Troubleshooting and Other Tips

This document outlines the key software-related issues encountered during the development and deployment of the unified monitoring system, particularly those related to configuring and interfacing with sensor nodes and firmware flashing on microcontrollers. Each problem is accompanied by practical, tested solutions aimed at helping future users or developers quickly overcome similar challenges. The goal is to provide a clear and reusable reference to streamline the setup and debugging process, ensuring smoother integration and operation of the system.

This document is organized by hardware platform and topic, providing step-by-step solutions to common software and connection issues encountered when working with the different technologies used in this project.

# ESP8266 Issues

**If you get either of the following error messages: "A fatal esptool.py error occurred: Cannot configure port, something went wrong. Original message: PermissionError(13, 'A device attached to the system is not functioning.', None, 31)" OR "Port monitor error: command 'open' failed: Invalid serial port. Could not connect to COM3 serial port"**

* Refer to https://learn.sparkfun.com/tutorials/how-to-install-ch340-drivers/all . Follow the article or their YouTube video on the subject. This guide provides a clean installation of the CH340 driver and an easy to follow tutorial on how to install it. From what I've gathered from several forum posts, those errors happen due to a recent (post 2019, as the older posts are from that year) driver update that messes up the connection between PCs and this specific Arduino model. I'm using version 3.4.2014.8 and it seems to solve that issue and this is the version that is automatically installed using Sparkfun's installer. The driver problem seems to be reoccurring. If the ESP8266 starts printing squares/gibberish again, simply run the driver installer again and repeat the steps.

**If you can't upload your code to the ESP8266 D1 mini R2 (output prints out "Connecting ......\_\_\_\_\_...." followed by "A fatal esptool.py error occurred: Failed to connect to ESP8266: Timed out waiting for packet header")**

* Downgrade esp8266 by ESP8266 Community to **version 2.5.0**
* Disable PROGMEM for ArduinoJson - #define ARDUINOJSON\_ENABLE\_PROGMEM 0 - on the line before importing the library. This increases RAM usage, but for this particular project it seems to work fine.
* Despite online advice, DO NOT downgrade ArduinoJson to version 5 - the code for the second microcontroller uses objects from version 6 and above.
* VERY IMPORTANT: MAKE SURE THE ESP8266 IS NOT CONNECTED BY UART TO ANYTHING ELSE !!!! If the board is connected to a second microcontroller, communicating by UART, you will NOT be able to upload new code onto it. Make sure only the micro-USB cable is connected to the board and nothing else.

# ESP32 WROOM

**If you can't find the COM port to which the ESP32 is connected in the port list in Arduino IDE AND are running Windows 11:**

Installing the following driver has solved the problem: https://www.silabs.com/developer-tools/usb-to-uart-bridge-vcp-drivers?tab=downloads. I installed Silicon Labs' CP210x universal windows driver (v11.4.0) and it solved the problem. I did not have this issue when working on Windows 10. A Mac OSX driver is also available, but I can't tell if this is an issue on other OS.

# General Arduino IDE Tips

* Before reinstalling your drivers, try unplugging the boards and connecting them again. After plugging the boards back again, reset them using the reset button. Sometimes this solves the issue, other times it doesn't. Behavior is not consistent. Useful when you get garbage on the serial monitor.
* Sometimes the COM ports stop showing up in Arduino IDE. This happened to me after turning off USB ports in the hub and when turning them on again the ports wouldn't come up in the IDE. The only way I found to fix this was restarting my computer. Again, it is not consistent and might be a problem with the USB hub.
* If the program doesn’t compile due to issues with **default** libraries, make sure the correct board model is selected on the top left hand side corner. While working with the IPI files, select LOLIN(WEMOS) D1 R2 & mini. As for the distance-vibration module, select the DOIT ESP32 DEVKIT V1 model.

# General IPI Prototype Tips

* If you start getting blank readings being sent to the broker (...,topic=inc\_data/I1 aX=0,aY=0,aZ=0, ...), make sure the sensor node is powered on. The problem might not be the software and even if the node is plugged in a wall plug/extension, the small red and black cables can fail. To check if power is reaching the node, unplug it and plug it back again. A blue LED light should blink a couple of times. If it doesn't, the board isn't getting any power. If it does blink, conduct further testing to ensure it stays on after those initial moments. If you start getting blank reading again it probably lost connection again.
* If you need to disassemble the inclinometer, make sure you unscrew the different components without twisting the cables. Twisting the metal tubes (sensor nodes) WILL twist the power cables, potentially breaking them. Start by making sure the white plastic tubes are completely loose, then disconnect the power for the end node and then you can twist it.
* It is recommended that one should use the MAC address designated to each microcontroller from the factory. To get the MAC address of an esp32/esp8266, insert the following line of code to your setup() function: Serial.println(WiFi.macAddress());
  + Make sure you set the device as a Wi-Fi station before you try to get its address