Technical Specifications

RobotShop Inc. March 8, 2021

NDA

The information contained in this document is shared with potential candidates for the proper evaluation of the “project” and must not be distributed or shared with other parties.

Overview

This PCB project is part of a larger project which has the mandate of developing a new line of all-in-one, modular, user-programmable actuators. There are currently three sizes of actuators in development which all make use of hybrid / bipolar stepper motors and gearing of different sizes. A goal is to design the PCBs in such a way that a single design can be shared between all three actuators.

This project has been under development for almost two years and the electromechanical side (selection of gearing, motors, mechanical frame, electronic mounting etc.) has been designed, sourced and prototyped. A second prototype of the electronics had been underway but was not finished when the lead EE on the project left the company. It is therefore important that the PCB design fit the existing physical prototypes, and the design be completed within a few months.

These actuators will be daisy chained on a CAN BUS so wiring (power and communication) connects from one actuator to the next so that each actuator does not need to be individually connected to a central “hub” or controller; the first in the bus receives communication input via USB, which is then shared along the bus.

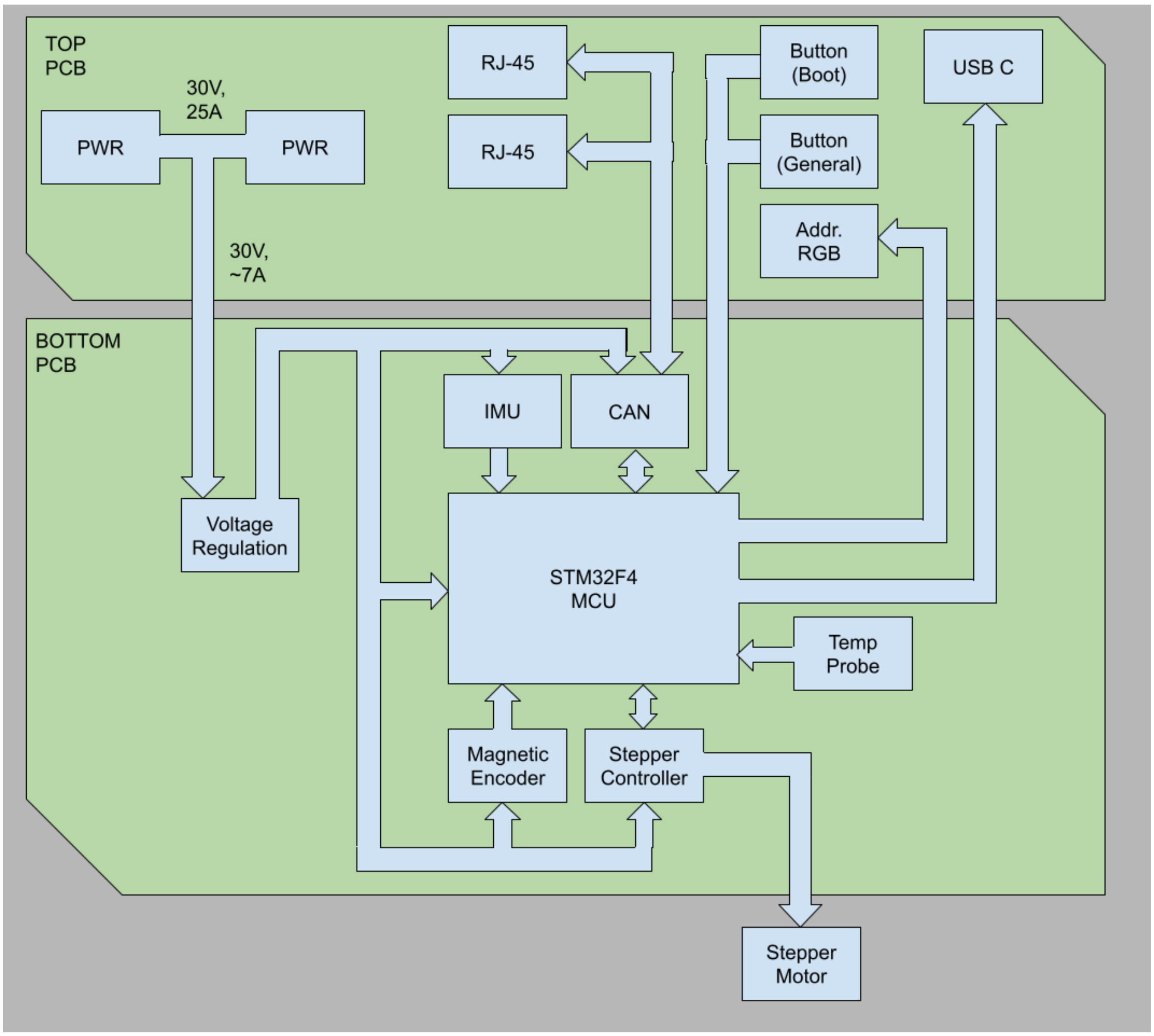
Input Specifications

- Vin (max): Charged 8S LiPo
- Vin (nom): 30V (custom wall adapter with 30V, 30A output)
- Vin (min): Discharged 8S LiPo
- Current (max continuous) to motor: 6A
- Current on power bus (max): ~25A*
- Communication: CAN BUS & USB

* Since the actuators are daisy chained (one to the next to the next), the sum total of all the current required by all actuators in the BUS will need to pass through the first actuator, therefore the design of the top PCB power traces needs to accommodate this maximum current since any actuator can be used as the “first” actuator.

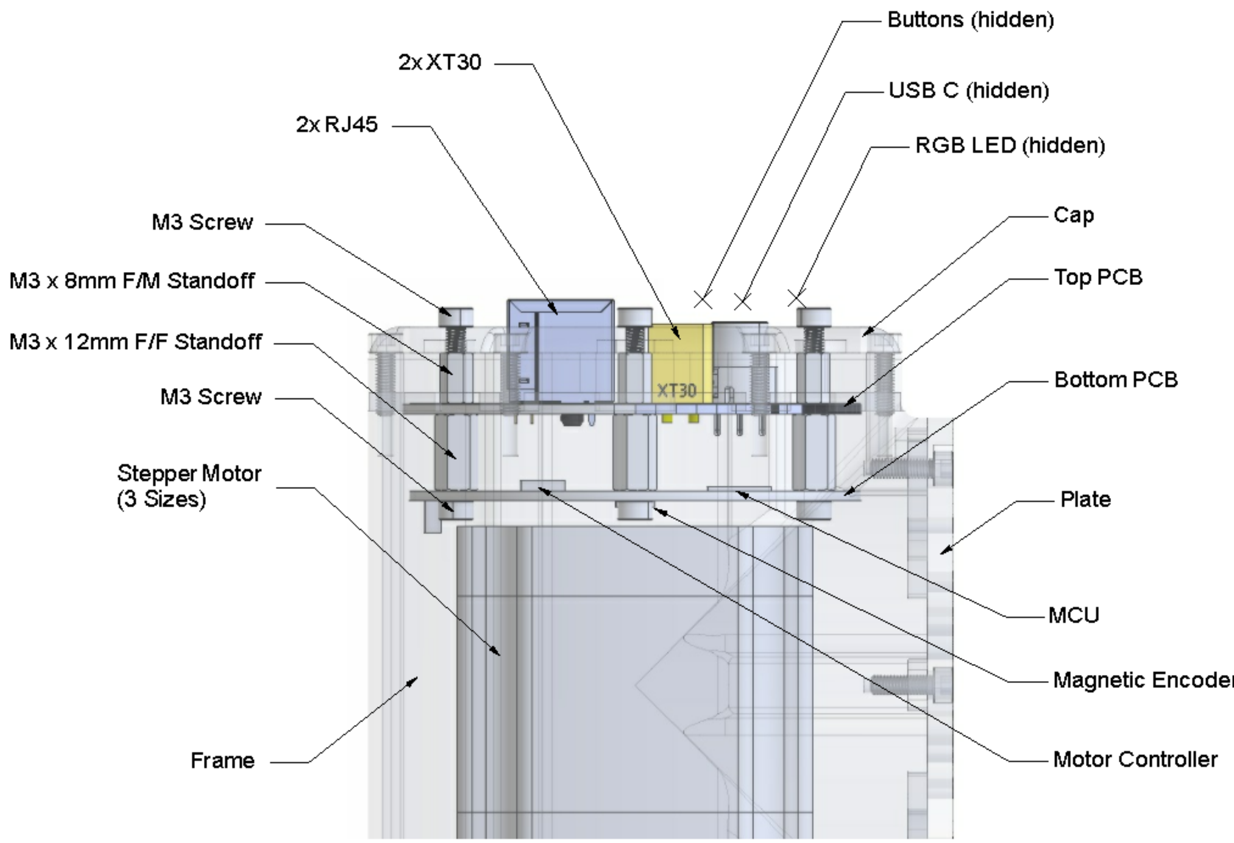
Component Diagram

The following diagram is preliminary, does not include details and may contain errors. It is subject to review. Block diagram of MCU not available. There is no electromechanical brake or optical encoder.



Layout

The layout shown below is intended to help the reader better understand the information presented in this document. Exact component placement and selection will be reviewed with the designer upon successful awarding of the contract.



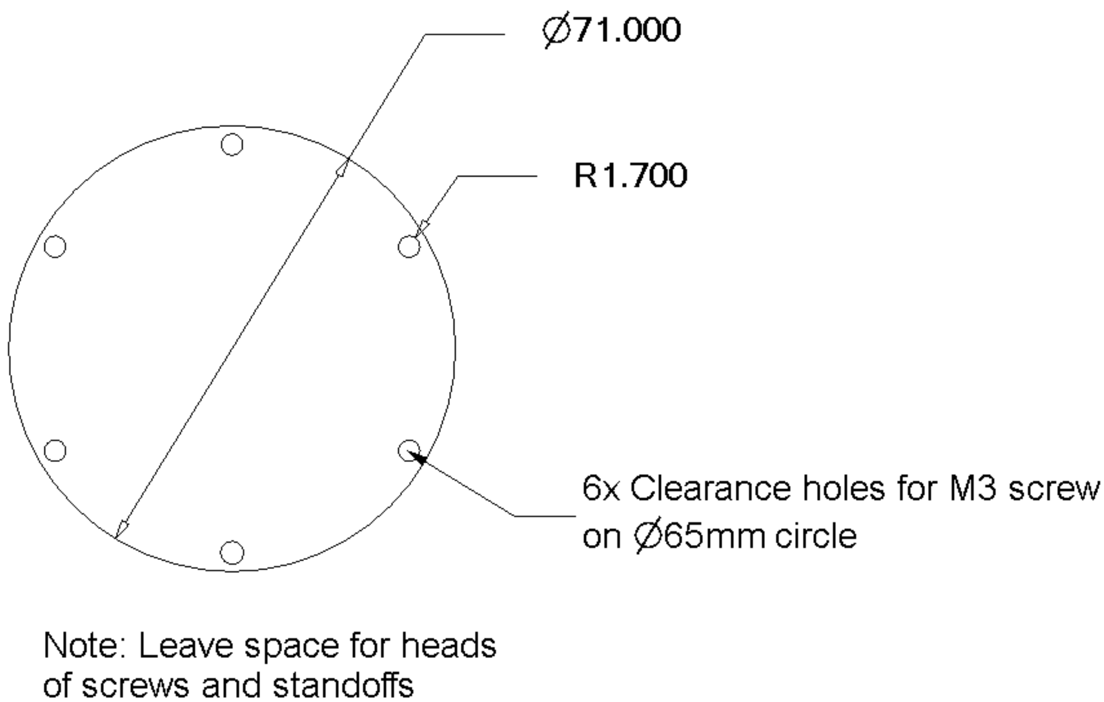
Not shown / included:

- Connection between top and bottom PCBs
- Motor controller heat sink (if needed)
- Cable between stepper motor and bottom PCB
- Location of temperature sensors, LEDs
- CAN transceiver
- Shaft with magnet

1. Top PCB

The connectors on the top PCB will be accessible via cutouts in the cap. It is at the discretion of the designer if any additional key parts need be added beyond those indicated below, though most connectors on the top PCB are through-hole which impedes components from being easily placed on the underside of the board. The top PCB connects to the bottom PCB using direct pins / connectors, several of which can be used to be able to handle the ~8A current of the most powerful of the three stepper motors. The main connectors (2x Ethernet, 2x XT30) as well as the two buttons and LEDs should be placed symmetrically. The USB connector can (ideally) be placed centrally along the axis of symmetry. Access to the connectors is only available from the top, not the sides of the frame or cap.

PCB Dimensions



- The dimensions above apply to both the top and bottom PCBs.
- Mounting holes: 6x Ø3.4mm (M3 clearance) holes on a Ø65mm circle
- Standoffs (cap to top PCB): M/F M3 x 8mm hex, 6mm hex size
- Standoffs (top to bottom PCB): F/F M3 x 12mm hex. 6mm hex size (interconnect)
- PCB Diameter: 71.0mm (breakoff points must be within this and cannot protrude)
- PCB Thickness: 1.60mm required for the frame and to ensure the magnet is within the appropriate distance of the magnetic encoder chip.
- Four layer PCB design strongly suggested (High voltage (Vin); GND; Signal (two layer))

Power In / Out

2x **XT30 connectors**

- Why? Low profile, secure friction fit, inexpensive, can handle high current
- Alternatives were M5 or M12, but physical size and cost impeded their use
- Vertically oriented, close to each other to pass high current (~25A) current on PCB
- Relatively inexpensive and available from many suppliers
- Vin: ~25V to 45V
    - Nominal: 30V from wall adapter
    - Max: 43V from charged 10S LiPo
    - Min: 27V from discharged 9S LiPo

Communication In/Out

2x **RJ45 connectors**

- Why? Second most common connector used on the market for CAN communication
- Alternative was DB9, but large size, heavy and expensive cabling impeded their use
- CAN will be used for communication between actuators (allows daisy chain / BUS)
- Position on the PCB must take into account the power connector and interference

1x **USB type C**

- Why? Easily upload code and most popular “next gen” USB connector
- Can be used to power the logic, but NOT the stepper motor (TBD)
- Used for communication between a computer and the first actuator in the BUS

User I/O

1x **Button**

- Why? General purpose, user-programmable input
- Due to the height of the XT30 and RJ45 connectors, the button likely also needs to be high to try to have them all on roughly the same plane of the frame.

1x **Bootloader (Button or switch)**

- Why? Enter DFU Bootloader mode
- Open to suggestion on implementation

LEDs

1x **RGB LED**

- Why? User controllable and/or preprogrammed (ex. gives the actuator’s status)
- Possibly addressable RGB like neopixels?
- Note that just like the button, the frame is ~14mm above the top PCB, so a vertical, rigid light pipe may be needed.

1x **Power LED**

- Why? Lit when actuator is receiving power
- Note that just like the button, the frame is ~14mm above the top PCB, so a light pipe ~10-12mm may be needed. Consider this when selecting components.

2. Bottom PCB

The bottom PCB contains the MCU, motor controller, IMU, voltage regulation on the top, and the magnetic encoder and temperature sensors mounted on the bottom. This PCB will be in very close proximity to a hot stepper motor, so temperature-resistance components need to be selected and placed accordingly.

Board Dimensions

Identical to top PCB dimensions.

PCB

- Must be 1.60mm thick given space and dimensional requirements.
- Four layer design
  - High voltage (Vin)
  - GND
  - Signal (two layer)

CAN Transceiver: Communication to serial

- Converts CAN communication to serial for use by the MCU.
- No specific chip selected and open to suggestions.

Microcontroller

Arduino IDE compatible Cortex-M microcontroller (using STMDuino or other?). The STM32F405 seems quite versatile (ex. Adafruit Feather is based on this chip and can be programmed using the Arduino IDE, MicroPython or CircuitPython). The Cortex chip used in previous prototypes is the STMF413, which does not seem Arduino compatible (please verify). The selected candidate should explain their suggestion of MCU given these criteria before proceeding with the design.

Stepper Motor Controller

The stepper motor controller can provide far more current than is needed for two out of the three motors, but in order for the electronics to be interchangeable between actuators, the PCBs must be identical. The stepper motors have the following specs, and are intended to operate at 30V input.

“Small” Stepper motor

- Type: Hybrid / Bipolar / 4-wire
- Phase: 2
- Voltage: 2.8V
- Current: 1.33A
- Resistance: 2.1 Ohms
- Inductance: 2.5 mH

“Medium” Stepper motor

- Type: Bipolar / 4-wire Stepper
- Phase: 2
- Voltage: 2.4V
- Curent: 3A
- Resistance: 0.8 Ohms
- Inductance: 2.4 mH

“Large” Stepper motor option 1

- Type: Bipolar / 4-wire Stepper
- Phase: 2
- Voltage: ___
- Current: 5.5A
- Resistance: ___
- Inductance: ___
- NOTE: Different connector suggested due to high current

“Large” Stepper motor option 2

- Type: Bipolar / 4-wire Stepper
- Phase: 2
- Voltage: 2.17V
- Current: 7A
- Resistance: 0.31 Ohms
- Inductance: 2mH
- NOTE: Different connector suggested due to high current

8A+ Stepper Controller Chips

- Previous selection: STMicroelectronics PowerStep01
  - Serial only, complex setup using registers; not very “user friendly”
- Proposed: Trinamic Motion Control (Tmc5160 or Tmc2160-ta)
  - Allows for simple step/direction OR more complex trajectory control
  - To change between the two modes, perhaps a physical switch, or via IO?
- Possible alternative: Toshiba (TB67H420FTG)

Magnetic Encoder

- Why? Obtain absolute position of output shaft without zeroing upon power-up, and verify target position accuracy after motion.
- Broadcom AEAT-8800-Q24, 16-bit absolute magnetic encoder

IMU

- Why? Obtain orientation of the actuator in space, sudden motion and impacts
- InvenSense ICM-20600, 6 DoF IMU (2g or 4g range should be sufficient)

Temperature Sensor

- Why? Read the bottom PCB to verify it’s not too hot

Temperature Probe

- Why? Used to read the stepper motor’s temperature (up to 80C)

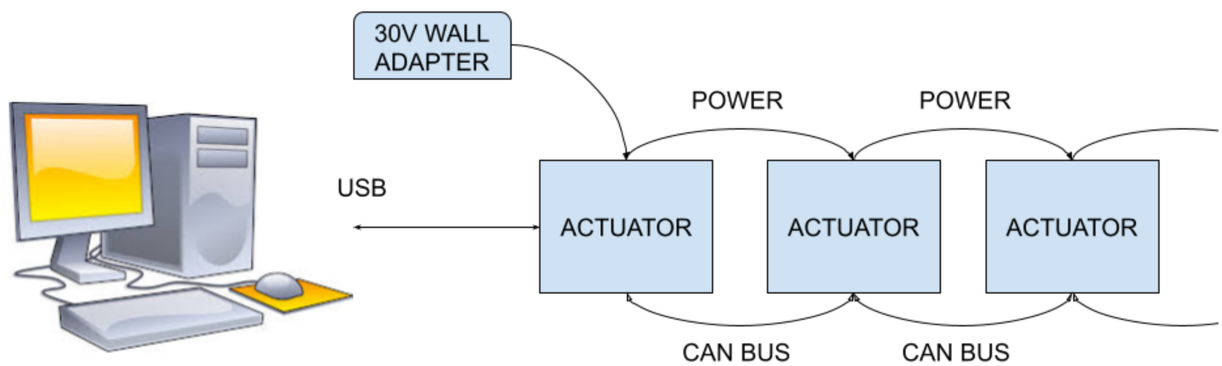
Voltage Regulation

- Why? Convert Vin to voltages required by the main components (3.3V, 5V, 12V…)
- Two voltage regulation circuits have already been prototyped (5V and 3.V) and can be reused at the discretion of the designer.
- TBD most efficient voltage regulation method: cascade; linear etc.

User Features

USB: Upload Code to each actuator

An end user should be able to upload custom code to the board. In a separate project, a simple software interface will likely be created to make use of sample code, allowing the user to modify key parameters such as the actuator's unique ID. If the chosen candidate for this PCB project has sufficient experience creating such code, they are invited to apply.



USB: Send / receive commands via CAN BUS

In a bus of N actuators which are connected from one to the next via CAN and power cables, the first actuator in the BUS is connected via USB to an external computer which allows the computer to send commands to that first actuator, which then communicates the commands on the BUS via CAN. It is not envisioned that many actuators on the BUS will be connected to many computer as this may cause issues. USB communication is "master" and takes priority over any BUS communication.

CAN Communication

CAN communication between actuators was selected over serial communication largely because of its widespread acceptance as a reliable method for communication between industrial devices (despite this actuator not being considered for the industrial market). The alternative, which is sometimes used for such "all in one" actuators is RS485. Sample code for end-users should be created to make use of CAN communication between actuators of different IDs.

Bootloader

Bootloader will depend on the MCU chosen. On the suggested MCU, in order to enter DFU bootloader mode, two pins need to be jumpered. This can be replaced by a small button or small switch. See how Adafruit makes use of one STM32F405:
https://learn.adafruit.com/adafruit-stm32f405-feather-express/dfu-bootloader-details
https://learn.adafruit.com/adafruit-stm32f405-feather-express/arduino-ide-setup
https://learn.adafruit.com/adafruit-stm32f405-feather-express/micropython-setup
https://learn.adafruit.com/adafruit-stm32f405-feather-express/circuitpython-setup

Motor Controller

The end user will need to be able to send commands to the stepper motor controller, ideally both in terms of step and direction ("simple" control) and, should a more complex chip be chosen, commands to make use of any advanced features (ex. step mode, motion profile etc.) should be provided as part of the example code.

Magnetic Encoder

Rather than installing a magnet on the rear shaft of the stepper motor, a shaft has been connected to the output of the gearbox and passes through the hollow shaft of the stepper motor, onto which a (diametrically-magnetized) magnet is attached. An absolute position magnetic encoder chip is placed centrally on the underside of the "Bottom" PCB to read the magnetic field, thus allowing a user to query the absolute angular position (within 360.00 degrees) of the output shaft. This inexpensive way of providing absolute position feedback without zeroing the actuator is a core feature of the overall product. Accuracy of the reading while in motion (i.e. dynamic) is less than that of static readings.

IMU

A six-axis (3x accel, 3x gyro) inertial measurement unit is included to know the actuator's absolute orientation with respect to earth. Additional user-created functionality can potentially be derived from this inexpensive feature, including sudden loss of steps, impacts, ensuring an assembly is oriented correctly in space etc. The chip should be placed as centrally as possible.

Temperature Sensor(s)

Given that stepper motors can run very hot (up to 80C), it is important to include a temperature probe which can be glued to the stepper motor during assembly, as well as a surface-mount temperature sensor on the underside of the Bottom PCB which has the most direct exposure to the stepper motor.

3. Sample Code

Production / Quality Control Test Code & Procedure

The first code to be written is to verify that the PCB and various components is working correctly. This code is not only to verify the prototype PCBs work, but will also be used in production as quality control to ensure the correct components have been chosen, properly oriented on and soldered to the PCB. The PCBA manufacturing company will require a procedure to make use of this code. The test code /procedure can involve the creation of a jig, as well as manual input and visual verification.

User-Oriented Example Code

The sample code intended for an end-user should be well commented. If necessary, an Arduino library should be created in order to make it easier for an end user..

- ❏ Set the actuator’s unique ID on the BUS
- ❏ Send steps and direction to the motor controller
- ❏ Change step mode (half step, quarter step etc.)
- ❏ Advanced motor controller configurations (with comments to describe each)
- ❏ Read absolute magnetic encoder value
- ❏ Read IMU values
- ❏ Read temperature sensors
- ❏ Send communication via CAN Bus
- ❏ Read and parse communication received via CAN Bus

Uncertainties

1. Microcontroller: The MCU selected needs to be Arduino IDE compatible, relatively fast, and able to store sufficient code, though its size (10x10mm, 14x14mm etc) needs to fit the board.
2. CAN transceiver: This will also depend on the MCU chosen, though most seem to require a separate transceiver chip
3. Voltage Regulation: What is the best method for highest efficiency and lowest heat generation
4. Motor Controller: Advanced motor controller chips will allow higher efficiency and include marketable features like low current braking, smooth motion trajectories and more
5. Loss of steps resulting in free spinning: Should the load be too great, can the motor controller stop the actuator from spinning freely (i.e. brake the stepper without using a physical electromechanical brake)