# LUMS Stepper – universal 3-axis stepper motor controller

Kacper Oreszczuk

# 1 Motivation

Stepper motors are commonly used in laboratory. They are powering x-y-z translation stages, tunable filters, polarisation optics and many other devices. The goal of this project is to replace commercially available drivers with a cheaper, more reliable and more universal solution.

### Cost:

Commercially available stepper motor drivers designed for laboratory use can reach prices of a few hundred US Dollars for a single axis. Proposed solution costs about $30 per axis with small labour cost.

### Quality:

Personal experience in our laboratory revealed problems with microstep precision in drivers supplied by one of the leading laboratory equipment manufacturers. I have tested thoroughly stepper motor chip before choosing it for this project.

### Features:

Custom solution allowed me to implement important additional features. The most important is automatic compensation of mechanical hysteresis.

# 2 Hardware design

Single device is able to control three bipolar stepper motors with 1/64 microsteps precision. Maximum current is about 1.5A per motor and the supply voltage must be between 6V and 30V. Low level motor driving operation is performed by*AMIS-30543* driver on *Pololu* breakout board. Three such drivers are connected to *NUCLEO32* board with *STM32L432KC* microcontroller. Nucleo board connected via USB to computer provides virtual serial port for communication and power for logic layer of device. Separate power connector is provided on main PCB for motor operation and three high-density D-SUB15 female connectors (ICD15S13E4GV00LF) for motors. Full schematics of the device is presented in Figure 2. PCB board is double-sided. See Figure 3 for PCB layout. This design, which effectively consists of 8 elements makes the project easy and quick to deploy in several copies. The driver is confined in 3D-printed case (Figure 4) protecting it and allowing mounting to optic table. Picture 1 shows assembled device with the top lid removed. Design files for PCB and case along with source code for microcontroller are available at project page: `github.com/kacperoreszczuk/lums_stepper/`.

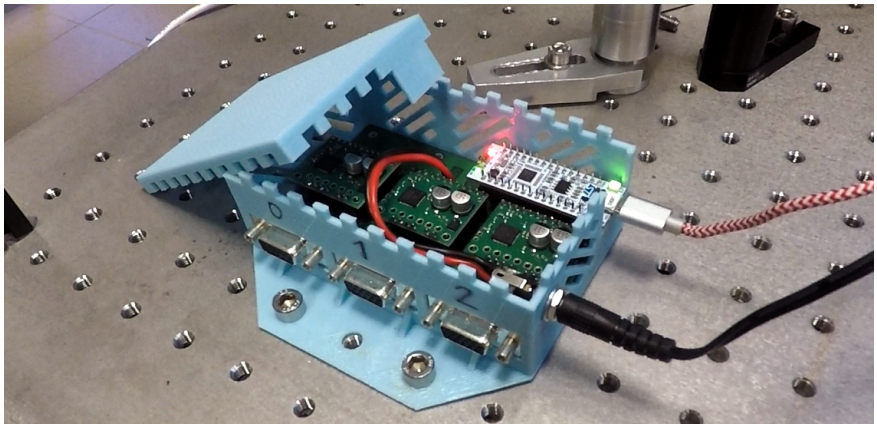

***Figure 1:*** *Assembled LUMS Stepper.*

# 3 Software and protocol

Driver has two basic operation modes - movement with given velocity or movement to given position. In either of the cases the movement of motor is smoothed at starting and stopping. For user convenience speeds and positions are expressed in real units (for example millimeters in case of linear travel stage), not motor steps. End switches (limit switches) of different types are supported by this driver. They provide additional safety, as the driver will stop at the edge of the movement range of, for example, linear translation stage. Limit switches are also used to automatically calibrate absolute position of the stage.

When the mechanical setup of driven actuator is not completely rigid, the mechanical hysteresis can occur. Due to this effect real position of the actuator depends on the direction from which set position is approached. In this project user can specify the amplitude od mechanical hysteresis. Algorithm will track previous movements and compensate the offsets in a manner transparent for the user.

User can communicate with LUMS Stepper via virtual serial port provided by Nucleo board. Protocol of communication is based on single line plain-text commands initiated from computer and on single line reply from driver. For protocol description please refer to Table 1. Full list of accepted commands is presented in Table 2

| Command: | ID header value \r |
|---|---|
| ID | Single digit number – number of motor, counted from 0. Defaults to 0 if not specified. |
| header | Two lower case characters specifying command type (see table 2). |
| value | Numerical argument (optional depending on command type). Defaults to 0 if not specified. |
| \r | Carriage return as endline character. Other whitespace before, after or between command parts is ignored. |
| Reply: | header value \r |
| header | Driver returns header after every command |
| value | Optional depending on command type |
| Error reply: | ?\r |

***Table 1:*** *Protocol of communication between PC and LUMS Stepper.*


## Acknowledgements

This work was supported by the Polish Ministry of Science and Higher Education in years 2017-2021 as research grant "Diamentowy Grant" under decision 0149/DIA/2017/46.

| Command | Abbreviation origin | Example | Description |
|---|---|---|---|
| id | ID | id | Returns unique ID of device, between 101 and 199. |
| ac | Axis Count | ac | Returns number of supported motors. |
| sc | Set Current | 0sc600 | Sets motor current, in milliamperes. Rounds down to nearest value: 132, 245, 355, 395, 445, 485, 540, 585, 640, 715, 780, 870, 955, 1060, 1150, 1260, 1405... (mA) |
| ss | Set Step | 0ss0.005 | Defines the size of motor full step in user units, for example in millimetres or degrees. |
| sh | Set Hysteresis | 0sh0.001 | Sets mechanical hysteresis compensation distance. Algorithm applies corrections to target positions when necessary (f. eg. when changing direction). |
| sm | Set Max Velocity | 0sm1.5 | Sets maximum allowed velocity in move velocity mode, in user units. |
| sv | Set Velocity | 0sv1 | Sets velocity in move absolute mode, in user units. |
| sa | Set Acceleration | 0sa0.5 | Sets time that it takes to accelerate from zero to move absolute velocity. |
| sl | Set Limit | 0sl2 | Sets limit switch type. 0 – no limit switch, 1 - active limit switch (active high). 2/3 – mechanical, default NC/C. 4/5 similarly to 2/3, but enabled only for homing. |
| sr | Set Reversed | 0sr1 | Reverses direction of motor rotation and swaps limit switches. |
| so | Set Offset | 0so4.2 | Specifies the position of the limit switch for homing. |
| hm | HoMe | 0hm | Starts homing procedure. If limit type is set to zero, then it only corrects current position to be zero. |
| mv | Move Velocity | 0mv0.5 | Sets target velocity. Driver will smoothly increase or decrease speed to given value. |
| ma | Move Absolute | 0ma3.2 | Sets target position. Driver will smoothly start and halt motor to achieve goal position. |
| mr | Move Relative | 0mr0.1 | Changes target position by given value, even if target position was not achieved yet. In „move velocity" mode sets target relative to current position. |
| tp | Tell Position | 0tp | Returns current position of given axis. |
| ts | Tell Status | 0ts | Returns status of current axis. 0 – stopped, 1 – move velocity, 2 - move position, 3 – homing. |
| ta | Tell All | ta | Returns current positions of all axes, separated by spaces. Next, after another space, returns statuses of all axes in single joined string. |

***Table 2:*** *Commands in communication protocol.*

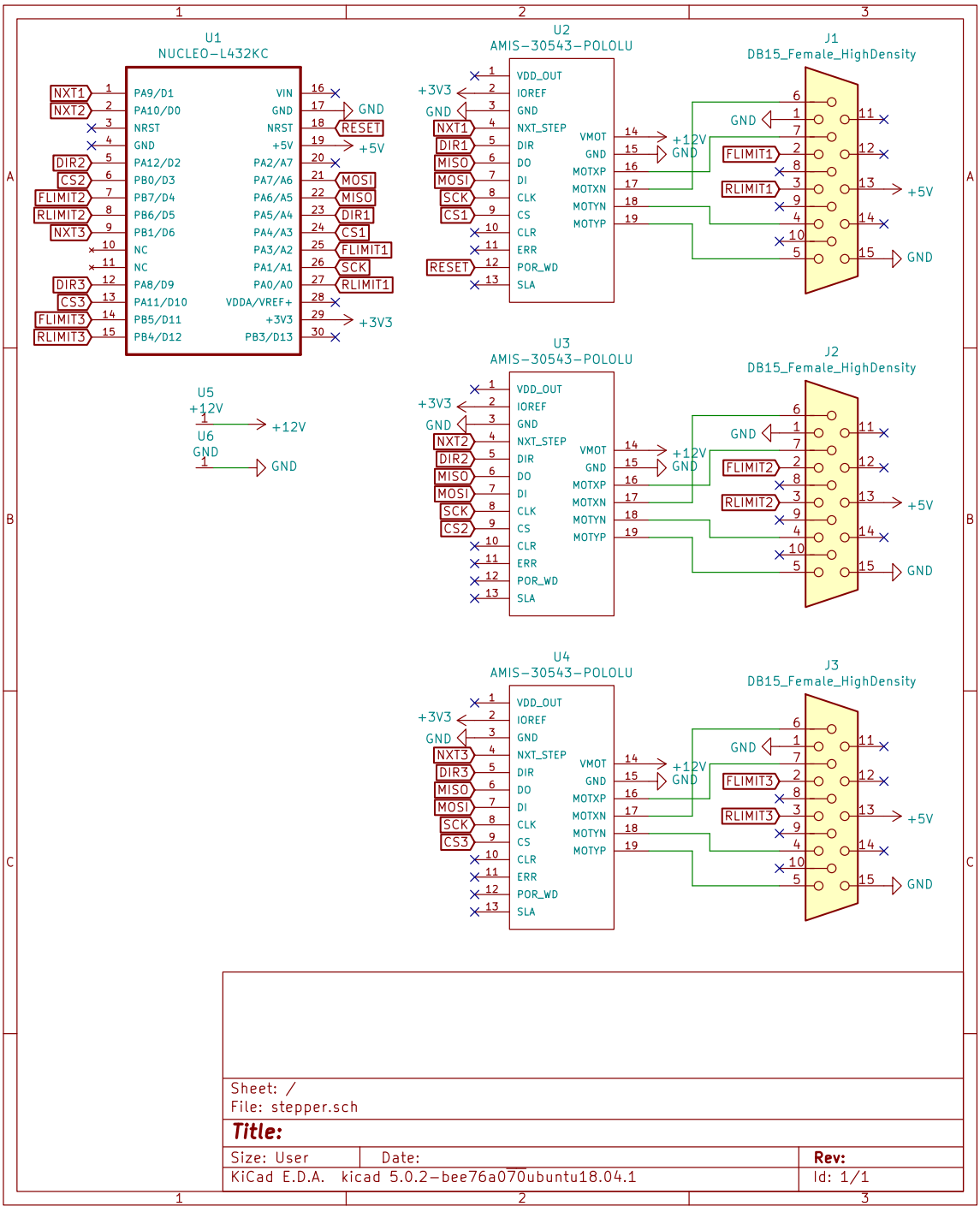


***Figure 2:*** *Schematics of the LUMS stepper.*

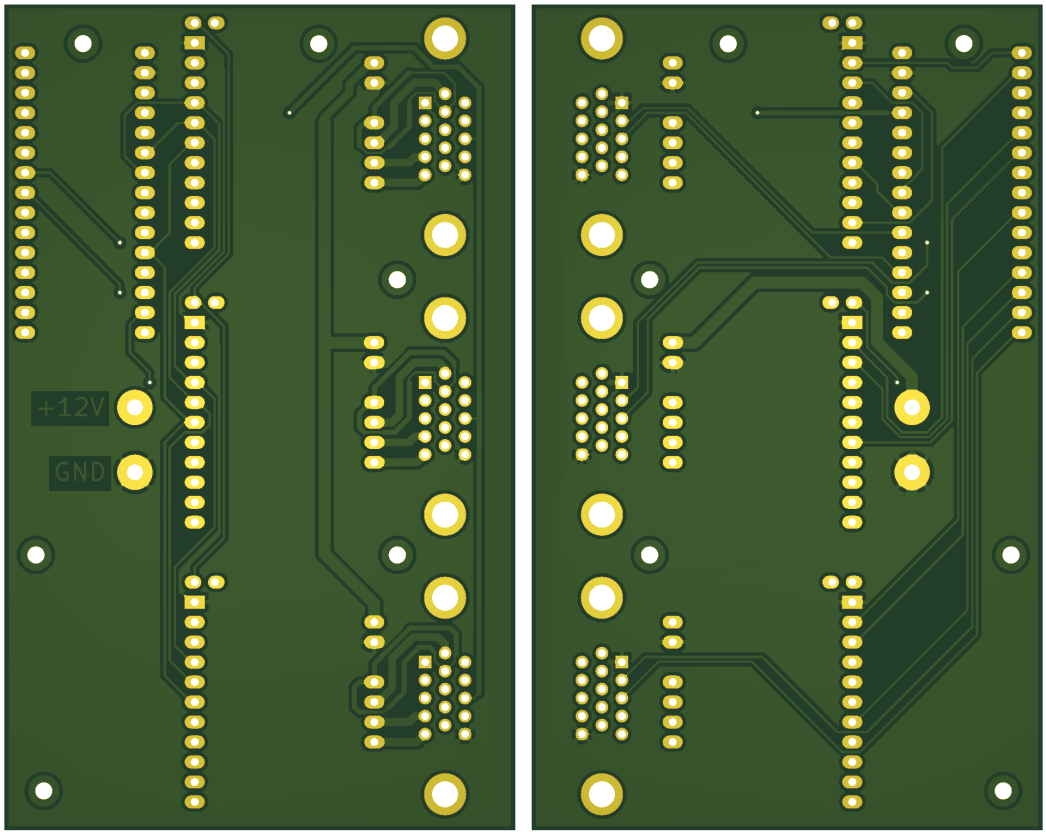


***Figure 3:*** *PCB design. View from top and bottom, respectively.*

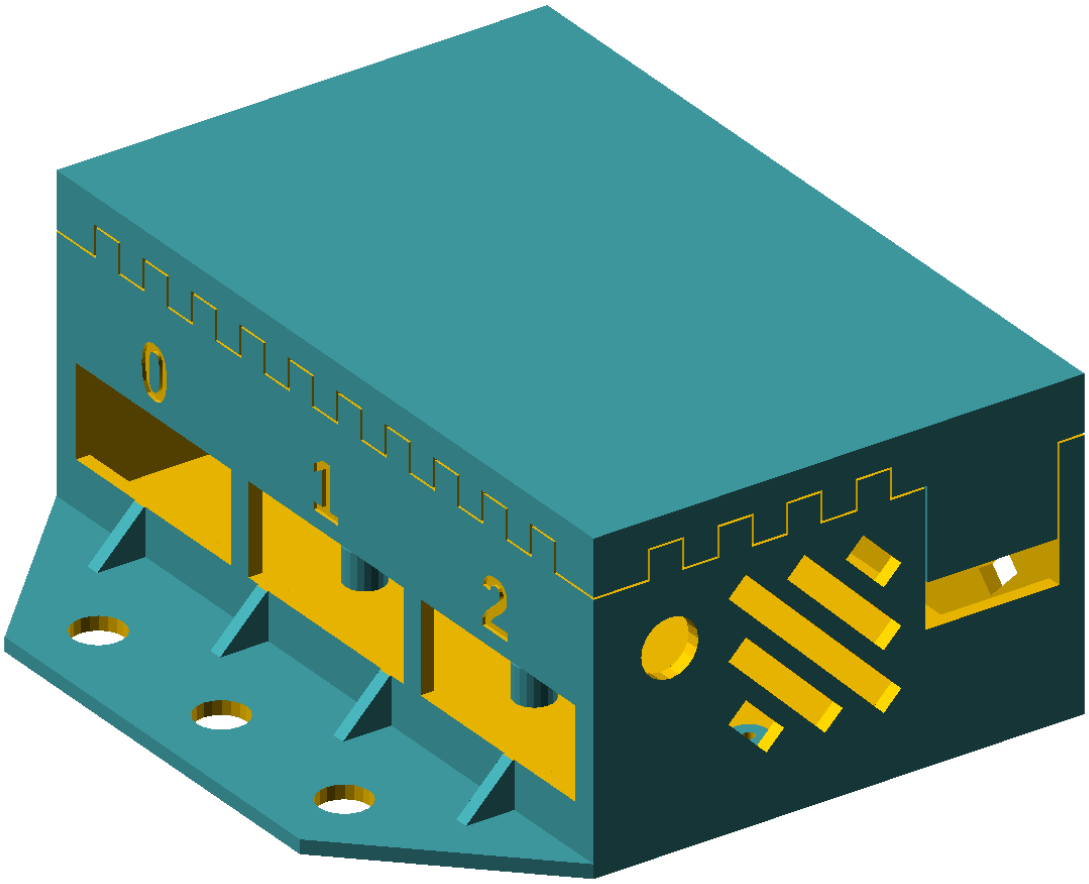


***Figure 4:*** *Case, model for 3D printer.*