MIGNON ROBOTICS
KAZAKHSTAN, ASTANA

Kiosk with drone hub

Supervised by: Adilzhan Adilkhanov

Drone Hub

1 Introduction

The main objective of the Start Up project is to develop an intellectual construction for automating the delivery process. The construction consists of kiosk and drone hub. To begin with, the inside of the kiosk is designed to fit cells for storing orders and rail is built in the middle for linear motion of the gripper, a robot designed to distribute orders into cells and deliver them to the parcel window. On the exterior of the kiosk, there are a monitor and a pick-up window. It is needed to scan the QR code on the monitor in order for your order to be collected by gripper and transferred to the pick-up area. The question that arises is 'How are parcel orders loaded inside? '. Orders are loaded through a drone hub located on top of the kiosk. The drone landing area is covered by a dome that opens after receiving a signal from the incoming drone. The main part of the hub is a centering platform. The centering is crucial to ensure that the delivery box is placed above the lifting mechanism. Once aligned, the package is lowered by the mechanism and transferred to the gripper. Lastly, the package is loaded into the assigned cell in the kiosk. The electronics part of drone hub control is explained in detail in this report: starting with load cells for detecting the weight/pressure change on platform, followed by stepper motors for centering, and concluding with linear actuators for dome mechanism control.

2 Landing platform

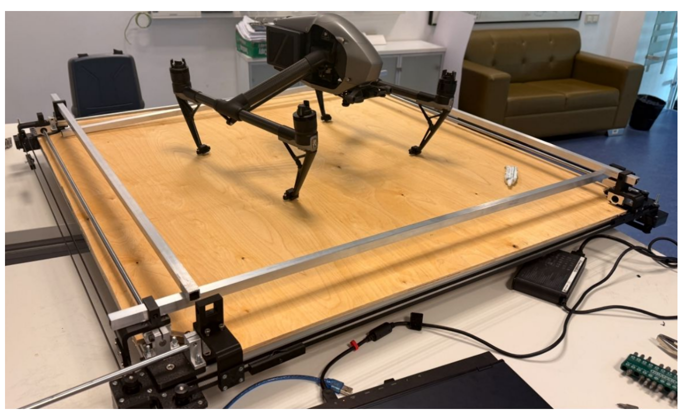

Figure 1: Centering Platform

In Figure 1, the centering platform is illustrated. 2 stepper motors are fixed to one of platform corners. The motor belts are wrapped around the motor shafts. When the motors are activated, they rotate in either a clockwise or counterclockwise direction, causing the belts to move accordingly. This belt movement drives the aluminum profiles, which are assembled with the help of connectors.

2.1 Weight Control

The platform centers the drone as it lands on it. In order to detect the weight change on the platform, 2 load cells are attached under it. The program is written such that once the load cell measurement exceeds a certain threshold, the stepper motors are engaged to activate the platform centering.

2.1.1 Load Cells

Load cell is a sensor that converts the amount of deformation into an electrical signal. The deformation can be small, therefore amplifier HX711 is connected between Arduino and load cell to amplify and convert the small voltage signals from load cells into digital values .

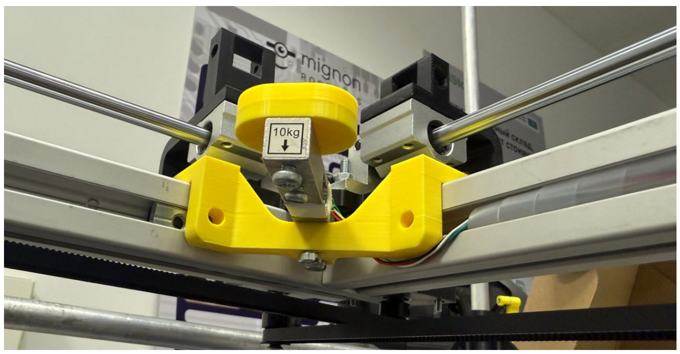

Figure 2: Load cell fixed to the platform

2.1.2 HX711 Amplifier

LINK: HX711 Info

- Power supply voltage: 5 V
- Current consumption: up to 10 mA
- Sampling rate: 80 Hz
- ADC resolution: 24 bits
- Gain factor Input A: 64 or 128
- Input B: 32
- Number of load cell input channels: 2
- Operating temperature: -40 °C to +85 °C

Pin configurations:

- E+ : Provides positive excitation voltage to the load cell. Red wire.
- E- : Provides negative excitation voltage (ground) to the load cell. Black wire.
- A+ : Positive differential input for Channel A. Green wire.
- A- : Negative differential input for Channel A. White wire.
- GND : Common ground for power and signals.
- VCC : Supply voltage for the HX711 module. Accepts 2.7V to 5.5V.
- DT : Serial data output for communication with a microcontroller.
- SCK : Serial clock input for communication with a microcontroller.

2.1.3 Load cell wiring the circuit

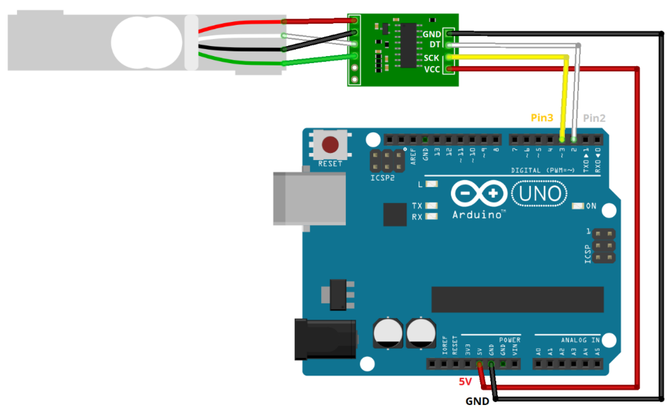

Figure 3: Load cell, HX711 amplifer connected to arduino

To begin with, it is needed to wire up your circuit as shown above. Before doing any measurements, calibration is needed. The calibration factor is different for each load cell, therefore calibrate both. Do not use the the same factor for all of them.
In order to calculate the calibration factor, use the following code. Take the object with a known weight. After uploading a code first, it is asked to tare and afterwards to place the load with a known weight and enter the weight in grams. All needed steps are described in the comments of the code.

THE CODE LINK: The code to determine the calibration value

After determining the calibration factor for each load cell, the values should be inserted in the following code for reading two load cells. The load cells should work in a way that when the readings go above the defined threshold, it gives a command for the stepper motors to move, so the drone will be centered.

CODE FOR READINGS OF TWO LOAD CELLS: 2x LOAD Cell Readings
Note: The codes for load cell are from library in Arduino IDE: HX711_ADC by Olav Kallhovd.

One problem was faced when working with load cells, the readings were decreasing; leading to improper values. This problem can activate or deactivate centering platform because of false alarms. Therefore, I have connected tare button to Arduino, taring load cells 2-3 times in the beginning made the reading stable.

2.2 Centralizing mechanism

As it is described above, the centralizing mechanism is responsible for adjusting the drone to the certain position (on top of the lift). The platform is controlled by Nema17 Stepper motors.

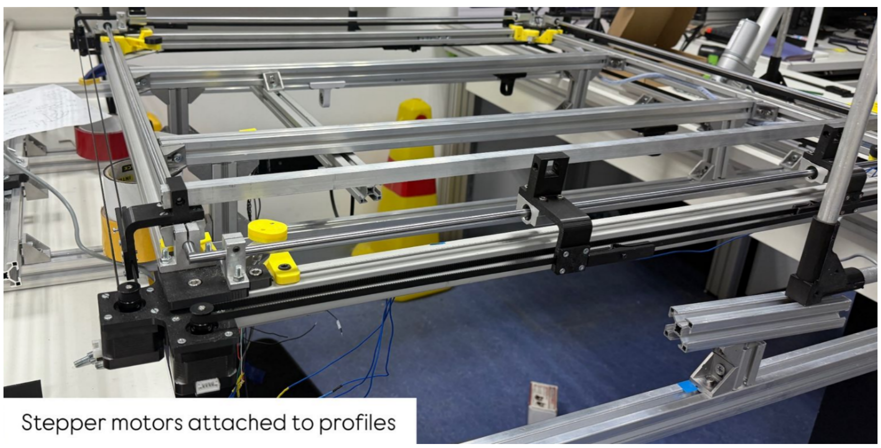

Figure 4: Stepper motor placement on the profiles

2.2.1 Nema 17 Unipolar Stepper motor

The stepper motor of the "DQ-42HB34A" model is used. It is a 2-phase unipolar stepper motor with a step angle of 1.8°, a holding torque of 0.31 N·m, and operates at 1.4 A per phase with a coil resistance of 2.3 Ω . The stepper motor's movement is controlled by Arduino code. However, microcontoller (5V output and 20 mA current) cannot be directly connected to the stepper motor as it is not able to provide necessary voltage and current, therefore it is crucial to install motor driver between arduino and the stepper motor. Stepper motor driver gets the power of 12V from the external power supply, and it gets control signal from Arduino. Motor driver amplifies the control signal and sends it to the motor, so it operates properly.

2.2.2 6 Wires Unipolar as Bipolar motor

6-wire stepper motor has 4 Coil End Wires and 2 Center tap wires. In order to use it as bipolar, center taps must be isolated.
They can be identified by measuring the resistance between wires. If there is a resistance, the wires are part of the same winding, otherwise the wires are not part of the same coil.
After determining the wires of the same coil, the 'center tap to coil end' resistance is half the 'coil end to coil end' resistance.
They were identified already, refer to the following Figure 2.

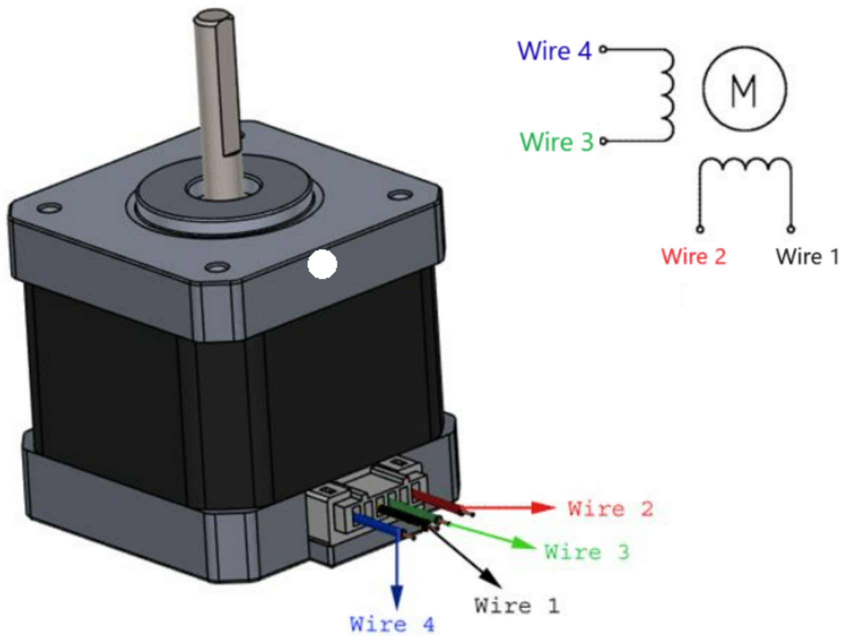

Figure 5: Stepper motor coil ends as bipolar use

2.2.3 TB6600 Driver

For this project's setup the stepper motor is handled by TB6600 stepper motor driver. It is designed to drive 2-phase and 4-phase hybrid stepper motors with operating voltage ranging from 10V up to 42V and a maximum current of 4A. It supports microstepping mode with step division up to 1/32, which ensures high positioning accuracy and control of the output current. The advantage of using this driver is that it has built-in protection against overheating, over-current and short circuit.

Technical specifications

- Input Voltage: 10-42V
- Maximum Output Current: 4A

- Supported Stepper Motors: 42mm - 86mm (Nema17 - Nema34)
- Macro Step: 1 / 2 / 8 / 16 / 32
- Current Adjustments: 8 Levels
- Operating Frequency: Up to 20kHz

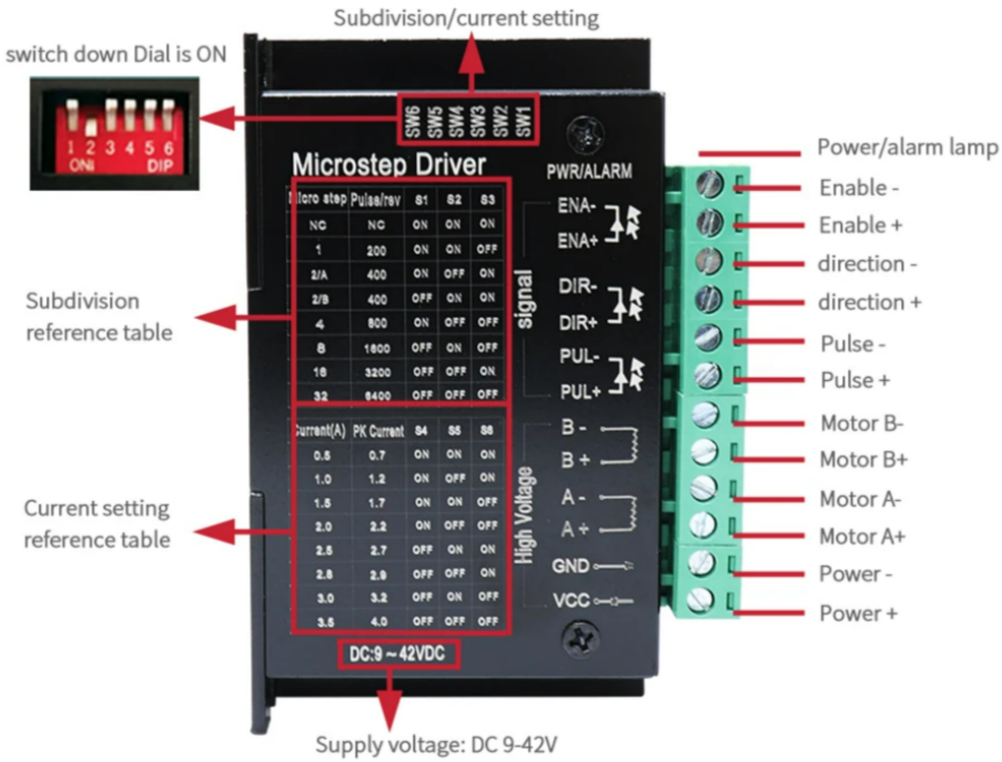

Figure 6: A stepper motor driver TB6600 configuration

1. SW1, SW2, SW3 - alter micro-step resolution from full step to 1/32 step.
2. SW4, SW5, SW6 - adjust driver current from 0.7A to 4A for continuous motor operation.

2.2.4 The circuit connection

In Figure 2, it illustrates where motor's windings must be connected. 2 windings of the same coil should be put into B+ and B-, other 2 of another coil into A+ and A-.

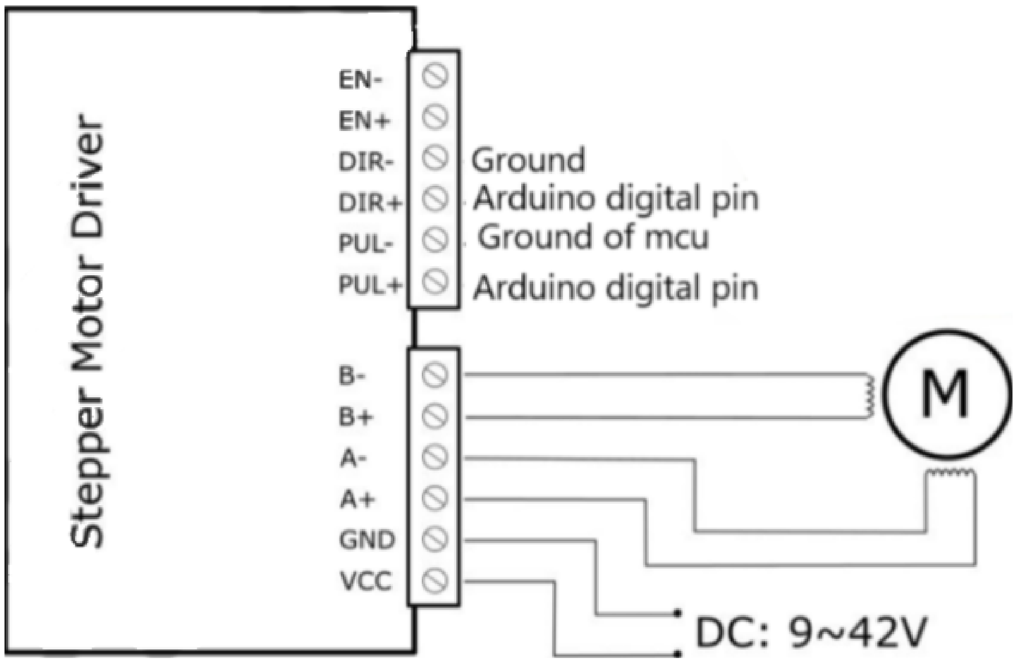

Figure 7: TB6600 driver connection to the stepper motor

1. I have connected one motor driver to digital pins 7(PUL+) and 6(DIR+);
2. Second motor driver is connected to digital pins 9(PUL+) and 8(DIR+).
3. SW1, SW4, SW6 are on.

The CODE for testing the adjustments of the profiles: Code for testing stepper motors

2.3 Elevator

The elevator serves as the link between the roof and the kiosk. Its main task is to receive packages delivered by drones and lower them into the main assembly area. Thereafter, a gripper retrieves the parcel and carries it into one of the designated storage cells until it is collected by a customer. Furthermore, it is planned to make the elevator two-way structure: it should be able to take the packages dropped off by drone and vice versa lift them for the drone to pick it up.

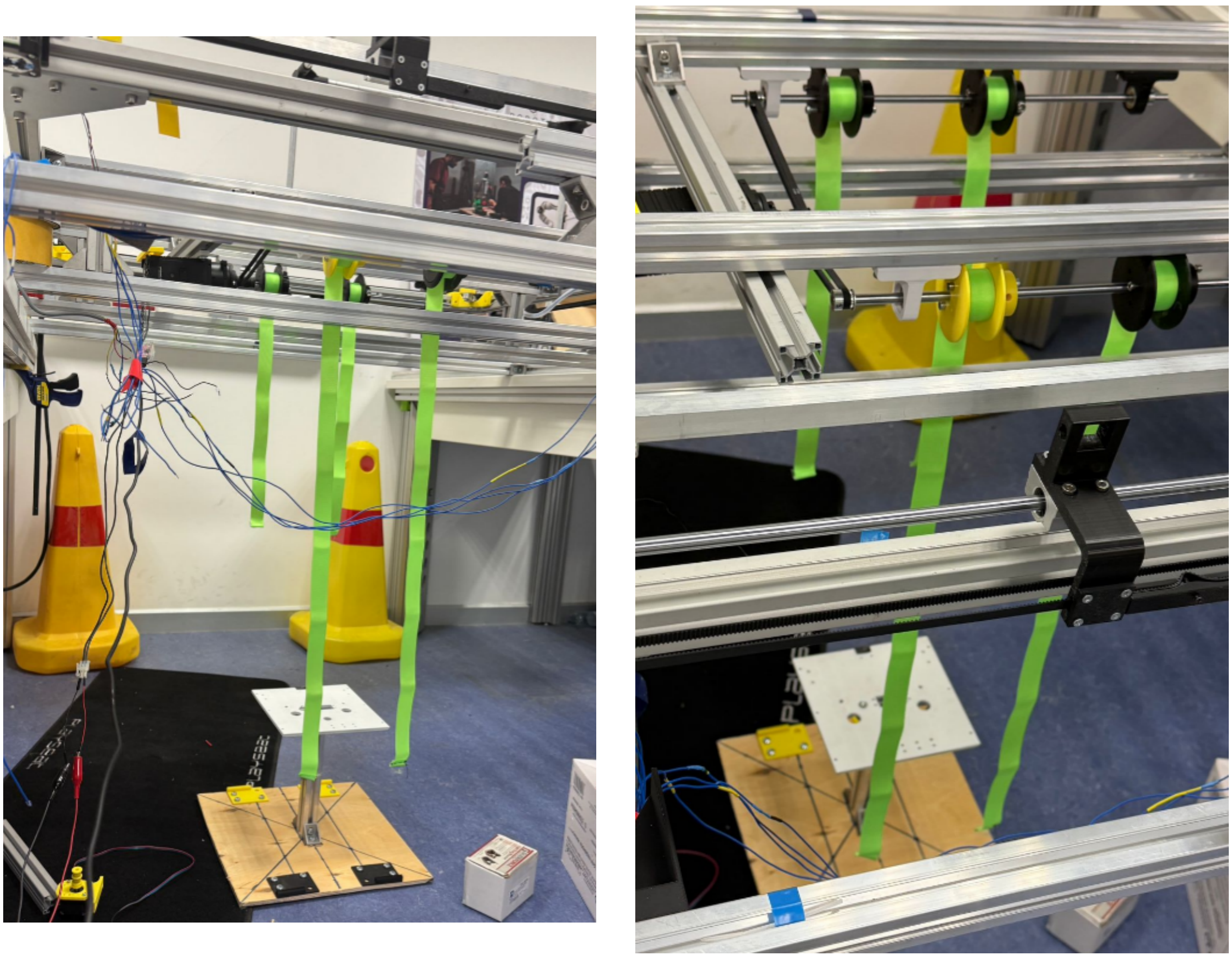

Figure 8: The elevator structure under construction

Firstly, the lifting mechanism was planned to be controlled by stepper motor and arduino microcontroller, but nema17 motor cannot provide enough torque, so it was decided to use dynamixel.

3 Dome mechanism

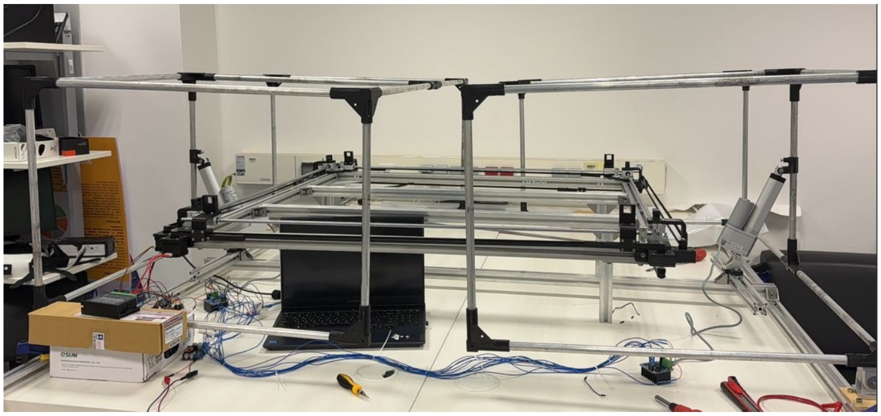

Figure 9: The domes in a closed state

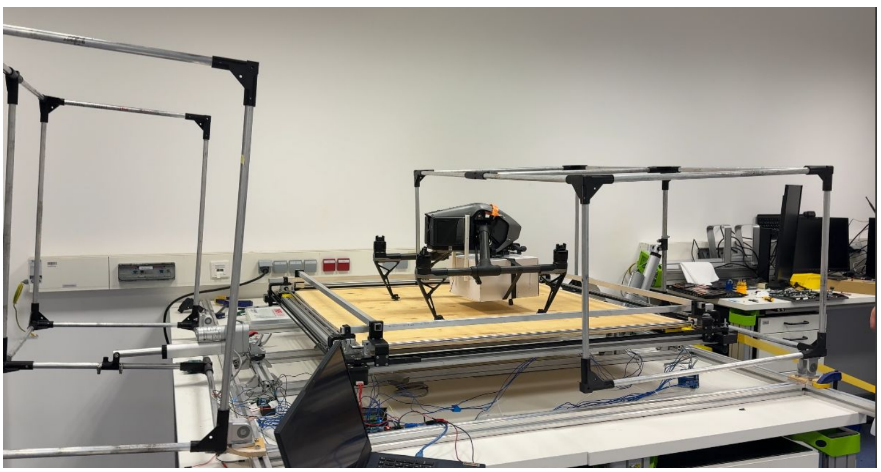

Figure 10: One of the domes open

The dome construction can be explained as the roof of the centering platform in simple words. 2 Linear actuators are attached to the domes of the drone port. When the command 'open' is sent, actuators start extracting one after another with 1 second in between, opening to the certain position. After, the drone is landed, by retracting actuators, the dome is closed. The domes are designed to shield the platform against environmental factors while also concealing the internal mechanism.

3.1 Linear actuator specifications

A short overview: Glideforce linear actuators are 12V DC gearmotors that use a lead screw to move a shaft back and forth along its length. The shaft will hold its position even when unpowered and subjected to static loads. Two limit switches safely stop the motor at either end of its range, while integrated diodes allow it to reverse direction after reaching a limit point if the supplied voltage is reversed.

1. NO. LACT4P-12V-5
2. Stroke length: 4 inches $- >$ 101.6 mm
3. Input voltage: 12VDC
4. Average Speed: 1.7 inches/ second $- >$ 43.18 mm/ second.
5. Dynamic Load: 34 lbs $- >$ 15.4 kg.
6. Static Load: 450 lbs $- >$ 204 kg.

3.1.1 How to use the actuator

First, in order to check if the linear actuator works, it can be tested easily. To do so, a power source of up to 12 V needs to be connected to the motor leads. In order to move it forth, the red wire of the motor should be connected to the ground of the power supply while black wire to the positive rail. In order to reverse the direction of motion, simply reverse the applied voltage.

The linear actuator, except power leads, have other 3 wires which are needed to get the feedback information about the position of the actuator. **3 WIRE CONFIGURATIONS:** Blue - feedback (must be connected to analog pin of the arduino); White - should be connected to ground; Yellow - should be connected to 5V from microcontroller.

In order to control the speed and direction of the actuator IBT-2 motor driver is used.

3.2 IBT - 2 motor driver

The IBT-2 H-Bridge Motor Driver is an electronic component designed for controlling DC motors' direction and speed. It is capable of driving high-power motors in both forward and reverse directions.

3.2.1 Motor driver specifications

1. Supply Voltage: 6V to 27V
2. Logic Voltage: 5V (compatible with Arduino)
3. Continuous current: Up to 43A
4. Peak Current: 100 A (for a feew seconds) - not recommended

3.2.2 Pin configurations and Descriptions

1. RPWM - Right PWM input for motor speed and direction
2. LPWM - Left PWM input for motor speed and direction
3. R_EN and L_EN - Right and Left Enable Inputs respectively
4. Vcc - 5V arduino
5. GND - Ground
6. B+ : power supply positive
7. B- : power supply negative
8. M+ : motor positive rail
9. M- : motor negative rail

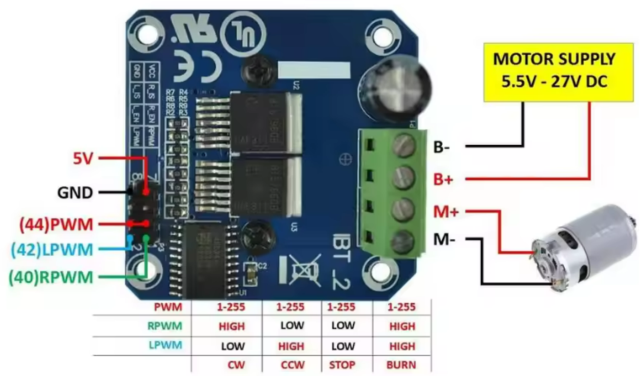

Figure 11: IBT - 2 pins

`https://docs.cirkitdesigner.com/component/2097d705-7df9-466d-9fd0-5dc90909a237/ibt-2-h-bridge-motor-driver` - Useful link for IBT-2 with different circuit examples

3.3 Wiring of the setup

3.3.1 Wiring one linear actuator

This setup is useful for testing each actuator individually. If you have problems like: one of them are not moving, giving incorrect Analog readings $->$ Test each actuator to find the problem.

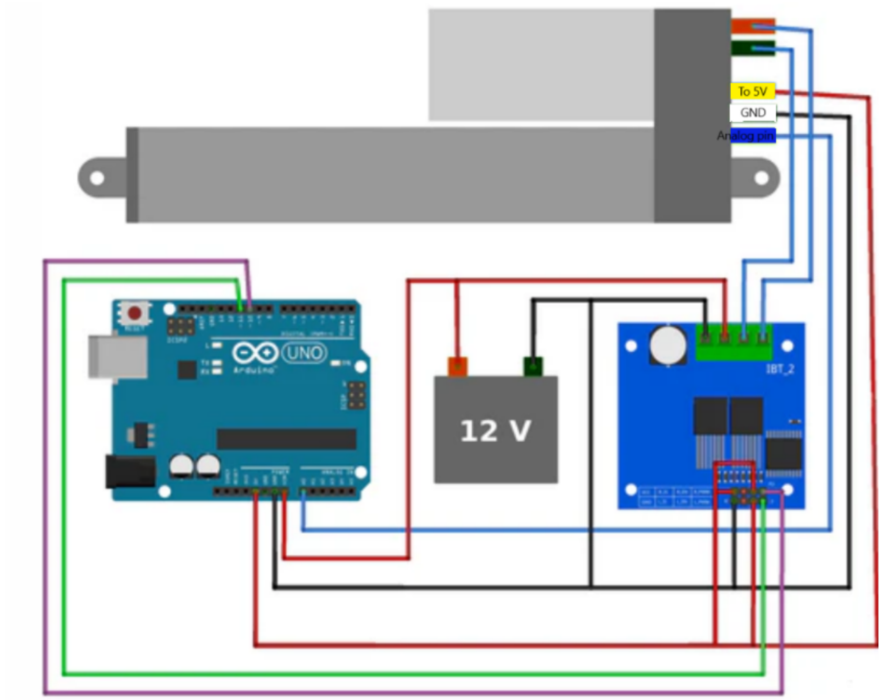

Figure 12: Single Linear Actuator wiring

1. Blue(feedback wire) of the actuator is connected to analog pin on microcontoller. Check to which pin connected, and change the pin in the code. Otherwise, analog readings will not be proper.
2. White wire is connected to GND.
3. Yellow wire is connected to 5V.
4. RPWM of motor driver connected to PWM pin on Arduino.
5. LPWM of motor driver connected to PWM pin on Arduino.
6. R_EN and L_EN are connected to 5V (Arduino).
7. V_cc to 5V.
8. GND to ground.
9. R_is and L_is: no need to wire up them.

CODE for testing feedback of one actuator: Code Link for one linear actuator

3.3.2 Wiring two linear actuators

In general, the wiring is the same as for the one linear actuator, just adding analog pin, vcc and gnd connections.

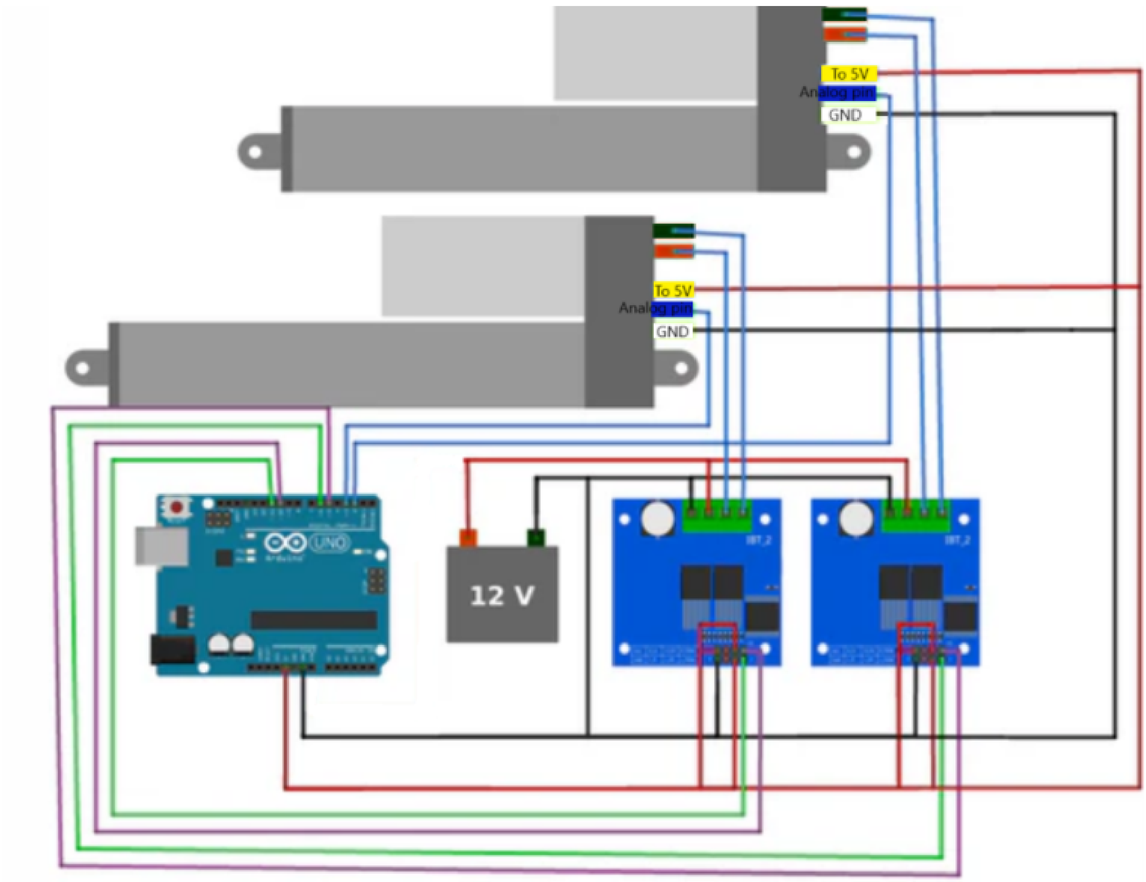

Figure 13: 2 Linear actuator wiring

CODE for opening the domes with 2 linear actuators: Two Linear actuators
Note: the voltage more than 12V should not be supplied (motor can be damaged). In order to prevent step down high voltage, Buck-Boost DC-DC Converter (LTC 3780) is used. The power from power supply goes to the converter, the limit of the output voltage is adjusted by rotating the constant voltage control.

- Model Number: LTC3780 Module.
- Operating Mode: Buck-Boost (Auto Step-up/Step-Down).
- Control Mode: Constant Voltage (CV)/Constant Current (CC).
- Input Voltage: (5-32) VDC.

- Output Voltage: (1-30V) VDC Continuous Variable with Trimpot.
- Output Current: 10A maximum, 7A continuous.
- Output Power: 80W continuous, peak 130W.
- Input Fused: 15A; removable SMD fuse.
- Short Circuit Protected.
- Efficiency: 98
- Input Under Voltage (UV) Cut-off.

NOTE: Once, there was a problem with one of the linear actuators. It was not working after the test. After test-driving, it was illustrating that actuator is not taking any voltage (short circuiting somewhere). Firstly, it was assumed that the motor is burnt and it cannot be used anymore. However, the limit switch was stalling, it was always trying to pull the motor down. Therefore, if this type of problem occurs again, try to check limit switch.

4 The final wiring/code/how small systems connected into one

In order to prevent electronics from the damage, they are placed into enclosure which is fixed to the inside of the profiles with the help of printed corners.

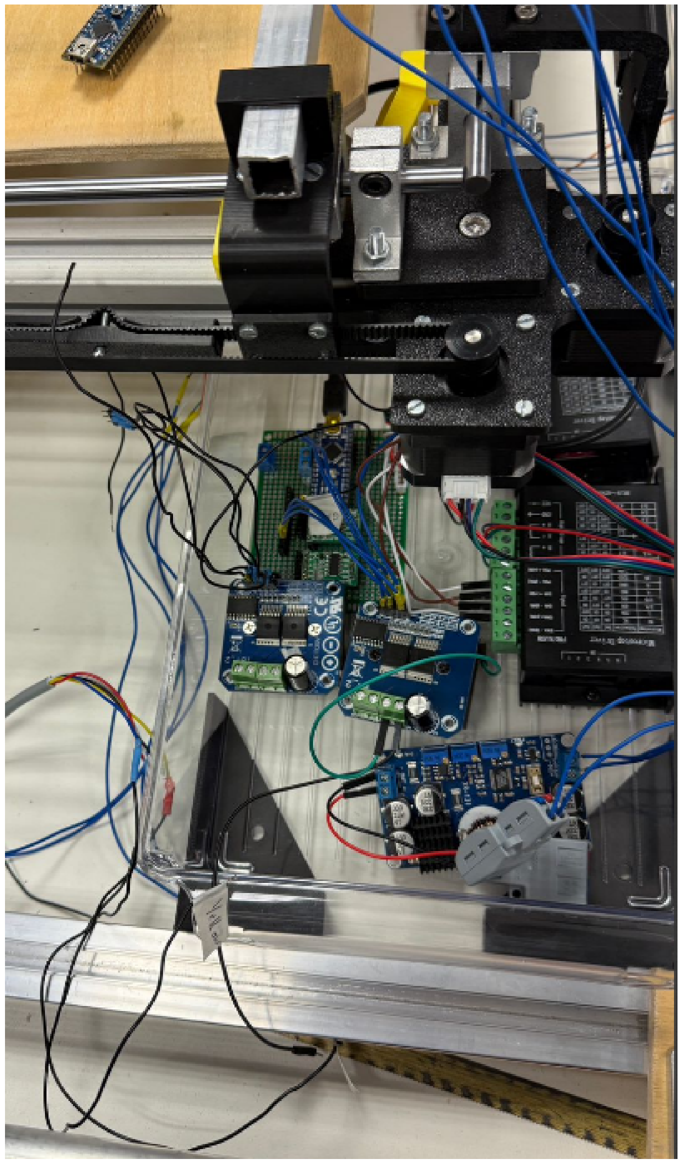

Figure 14: Electronic enclosure

4.1 The wiring to the arduino

4.1.1 Stepper motors

- The wiring is the same as in Figure 5 and 7.
  The digital pins used for Pul $+$ and Dir $+ - >$9 and 8 (motor 1); 7 and 6(motor 2).

4.1.2 Load cells

- The wiring of the load cell is the same as in Figure 3. Vcc, Clk, Dt, Gnd pins are soldered to arduino pins. Needed to connect: load cell wires(red, black, green, white).

4.1.3 Linear actuators

- Refer to Figure 13. RPWM - 5, LPWM - 3; RPWM -11, LPWM - 10 (Must be connected to PWM pins).
- The Arduino Nano has six PWM (Pulse Width Modulation) pins: 3, 5, 6, 9, 10, and 11. These pins can be used to generate analog-like output using the analogWrite() function, allowing for control of things like LED brightness or motor speed.

4.2 How the whole system works

- First, the whole setup of the drone hub includes controlling centering platform (stepper motors), linear actuators, load cells, tare button, and push button.
- For now, push button is used to open and close domes. (Temporary solution)
- Once drone landed, the load cell readings will increase rapidly, giving an alarm to the centering mechanism. After centering, the drone releases the box, so it is relocated on the elevator, which goes down to the storage system.

THE CODE: Final code

5 References/ Links

1. IBT -2 datasheet
2. Stepper motor datasheet
3. TB6600 datasheet
4. DC to DC Buck converter LTC 3780
5. Linear actuator datasheet
6. Linear actuator wiring
7. Stepper motor wires/ how to identify center taps
8. **All codes are in the following link**: Codes - each section includes a direct link to the relevant code.