

## **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

<b>1</b>	<b>STSW-L99SM81V GUI</b>	<b>4</b>
1.1	Main GUI window	4
1.1.1	Step CLK profile	5
1.1.2	Current profile	5
1.1.3	General settings	7
1.1.4	Stall detection	8
1.1.5	SPI registers	9
1.2	Online command editor	13
1.2.1	Copy, Cut, Paste and Delete	14
1.2.2	Insert Comment	14
1.2.3	Insert Delay	14
1.2.4	Insert GoTo command	14
1.2.5	Insert CompareResponseTo command	14
1.2.6	Insert CompareResponseTo + If command	15
1.2.7	Insert WaitForResponse command	15
<b>2</b>	<b>Revision history</b>	<b>16</b>

## List of figures

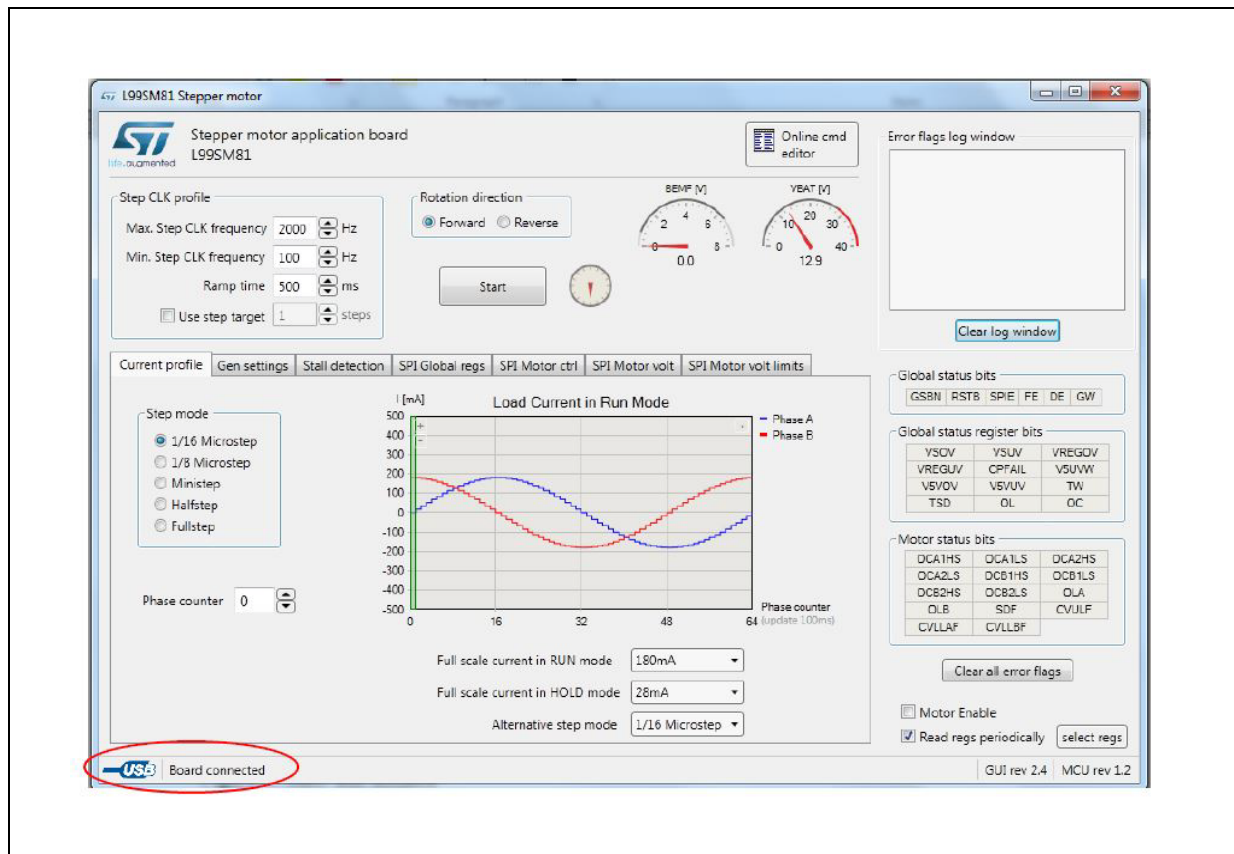
Figure 1.	Main GUI window . . . . .	4
Figure 2.	Step CLK profile . . . . .	5
Figure 3.	Step CLK profile graph . . . . .	5
Figure 4.	Current profile for MX1=1 . . . . .	6
Figure 5.	Current profile for MX1=0 . . . . .	7
Figure 6.	General settings . . . . .	8
Figure 7.	Stall detection . . . . .	9
Figure 8.	SPI global regs . . . . .	10
Figure 9.	SPI Motor control . . . . .	10
Figure 10.	SPI Motor voltage . . . . .	11
Figure 11.	SPI Motor voltage limits . . . . .	11
Figure 12.	Read regs periodically . . . . .	12
Figure 13.	Registers selection . . . . .	12
Figure 14.	Online command editor . . . . .	13
Figure 15.	Context menu . . . . .	14

# 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

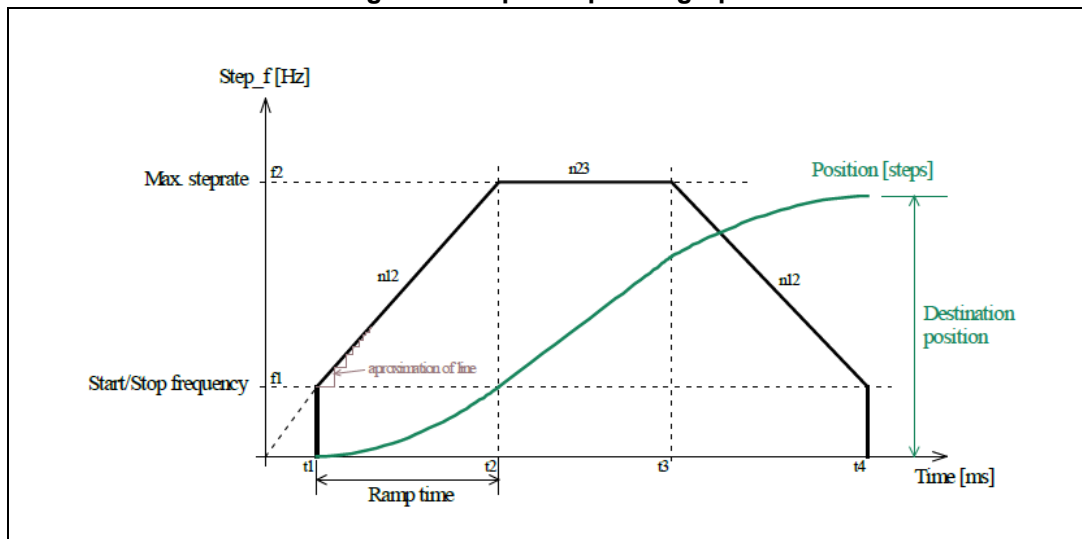
The screenshot shows a window titled "Step CLK profile" with the following settings:

- Max. Step CLK frequency: 2000 Hz
- Min. Step CLK frequency: 100 Hz
- Ramp time: 500 ms
- Use step target: 1 steps

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

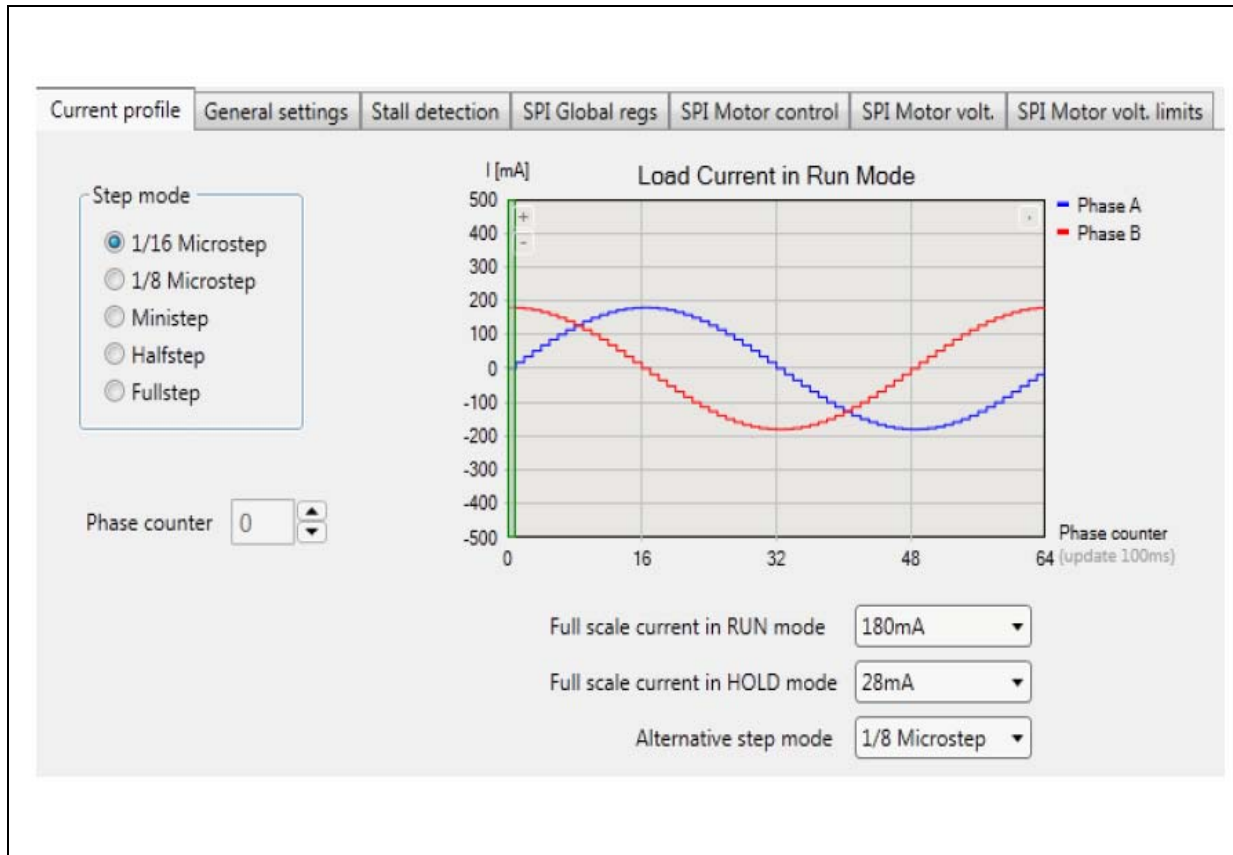


Figure 5. Current profile for MX1=0



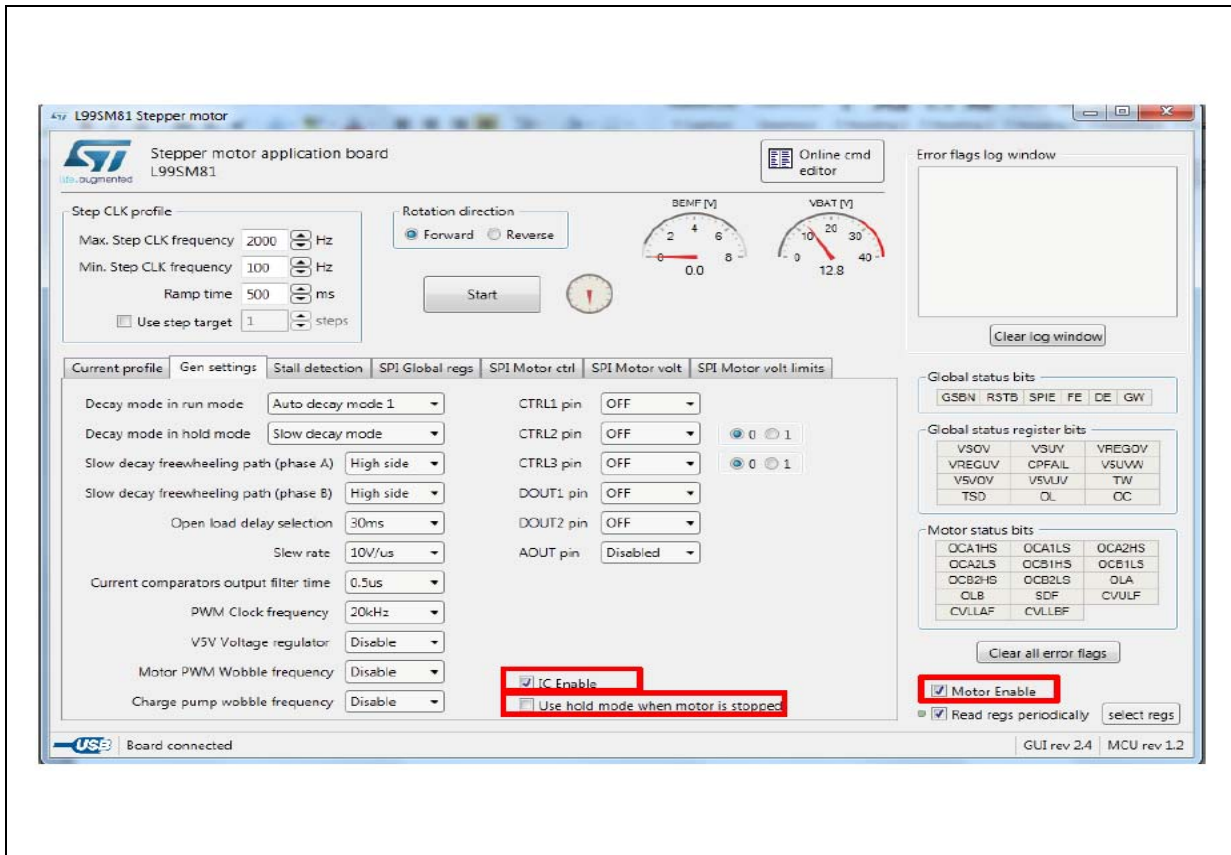
### 1.1.3 General settings

**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



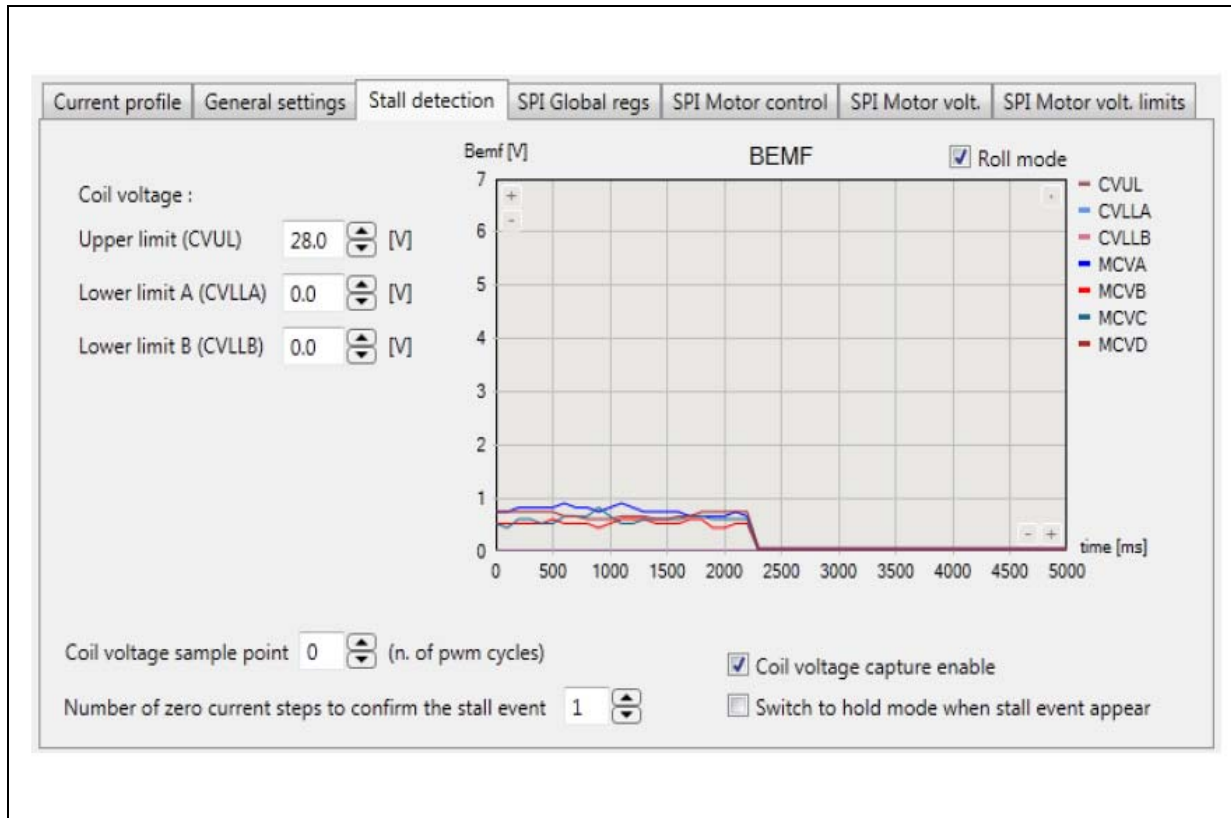
### 1.1.4 Stall detection

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

Current profile		General settings		Stall detection		SPI Global regs		SPI Motor control		SPI Motor volt.		SPI Motor volt. limits			
<b>GSR</b> [0x0000]				<b>MSR</b> [0x0000]				<b>GCR1</b> [0x0020]				<b>GCR2</b> [0x0000]			
15	0	VSOV		15	0	OCA1HS		15	0	CPWBE		15	0	-	
14	0	VSUV		14	0	OCA1LS		14	0	MWBE		14	0	-	
13	0	VREGOV		13	0	OCA2HS		13	0	-		13	0	-	
12	0	VREGUV		12	0	OCA2LS		12	0	AOUT1		12	0	-	
11	0	-		11	0	OCB1HS		11	0	AOUT0		11	0	DOUT11	
10	0	CPFAIL		10	0	OCB1LS		10	0	VSVE		10	0	DOUT10	
9	0	V5UVW		9	0	OCB2HS		9	0	-		9	0	-	
8	0	V5VOV		8	0	OCB2LS		8	0	-		8	0	DOUT21	
7	0	V5VUV		7	0	OLA		7	0	-		7	0	DOUT20	
6	0	TW		6	0	OLB		6	0	-		6	0	-	
5	0	TSD		5	0	-		5	1	MX1		5	0	-	
4	0	OL		4	0	-		4	0	-		4	0	-	
3	0	OC		3	0	CVULF		3	0	MX2		3	0	-	
2	0	SDF		2	0	CVLLAF		2	0	MX31		2	0	-	
1	0	-		1	0	CVLLBF		1	0	MX30		1	0	-	
0	0	Parity		0	0	Parity		0	0	Parity		0	0	Parity	
adr=1h [R&C] [R]				adr=2h [R&C] [R]				adr=3h [W] [R]				adr=4h [W] [R]			

Read Device Info Regs

Figure 9. SPI Motor control

Current profile		General settings		Stall detection		SPI Global regs		SPI Motor control		SPI Motor volt.		SPI Motor volt. limits			
<b>MCR1</b> [0x3800]				<b>MCR2</b> [0xFC81]				<b>MCR3</b> [0x8031]				<b>MCREF</b> [0x0000]			
15	1	ME		15	1	FREQ1		15	1	CVE		15	0	HC3	
14	0	HOLDM		14	1	FREQ0		14	0	D5		14	0	HC2	
13	0	ASM2		13	1	FTOCE		13	0	D4		13	0	HC1	
12	0	ASM1		12	1	TBE		12	0	D3		12	0	HC0	
11	1	ASM0		11	1	FT1		11	0	D2		11	0	-	
10	0	SM2		10	1	FT0		10	0	D1		10	0	-	
9	0	SM1		9	0	SR1		9	0	D0		9	0	-	
8	0	SM0		8	0	SR0		8	0	SD2		8	0	-	
7	0	DIR		7	1	DMR1		7	0	SD1		7	0	-	
6	0	PH5		6	0	DMR0		6	0	SD0		6	0	-	
5	0	PH4		5	0	SDAFW		5	1	CVLUR1		5	0	-	
4	0	PH3		4	0	SDBFW		4	1	CVLUR0		4	0	CA3	
3	0	PH2		3	0	OLDLY		3	0	AHMSD		3	0	CA2	
2	0	PH1		2	0	DMH		2	0	-		2	0	CA1	
1	0	PH0		1	0	-		1	0	-		1	0	CA0	
0	0	Parity		0	1	Parity		0	1	Parity		0	0	Parity	
adr=5h [W] [R]				adr=6h [W] [R]				adr=7h [W] [R]				adr=8h [W] [R]			

Figure 10. SPI Motor voltage

Current profile		General settings		Stall detection		SPI Global regs		SPI Motor control		SPI Motor volt.		SPI Motor volt. limits			
<b>MCVA</b> [0x0000]				<b>MCVB</b> [0x0000]				<b>MCVC</b> [0x0000]				<b>MCVD</b> [0x0000]			
15	0	-		15	0	-		15	0	-		15	0	-	
14	0	-		14	0	-		14	0	-		14	0	-	
13	0	-		13	0	-		13	0	-		13	0	-	
12	0	-		12	0	-		12	0	-		12	0	-	
11	0	-		11	0	-		11	0	-		11	0	-	
10	0	CV9		10	0	CV9		10	0	CV9		10	0	CV9	
9	0	CV8		9	0	CV8		9	0	CV8		9	0	CV8	
8	0	CV7		8	0	CV7		8	0	CV7		8	0	CV7	
7	0	CV6		7	0	CV6		7	0	CV6		7	0	CV6	
6	0	CV5		6	0	CV5		6	0	CV5		6	0	CV5	
5	0	CV4		5	0	CV4		5	0	CV4		5	0	CV4	
4	0	CV3		4	0	CV3		4	0	CV3		4	0	CV3	
3	0	CV2		3	0	CV2		3	0	CV2		3	0	CV2	
2	0	CV1		2	0	CV1		2	0	CV1		2	0	CV1	
1	0	CV0		1	0	CV0		1	0	CV0		1	0	CV0	
0	0	Parity		0	0	Parity		0	0	Parity		0	0	Parity	
adr=9h <span style="border: 1px solid black; border-radius: 50%; padding: 2px;">R</span>				adr=Ah <span style="border: 1px solid black; border-radius: 50%; padding: 2px;">R</span>				adr=8h <span style="border: 1px solid black; border-radius: 50%; padding: 2px;">R</span>				adr=Ch <span style="border: 1px solid black; border-radius: 50%; padding: 2px;">R</span>			

Figure 11. SPI Motor voltage limits

Current profile		General settings		Stall detection		SPI Global regs		SPI Motor control		SPI Motor volt.		SPI Motor volt. limits			
<b>MCVLLB</b> [0x0024]				<b>MCVLLA</b> [0x0034]				<b>MCVUL</b> [0x0100]							
15	0	-		15	0	-		15	0	-		15	0	-	
14	0	-		14	0	-		14	0	-		14	0	-	
13	0	-		13	0	-		13	0	-		13	0	-	
12	0	-		12	0	-		12	0	-		12	0	-	
11	0	-		11	0	-		11	0	-		11	0	-	
10	0	CVLLB9		10	0	CVLLA9		10	0	CVUL9		10	0	CVUL9	
9	0	CVLLB8		9	0	CVLLA8		9	0	CVUL8		9	0	CVUL8	
8	0	CVLLB7		8	0	CVLLA7		8	1	CVUL7		8	1	CVUL7	
7	0	CVLLB6		7	0	CVLLA6		7	0	CVUL6		7	0	CVUL6	
6	0	CVLLB5		6	0	CVLLA5		6	0	CVUL5		6	0	CVUL5	
5	1	CVLLB4		5	1	CVLLA4		5	0	CVUL4		5	0	CVUL4	
4	0	CVLLB3		4	1	CVLLA3		4	0	CVUL3		4	0	CVUL3	
3	0	CVLLB2		3	0	CVLLA2		3	0	CVUL2		3	0	CVUL2	
2	1	CVLLB1		2	1	CVLLA1		2	0	CVUL1		2	0	CVUL1	
1	0	CVLLB0		1	0	CVLLA0		1	0	CVUL0		1	0	CVUL0	
0	0	Parity		0	0	Parity		0	0	Parity		0	0	Parity	
adr=Dh <span style="border: 1px solid black; border-radius: 50%; padding: 2px;">W</span> <span style="border: 1px solid black; border-radius: 50%; padding: 2px;">R</span>				adr=Ah <span style="border: 1px solid black; border-radius: 50%; padding: 2px;">W</span> <span style="border: 1px solid black; border-radius: 50%; padding: 2px;">R</span>				adr=Fh <span style="border: 1px solid black; border-radius: 50%; padding: 2px;">W</span> <span style="border: 1px solid black; border-radius: 50%; padding: 2px;">R</span>							

### 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 periodically

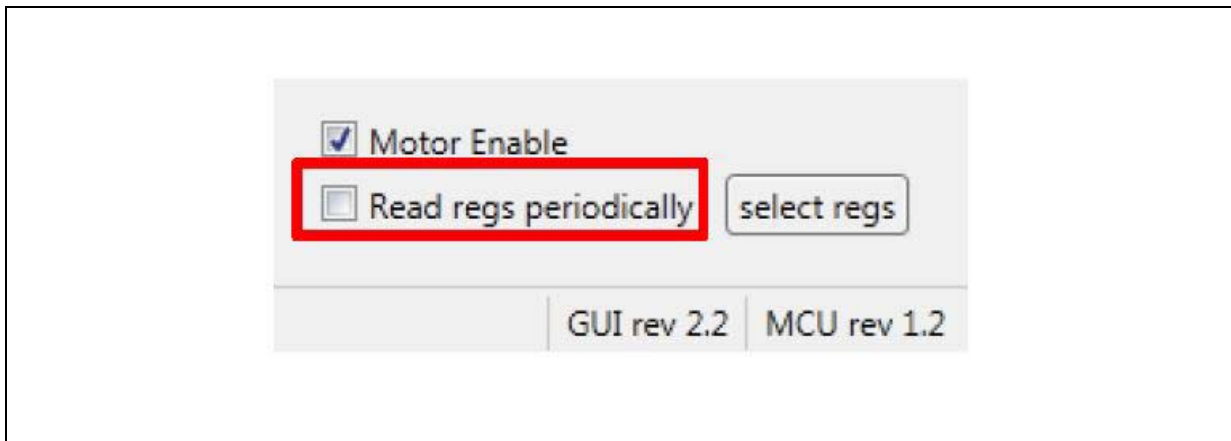
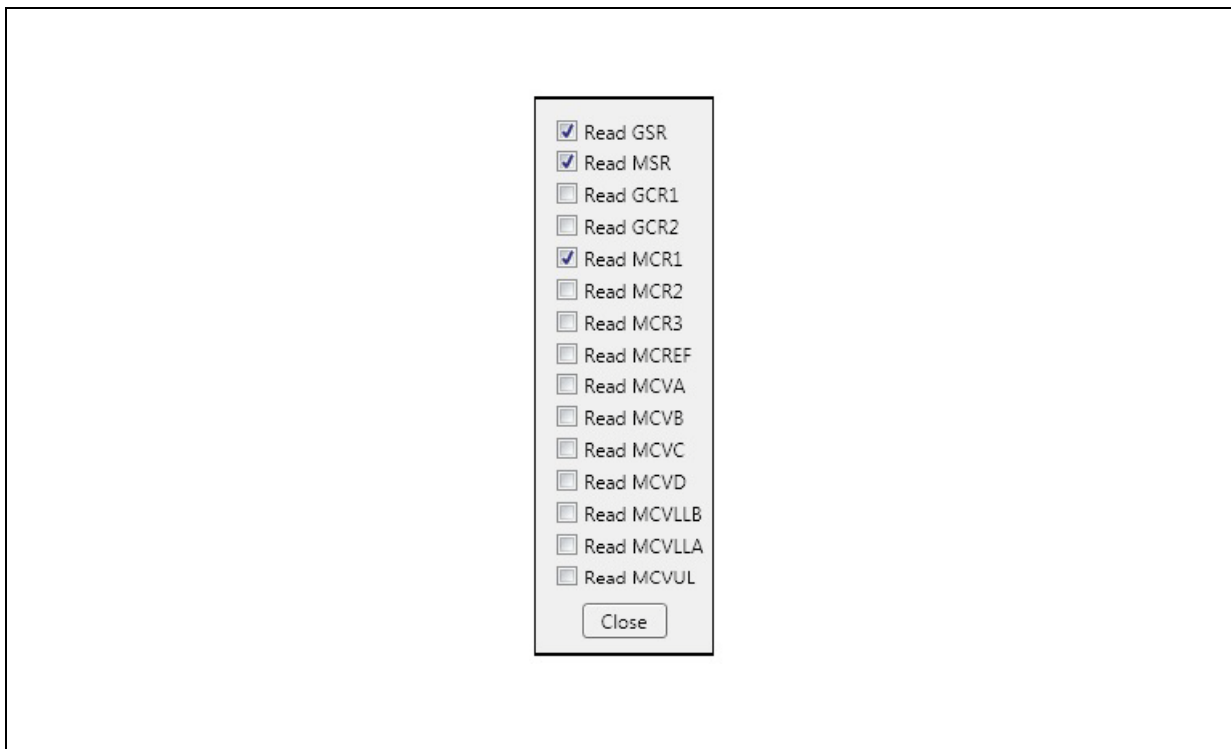


Figure 13. Registers selection

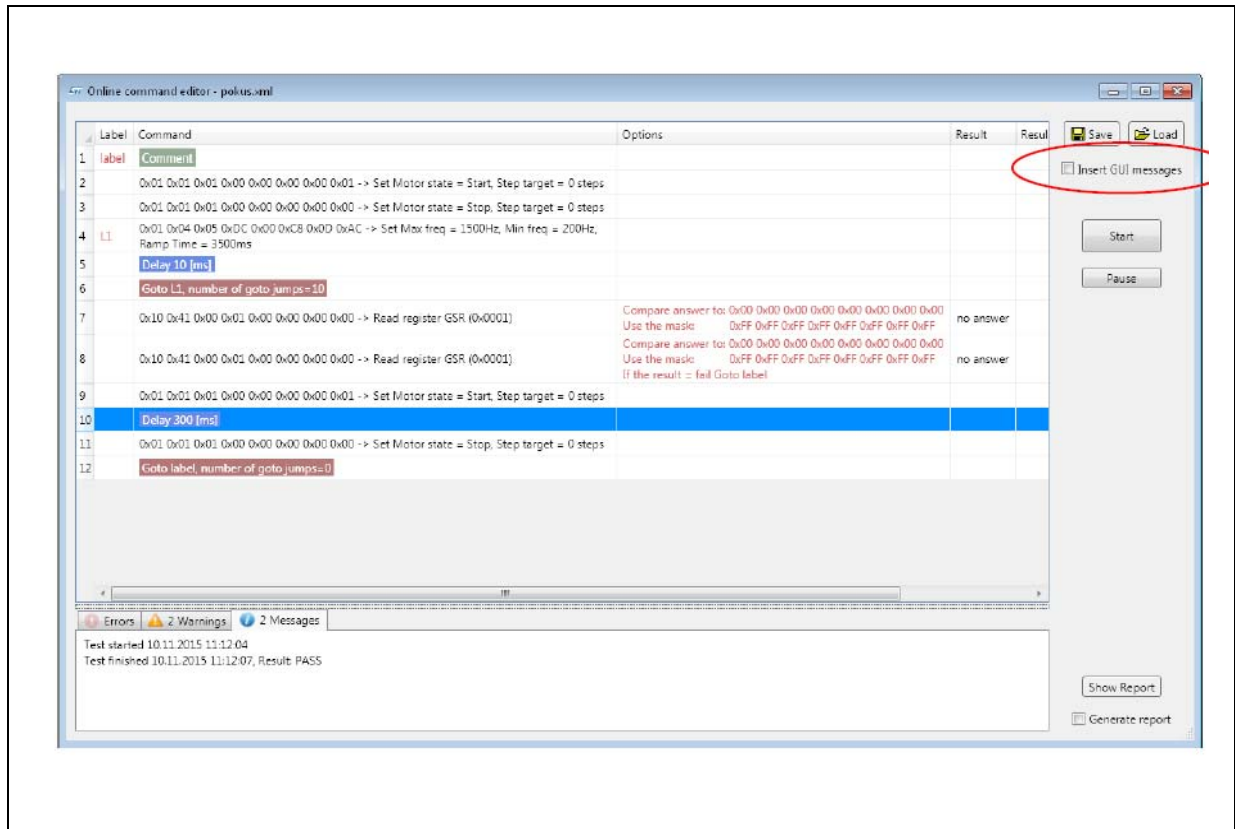


## 1.2 Online command editor

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:

8	0x10 0x41 0x00 0x01 0x00 0x00 0x00 0x00 -> Read register GSR (0x0001)	Compare answer to: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 Use the mask: 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF
---	---	---

→ 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:

9	0x01 0x01 0x01 0x00 0x00 0x00 0x00 0x01 -> Set Motor state = Start, Step target = 0 steps	Compare answer to: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 Use the mask: 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF If the result = pass Goto Label1
---	---	---

→ 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).

9	0x01 0x01 0x01 0x00 0x00 0x00 0x00 0x01 -> Set Motor state = Start, Step target = 0 steps	Wait for response
---	---	-------------------

## 2 Revision history

Table 1. Document revision history

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



**IMPORTANT NOTICE – PLEASE READ CAREFULLY**

STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST's terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers' products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2018 STMicroelectronics – All rights reserved