

# *STSPIN233 Click*

---

## **FIRMWARE GUIDELINES**

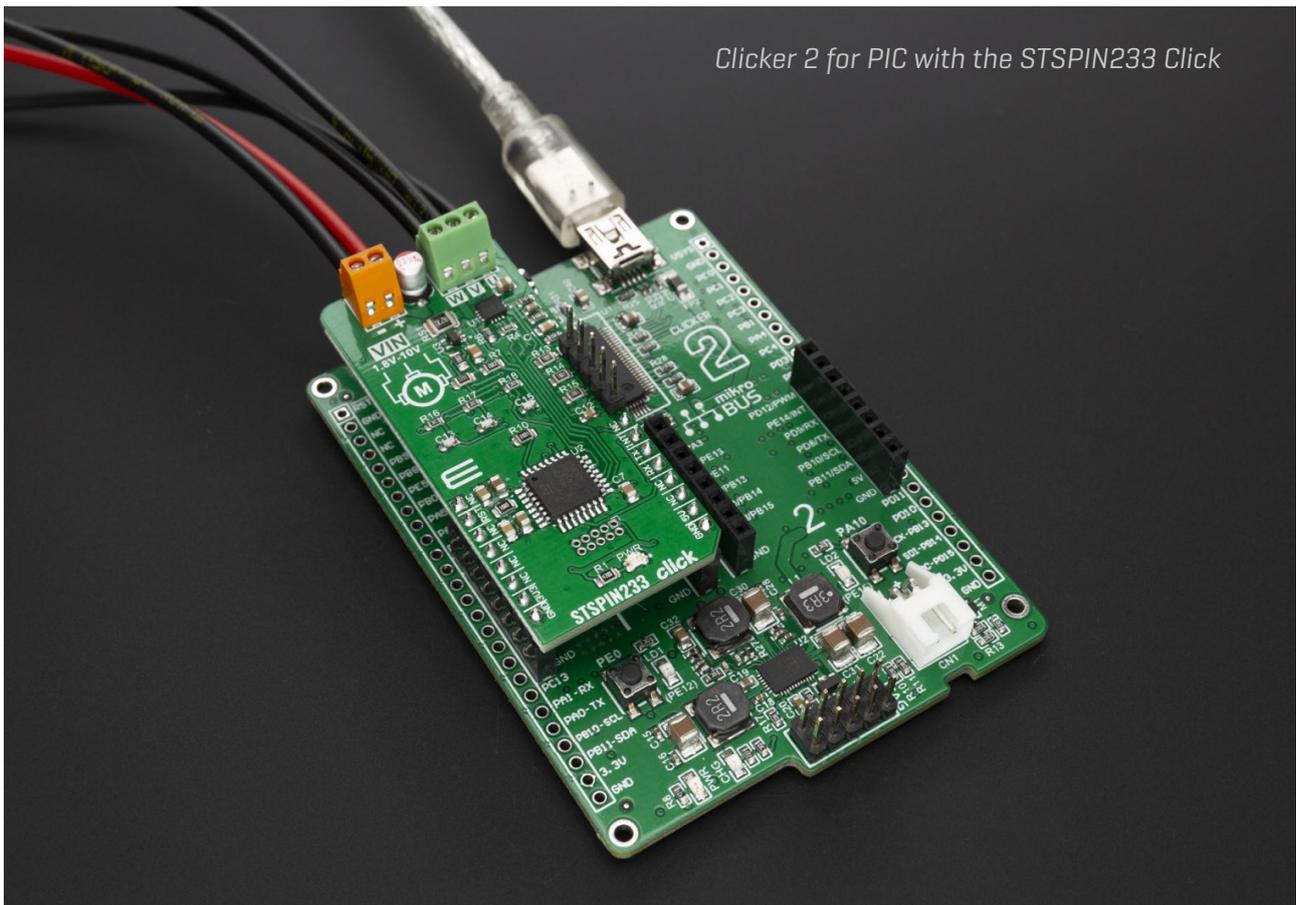
VERSION 1.0

**MIKROE**

# 1. Document scope

---

This document describes commands in detail, as well as the message responses, used in the communication with the STSPIN233 Click board. Some additional specifics are covered, such as the UART serial communication configuration, which is necessary for the motor control, firmware flow on the startup, as well as error and fault messages.



## 2. Supported Command Types

---

There are several different types of commands. They can be categorized as the **ACTION**, **SET**, **GET**, or **HELP** commands. In most cases, the same command may be used to perform a specific action and set up some parameters, but also to read back the parameters, or test the specific command to find out about the parameters that can be used with it. Also, commands acknowledge the execution status with short response messages.

### 2.1 Command Response String

Each command sends back a command response string, acknowledging the command reception. The command response string varies depending on the used command and the command type. In most cases, the command response string is just a status message which confirms the execution, but the command response message can also contain required information, such as in the case of **SET** and **HELP** type of commands.

## 3. Serial communication configuration

---

Send commands through the USART terminal to the board for testing it or directly from the main board. The STSPIN233 Click accepts commands, which can be sent over the UART interface pins of the mikroBUS™ or through COM port on a personal computer.

### *UART interface – configuration parameters*

**Baud rate:** 230400

**Parity:** NO

**Data bits:** 8

**Stop bit:** 1

**Flow control:** NONE

**End Character:** CR+LF

# 4. List of commands

---

## 1. BASIC MOTOR COMMANDS

- STARTM** – Start Motor
- STOPMT** – Stop Motor
- DIRECT** – Set Motor direction to CW or CCW
- SETSPD** – Set Motor Speed
- STATUS** – Get Status
- HELP** – Show HELP menu

## 2. MAIN MOTOR PARAMETERS

- INIREF** – Startup reference [0 - 4095]
- COMDEL** – Commutation delay
- POLESP** – Set the Motor pole pairs

## 3. PI REGULATOR PARAMETERS – SPEED LOOP

- KP-PRM** – Set PI proportional term
- KI-PRM** – Set PI integral term

At start-up [or after a “HELP” command is received] the list of commands is sent through UART. After this firmware goes into IDLE state waiting for a command.

## 5. Command description and default state

---

COMMAND	DESCRIPTION	DEFAULT
STARTM	Runs the motor	
STOPMT	Stops the motor	
DIRECT	Sets the motor direction	0 [CW direction]
SETSPD	Sets the motor speed value	1000 RPM
GETSPD	Returns the motor speed value	
STATUS	Returns the state machine status	
HELP	Sends the list of commands	
INIREF	Sets the start-up PWM duty cycle value	819
COMDEL	Sets the commutation delay	7
POLESP	Sets motor pole pairs	6
KP-PRM	Sets the proportional term – speed loop controller	500
KI-PRM	Sets the integral term – speed loop controller	250

### 5.1 STARTM command

- When a STARTM command is received an acknowledgement is sent.
- The motor goes into STARTUP state.
- After validation of the proper functioning of the Hall sensors, it goes into RUN state using the values loaded in the registers before the STARTM command reception.

**Example:**

STARTM

START MOTOR COMMAND RECEIVED !

## 5.2 STOPMT command

- When a STOPMT command is received an acknowledgment is sent.
- The motor goes into STOP state.
- All PWM signals are stopped, outputs are set to high impedance and the registers SPEED, INIREF, KP, KI and COMDEL are reset to default value.
- The DIRECTION and POLE PAIRS values are not reset.

### Example:

```
STOPMT
STOP MOTOR COMMAND RECEIVED !
```

## 5.3 DIRECT command

- When a DIRECT command is received an answer is sent and an additional input is expected [0 = clockwise direction, 1 = counter-clockwise direction]
- After 0 or 1 is received, the register is updated and the command can be sent while in STOP or in RUN state and it applies immediately.
- A number different from 0 or 1 is interpreted as 1.
- An acknowledgment is sent at the end of the communication.

### Example:

```
DIRECT
CW <0> CCW <1>:
1
OK
```

## 5.4 SETSPD command

- When a SETSPD command is received an answer is sent and an additional input is expected [target speed value in RPM].
- After a value is received, the register is updated.
- The command can be sent while in STOP or in RUN state and it applies immediately.
- An acknowledgment is sent at the end of the communication.
- The register value is reset to the default one after the motor returns into STOP state.

### Example:

```
SETSPD
Insert the value:
2000
OK
```

## 5.5 GETSPD command

- When a GETSPD command is received the firmware returns the instantaneous calculated speed value.

### Example:

```
GETSPD
```

```
Motor Speed: 2143 RPM
```

## 5.6 STATUS command

- When a STATUS command is received the firmware returns the present value of the state machine.

### Example:

```
STATUS
```

```
Status: RUN
```

## 5.7 INIREF command

- When INIREF command is received an answer is sent and an additional input is expected [PWM duty cycle at start-up before speed loop activation].
- Allowed maximum value is 4096. Duty-cycle = INIREF/4096. Default value [819] corresponds to 20% duty cycle.
- After a value is received, the register is updated.
- The command has no effect if sent while in RUN state.
- An acknowledgment is sent at the end of the communication.
- The register value is reset to the default one after the motor returns into STOP state.

### Example:

```
INIREF
```

```
Insert the value:
```

```
1000
```

```
OK
```

## 5.8 COMDEL command

- When a COMDEL command is received an answer is sent and an additional input is expected [commutation delay between Hall sensor signal and step change].
- The value is considered as DIGIT counts of the low frequency timer. The COMDEL value that corresponds to a delay of 1 PWM cycle may be calculated according to the formula:

**$SYSCLOCK\_frequency / (PWM\_frequency - 1) / LF\_timer\_PSC$**

- After a value is received, the register is updated.

- The command has no effect if sent while in RUN state.
- An acknowledgment is sent at the end of the communication.
- The register value is reset to the default one after the motor returns into STOP state.

**Example:**

COMDEL

*Insert the value:*

10

OK

## 5.9 POLESP command

- When a POLESP command is received an answer is sent and an additional input is expected [motor pole pairs].
- After a value is received, the register is updated.
- The command has no effect if sent while in RUN state.
- An acknowledgment is sent at the end of the communication.

**Example:**

POLESP

*Insert the value:*

7

OK

## 5.10 KP-PRM command

- When a KP-PRM command is received an answer is sent and an additional input is expected [value of the proportional term of the speed loop controller]
- After a value is received, the register is updated.
- The command can be sent while in STOP or in RUN state and it applies immediately.
- An acknowledgment is sent at the end of the communication.
- The register value is reset to the default one after the motor returns into STOP state.

**Example:**

KP-PRM

*Insert the value:*

400

OK

### 5.11 KI-PRM command

- When a KI-PRM command is received an answer is sent and an additional input is expected (value of the integral term of the speed loop controller).
- After a value is received, the register is updated.
- The command can be sent while in STOP or in RUN state and it applies immediately.
- An acknowledgment is sent at the end of the communication.
- The register value is reset to the default one after the motor returns into STOP state.

**Example:**

```
KI-PRM
Insert the value:
200
OK
```

### 5.12 Message Error

- When a non-valid command is sent, an error message is returned and the command is ignored.

**Example:**

```
TEST
ERROR - PLEASE TYPE AGAIN !
```

### 5.13 Fault

- If a fault condition occurs [i.e. motor stall] the firmware returns an error message with the motor status.
- The motor goes into STOP state.

**Example:**

```
Status: SPEEDFBKERROR
```

### 5.14 Overcurrent Error

A complete list of the errors caused by over current that may be returned is:

**SPEEDFBKERROR** - Hall sensor reading error (motor stall)

**LF\_TIMER\_FAILURE** - If timer overflow

**OVERCURRENT** - Overcurrent event

# MIKROE

[mikroe.com](http://mikroe.com)



[mikroe.com/mikrobus](http://mikroe.com/mikrobus)



[libstock.com](http://libstock.com)

---

## DISCLAIMER

All the products owned by MikroElektronika are protected by copyright law and international copyright treaty. Therefore, this document is to be treated as any other copyright material. No part of this document, including product and software described herein, may be reproduced, stored in a retrieval system, translated or transmitted in any form or by any means, without the prior written permission of MikroElektronika. The document PDF edition can be printed for private or local use, but not for distribution. Any modification of this document is prohibited.

MikroElektronika provides this document 'as is' without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties or conditions of merchantability or fitness for a particular purpose.

MikroElektronika shall assume no responsibility or liability for any errors, omissions and inaccuracies that may appear in this document. In no event shall MikroElektronika, its directors, officers, employees or distributors be liable for any indirect, specific, incidental or consequential damages (including damages for loss of business profits and business information, business interruption or any other pecuniary loss) arising out of the use of this document or product, even if MikroElektronika has been advised of the possibility of such damages. MikroElektronika reserves the right to change information contained in this document at any time without prior notice, if necessary.