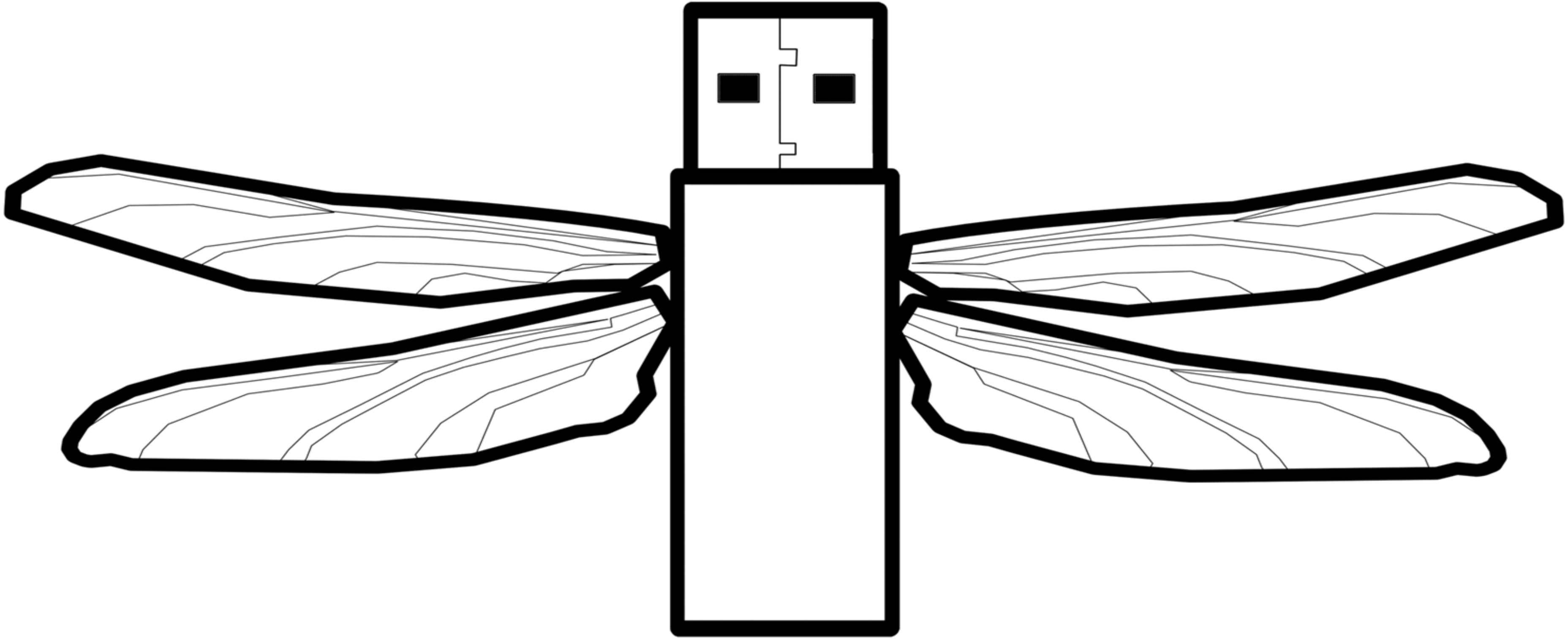
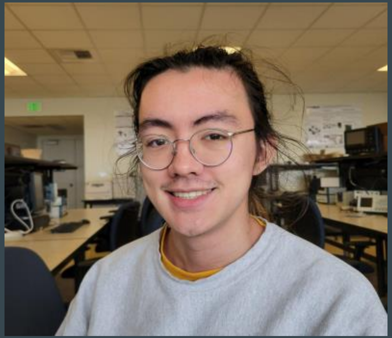
# Project Dragonfly



UCSB CE CAPSTONE 2022-2023

# Development Team

Danny Cardenas



Teagan
Connon

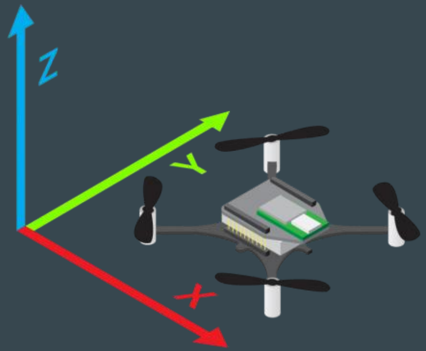
Noah
Lutz

Jesus Oviedo

Marko Ristic

# What is Project Dragonfly?

- Project Dragonfly serves as a way to consolidate several low-profile sensors into a single, peripheral device, which attaches via USB to an drone in order to provide an estimation of state while keeping the device as small as possible
- In doing so, we hope to create a modular, more cost effective way of providing state estimation, reducing the individual sensor configuration workload for drone manufacturers and hobbyists



# Project Overview

| Hardware | Firmware | Software |
| --- | --- | --- |
| | | |

# Project Overview

| Hardware | Firmware | Software |
| --- | --- | --- |
| PCB to connect all necessary data lines and signals to MCU from sensors as well as power | | |

# Project Overview

| Hardware | Firmware | Software |
|---|---|---|
| PCB to connect all necessary data lines and signals to MCU from sensors as well as power | Manages data-ready signals from sensors<br><br>Packages data w/ timestamp and sends over USB | |

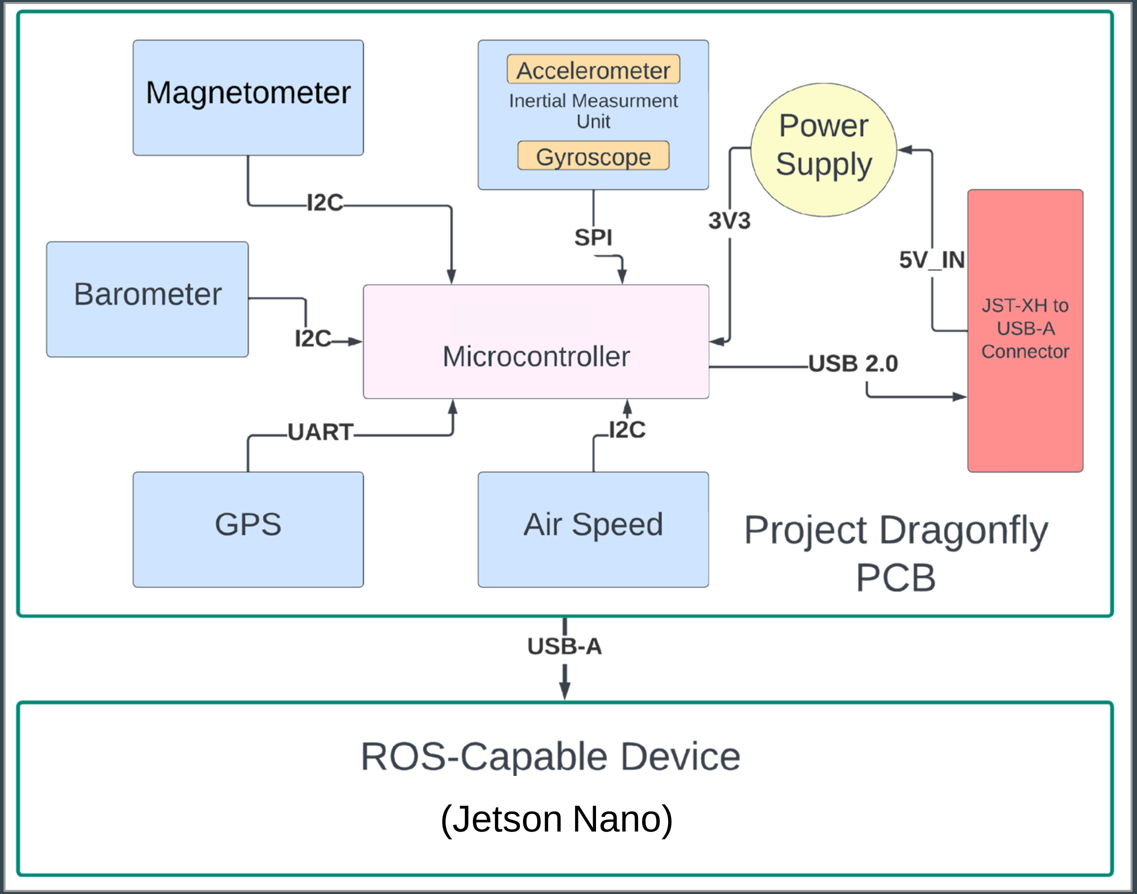
# Project Overview

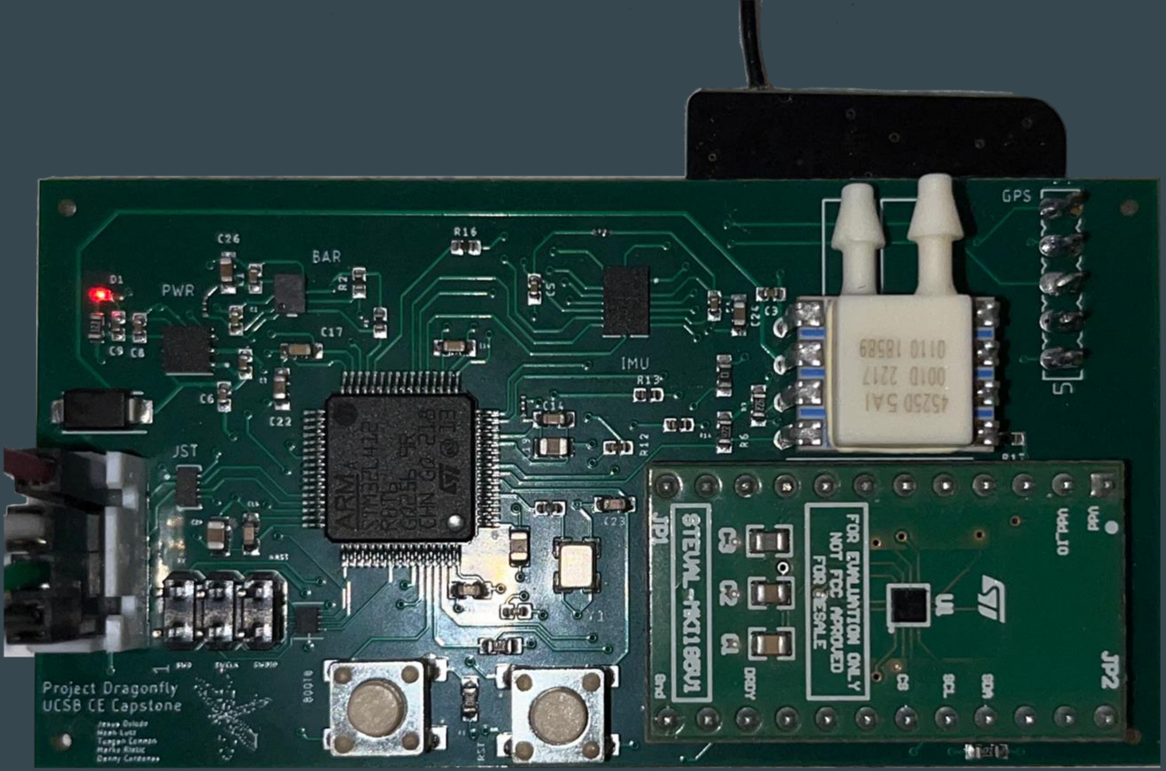
| Hardware | Firmware | Software |
|---|---|---|
| PCB to connect all necessary data lines and signals to MCU from sensors as well as power | Manages data-ready signals from sensors<br><br>Packages data w/ timestamp and sends over USB | Retrieves data and implements ROS for virtual drone flight visualization as well as state estimation. |

# Hardware Overview

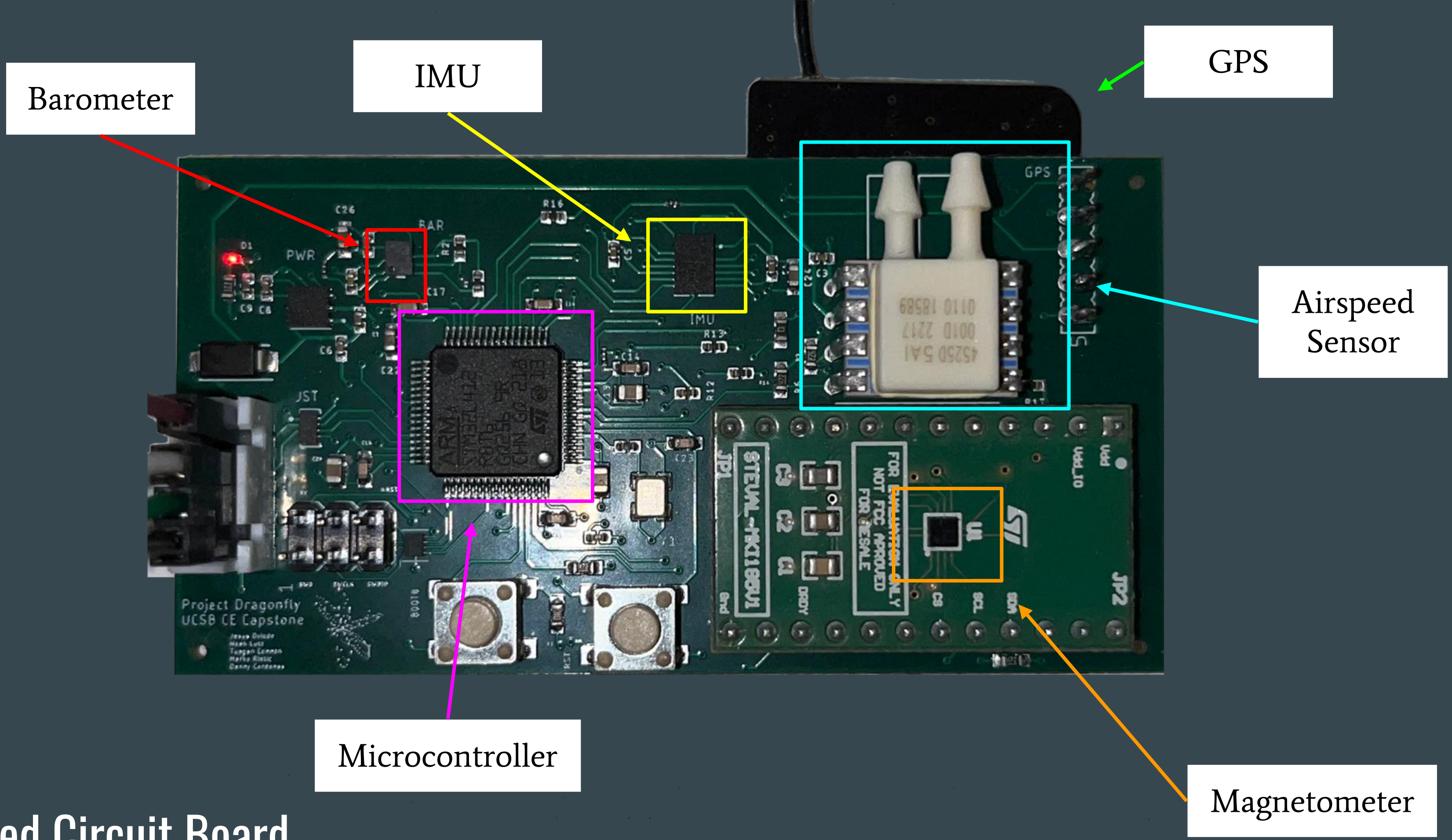
- Block Diagram
- Printed Circuit Board Overview & Challenges
- Sensor Functionality & Motivation

# Block Diagram





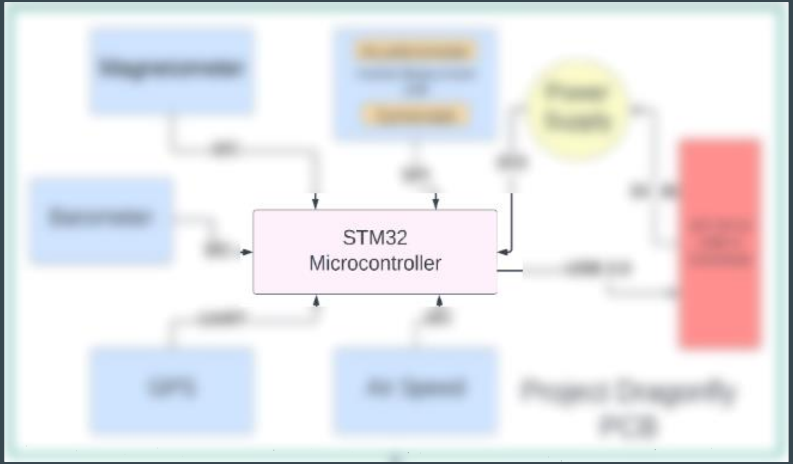

# Printed Circuit Board

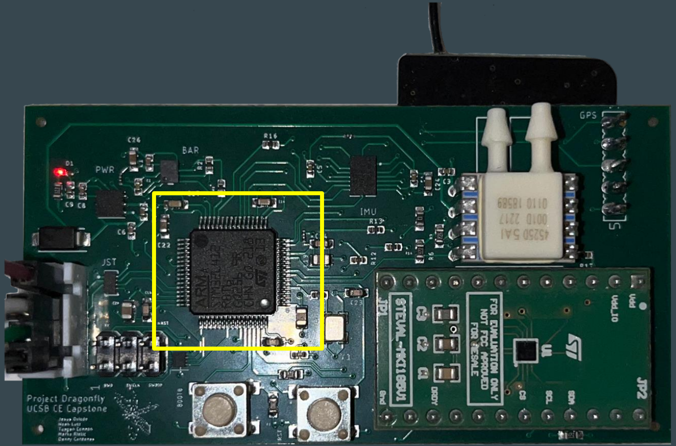



# Printed Circuit Board

# Microcontroller Specifications

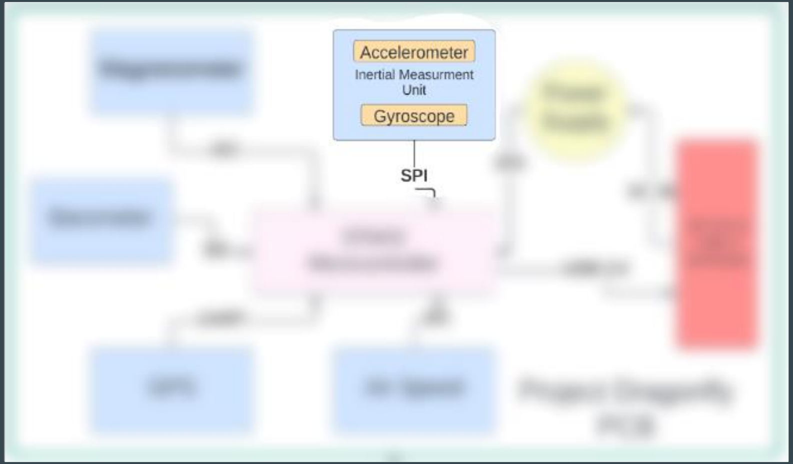
- STM32L412RBT6
  - 64 pins
  - 10mm x 10mm
  - 128Kb Flash
- Supported Communication Interfaces
  - 3x I2C
  - 3x USART
  - 2x SPI
- Price: $4.13

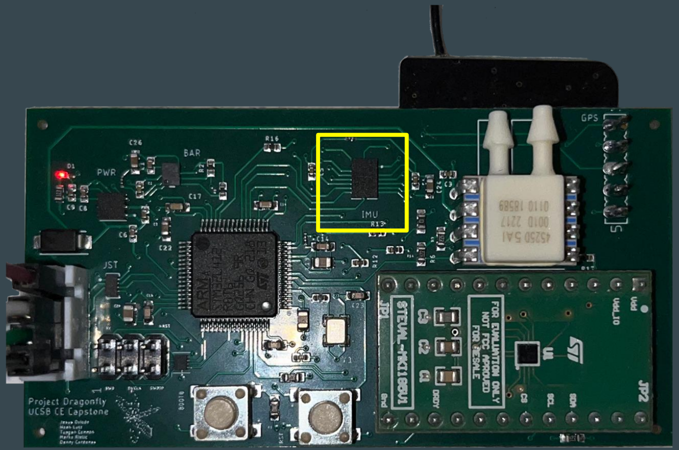


# Inertial Measurement Unit -BMI088

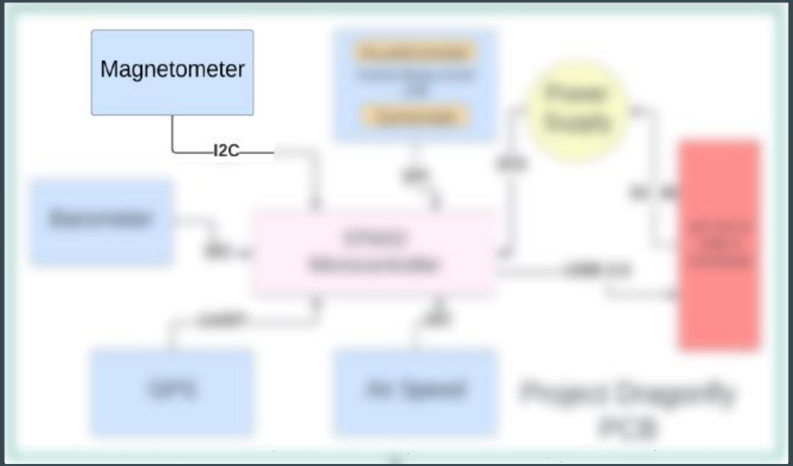
- Accelerometer:
  - 16-bit precision
  - ±2, ±4, ±8 or ±16 g range
- Gyroscope:
  - 16-bit precision
  - ± 125°/s, ± 250°/s, ± 500°/s,± 1000°/s, or ± 2000°/s range
  - Data Output Rates: 12.5 Hz ... 2 kHz
- SPI protocol
- Dimensions: 3.0mm x 4.5mm x 0.95mm
- Price: $3.46

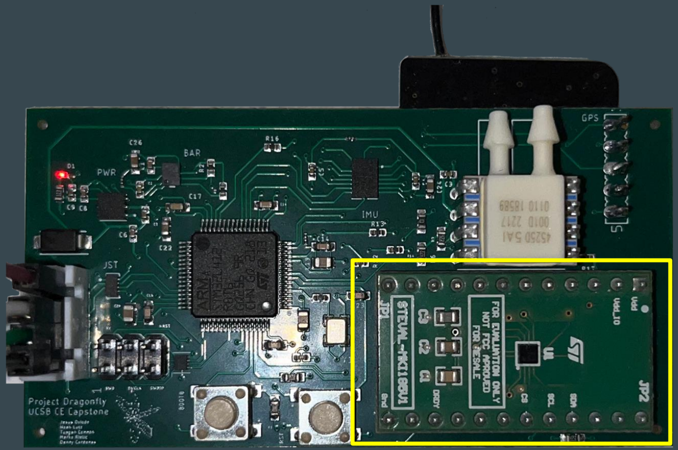


# Magnetometer - IIS2MDCTR

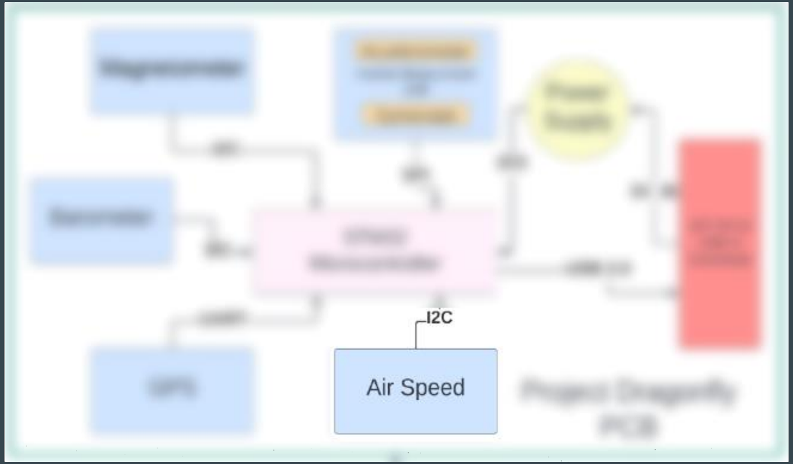
- Gives 3-axis digital magnetic direction
- I2C protocol
- 16 bit data output
- Price: $2.82

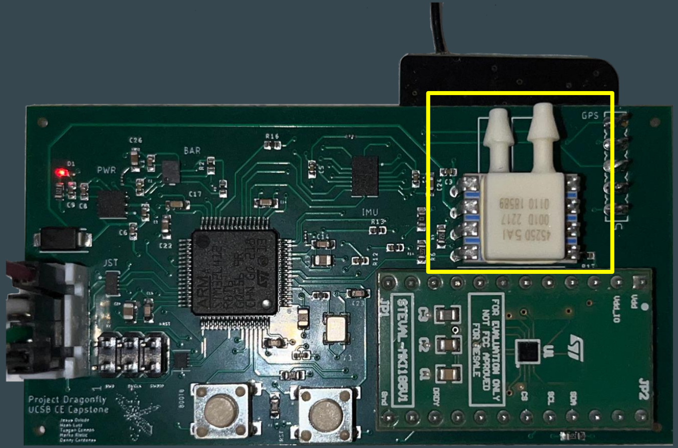


# Air Speed Sensor - 45525DO

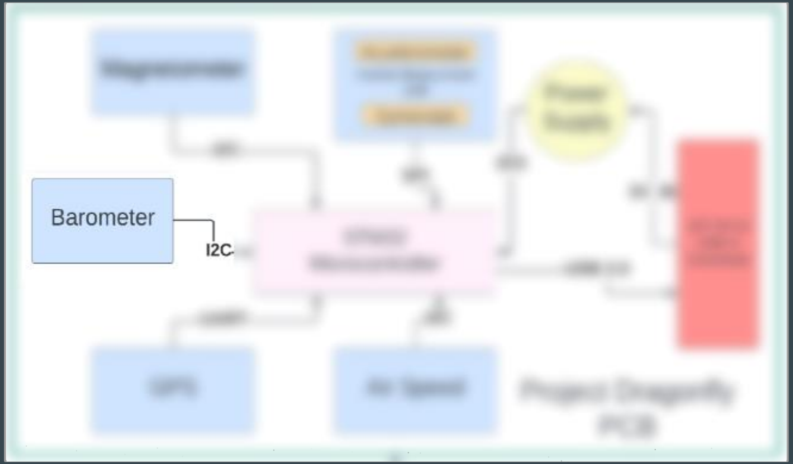
- Differential pressure sensor that is used to find airspeed
- I2C protocol
- Output: 14 bit differential pressure, 11 bit temperature
- V = ½ (K(Δ p))^1/2
- Accuracy: ±0.25% of span
- Dimensions: 24.7mm x 16.8mm
- Price: $72.25

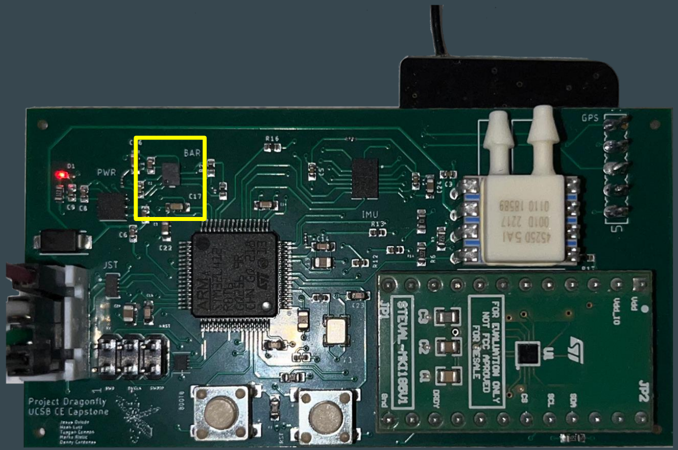


# Barometer
# - DPS310XTSA1

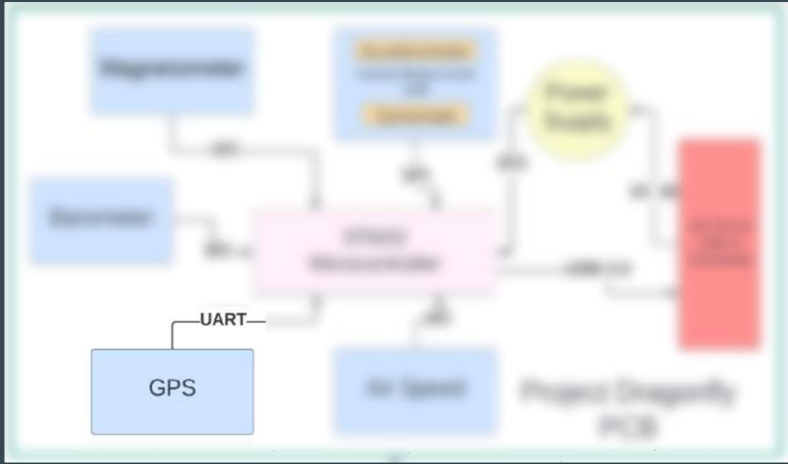
- Temperature and pressure readings used to calculate altitude
- I2C protocol
- 24-bit data output
- Accuracy: ±0.06 hPa ±0.5°C
- Dimensions: 2mm x 2.5mm x 2mm
- Price: $2.83

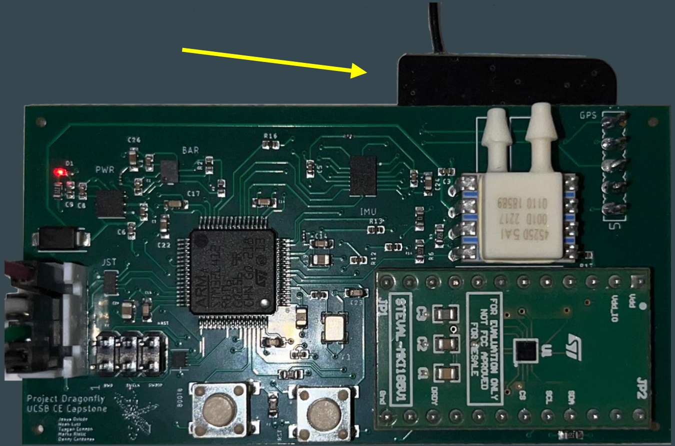


# GPS
# - NEO 6M

- Outputs Latitude and Longitude of current position
- UART protocol
- Accurate within 2.5 meters
- 5 Hz update rate
- Optionally get current speed
- Price: $10.99



# Firmware Overview

- USB Data Packet Structure
- Software Flow Diagram
  - Sensor Timing & Interrupt Handling
  - USB Data aggregation
  - USB packet timing

# USB Packet Header Fields

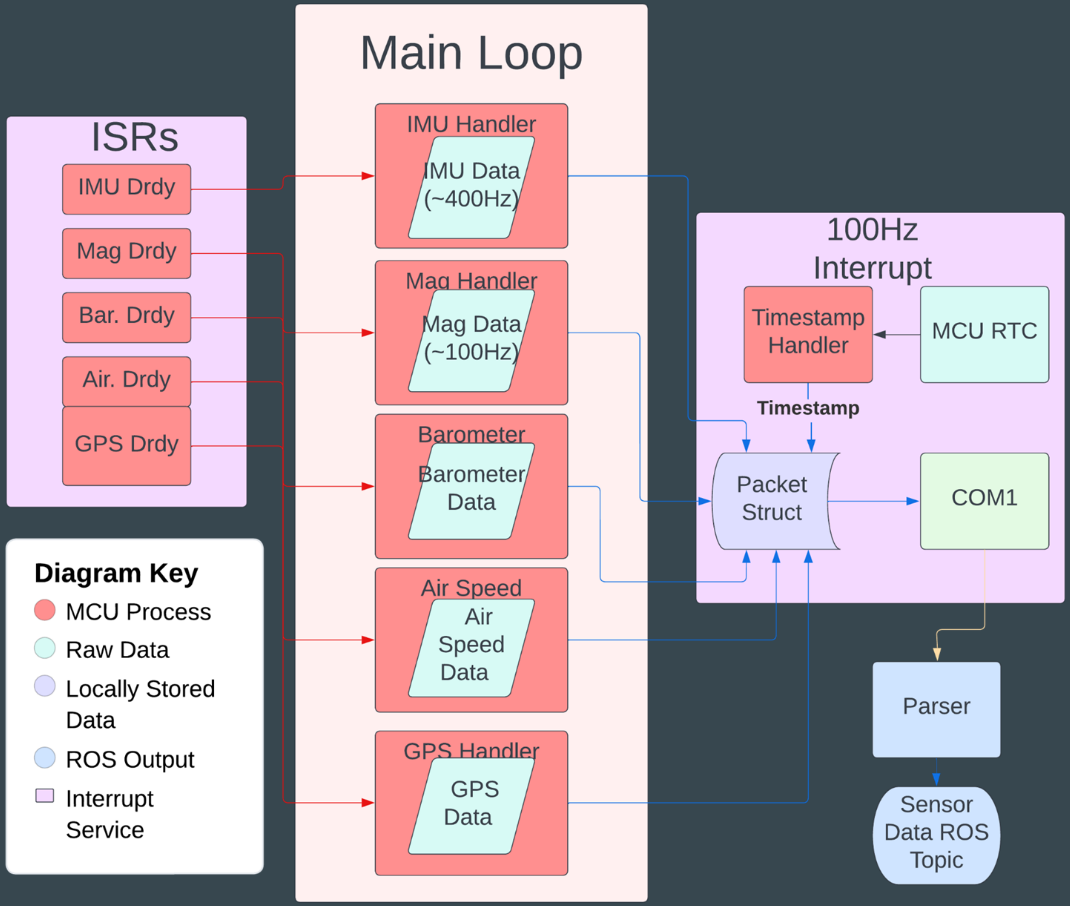
| Device ID | | | | | | | |
|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |

| Stale Bits | | | | | | | |
|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| X | X | IMU Accel | IMU Gyro | Mag | Airspeed | Barometer | GPS |

# General USB Packet Structure

| | | | | | | |
|---|---|---|---|---|---|---|
| Device ID | Stale Byte | IMU Data | Mag Data | Barometer Data | DP Data | GPS Data |

- IMU data: 24 bytes
- Magnetometer Data: 12 bytes
- Barometer Data: 12 bytes
- Differential Pressure Data: 8 bytes
- GPS Data: 8 bytes
- Total Packet Size: 72 bytes




# Software Flow

# Software Overview

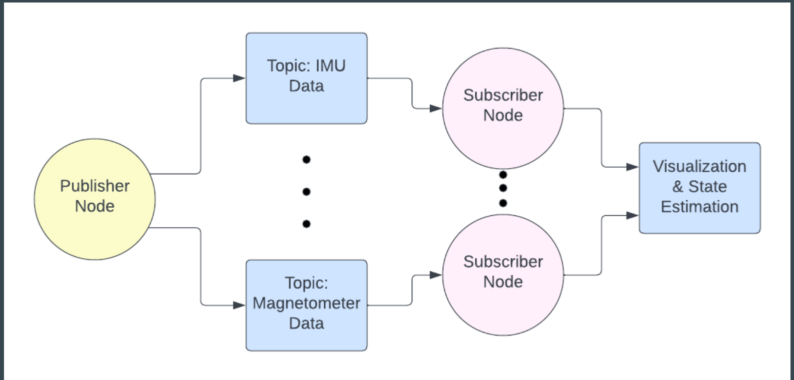
- ROS Background
- ROS Overview
- State & State Estimation Background
- Using robot_localization and RViz for Visualization

# What is ROS (Robot Operating System)?

- NOT an operating system
- Set of software frameworks for robot software development
- Open-sourced collection of software libraries and pre-built packages
- Used for robot functionality
    - Control, perception, simulation, state estimation, etc

# How We Use ROS

- The data packet reaches Jetson Nano via USB
- Use ROS to publish data for each sensor as its own topic
- Subscriber nodes subscribe to the sensor they wish to read from
- Data is sent to the visualization setup for the virtual drone to mirror the movement of the PCB
- Robot_localization nodes utilize sensor data to perform state estimation

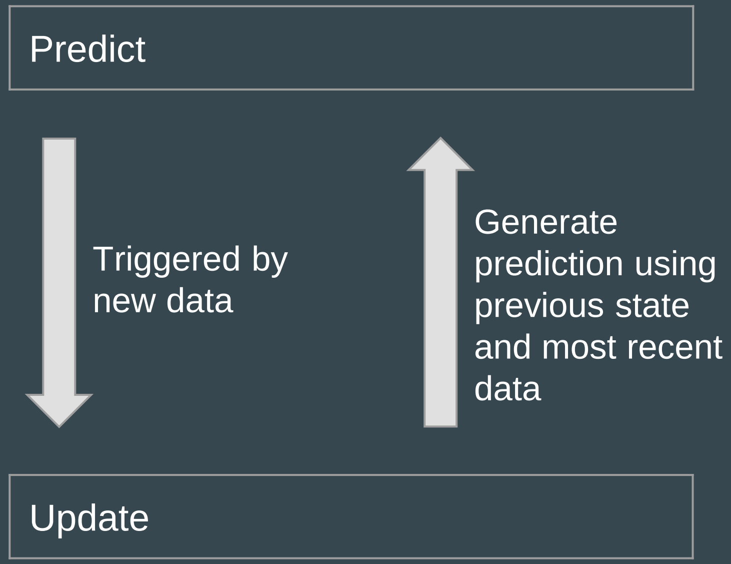


# What is State?

- In UAV systems, the state (x) of the aircraft is represented by position and orientation, as well as its first and second order derivatives
- We define $x \in R^{5x3}$,
    - $x = (\langle x,y,z \rangle, \langle yaw, pitch, roll \rangle, d\langle x,y,z \rangle/dt, d\langle yaw, pitch, roll \rangle/dt, d^2\langle x,y,z \rangle/dt^2)$
- We use an Extended Kalman Filter to process noisy sensor data and create an estimation of state

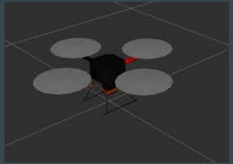
# State Estimation

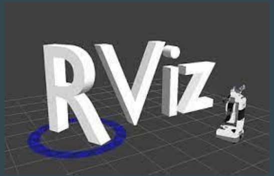
- Our device performs state estimation using an Extended Kalman Filter, which seeks to minimize the equation that defines our state estimation covariance
- The algorithm works in 2 phases: predict & update
- Using robot_localization ROS library

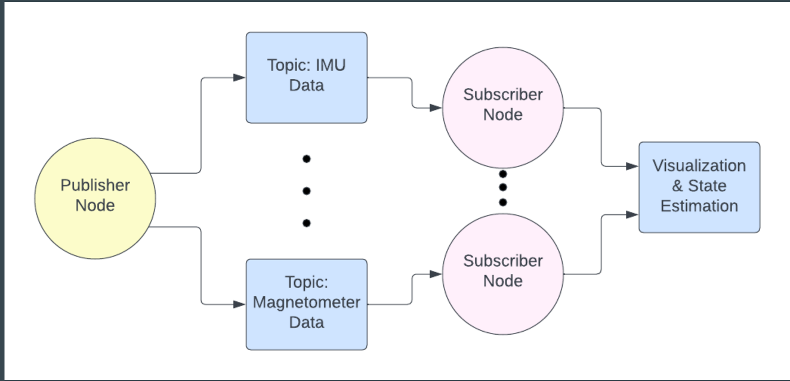


# How We Use ROS (cont.)

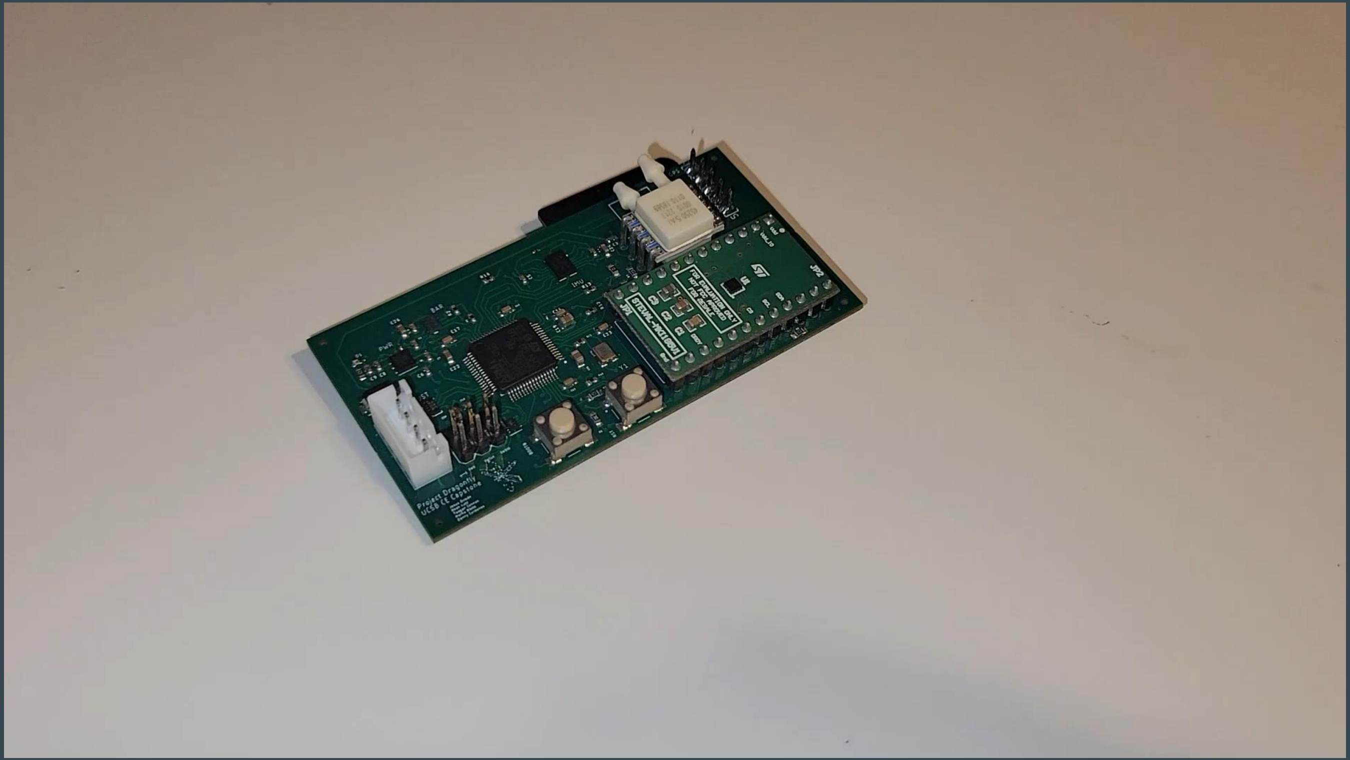
- Open-source and readily available ROS Melodic packages
  - robot_localization EKF and NavSatTransform Nodes
  - ROS IMU+Mag Filter Node
  - RViz visualization of state
  - Standard ROS message formats (IMU, MagneticField, NavSatFix, etc.)





# DEMO VIDEO



## Acknowledgements

Dr. Yogananda Isukapalli

Jimmy Kraemer

Alex Lai

Venkat Krishnan

Phil Tokumaru

Tiziano Fiorenzani

Warren Ward

AeroVironment

# Thank You

Team Sponsor

AeroVironment™

- Phil Tokumaru
  - AeroVironment Project Advisor

# Any Questions?