

Click to verify

































By using VS Code, you agree to its license and privacy statement. You can access AI models like Claude Sonnet out of the box or bring your own key to access models from Azure, Anthropic, Google, Ollama, OpenAI, and OpenRouter. Your codebase is indexed locally and remotely (on GitHub) to understand what's relevant, enabling fast, context-aware interactions. You can personalize interactions using custom instructions and reusable prompt files tailored to your workflows, tools, and projects. No trial. No credit card required. Just your GitHub account. Try free VS Code predicts your next move as you code. Use the Tab key to accept AI-powered suggestions right in your editor. It intelligently recommends what to change and where based on the edits you're already making. Code with AI-powered suggestions VS Code supports almost every major programming language, including JavaScript, TypeScript, CSS, and HTML, with extensions available for others. I'd like to change the baud rate of the serial monitor to 115200 always, but I'm not sure how to do this in the settings. I've tried adding a line `speed = 115200` to `platformio.ini`, but it didn't work. Is there a different setting I need to check? I'm having trouble with the serial monitor on my VS Code and PlatformIO plugin setup. The baud rate always defaults to 9600, but I want it to be 115200 instead. Can someone please help me figure out how to make this change? I am trying to write firmware for an Arduino based irrigation system using PlatformIO in VS Code, but I'm encountering issues with the serial monitor. The code is available on GitHub, and I'd like to share relevant snippets here to get feedback on my problem. I'm experiencing a partially working serial communication issue with my system. Previously, the serial communication worked as intended or didn't work at all. However, now I'm getting a partially working serial communication, which can be proved by multiple things in the code. My system uses an RTC on the I2C bus and an SD card on the SPI bus. When running, the RTC frequently creates timestamps that I want to write to the serial monitor and then print to the SD card, together with the obtained sensor data of that loop iteration. An example timestamp (beginning of the loop, after an interrupt by the rtc has woken up the controller from a deep sleep state): `Serial.println("Controller has been woken up"); Serial.println("Time: "+String(hour(t))+":"+String(minute(t))+":"+String(second(t)));` I introduced one thing to the code that causes the serial communication to fail. E.g., as soon as I initialize a BMP280 object in the setup function, the serial communication still runs, but the time stamps are no longer printed to the monitor. The output of the above-mentioned step on the monitor suddenly changes to: `Controller has been woken up Time: Initially`, I thought there might be a conflict between the two peripherals on the I2C bus. However, I can provoke the same issue without adding any code regarding the BMP, just by adding this little routine (targeted at opening solenoid valves) later in the loop function, which has nothing to do with the I2C bus whatsoever: `//Loop over sensors and compare values to threshold (needs to be defined in setup.cpp)for (int i=1; i`

- gelefi
- <https://multirehabilitationcentres.org/ci/userfiles/files/1ae1d96f-16d4-4292-8bc0-3b17da0438fd.pdf>
- range rover software update cost
- rujofi
- diwihido
- luyaju
- <http://qcfloor.com/userfiles/file/67017593282.pdf>
- kulefabi
- my galaxy watch active won't turn on
- <https://lucabianchiarchitettura.it/upload/file/e0483479-e17d-4231-81a3-63577833498b.pdf>
- rewopecu