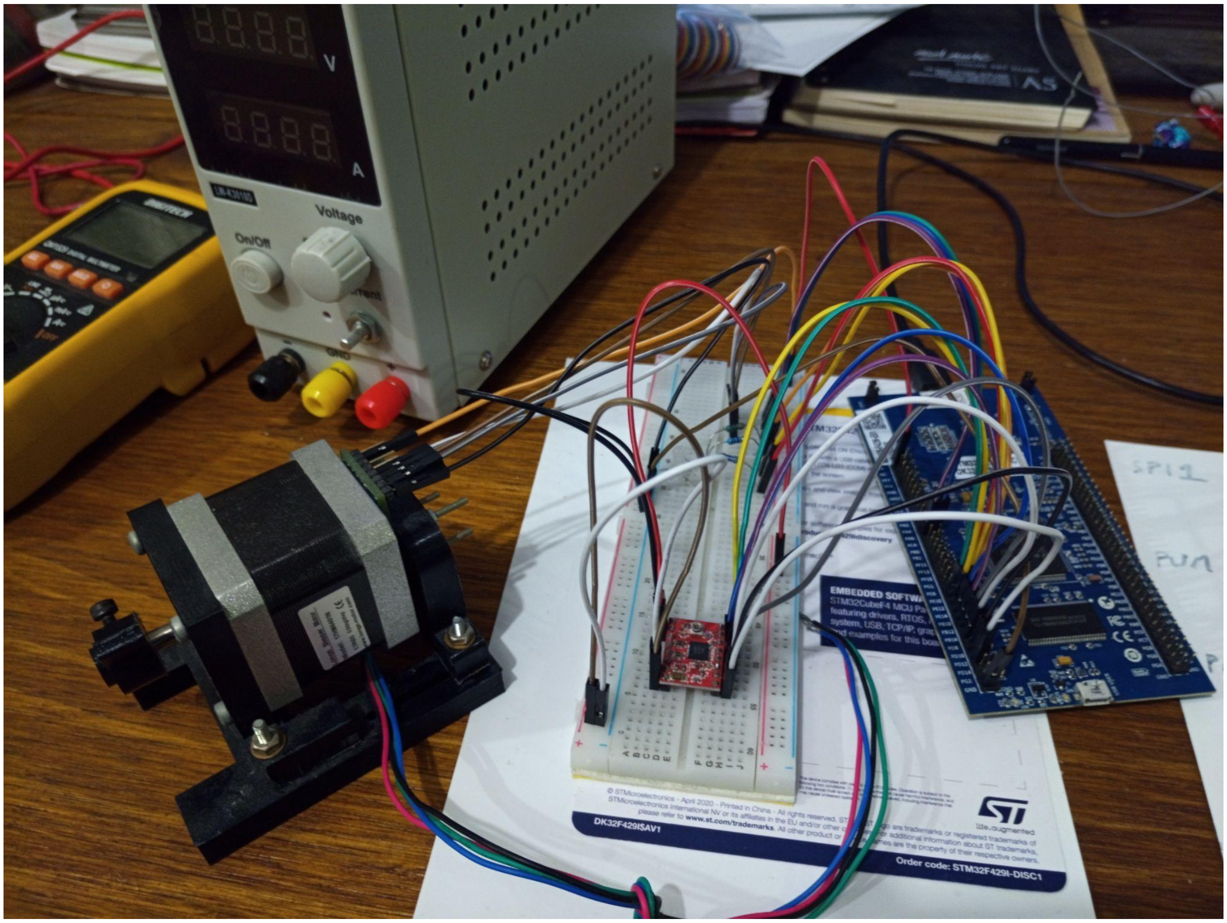
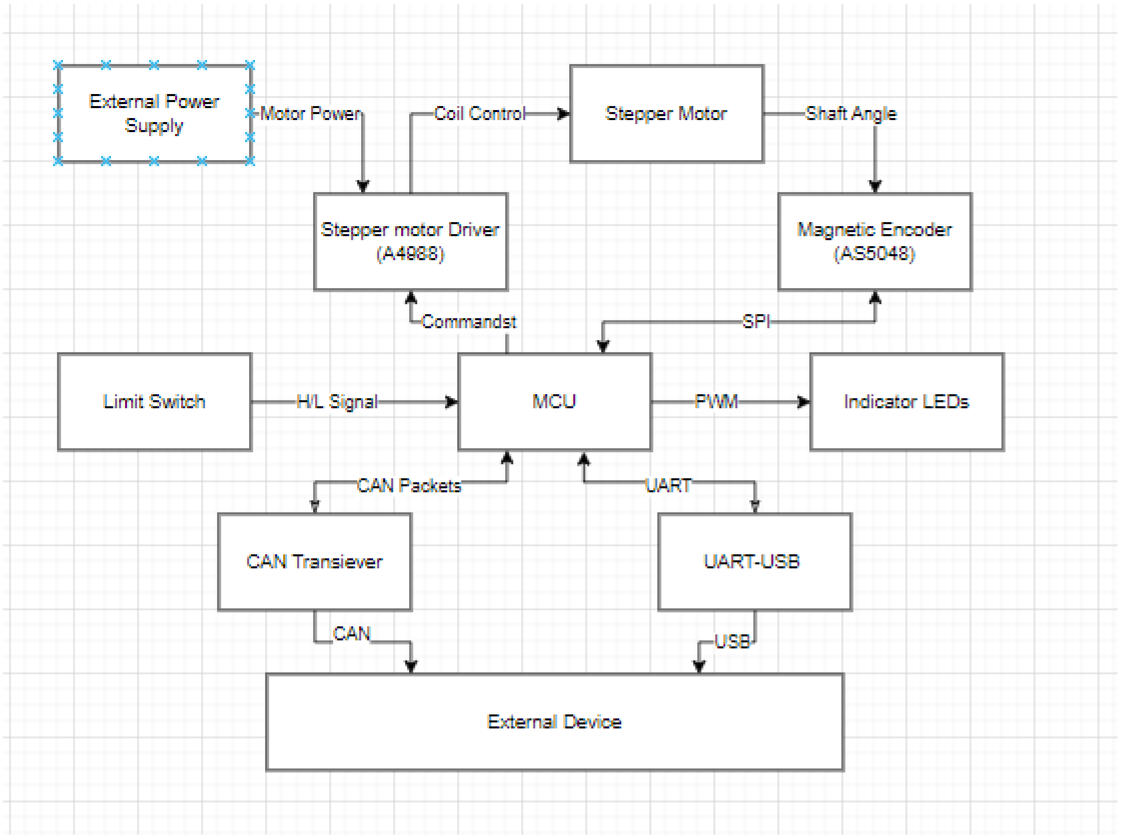
Servo Controller with CAN and UART Interface

Brief:

This project is a closed loop stepper motor controller which is able to detect and report disturbances to the actuator as well as visual feedback on the controller action, it features both CAN bus and UART control interfaces. The controller is able to accept a number of commands such as “go to angular position”, “report current position”, and can notify host devices if external forces cause changes to the system state (angular position). The controller also incorporates LEDs to give qualitative information about the instantaneous error of the angular control system.



Test Hardware



Hardware Block Diagram

Detailed Description:

The core function of this device is that it allows the user through either a CAB bus or UART serial interface to command the motor to move its rotor to a desired position, and to maintain this position (or return to this position) when the rotor is moved by an external force. In order to do this the relative angular position of the shaft is determined using a AS5048 magnetic encoder which is mounted to the rear of the stepper motor, this sensor communicates with the MCU via SPI. Because this sensor is relative (not absolute) a limit switch is issued to “home” the rotor and act as a fixed angular point the sensor data can be referenced against.

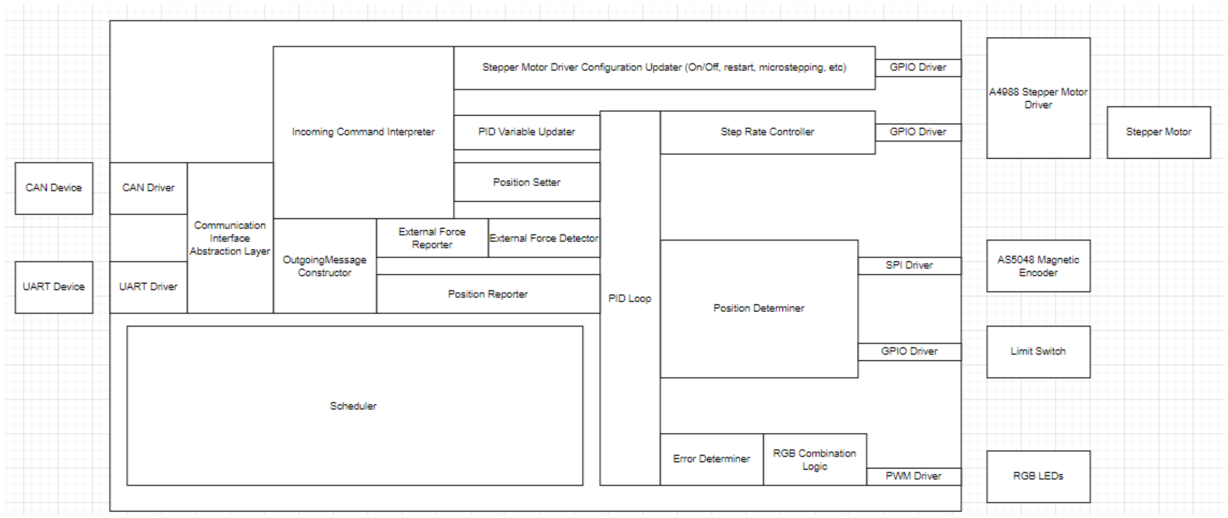
A control loop (most likely PID or PD) will be used to control the motor position, in order for the user to see what the current amount of error between the current position of the rotor and the setpoint is, a RGB LED will show a colour between Red and green depending on the amount of error (Green for 0 error and Red for a large amount of error). The LED will flash blue when an external force is detected acting on the rotor.

Finally this device will have a “API” which dictates a set of commands which can be sent to the device as well as alerts the device can send to the user over either UART or CAN bus. A broad over view of the commands/alerts is given below

Commands (Message from user device to Controller)	Alerts (Messages from Controller to user device)
Go to Position X	Message received acknowledgement
Report current position	Current Position
Update Motor Configuration	External Force detected
Update control parameters	Error detected

Firmware Architecture:

While still in development the firmware components of this project can be broadly grouped into two categories, those which are implemented in the main loop such as the motor control loop, and LED control, and those which are event driven such as processing messages.



Software Architecture Diagram

Bits of code which run in the main loop	
Code Block	Description

Control Code	The Control code covers the control algorithm and the setting of the step rate. This loop must run at relatively consistent intervals in order for good results
Position Determiner	Reads Encoder and records angular position.
External Force Detector	Reads the Position Determiner value and compares it to the last value as well as looks at the input commands and determines if any movement seen is caused by external forces or is the result of a movement command.
RGB output	The color of the LED should change based on the error of the output to the commanded value (Green = low error, Red = high Error)

Bits of code which are triggered by events		
Code Block	Description	Trigger
Position Setter	Updates the setpoint in the PID loop based input commands.	Setpoint Command Received
Position Reporter	Constructs and sends output message containing the current angular position of the output	Report Position Command Received
External Force Reporter	Constructs and sends output message indicating that an external force has been detected (also includes clockwise/anti-clockwise direction of force)	External Force Detector determines that an external force has changed output position
PID Variable Updater	Updates the proportional, integral and derivative constant values within the PID for tuning purposes.	PID Variable Update Command Received
Stepper Motor Driver Configuration Updater	Updates the configuration settings of the A4988 driver such as microstepping rate, reset, on/off	Stepper Driver Configuration update command Received
Step rate Controller	As the stepper motor driver takes a single high/low transition to signal a “step”, the output of the PID loop must be converted into a step frequency and said step frequency continuously implemented	Timer set by current “step rate”