## Flight Controller Data Acquisition

The following steps guide you to establish the communication between the flight controller and the ESP32 board:

1. Plug your flight controller to the computer, then open Betaflight Configurator.
2. Go to the **Port** tab, turn on MSP on **UART2**, click Save and Reboot on the bottom left corner (**Do not turn off the first MSP**).
3. Upload the *MSP\_Communication\_Example.ino* onto the ESP32 board (UM Tinypico), then connect the UART pin to the board.

**Green -> TX, Yellow -> RX**

1. Turn on the serial port, you should be able to see the throttle commands.

Tasks:

1. Go through the MSP library, try implementing your own code to get **MSP\_ATTITUDE** and **MSP\_RAW\_SENSOR**
2. Request different messages at the same time, how does the refreshing rate changes?

## Getting data from external sensor

Racing drones don’t typically come with many sensors, yet things like range sensor would be very helpful in autonomous flights. The testing deck we provided comes with a ToF sensor as an example. Connect **Blue**(SDA) and **White**(SCL) wire to the board on pin

Tasks:

1. Upload *ToF\_Sensor\_Example.ino* onto the ESP32 board, check data input using Serial port/Serial plotter.
2. Combine *MSP\_Communication\_Example.ino* and *ToF\_Sensor\_Example.ino*, show or
3. plot ToF sensor data and Accelerometer data together.

## Communicating with ground station

It’s crucial to collect and log onboard data / establish offboard control when needed. ESP32 comes with a Wi-Fi module which is helpful on communication with the ground station(laptop). We use UDP protocol to establish the communication between the board and local host.

Tasks:

1. Based on the provided *UDP\_Example\_ESP32* and *UDP\_Example\_Python*script, try establishing communication between the ESP32 board and the PC. You will need to download **Telelink.py** and configure your Wi-Fi hotspot to do this.
2. Modify the ESP32 code based on your Wi-Fi setting, try send the telemetry data back to your PC.
3. Try send data to the ESP32 board. You can confirm that either using the Serial Port or with the LED light.

# Board Pinout:

