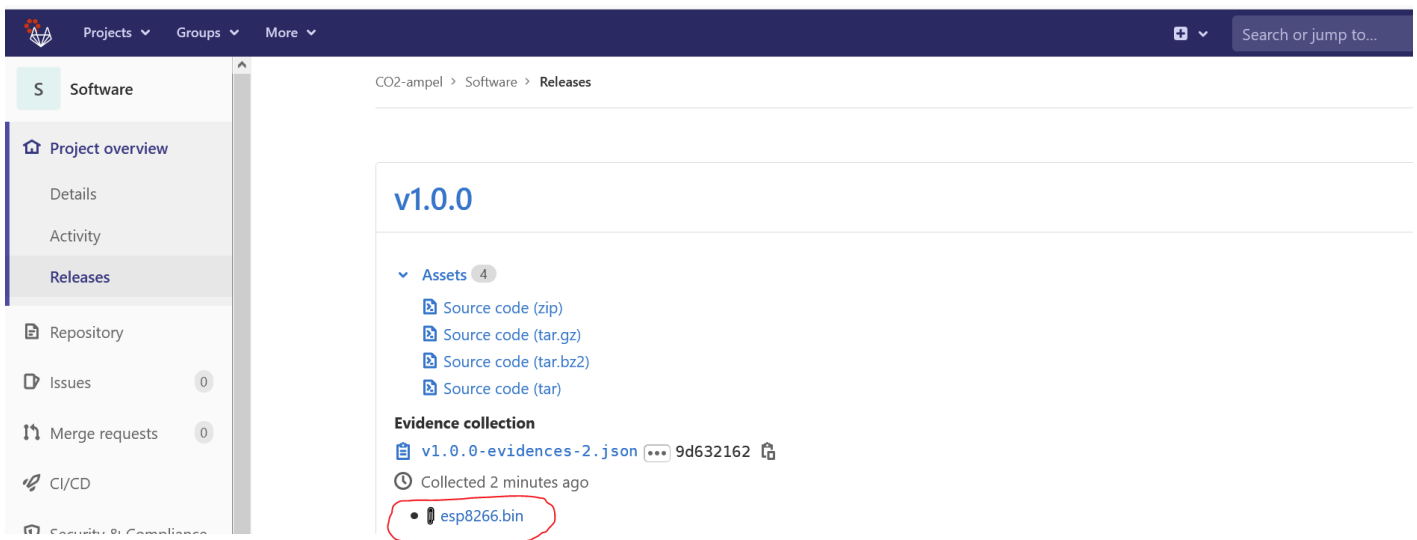


3. Software aufspielen

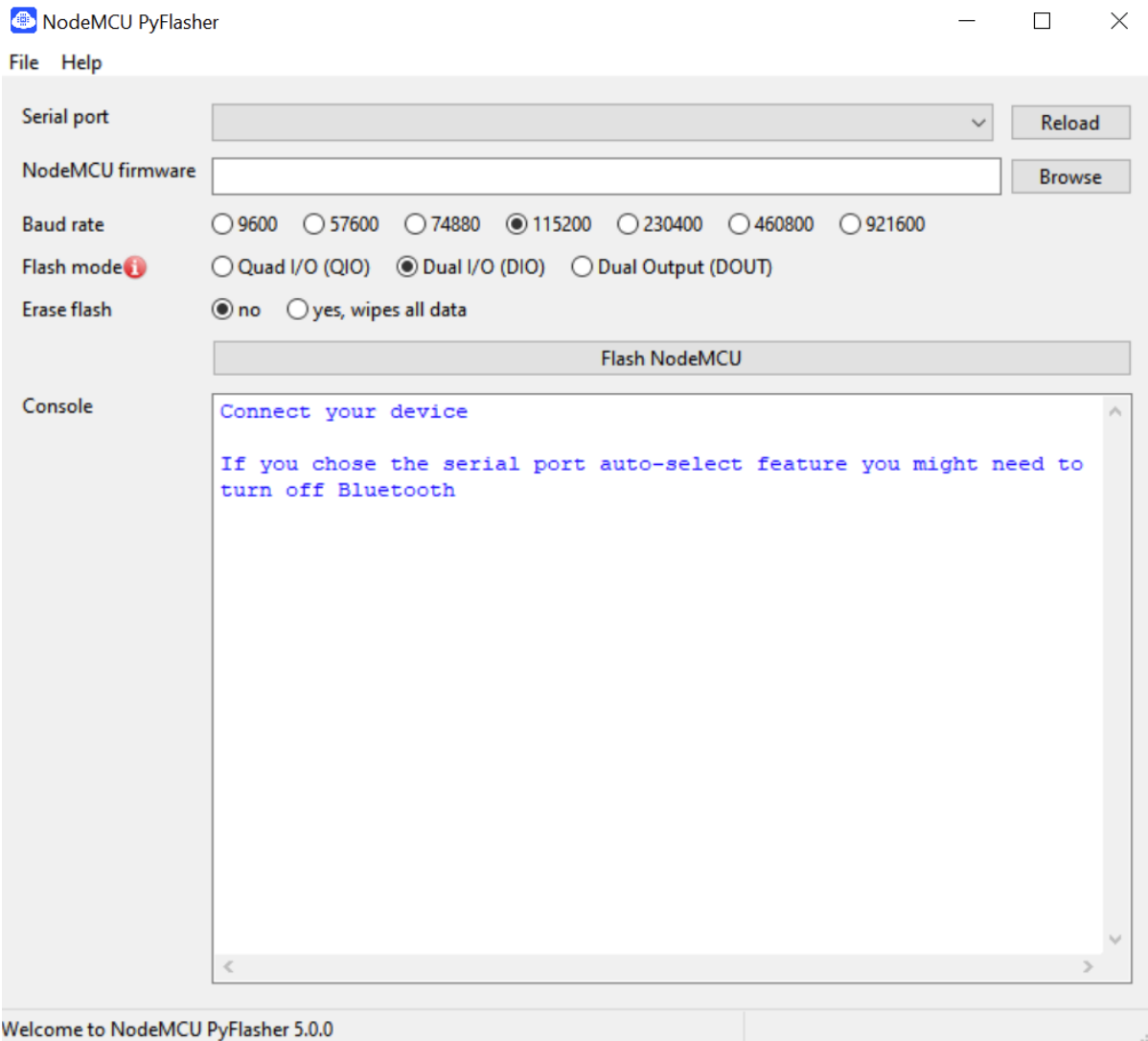
Die CO2 Ampel muss noch mit der richtigen Software bespielt werden. Das geht auf verschiedene Arten - zwei sind hier beschrieben. Wenn du die Software noch anpassen möchtest oder dir den Quellcode erst anschauen und dann aufspielen möchtest, folge den Beschreibungen weiter unten - im Abschnitt [Software per IDE aufspielen](#). Der folgende Abschnitt erklärt wie du relativ einfach, die Software mittels einer von uns bereits erstellten Datei auf den Mikrocontroller bekommst.

Software direkt flashen

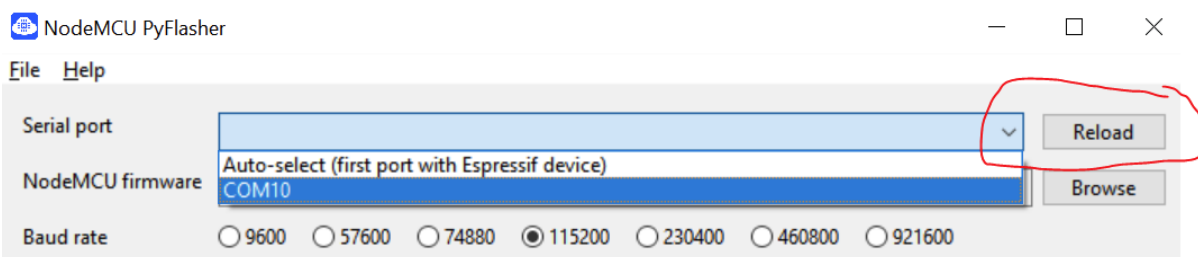
Du benötigst die esp8266.bin aus unserem Repository. Lade dir die Version aus dem letzten Release (hat die größte Nummer) [hier](#) herunter, in dem du die Datei anklickst.



Um diese Datei jetzt auf den Microcontroller zu flashen, brauchst noch ein Flashtool - z.B. den [nodemcu-pyflasher](#). Lade dir den letzte Release herunter und führe dann die heruntergeladene NodeMCU-PyFlasher.exe Datei aus. Windows beschwert sich, dass die Datei evtl. bösartig ist, da sie von einem unbekanntem Hertseller stammt. Klicke in dem Fenster auf "Weitere Informationen" und dann unten auf "Trotzdem ausführen". Es öffnet sich ein Fenster:

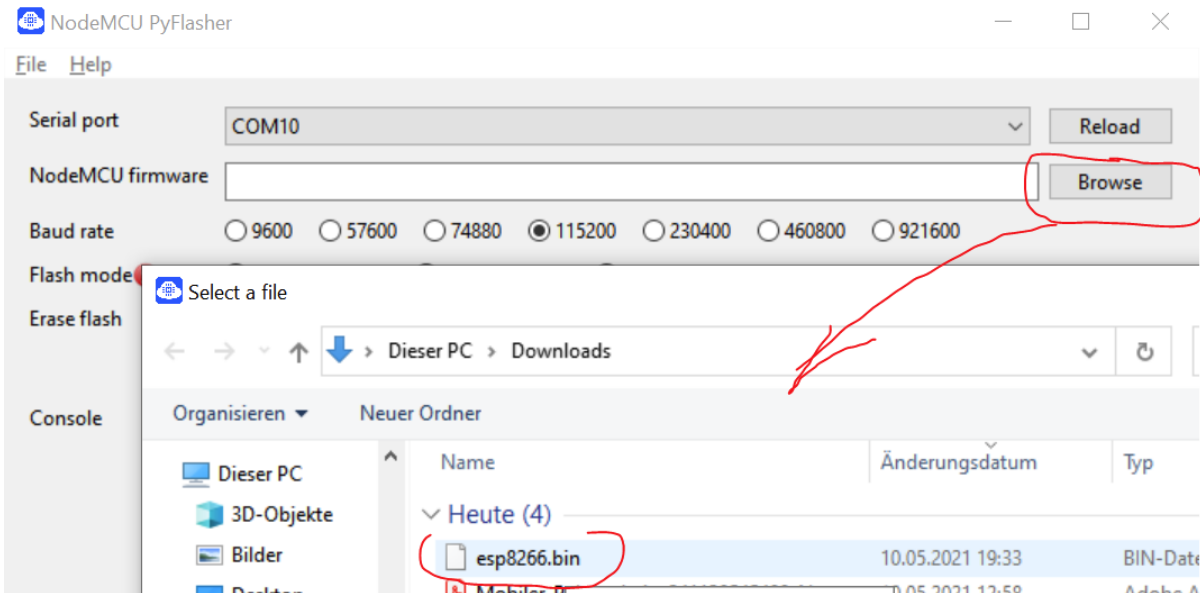


Falls nicht schon geschehen, verbinde nun den Microcontroller der CO2-Ampel per USB Daten Kabel mit deinem PC. Ein Klick auf den "Reload" Button und dann auf das kleine Dreieck im Flasher Tool sollte jetzt ein COM Port anzeigen, an dem der Microcontroller angeschlossen ist.

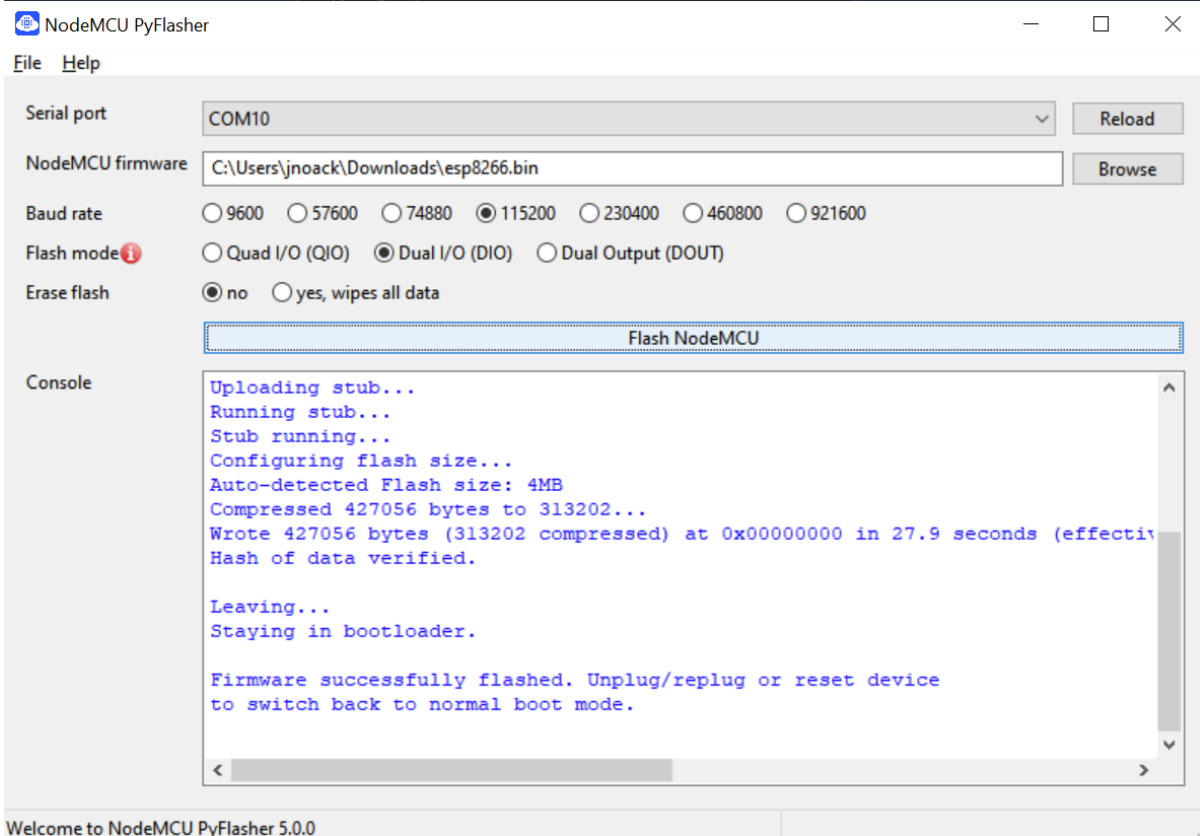


In unserem Fall an COM10. Das kann bei dir auch ein anderer COM-Port sein. Wähle den Port aus indem du drauf klickst.

Jetzt klicke auf "Browse" und suche die esp8266.bin Datei, die du weiter oben bereits aus dem Repository geladen hast - in der Regel findest du sie unter "Downloads". Wähle die Datei aus und klicke unten auf "Öffnen".



Im nächsten Schritt wird der Inhalt der esp8266.bin Datei auf den Micrcontroller geflashed. Dazu klicke auf "Flash NodeMCU". Im unteren Teil des Fensters erscheinen jetzt einige Meldungen und es sollte eine Prozentzahl langsam hochlaufen. Sobald 100% erreicht sind bist Du fertig:

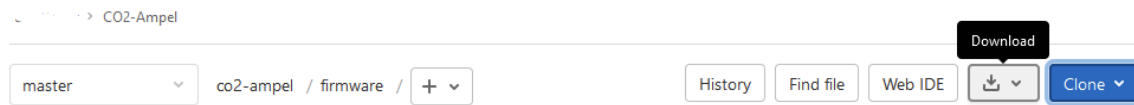


Die Ampel ist jett einsatzbereit. Du kannst sie vom PC trennen und an das USB Netzteil anschließen.

Viel Spaß bei richtig Lüften ;-) ... dabei hilft dir auch unsere Bedienungsanleitung.

Software per IDE aufspielen

Die Software (Firmware) für unsere Ampel findest du [in unserem MakerLab GITLab Repository](#). Du kannst dir das komplette Repository per DownloadButton herunterladen oder eine lokalen Clone anlegen.



Je nachdem, welche Entwicklungsumgebung (IDE) du benutzt, ist der Weg, die Software auf den Microcontroller zu spielen verschieden. Die wohl am häufigsten benutzten IDEs sind die ArduinoIDE und VisualStudioCode mit PlattformIO. Die ArduinoIDE ist gut für Einsteiger geeignet - VSCode hat schon sehr professionelle Features.

Für den Moment geht es hier nicht um Code Entwicklung - sondern darum, die Firmware irgendwie auf den Microcontroller zu bekommen - zu flashen.

Unsere CO2 Ampel basiert auf ein Projekt der Hochschule für Technik in Stuttgart. Dort gibt es auch [eine sehr gute Anleitung](#) die ab Seite 10 sehr detailliert beschreibt, wie du mit der Arduino IDE die Software auf die CO2-Ampel flashen kannst. Falls du dich noch mit dem Thema IDE und flashen von Software beschäftigt hast, möchte wir dir diese Anleitung sehr empfehlen.

Das Repository kann sowhol mit der VSCode/PlattformIO als auch der ArduinoIDE genutzt werden.

Sobald du deinen Sensor geflashed hast, ist die CO2-Ampel einsatzbereit.



Version #6

Erstellt: 2021-05-08 10:26:26 UTC von Jens Noack

Zuletzt aktualisiert: 2021-05-10 16:12:23 UTC von Dorian Zedler