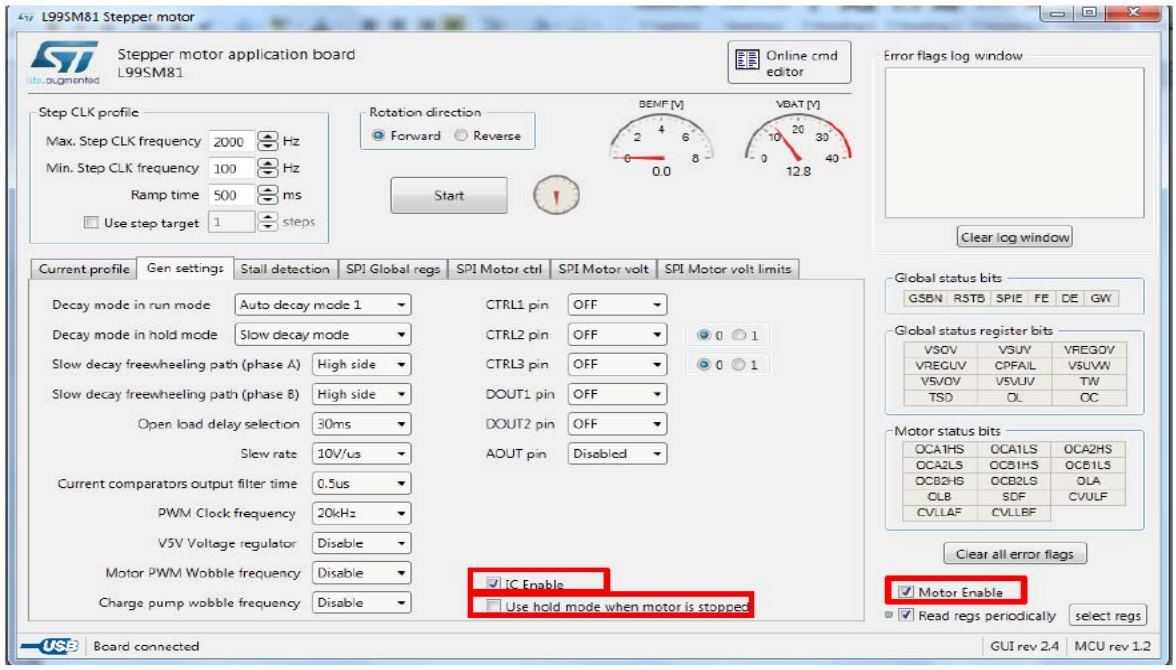
1.1.3 General settings

IC Enable: this check box controls the device enable pin. When the check box is set the enable pin will be set high.

Motor Enable: this check box controls the ME bit of MCR1 register. When the check box is set the outputs will be controlled according to the selected operating mode. When it is unchecked all outputs will be in high impedance.

Use hold mode when motor is stopped: this check box controls whether to put or not the driver in Hold mode whenever the motor is stopped.

Figure 6. General settings

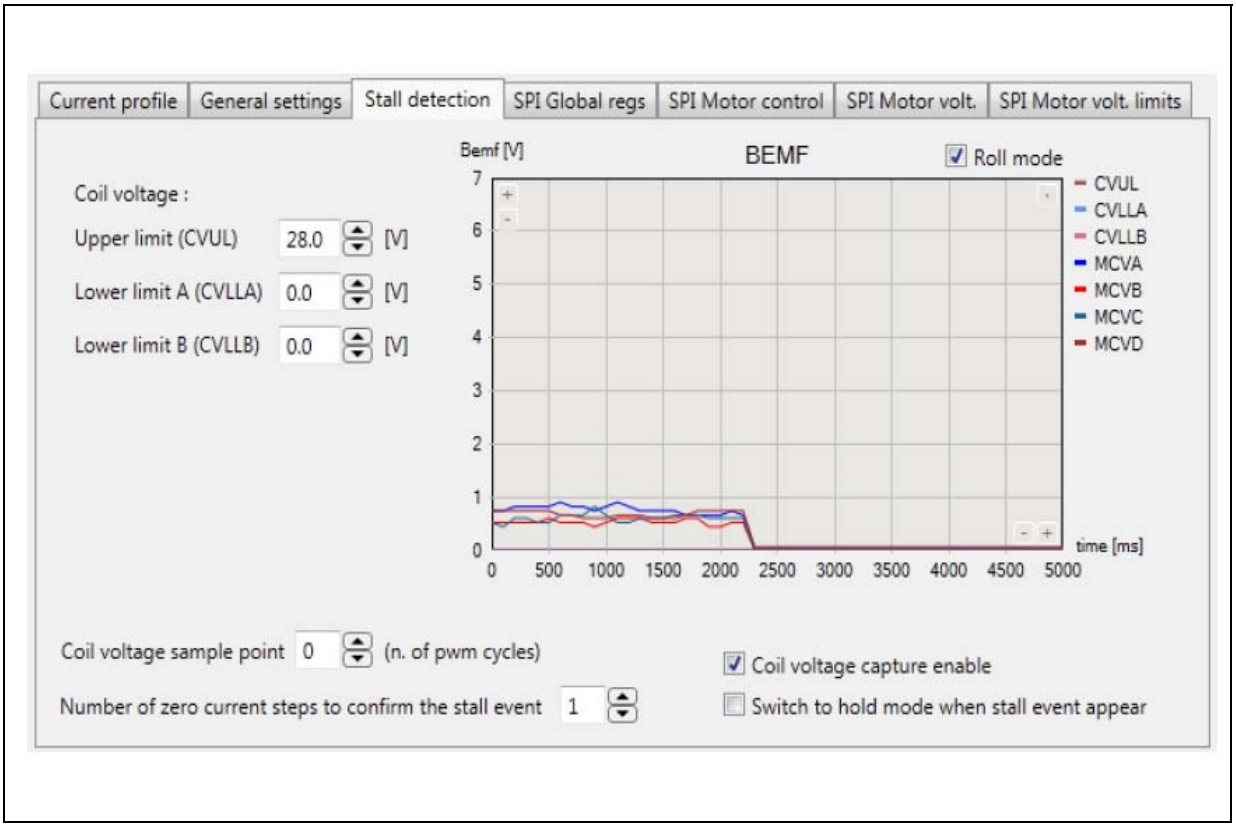


1.1.4 Stall detection

In this window there are the settings related to the Stall detection IP.

The graph shows the actual BEMF voltage measured at 0° (MCVA), 90° (MCVB), 180° (MCVC) and 270° (MCVD).

Figure 7. Stall detection



1.1.5 SPI registers

The control registers can be changed by direct clicking on the bits. This event will generate new write message command sent to the micro. Another possibility is to write register value to the text box which is placed on the upper right corner of the register. The registers can be read manually by clicking on “R” button or automatically by periodically reading of the selected registers.

Figure 8. SPI global regs

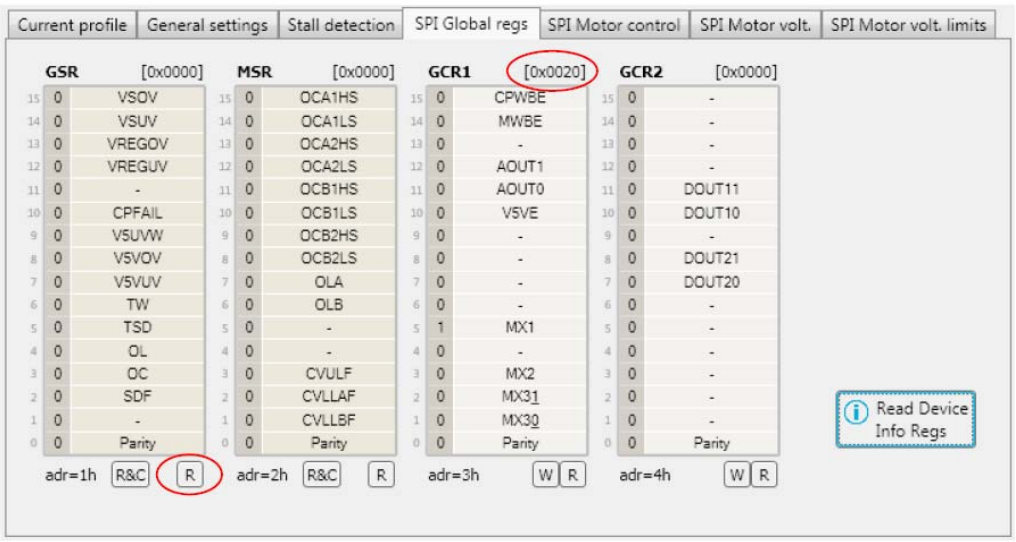


Figure 9. SPI Motor control

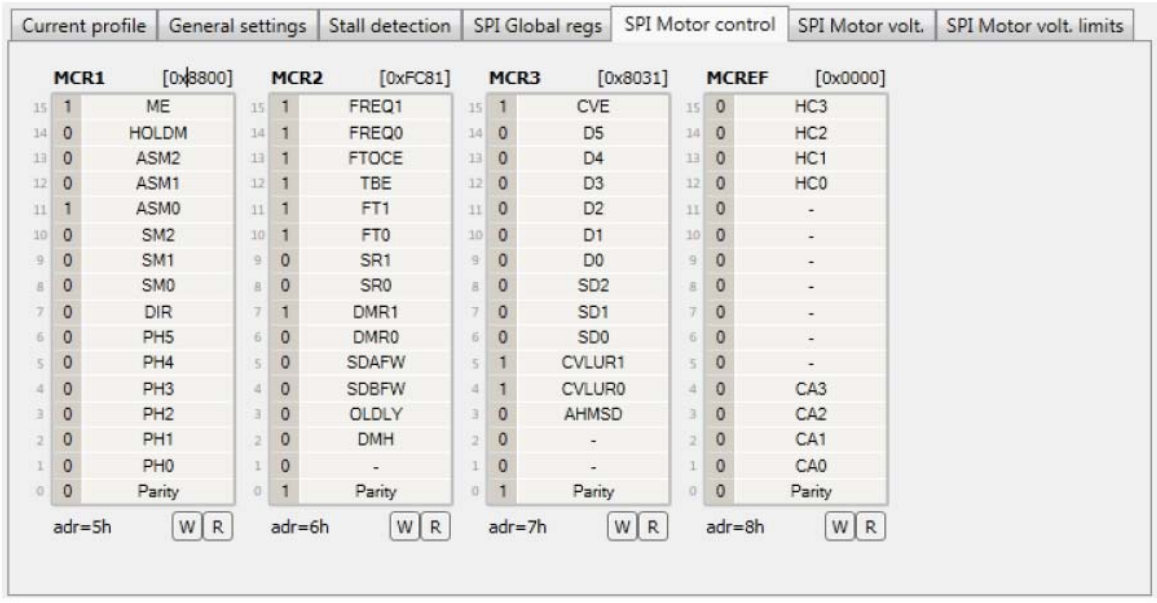


Figure 10. SPI Motor voltage

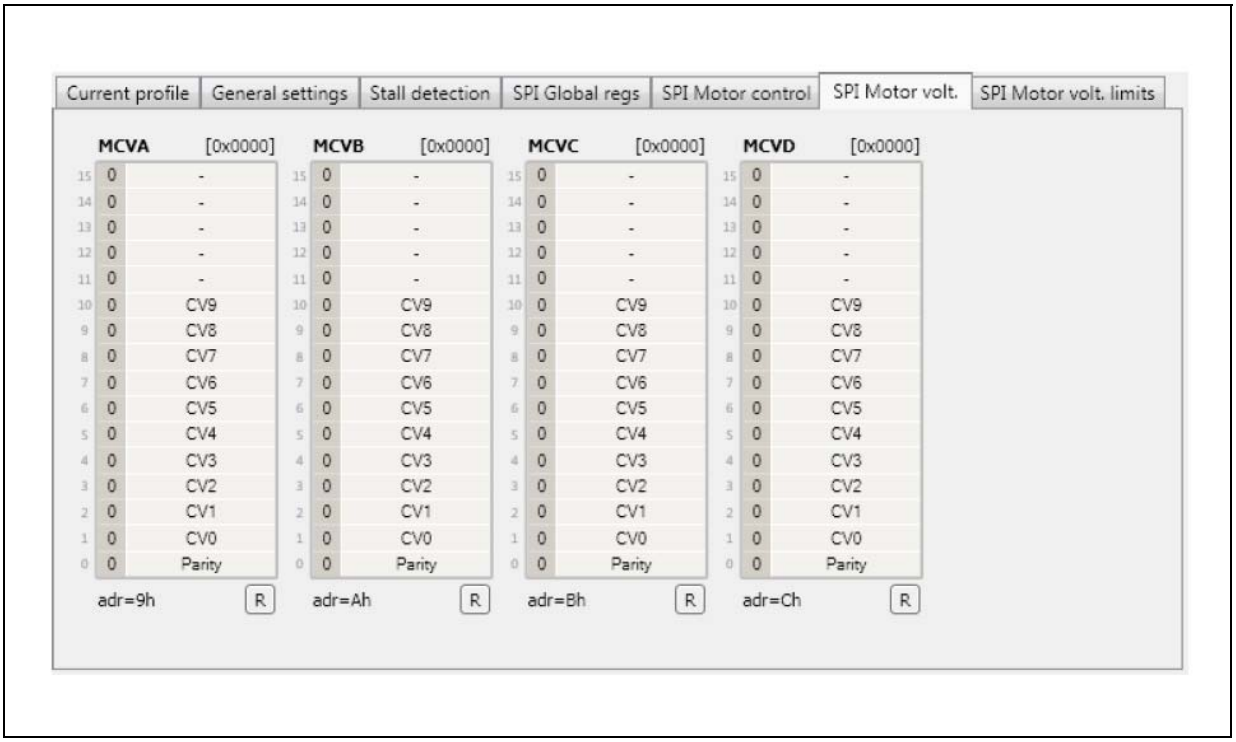
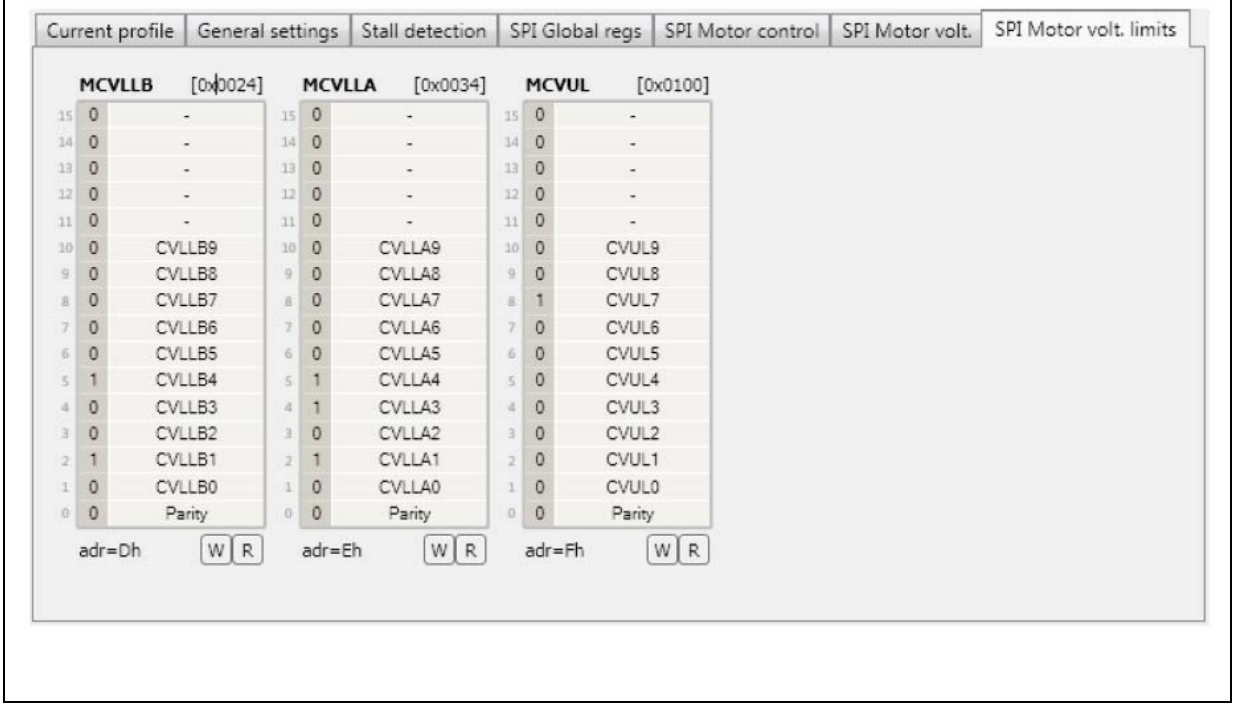


Figure 11. SPI Motor voltage limits



Periodical read of SPI registers

The registers can be read periodically when “Read regs periodically” check box is set (time period=100ms). Mainly it is dedicated for periodical read of status registers but also control registers can be read. Every time the GUI receives register values it will update all related GUI components. For example to get Phase counter value updated, the MCR1 register should be selected.

Figure 12. Read regs periodicaly

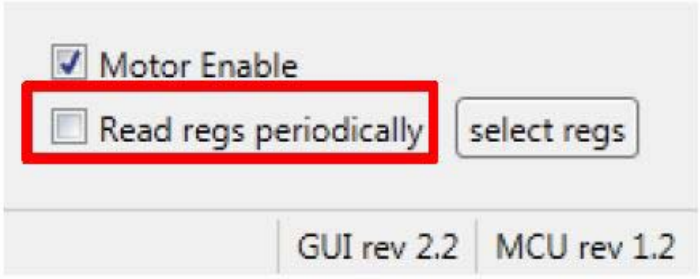
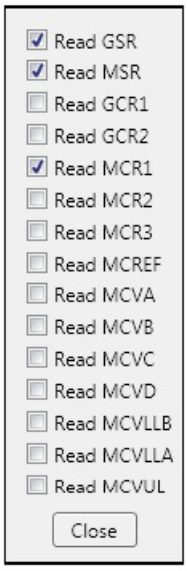


Figure 13. Registers selection

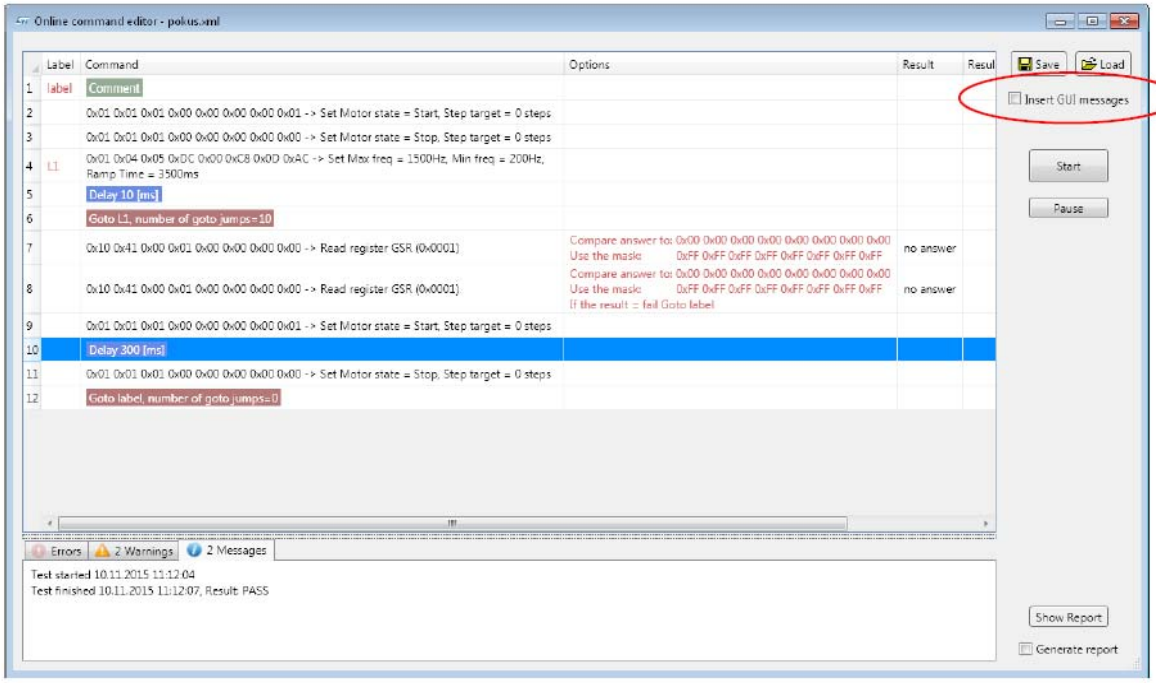


1.2 Online command editor

The online command editor window allows creating a command list to test the device.

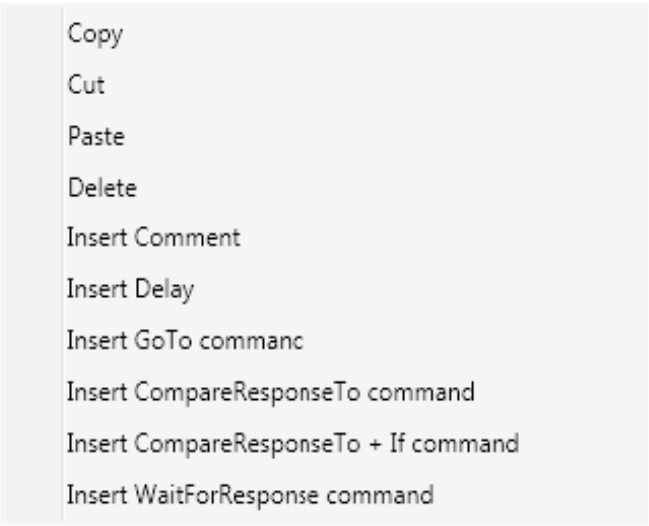
When the check box “Insert GUI messages” is set, then all the messages are logged into the online command editor window. It means that when the user will click for example on motor start button then the related message/command sent to the micro is also logged to the online cmd editor window.

Figure 14. Online command editor



By right clicking with the mouse over the command list window the context menu will appear making further features visible.

Figure 15. Context menu



1.2.1 Copy, Cut, Paste and Delete

Every line can be edited with Copy, Cut, Paste and Delete options.

1.2.2 Insert Comment

With this option it is possible to insert comment text.

1.2.3 Insert Delay

Insert delay in [ms]. Smallest value 10 ms.

Delay 10 [ms]

1.2.4 Insert GoTo command

With this command it is possible to jump to the defined label name.

Example:

Goto Label1, number of goto jumps=1

→ Jump to “label1”, number of jumps=1

Note: When it is set number of goto jumps = 0 → never ending loop

1.2.5 Insert CompareResponseTo command

With this command it is possible to wait for message answer and compare it to “Compare answer to” data. Then the result can be pass or fail. With the mask it can be selected which bits should be compared to data. (“0”=ignore, “1”=compare).

Example**:**

→ Compare message answer to “0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00” (it will be compared with all bits because mask is 0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF)

1.2.6 Insert CompareResponseTo + If command

With this command it is possible to wait for message answer and compare it to “Compare answer to” data. When the result is pass (fail) it will jump to defined label name.

Example:

→ Compare message answer to “0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00” (it will be compared with all bits because mask is 0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF). If the result is “pass” jump to “Label1”.

1.2.7 Insert WaitForResponse command

Wait for response message (no comparison is proceed).

2 Revision history

Table 1. Document revision history

Date	Revision	Changes
17-Dec-2018	1	Initial release.