THRUST BENCH

User Manual

Author
Anirudhhan Raghuraman
Dept. of Aeronautics, Indian Institute of Science, Bangalore
Feb 2024

Contents

1 Introduction

This documentation serves to introduce the purpose, design, and applications of the thrust test bench developed specifically for drones. This specialized apparatus is meticulously designed to evaluate and measure the thrust generated by drone propulsion systems, playing a pivotal role in the enhancement and optimization of drone performance.

The primary objective of the thrust bench is to provide engineers and drone enthusiasts with a dedicated platform for the comprehensive assessment and refinement of propulsion systems. By subjecting drone motors and propellers to controlled conditions, this test bench facilitates the systematic collection of crucial data, enabling the optimization of power efficiency, performance, and overall reliability of drones across various operational scenarios.

The thrust bench plays a pivotal role in the research and development phase, providing engineers with invaluable insights into the thrust output, efficiency, and overall functionality of propulsion systems. By subjecting engines to controlled conditions that simulate real-world scenarios, researchers can gather precise data to refine designs, enhance reliability, and ultimately propel the boundaries of aerospace technology

.

2 Before You Begin

The following requirements should be fulfilled before successfully installing and running the thrust bench.

- Software Requirements
  - **Arduino C** is the language used to control all the hardware present in the thrust bench. The required libraries are:
    - HX711.h
    - Servo.h
  - **Python** serves as the programming language employed for both retrieving and displaying data within a Graphical User Interface (GUI). The required libraries are:
    - PyGame
    - Serial
    - Matplotlib
    - pip
- Hardware Requirements
  - BLDC motor
  - Electronic Speed Controller (ESC)
  - Load sensor: 3
  - HX711 ADC: 3
  - ACS758LCB-050B or similar hall sensor: 1

- Jumper Cables
- LED: 2
- Potentiometer:1 (Optional)

3 Installation

This section gives all the necessary steps to install the libraries and packages required.

3.1 PyGame

Pygame is a Python library ideal for building straightforward graphical user interfaces (GUIs) and 2D games. It offers simplicity and versatility, making it a preferred choice for interactive applications, particularly for smaller projects and beginners.
Before installing pygame, you must check that Python is installed on your machine. To find out, open a command prompt (if you have Windows) or a terminal (if you have MacOS or Linux) and type this:

```
python - -version
```

If a message such as “Python 3.8.10” appears, it means that Python is correctly installed. If an error message appears, it means that it is not installed yet. You must then go to the official website and follow the instructions.

Once Python is installed, you have to perform a final check: you have to see if pip is installed. Generally, pip is pre-installed with Python but we are never sure. Same as for Python, type the following command:

```
pip - -version
```

If a message such as “pip 20.0.2 from /usr/lib/python3/dist-packages/pip (python 3.8)” appears, you are ready to install pygame! To install it, enter this command in the terminal:

```
pip install pygame
```

3.2 Serial

The serial library, also known as pyserial, is a fundamental Python module for serial communication. It offers a straightforward interface to interact with serial ports, making it easy to read and write data in various serial protocols. Widely used for connecting to microcontrollers, sensors, and other hardware, pyserial is a versatile and essential tool for seamlessly integrating serial communication into Python projects.
To install the serial library in python, enter this command in the terminal:

```
pip install pyserial
pip install serial
```

3.3 Matplotlib

Matplotlib is a popular Python library for creating diverse and customizable visualizations. It's widely used for generating static, animated, and interactive plots, making it a key tool for data analysis and visualization in various fields.
To install the matplotlib library in python, enter this command in the terminal:

```
pip install matplotlib.pyplot
```

3.4 HX711 Library

The HX711 library is used to interface with the high-precision HX711 ADC that is used to retrieve data from load sensors. To install this library, perform the following steps:

1. Open the Arduino IDE
2. go to **Sketch** > **Include Libraries** > **Manage Libraries**. Alternatively, use the shortcut **ctrl + shift + i** or use the menu in the right side of the window.
3. search and install the library *HX711 by Rob Tillaart*

4 Circuit Diagram

To get the hardware up and running, follow the connections made in fig. 1.

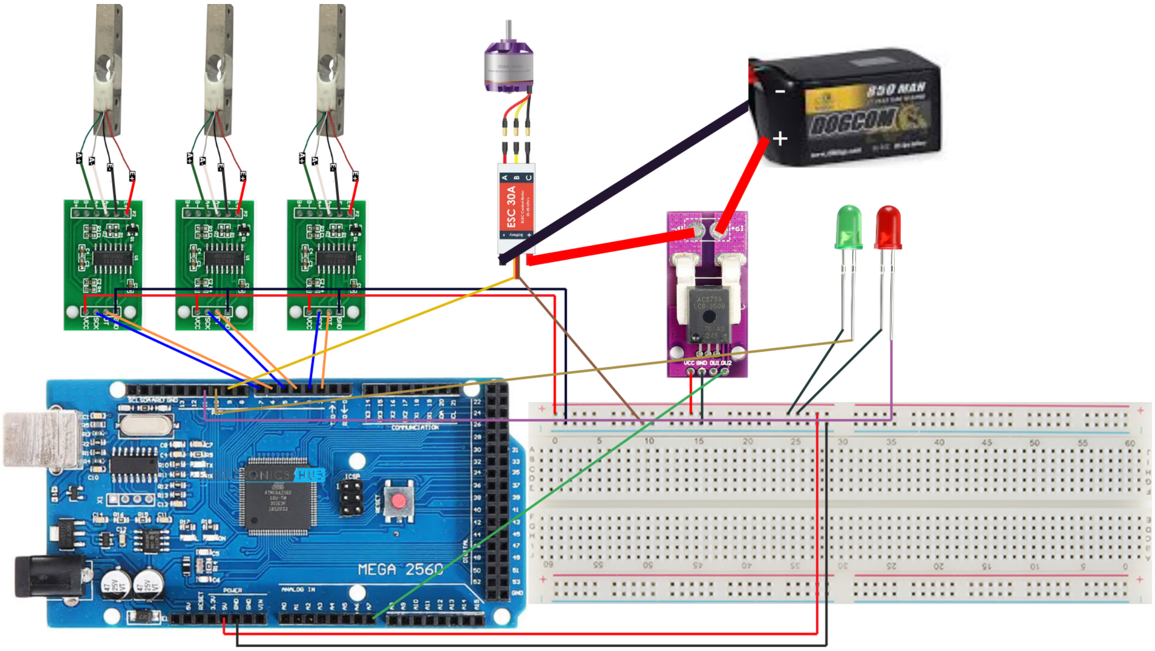

Fig. 1: Circuit Diagram

1. Load Sensor-HX 711 Interface
   - RED : E+
   - BLUE : E-
   - WHITE : A-
   - GREEN : A+
2. HX711-Arduino Interface
   - VCC : 5V
   - GND : GND
   - The SCL and SCK pins given below are for 3 sensors.
     - SCL : 2, 4, 6
     - SCK : 3, 5, 7

- ACS758-ESC interface
  - Connect positive of Battery to IP+ of ACS758.
  - Connect IP- of ACS758 to positive of ESC.
  - Connect negative of ESC to negative of Battery
- ACS758-Arduino Interface.
  - VCC : 5V
  - GND : GND
  - OU2 : A7
- LED-Arduino Interface
  - Connect short leg of LED to GND
  - Connect long leg of LED to Arduino Ports 10 and 11.

5 Calibration

Since the thrust bench uses physical sensors for data acquisition, they must be calibrated before usage. The source codes are provided in my GitHub page[1].

5.1 HX711 Calibration

Load cells can have slight manufacturing variations. Calibration helps account for these differences and ensures that the readings are accurate for each individual sensor.
Follow the steps to calibrate the load sensor.

1. Open and run the "Load_calibration" file.
2. Clear all weights from the load sensor.
3. Connect the load sensor to the Arduino and specify the DOUT and SCK pins in the calibration code. I have used pins 4 and 5 here but it will vary with respect to the pins used in your circuit.
4. Run the program without any weights on the scale.
5. Follow the instructions shown in the Serial Monitor.
6. Note down the number given by the serial monitor. This will be the calibration factor of the scale. This must be added to the source code before using the thrust bench
7. Repeat the same for all the 3 load sensors.

5.2 ESC Calibration

Calibration of the ESC separately is option as the calibration is done directly in the source code. If calibration is necessary for validation, make the connections as shown in the circuit diagram, run the code and follow the instructions shown in the serial monitor.
This calibration is taken from ELECTRONOOBS and the website link is embedded *here*. [2]

[1] https://github.com/AnirudhhanR/Thrust-Bench

[2] https://electronoobs.com/eng_robotica_tut5_1_code1.php

6 Running The Program

Once the prerequisites are completed, the source code must be updated. Follow the steps to complete the setup.

1. Open the "ThrustBench_Source" file.
2. Replace the calibration factors in the code with the values you obtained from calibration. This should be done in the "set_scale" function present in line 106, 109 and 112.
3. Upload the code to the Arduino and take note on the COM Port the board is connected to.
4. open the "main.py" file.
5. Replace the port in line 5 with the COM Port your Arduino is connected to.
6. Run the python file.

NOTE:

- Make sure the Arduino source code is uploaded before running the python file.
- While running the python file, make sure no instances of the Arduino IDE is open.