

Software manual

Version. 1.2

1. Available commands

Types of commands:

- Global – Can be called everywhere. (0)
- Group – Can be called just when the correct command group is selected. (0)

After startup of the MCU no command group is selected. To select command group type exact name of the group. After hitting “enter” the command group will be selected.

>> [command group]

After command group selection you can type desired command with argument (if specified). To exit the command group use the “back” command (0).

>> [command group] >> [command] [argument]

Global commands

- **clear** – Clears output terminal.
- **back** – Exits current command group.
- **status** – Prints out selected mode. Current(real) RPM and PWM duty cycle.
- **help** – Prints out all available commands. [Not implemented]

Command groups

CONTROL <ct>

- **mode <new mode>** – If second argument is given(optional: mode), then it specifies desired mode. Options:
 - **reg** – regulation mode
 - **man** – manual mode
- **start** – Starts the motor from 0 RPM. If *regulation* mode is selected – motor spins with 100% duty up to SETPOINT_DEFAULT rpm (config.h) after which the regulation starts.
- **stop** – Stops the motor.
- **duty <new duty>** – Prints out currently set duty. If second argument is given(optional: number), then it specifies new duty. (*man mode*)
- **rpm <new rpm>** – Prints out currently set RPM. If second argument is given(optional: number), then it specifies desired RPM. (*reg mode*)
- **s <n>** – Sets speed saved in: (#define SPEEDS { 1440, 1596, 2100, 2800 } – config.h), where n is array index + 1. (*reg mode*)
- **“+”** – Increases speed from current speed index n. (*reg mode*)
- **“-”** – Decreases speed from current speed index n. (*reg mode*)

Brake <brake>

- **on** – Brake is activated.
- **off** – Brake is deactivated.

Direction <dir>

- **cw** – Motor is set to turn clockwise.
- **ccw** – Motor is set to turn counter clockwise.
- **chDir** – Changes motor turn direction.

2. Configuration

Configuration is done by modifying **config.h** file.

2.1.PID tuning

Tuning is done by gains (gains include sampling time):

- **Kp** (proportional)
- **Ki** (integral)
- **Kd** (derivative)

ARW can be turned on/off by writing 0/1 to **ARW_EN** macro. Default setpoint value is defined by **SETPOINT_DEFAULT** macro.

2.2. Start boost

To start the regulation process after the rotor rpm reaches RPM(**MOTOR_RPM_REG_START** [float]) write 1 to **START_BOOST_EN**. This enables the motor to have higher initial torque but also causes rpm overshoot.

2.3.User rpm min and max

When using the **rpm <new rpm>** command the input interval is bounded by **RPM_MIN** and **RPM_MAX** macros.

2.4.User speed definitions

The **s <n>** command uses user defined speeds by macro **SPEEDS**, which holds array of RPMs .

2.5.Incremental encoder definition

Depending on type of incremental rotary encoder you need to write correct number of pulses for single revolution to **ENC_N_PULSES** macro.

2.5.1. Encoder signal watchdog

When the regulation is turned on, the PID regulator needs to always get the correct RPM from the encoder. In case of a failure (mechanical or electrical) the signal may be not present, the watchdog functionality halts the regulation and stops the motor after not receiving new pulse for defined time.

Writing 1 to **ENC_WDG_EN** enables this functionality and **ENC_WDG_MS** sets timeout [ms] after which the motor is stopped.

3. Software limitations and known issues

- Missing help command
- PWM frequency cannot be changed (20 kHz)
- Regulation period cannot be changed (2 ms)