Controlling SCF4 with G-code commands

FIZ+ controller expects industry-standard G-code formatted commands. Commands are text-based and human-readable. Some highlights and behavior notes:

- Controller has limited buffer/cache functionality, a user should rely on single command execution at a time. Each command ends with a response
- Controller never sends commands back to the computer without asking
- `G1` linear interpolation is not supported, but it can be implemented by setting motion speeds and feeds manually

G-code introduction

G-code is the common name for the most widely used numerical control (CNC) programming language. It tells the motors where to move, how fast to move, and what path to follow. G-code has many variants and each machine has unique behavior implementation. More details on G-code can be found at https://en.wikipedia.org/wiki/G-code

Index of supported G-code commands

Below is a list of supported commands. For a detailed explanation and usage examples see next section.

Version strings and identification commands

Command	Description	Returns

`$S`	Return version, sn, model and brand strings	See details below

Controller initialization

Command	Description	Returns
`$B1`	Reset motor driver	`OK`
`$B2`	Reset and initialize motor driver	`OK`
`$B3`	Reset STM32F103 processor	`OK`

Motion commands

Command	Description	Returns
`G0`	Rapid positionin	`OK`
`G4`	Wait / stall	after complete `OK`
`G90`	Absolute programming mode	`OK`
`G91`	Incremental programming mode	`OK`
`G92`	Set position	`OK`

Miscellaneous function

Command	Description	Returns
`M0`	Compulsory stop	`OK`
`M7`	DN function with filter (VIS)	`OK`
`M8`	DN function no filter (IR + VIS)	`OK`
`M230`	Set normal move	`OK`
`M231`	Set normal + forced move	`OK`
`M232`	Set PI low/high detection voltage	`OK`
`M234`	Set motor and DN drive current	`OK`
`M235`	Set motor idle current	`OK`

`M238`	PI LED on (some lenses leak light into sensor)	`OK`
`M239`	PI LED off	`OK`
`M240`	Set motor drive speed	`OK`
`M245`	Drive AUX output to low (high resistance output)	`OK`
`M246`	Drive AUX output to high (low resistance output)	`OK`
`M247`	Read power supply voltage value	`ADC=xxxx`

Advanced function

Command	Description	Returns
`M241`	Dividing setting 1	`OK`
`M242`	Pulse generation control 1	`OK`
`M243`	Microstepping	`OK`
`M244`	Dividing setting 2	`OK`

Status commands

Command	Description	Returns
`!1`	Return motor pos, limit sw, moving status	`4000, 20000, 0, 0, 0, 0, 0, 0`

Controlling SCF4 with G-code commands explanation and usage examples

Axis description

Driver support bipolar stepper motors for 3 axes. They are named A/B/C and usually have defined function:

- A - zoom
- B - focus
- C - aperture

`$S` - Return version string

Command returns version string concentrated into single line `EVB.1.0.2, SCF4-M RevB, Kurokesu, 5DDFF39-3739584E-xxxxxxxx` received information is comma separated:

- `EVB.1.0.2` - Module firmware version
- `SCF4-M RevB` - Module PCB revision
- `Kurokesu` - Manufacturer Brand
- `5DDFF39-3739584E-xxxxxxxx` - Unique serial number

`G0` - Rapid positioning

The main motor drive command. It moves motor specified number steps. Any or all axes can be specified at the same time. Minimal step size is 1. `G0` command replies as soon as the command is parsed and does not wait until motors stop.

- `G0 A100` - drives A-axis 100 steps
- `G0 A-100` - drives A-axis -100 steps
- `G0 A100 B-100 C1000` - drives all three axes by specified step count

`G4` - Wait

Instructs controller to delay (stall) defined interval in milliseconds [ms]. This is an only command which sends response not after parsing command but after prolonged execution time. During this time command parser is blocked and the only way to complete this command is to wait until it finishes.

`G90` - Absolute coordinates

Switch to absolute coordinate programming mode. In this mode, the range is limited by the internal 16-bit motor register and can be operated in a range of 0 ~ 65535.

`G91` - Relative coordinates

Switch to relative coordinate programming mode. Motors can be instructed to turn positive or negative 0 ~ 65535 steps (16-bits). When the counter exceeds this range, it will overflow. And will not affect motor movement.

`G92` - Set position

The controller sets an internal counter to the specified value.

- `G92 A0` - Set A axis counter to 0
- `G92 A100` - Set A axis counter to 100
- `G92 A100 B1000 C200` - Set all axis counters to specified counter values

`M0` - Compulsory stop

If the controller is not in G4 delay mode it will instruct motors to stop moving

`M7` and `M8` - Infrared filter switch commands

Instructs controller to shift filter to one of two fixed positions

`M230` and `M231` - normal / forced move

- Default is normal mode in which motors turn instructed step count and stop after
- In forced mode controller does not stop turning motor after specified step count is reached, instead seeks corresponding port PIN_x status state change. Internal step counter may overflow. Motion is stopped when input status changes, `M0` or reset command is issued. PIN_x is connected to internal 12bit Analog to Digital Converter (ADC), lower and upper thresholds can be controlled with `M232` command

`M232` - Set PI low/high detection voltage

Set 12bit comparator values for limit switch detection inputs. Upper voltage is 3.3V, lower - 0V

- A, B, C - set lower threshold values
- E, F, G - set upper threshold values
- `M232 A1242 E2483` - If the input voltage is below ~1V - set flag to `0`. If the value is above ~2V, set flag to `1`. Everything in between - is hysteresis and does not change the flag. Used to debounce and filter the input signal

`M234` - Set motor and DN drive current

Sets motor and coil drive current. Expects 8-bit value.

- `M234 A120 B120 C120 D80` - Set A, B, C stepper motors strength to 120 and IR filter driver to 80

`M235` - Set motor idle current

When motors are not moving, idle current can be reduced to eliminate heating and save energy. It is not advised to completely turn off motors as the driver can lose micro-step holding position

- `M234 A50 B50 C50` - Sets A/B/C motors holding current to value 50

`M238` and `M239` - control output pins

Opto-interrupter LEDs can be controlled by connecting them to PIS_AB and PIS_C pins. They can stay in either state, but some lenses in IR mode can leak light into the sensor so after referencing/calibration procedure is complete it is advised to switch them off

`M240` - Set motor drive speed

Each motor can drive at different speeds. Speed is 16-bit register. Register specifies internal timing interval, the lower value is the faster pulse rate will be.

- `M240 A800 B800 C1200` - Set motion speeds for each axis to corsponding value

`M241`, `M242`, `M243`, `M244` - advanced timing registers

- These registers control motor driver timing and should not be altered normally, however in some cases it can be necessary to make adjustments. Please seek the support if you

have any questions

`M245` - Drive AUX output to low

Command drives auxiliary output to high resistance state. This is default value after power up or CPU reset.

> Feature added in EVB.1.1.0 firmware. Used in **SCF4-L090 revC** and **SCF4-L087 revB** boards.

`M246` - Drive AUX output to high

Command drives auxiliary output to low resistance state.

> Feature added in EVB.1.1.0 firmware. Used in **SCF4-L090 revC** and **SCF4-L087 revB** boards.

`M247` - Read power supply voltage value

Command reads power supply voltage. Responds with 12bit decimal value.

Reference voltage is Vref=3.3

Symbol	Value	Units	Note
Vref	3.3	V	
ADC_TOP	4096		12bit max ADC value
SCALE	0.5		Boards uses resistance divider
ADC			Value returned by command

Conversion to voltage: Vin = ADC/4096*Vref/DIV

> Feature added in EVB.1.1.0 firmware. Used in **SCF4-L090 revC** and **SCF4-L087 revB** boards.

`! 1` - Return status

Command returns 9 values indicating:

- `4000, 20000, 0, 0, 0, 0, 0, 0` - example output
- A Motor position counter
- B Motor position counter
- C Motor position counter
- A Limit switch status
- B Limit switch status
- C Limit switch status
- A Moving status
- B Moving status
- C Moving status

Revision #18
Created 27 July 2020 17:59:02 by Saulius
Updated 1 July 2022 11:34:41 by Saulius