Hangboard Manual

Introduction

Other interesting project are [PiClimbing] and [ArduinoHangboard].

What you need

- Any hangboard (large list of supported hangboards below).
- A Raspberry Pi, force sensors and some basic skills to setup the software backend (no automation so far).
- Basic skills to create a board mount with the force sensors.
- Any mobile device (iOS / Android / WebApp) and some basic skills to deploy the debugging app (no Store so far)

Features

- Smart exercise timer
- Uses preexisting exercise files - easily extendable
- Measures hangtime, applied force, rate-of-force development, maximal load etc

List of implemented hangboards

- Beastmaker 1000
- Beastmaker 2000
- Cliffboard Mini
- Crusher 3
- Linebreaker Base
- Metolius Prime
- Metolius Project
- Metolius Simulator 3D
- Metolius Wood Grips 2 Compact
- Monster
- Mountain Rocks
- Redge Port
- Roots Baseline

- Simond Ballsy Board
- Topout Project
- Zlagboard Evo
- Zlagboard Mini

TIP	Your hangboard is not supported yet? It can be added easily. Just open a ticket: github.com/8cH9azbsFifZ/hangboard/issues/new

Software Design

This is a brief design layout of the project.

Frontend

- Web client (Running on the backend Raspberry Pi)
- iOS App
- Android App

Backend

- Running on a Raspberry Pi.
- Communicating to the frontend using websockets.

Software Used

- Flutter for the frontends
- Python backends
- Websockets for Communication
- JSON for Board configuration and finger grip positions
- SVG Layers for hold configuration

Currently inactive frontend implementations are - Python Flask for Web App - React Native for App

Hardware Design

- Raspberry Pi Zero W
- Sensors: as listed below

Force Sensors with HX711

Load cells are available widely with the HX711 signal amplifier module as a package [HX711LoadCellPackage]. We will use one of these packages as the force measurement sensors. The python module [HX711PythonModule] is slightly modified and contained in the backend sources.

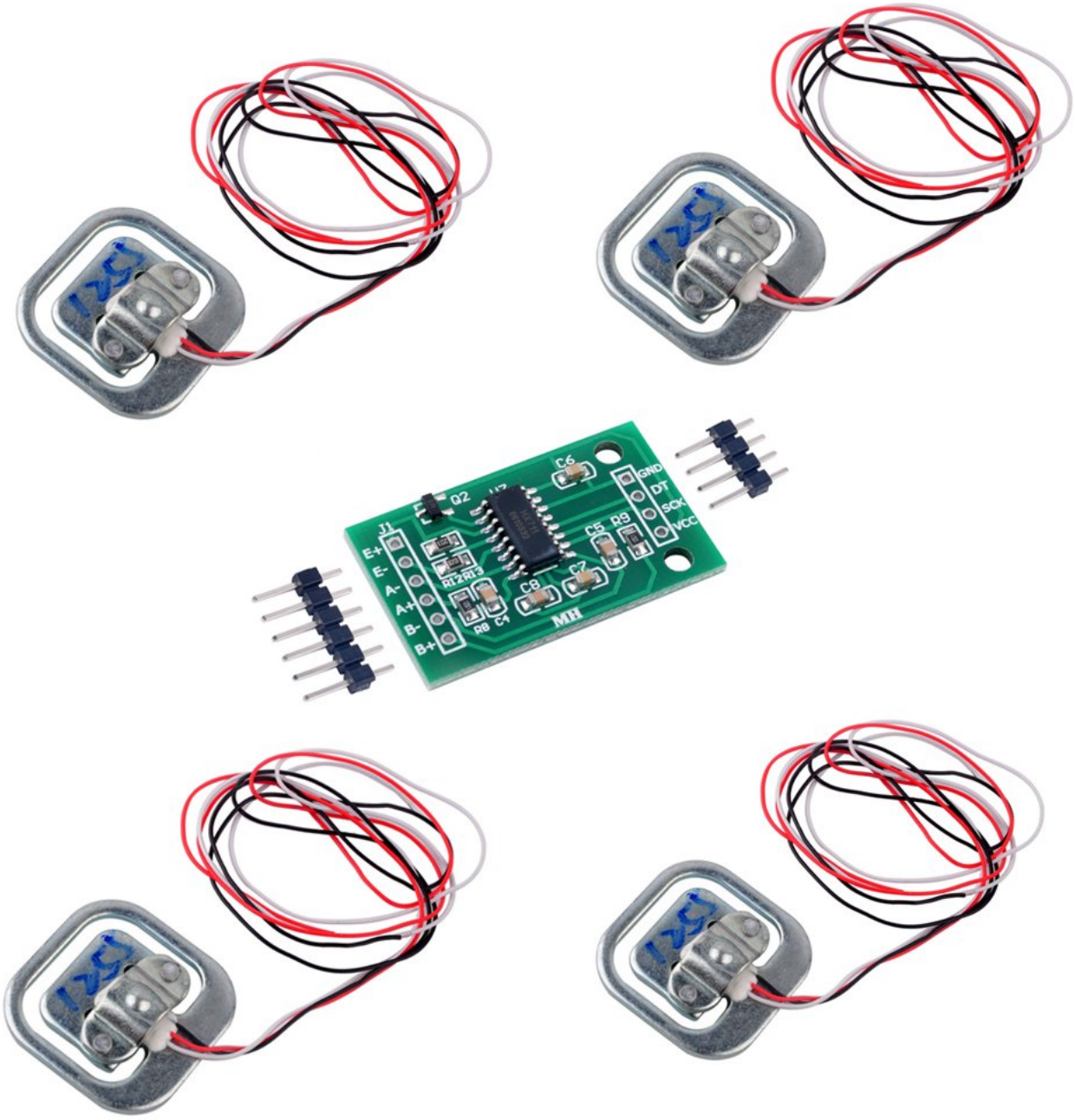

Figure 1. The HX711 with 4 load cells

- HX711 analog-to-digital converter [HX711Datasheet]
- Load Cells

NOTE	Some HX711 modules have a wrong grounding according to the application sheet: github.com/bogde/HX711/issues/172. This can be fixed with a small solder bridge.

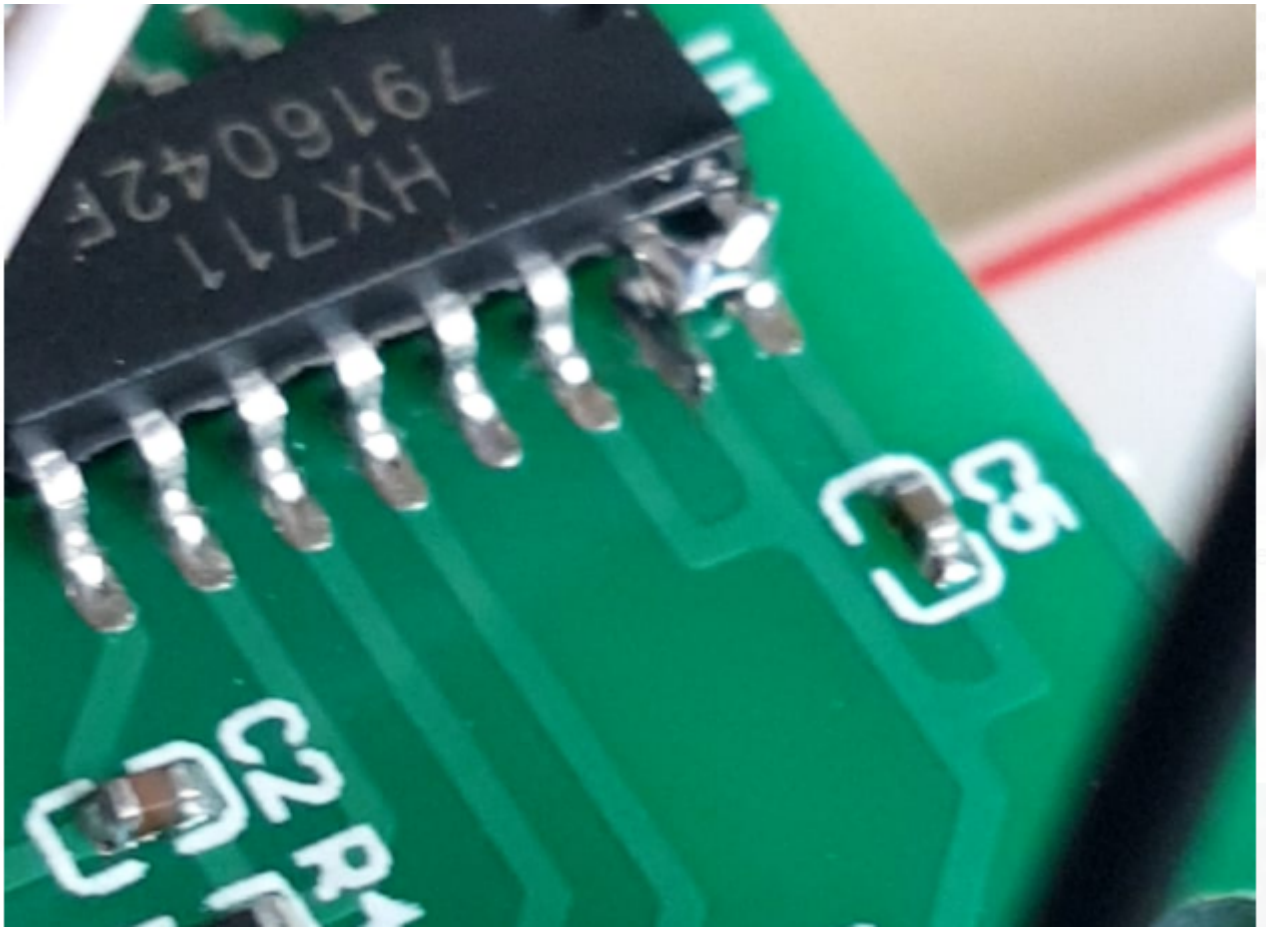

Figure 2. The HX711 Fix

Wire the HX711 module to the Raspberry Pi as follows:

Raspi GPIO	**Module**	**Module Pin**
3v3	HX711	Vcc
GPIO17	HX711	DT
GPIO27	HX711	SCK

Wire the 4 load cells as follows (according to the application sheet):

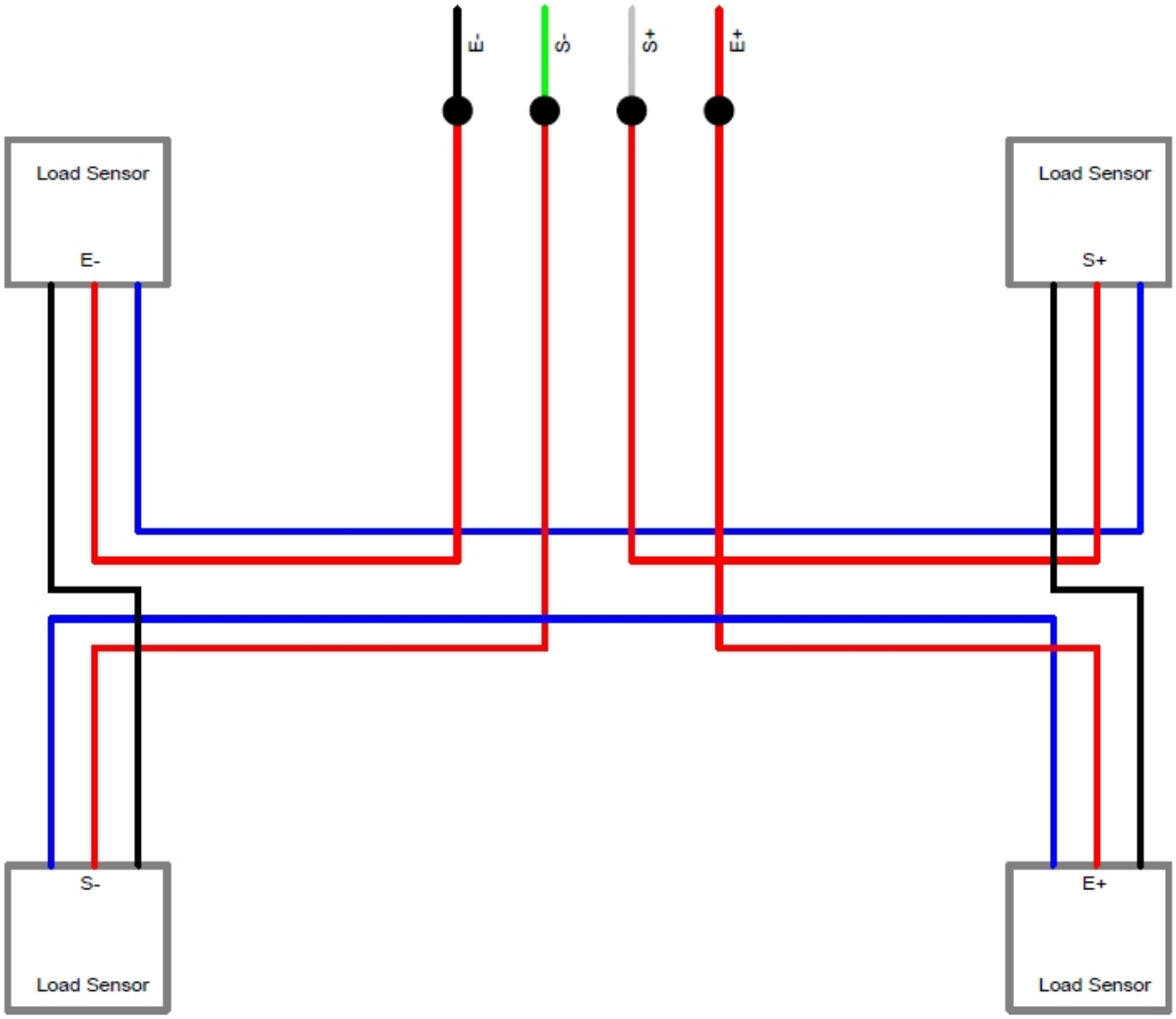

Figure 3. Wiring four load cells

Mounting the load sensors

Mounting the load cells in a zlagboard

1. Disassemble the 4 screws and the gyroscope mount
2. Place the 4 load cells at bottom
3. Create small "U-shaped" holds for the load cells (i.e. made from paper)

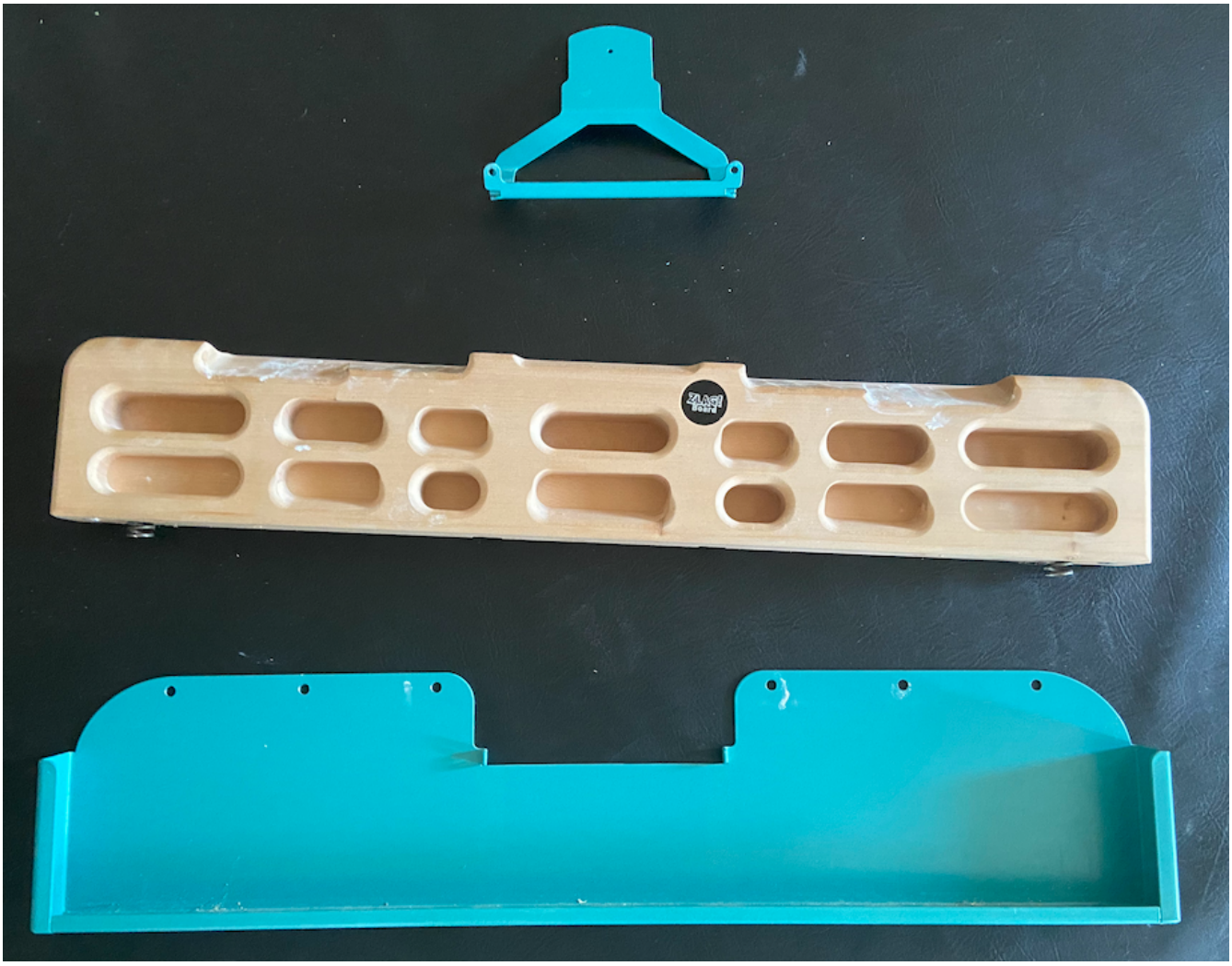

Figure 4. Zlagboard disassembled

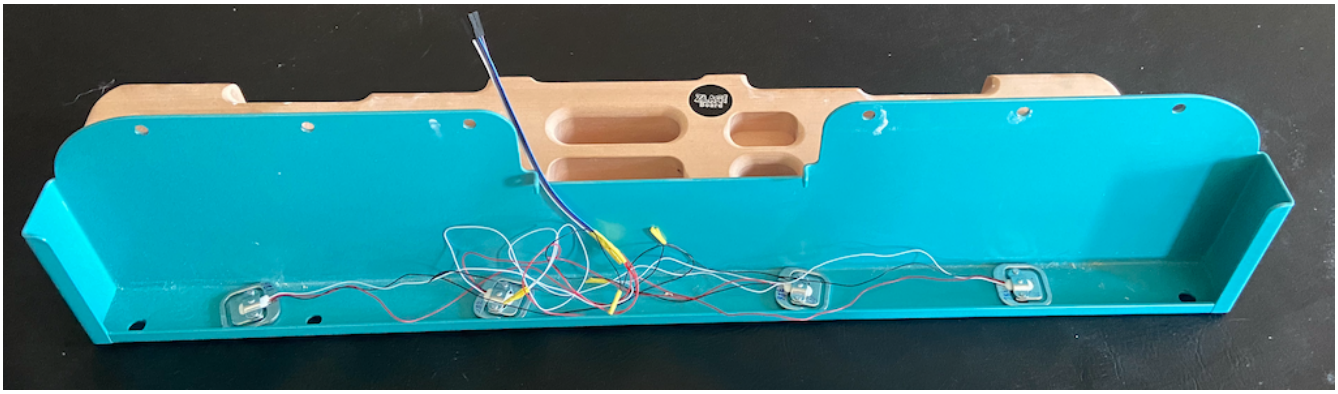

Figure 5. Zlagboard with load cells

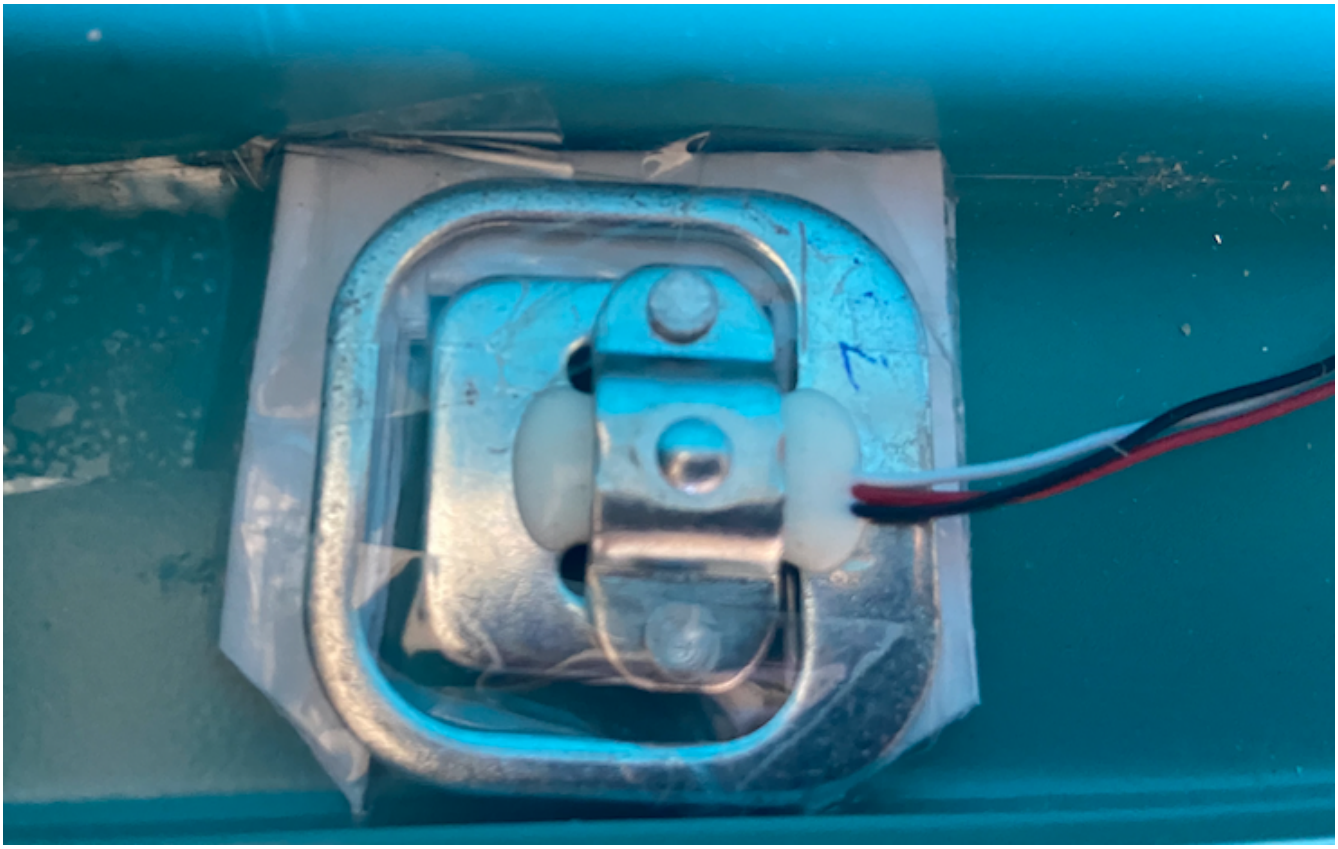

Figure 6. U-Shaped load cell mount

NOTE	Gyroscope mount disabled after placing the load cells…

Mounting the load cells for any existing hangboard

Any hangboard can be mounted on a wooden construction with the 4 load cells in between. This will provice force measurements for any existing hangboard.

An example construction of a hangboard mount is given here: Mount for Isometrix Board [ArduinoHangboard].

Figure 7. Mount for Isometrix Board [ArduinoHangboard]

Gyroscope Sensor: MPU-6050

Without further modifications a gyroscoope sensor can be mounted on an existing Zlagboard. Hangs can be measured with the gyroscope, too. We will use the widely used MPU6050 package [MPU6050Datasheet] with excellent documentations [MPU6050GettingStarted]. Obviously there will be noise in the measurements, so for accurate measurements in our setup a kalman filter is implemented in the backend, based on this implementation [MPU6050KalmanFilter].

CAUTION	Force measurements are not possible without the load cells.
NOTE	Modules with BLE are existing for further / future developments [MPU6050BLEVersion].

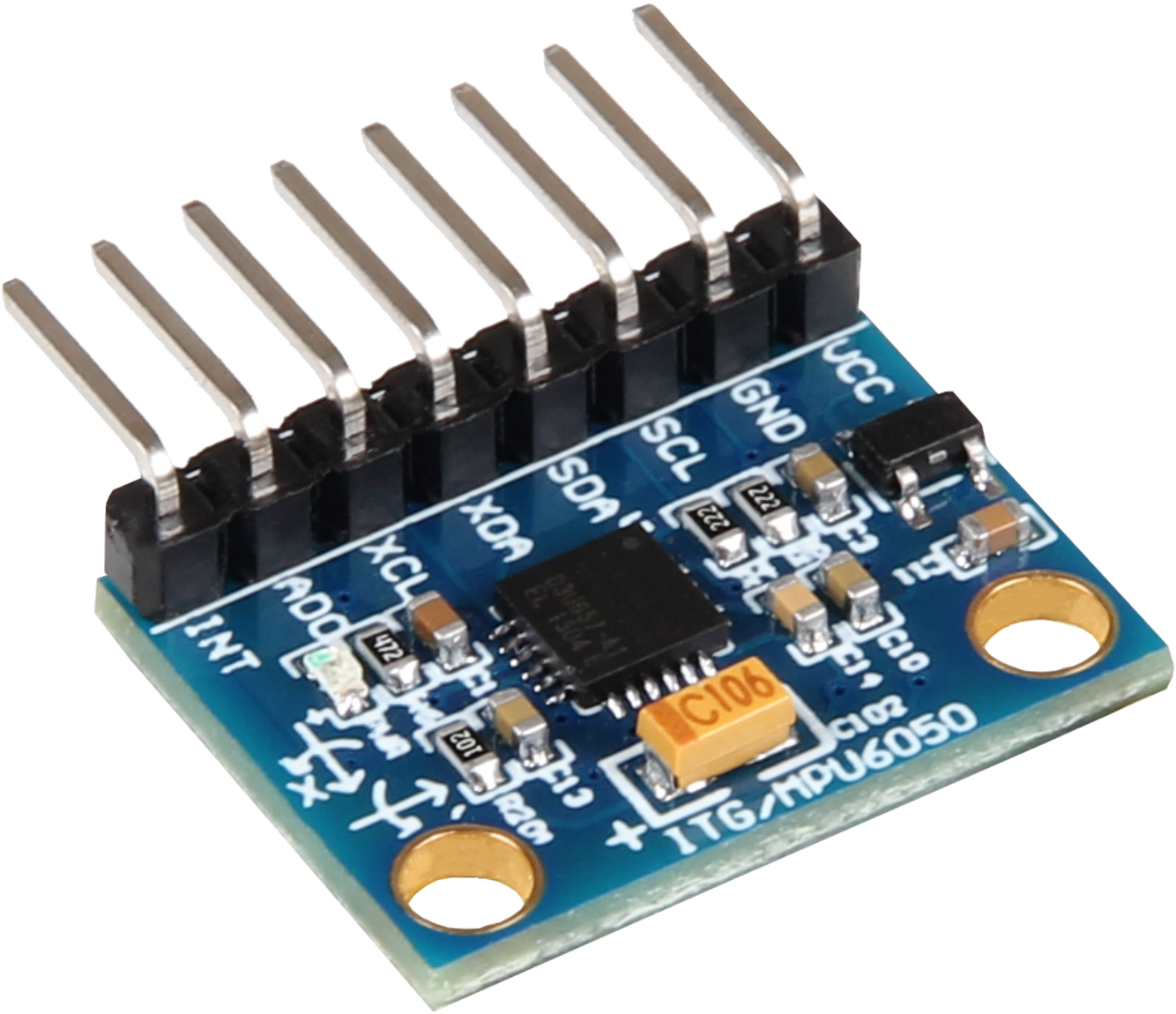

Figure 8. Sensor MPU-6050

Wire the Gyroscope sensor to the raspi as follows:

Raspi GPIO	**Module**	**Module Pin**
Pin 1 (3.3V)	MPU 6050	VCC
Pin 3 (SDA	MPU 6050	SDA
Pin 5 (SCL)	MPU 6050	SCL
Pin 6 (GND)	MPU 6050	GND

For getting started with the software for the Gyroscope, follow these steps

1. Enable I2C I/O `sudo sed -i 's/\#dtparam=i2c_arm=on/dtparam=i2c_arm=on/g' /boot/config.txt`
2. Load the user space module `grep i2c-dev /etc/modules ||echo i2c-dev |sudo tee -a /etc/modules`

3. Install I2C tools `sudo apt-get -y install i2c-tools`
4. Reboot `sudo reboot`
5. Check whether 68 exists in `sudo i2cdetect -y 1 | grep 68`

Distance sensor HC-SR04

WARNING	This sensor is not yet fully implemented in the backend.

For measuring distances (i.e. for pullups) we will use a HC-SR04 ultrasonic distance sensor [HCSR04Package]. There is excellent documentation on how to getting started [HCSR04GettingStarted]. For accurate measurements a kalman filter is implemented in the backend based on [HCSR04KalmanFilter].

Figure 9. Sensor HC-SR04

Wire the distance sensor to the raspi as follows:

Raspi GPIO	Module	Module Pin
Pin 2 (VCC)	HC-SR04	VCC
Pin 6 (GND)	HC-SR04	GND
Pin 12 (GPIO18)	HC-SR04	TRIG
	R1: 330Ω	ECHO
Pin 18 (GPIO24)	R1: 330Ω	
	R1: 330Ω	R2: 10kΩ
Pin6 (GND)		R2: 10kΩ

Training plans, Workouts, Exercises and Sets

The following definitions will be used:

Training Plan

A series of workouts, usually executed with at least of one day rest in between.

Workout

A series of exercise sets.

Excercise

A single exercise, i.e. hang, maximal hang, pull up, assisted pull up.

Set

A set of exercises with **Repetitions**, **Pause** between the exercises and a **rest to start** pause.

References

- [Website] Hangboard website: 8ch9azbsfifz.github.io/hangboard/
- [Discussions] Hangboard discussions: github.com/8cH9azbsFifZ/hangboard/discussions
- [Issues] Hangboard issues: github.com/8cH9azbsFifZ/hangboard/issues
- [PiClimbing] Raspi W Zero Hangboard: github.com/adrianlzt/piclimbing
- [ArduinoHangboard] Arduino Hangboard: github.com/oalam/isometryx
- [HX711Datasheet] HX 711 Datasheet: github.com/8cH9azbsFifZ/hangboard/raw/main/doc/force/hx711_english.pdf
- [HX711LoadCellPackage] Package of HX711 module and 4 load cells: www.amazon.ca/Bridge-Digital-Amplifier-Arduino-DIYmalls/dp/B086ZHXNJH

- [HX711PythonModule] The python modules for HX711: github.com/tatobari/hx711py or github.com/gandalf15/HX711/
- [MPU6050Datasheet] MPU 6050 Datasheet: github.com/8cH9azbsFifZ/hangboard/raw/main/doc/gyroscope/MPU-6000-Register-Map1.pdf
- [MPU6050KalmanFilter] Kalman filter implementation for MPU 6050: github.com/rocheparadox/Kalman-Filter-Python-for-mpu6050
- [MPU6050GettingStarted] Getting started with MPU6050 measurements: tutorials-raspberrypi.de/rotation-und-beschleunigung-mit-dem-raspberry-pi-messen/
- [MPU6050BLEVersion] MPU6050 BLE module: github.com/fundiZX48/pymotiontracker
- [HCSR04Package] HC-SR04 package: www.amazon.de/AZDelivery-HC-SR04-Ultraschall-Entfernungsmesser-Raspberry/dp/B07TKVPPHF/
- [HCSR04GettingStarted] Getting started with distance measurements using the HC-SR04: tutorials-raspberrypi.de/entfernung-messen-mit-ultraschallsensor-hc-sr04/
- [HCSR04KalmanFilter] Implementations of kalman filters for the HC-SR04 module: github.com/rizkymille/ultrasonic-hc-sr04-kalman-filter and github.com/NagarajSMurthy/Kalman-estimation-of-ultrasonic-sensor