

## ESP 32 CAM – PlatformIO IDE

---

Za početne projekte na vežbama je korišćen Arduino IDE (Integrisano razvojno okruženje). Postoje i druga okruženja. Jedno je PlatformIO IDE u Visual Studio Code. Cilj uvodnih vježbi jeste integracija PlatformIO u projektima udaljenog inženjerstva. Šta je PlatformIO?

PlatformIO je višepatformska, unakrsna arhitektura, višestruki okvir, profesionalni alat za inženjere embeded sistema i za programere softvera koji pišu aplikacije za embeded proizvode.

### PlatformIO

Ukratko, možete koristiti ovaj alat u programiranju bilo čega što je u vezi sa mikrokontrolerom i odabrati okvir po svom izboru ili koristeći ESP-IDF ili Arduino okvir. U radu ćemo se fokusirati na razvoj programa zasnovanih na Arduino okviru.

Takođe, koristićemo PlatformIO IDE ekstenziju u Visual Studio kodu u razvoju projekata. Visual Studio Code je uređivač teksta otvorenog koda, Integrisano razvojno okruženje (IDE) koji je ustanovio Microsoft. To je popularan uređivač teksta sa ugrađenim funkcijama za pomoć programiranju, što ga čini pogodnim za razvoj (Slika 1.).

Zašto bi trebalo da koristite PlatformIO?

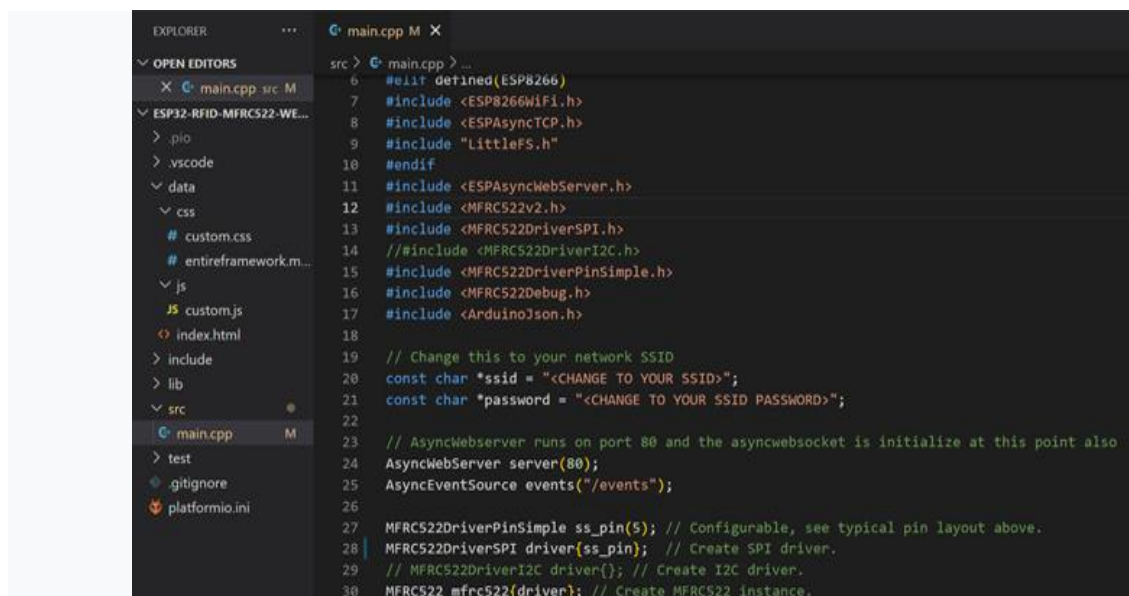
Slede neke od važnih karakteristika, ali je naravno to nisu sve.

Upravljanje bibliotekama/zavisnostima (dependencies)

Dodavanje zavisnosti je veoma jednostavno u PlatformIO IDE. Samo treba da izaberete biblioteku i dodate je u svom projektu iz PlatformIO spremišta. Ako biblioteka koja vam je potrebna nije prisutna, možete je dodati ručno. Sve zavisne biblioteke se automatski preuzimaju umesto vas.

Isticanje sintakse koda

Lako je kodirati kada znate koji deo uređujete. Funkcija isticanja sintakse koda je dobrodošla pomoć dok razvijate svoj program.



*Slika 1 Visual Studio Code razvojno okruženje i PlatformIO IDE ekstenzija*

## Intellisense

Funkcija IntelliSense vam pomaže u vašem razvoju jer će automatski dovršiti funkcije/metode za vas dok programirate.

## Code Assist

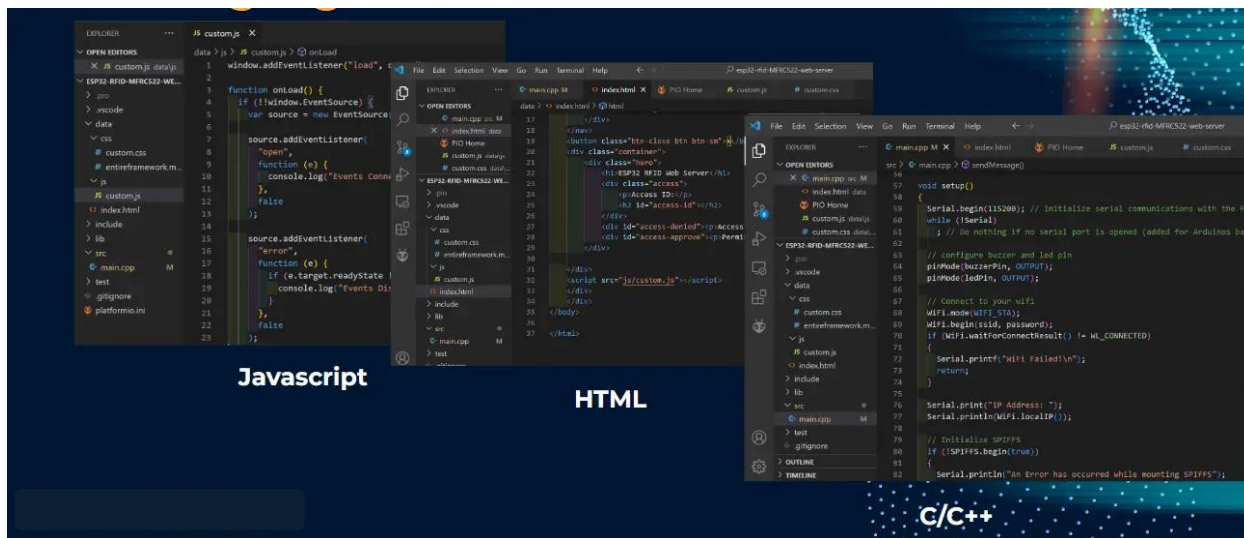
Funkcije brze pomoći kodu vam omogućavaju da dovršite sav kod jednostavnim unošenjem koda prečice.

## Otklanjanje grešaka

Kada stvari krenu naopako, lako je proveriti šta nije u redu jer vam je to istaknuto. Takođe, možete detaljnije istražiti šta uzrokuje problem tako da će vam biti lako da shvatite šta uzrokuje problem.

## Podrška za druge jezike

Danas u svojim projektima ne razvijate samo C/C++ ili Arduino kod. Kao takav, PlatformIO IDE pomaže u ovom pogledu jer je podrška za više jezika dostupna i drugim jezicima, kao što su Javascript ili HTML (Slika 2.).



Slika 2. Podržani programski jezici u okviru PlatformIO IDE

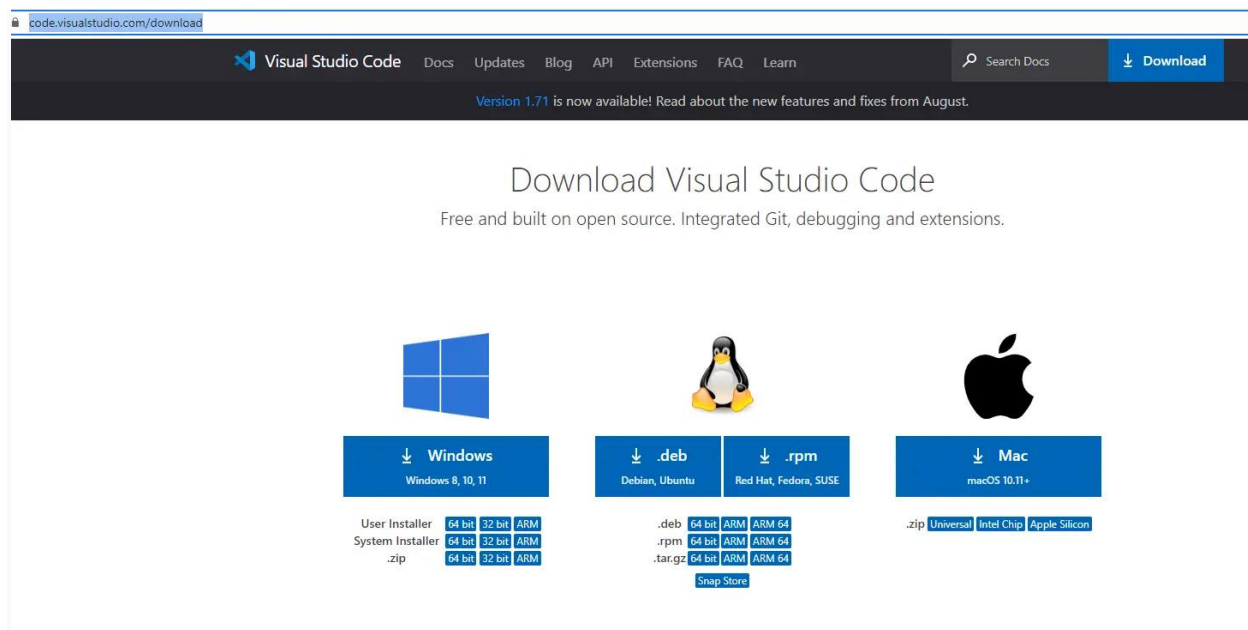
Kako početi da koristite PlatformIO?

Instalirajte Git

Moramo da instaliramo Git klijent da bismo mogli da preuzmemo odgovarajuću biblioteku za vaše projekte. Idite na ovaj link (<https://git-scm.com/downloads>) i preuzmite klijent za svoj operativni sistem.

Preuzmite Visual Studio Code

Idite na ovaj link (<https://code.visualstudio.com/download>) i preuzmite instalater za svoj operativni sistem. Podržava Windows, Linux i Mac OS (Slika 3).

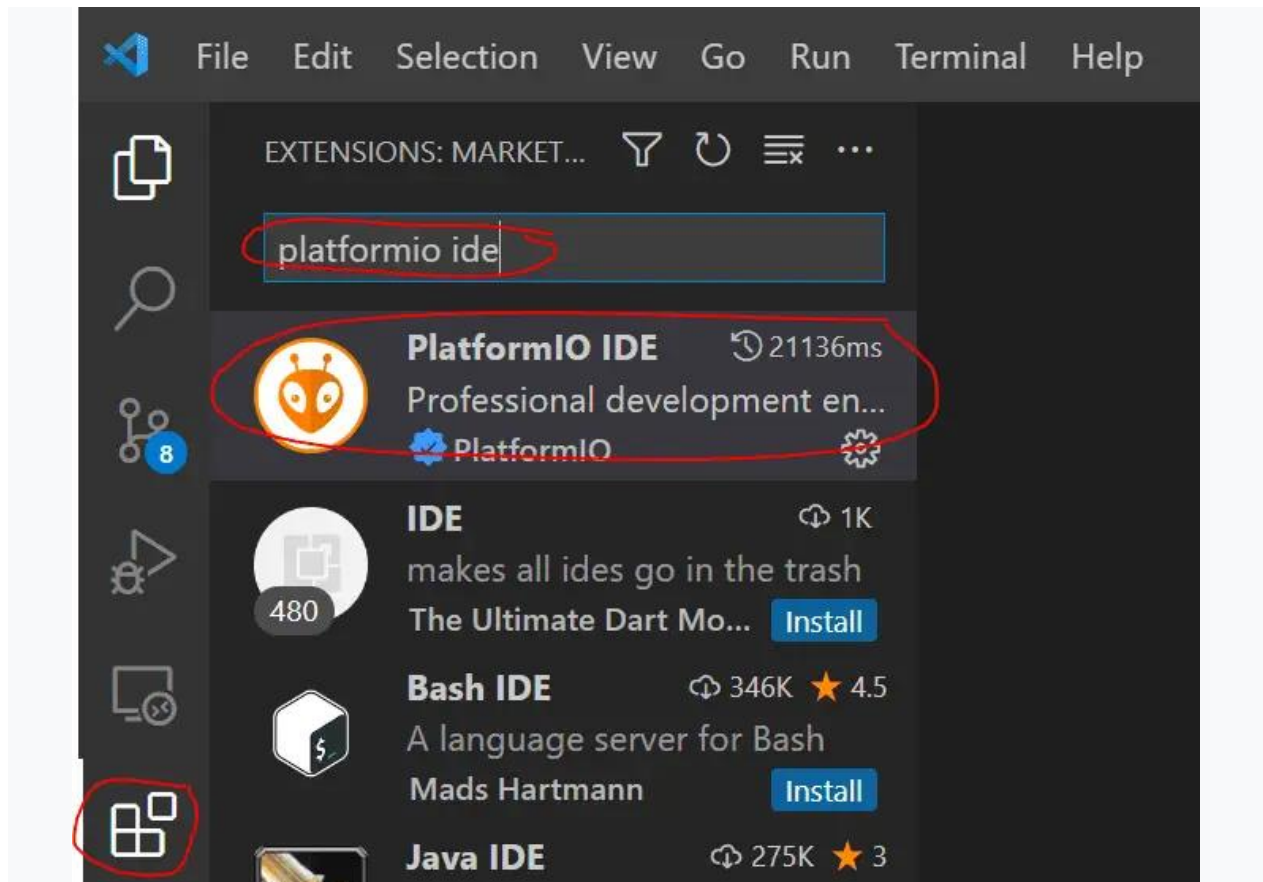


*Slika 3. Preuzimanje softvera Visual Studio Code*

Instalirajte proširenje PlatformIO IDE

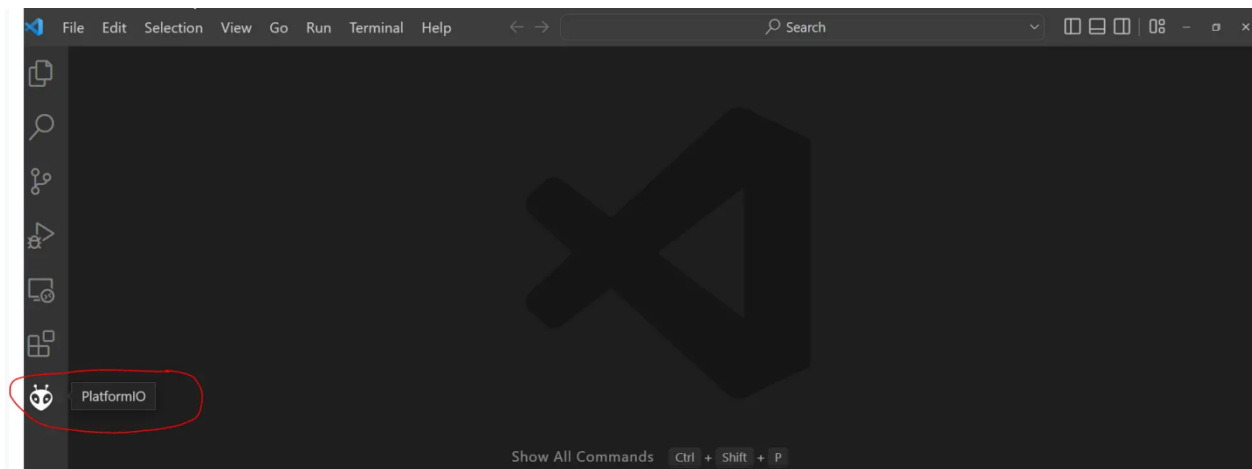
Kada se Visual Studio Code instalira, njega pokrenite i instalirajte PlatformIO IDE ekstenziju.

Na levoj strani ekrana Visual Studio Code izaberite karticu Ektensions, a zatim ukucajte „platformio ide“. Kliknite na dugme za instaliranje PlatformIO IDE da biste instalirali proširenje (Slika 4.). Ovo će potrajati neko vreme u zavisnosti od brzine vaše mrežne veze.



*Slika 4. Instaliranje PlatformIO IDE ekstenzije za Visual Studio Code*

Sada kada smo instalirali PlatformIO IDE možemo da počnemo da kreiramo naš test projekat. Ako je instalacija prošla dobro, trebalo bi da vidite alatku PlatformIO sa strane ekrana (Slika 5.).



*Slika 5. Alatka PlatformIO IDE ekstenzije*

Cilj vežbi?

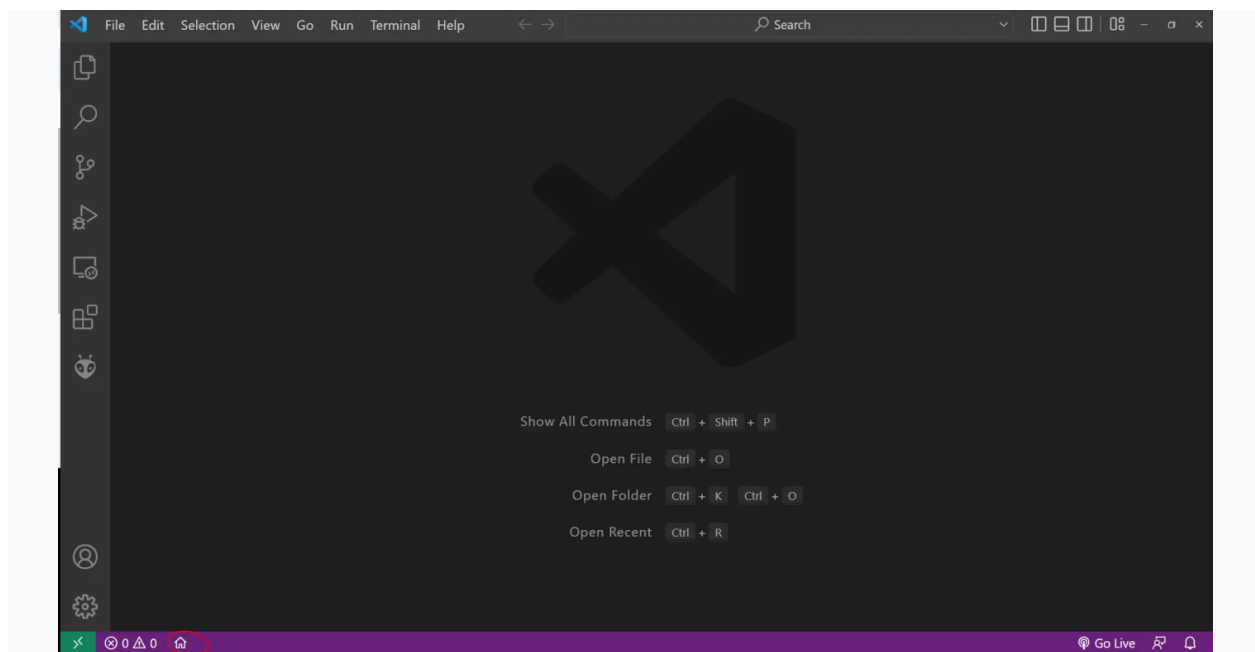
Kreiranje dva projekta uz pomoć PlatformIO.

Prvo, kreiramo popularni Blink projekat u Arduino IDE-u koji će vam pomoći da upoznate PlatformIO IDE-u.

Zatim se prethodni projekat poboljša i dodavanjem web servera za aktiviranje ugrađene LED diode (Light Emitting Diode) vaše mikrokontrolerske jedinice (MCU).

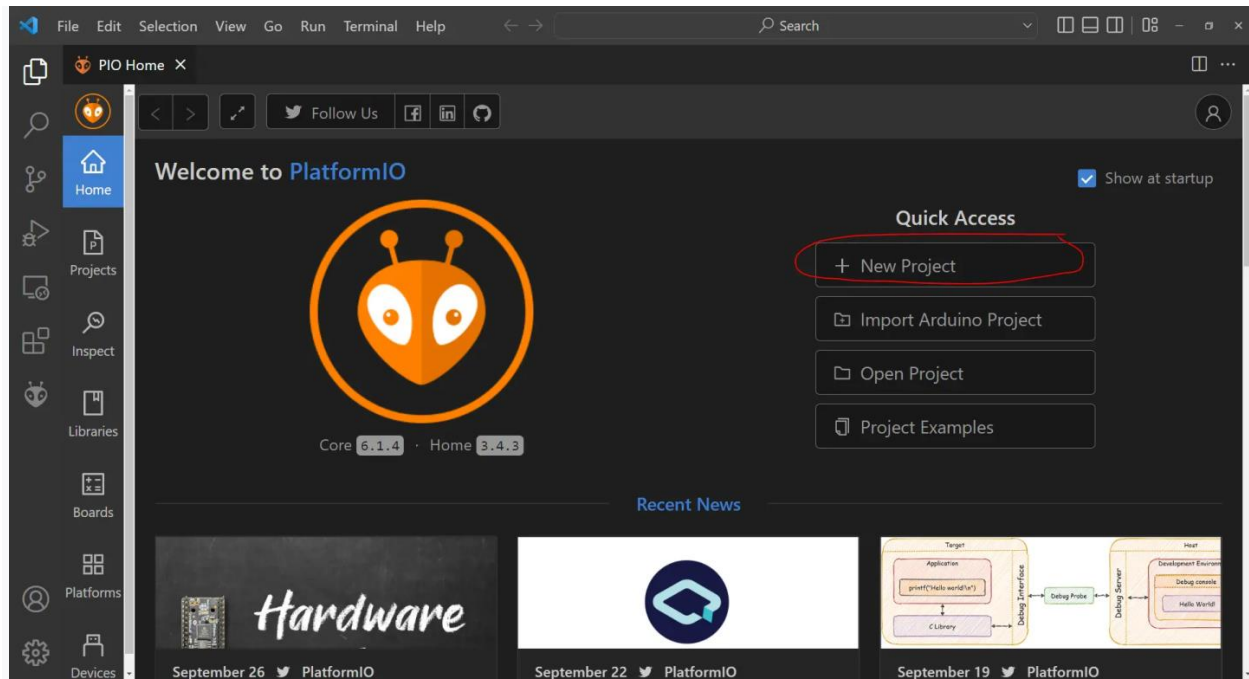
Kreirajte projekat

Kliknite na PlatformIO Home u lijevom donjem delu ekrana (Slika 6.).



*Slika 6. Početna stranica okruženja, PlatformIO Home tab*

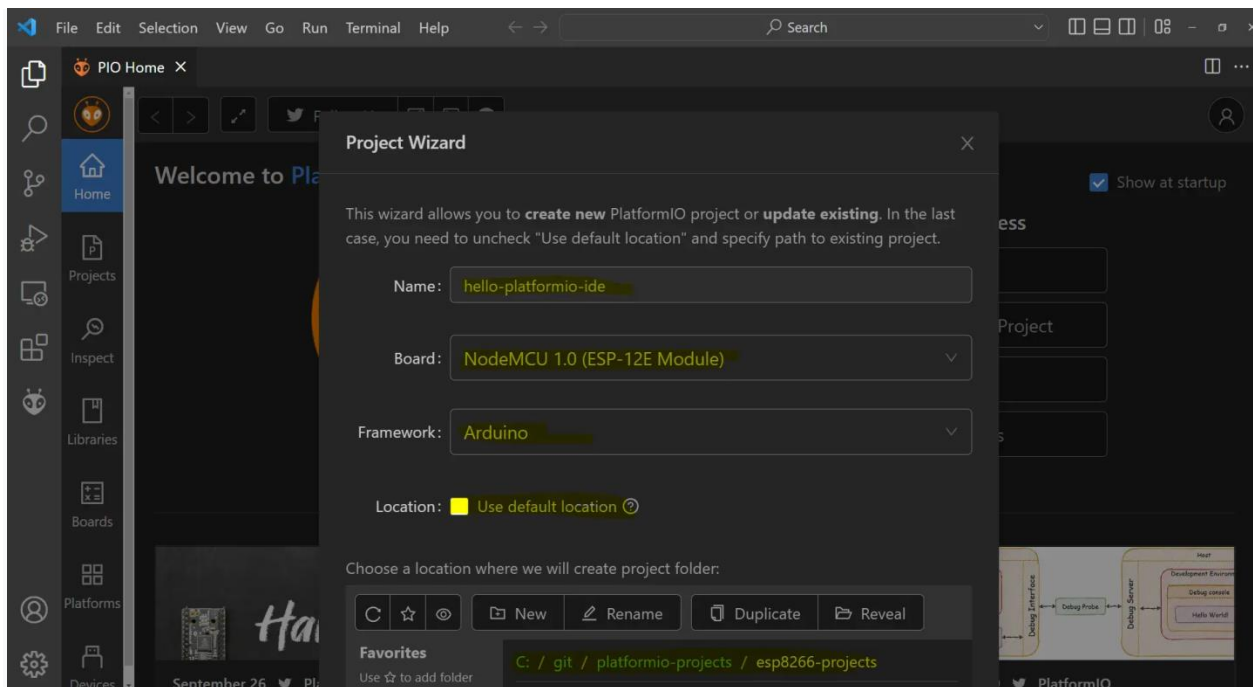
Kliknite na Novi projekat (Slika 7.).



*Slika 7. Kreiranje novog projekta*

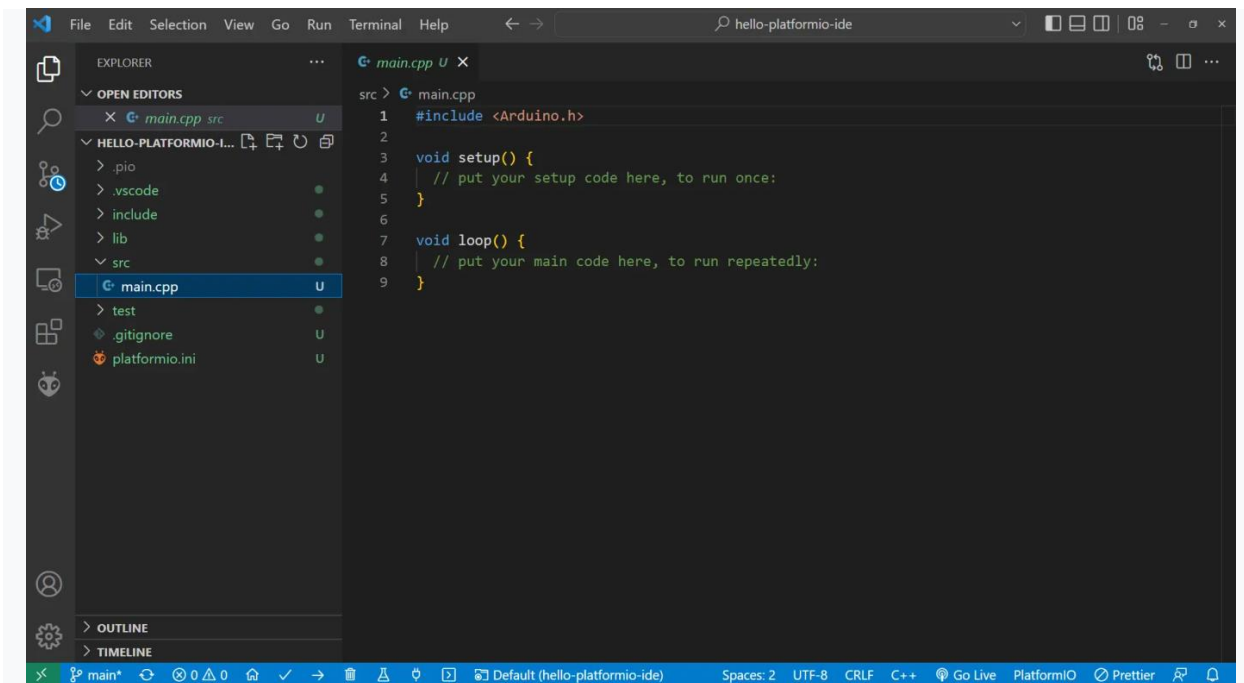
Na ekranu čarobnjaka za projekte unesite sledeće detalje (Slika 8.):

- Ime projekta
- Board: AI Thinker ESP-32 CAM
- Programski okvir: Arduino
- Lokacija čuvanja fajla



*Slika 8. Parametri novog projekta*

Kliknite na Završi (Finish) i sačekajte neko vreme dok se inicijalizacija ne završi. Trebalo bi da vidite kreiran sledeći projekat. Otvorite datoteku src/main.cpp. Ovo je polazna tačka našeg projekta (Slika 9.).



*Slika 9. Polazna tačka kreiranog projekta*

Treperenje ugrađene LED diode vašeg ESP-32 CAM.

Počnimo treperenjem ugrađene LED diode povezane sa našom mikrokontrolerskom jedinicom ESP-32 CAM. Povežite svoj mikrokontroler preko USB porta.

Unesite sledeći kod u „main.cpp“.

```
#include <Arduino.h>

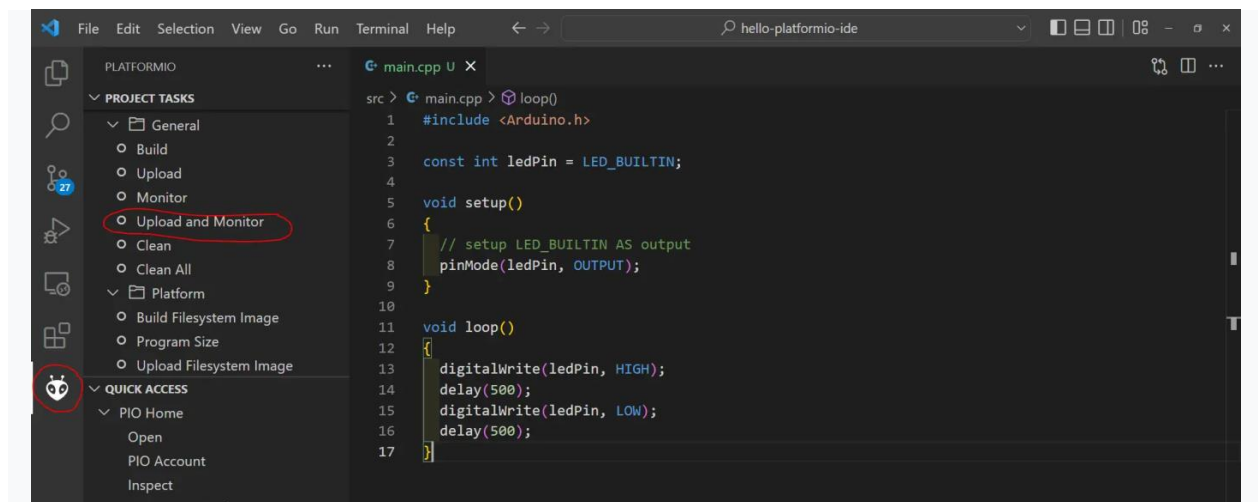
const int ledPin = LED_BUILTIN;

void setup()
{
    // setup LED_BUILTIN AS output
    pinMode(ledPin, OUTPUT);
}

void loop()
{
    digitalWrite(ledPin, HIGH);
    delay(500);
    digitalWrite(ledPin, LOW);
    delay(500);
}
```

Kod će dovesti do treperenja LED diode.

Zatim, otpremamo programski kod na ESP-32 CAM, a da bismo to uradili kliknite na ikonu PlatformIO sa strane i potražite Upload i Monitor (Slika 10.).



*Slika 10. Otpremanje programskog koda na mikrokontroler*

Ovo će otvoriti novi terminal i započeti kompajliranje vašeg projekta. Trebalo bi da vidite poruku o uspehu prikazanu ispod.

Proverite svoj mikrokontroler ESP-32 CAM i trebalo bi da vidite da dioda treperi.



## Nastavak - Štampanje na vašem serijskom monitoru u okviru PlatformIO

PlatformIO IDE sadrži integrisani serijski monitor. Otvorite datoteku platform.ini u svom projektu da biste konfigurisali brzinu prenosa. Ovo je konfiguraciona datoteka vašeg projekta i sadrži druge funkcije koje ovde nisu navedene. Takođe možete da postavite biblioteku koju želite da koristite u svom projektu, druge biblioteke će biti dodavane naknadno.

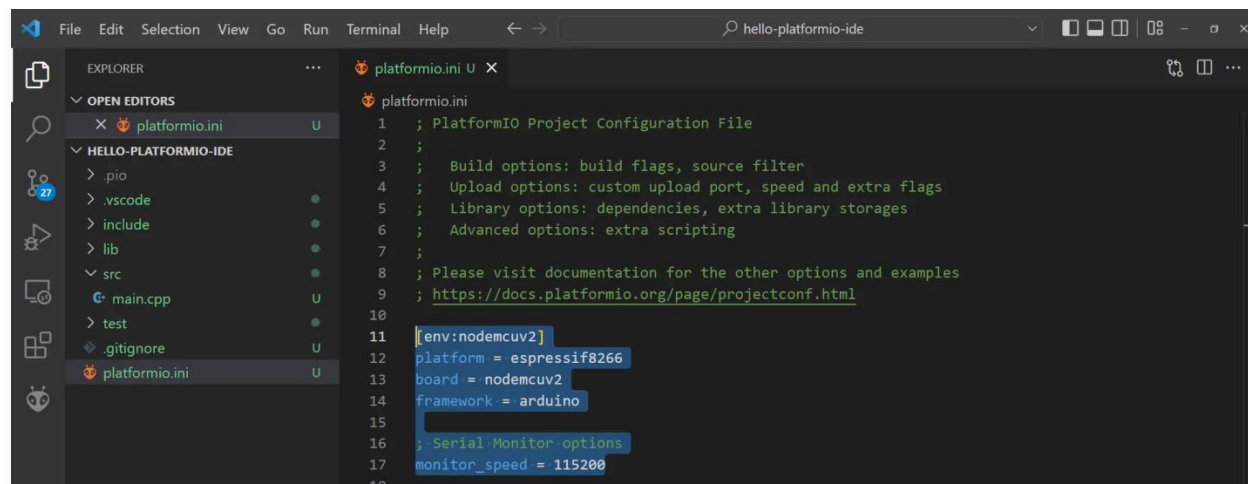
Za sada ćemo samo podesiti brzinu prenosa serijskog monitora tako što ćemo upisati odgovarajuću vrednost (Slika 11.). Monitor\_speed treba da odgovara brzini prenosa korišćenog kontrolera. Ukoliko je potrebno dodati i naredne komande u fajl:

```
monitor_rts = 0
monitor_dtr = 0
```

po potrebi upisati kombinacije 0 i/ili 1 za ova dva parametra, ukoliko prethodna kombinacija ne zadovoljava nesmetan prikaz na monitoru okruženja.

```
[env:nodemcu2]
platform = espressif8266
board = nodemcu2
framework = arduino
```

```
; Serial Monitor options
monitor_speed = 115200
```



Slika 11. Upisivanje brzine prenosa serijskog monitora

Proverite svoj kod i dodajte sledeće linije:

```
#include <Arduino.h>

const int ledPin = LED_BUILTIN;

void setup()
{
```

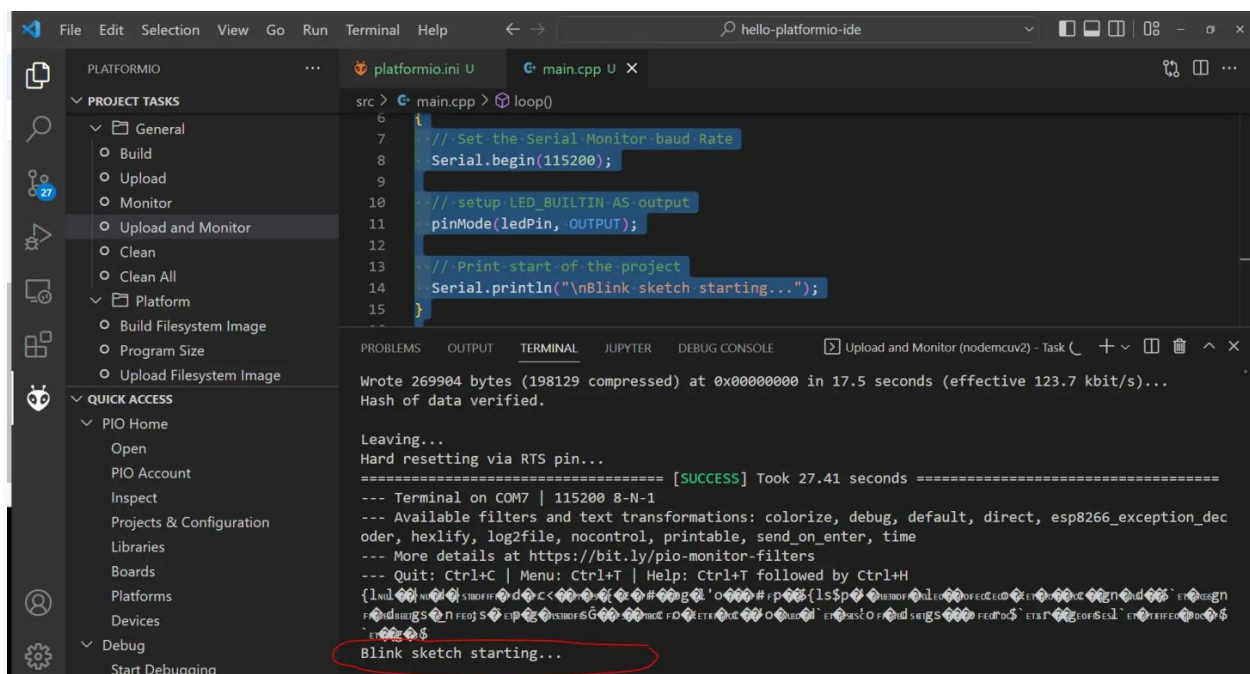
```
// Set the Serial Monitor baud Rate
Serial.begin(115200);

// setup LED_BUILTIN AS output
pinMode(ledPin, OUTPUT);

// Print start of the project
Serial.println("\nBlink sketch starting...");
}

void loop()
{
    digitalWrite(ledPin, HIGH);
    delay(500);
    digitalWrite(ledPin, LOW);
    delay(500);
}
```

Ovo bi trebalo da odštampa sledeće poruke u vašem terminalu (Slika 12.).



*Slika 12. Ispis poruka na serijskim monitoru*

Ako se ispisuju zadate poruke iz koda u serijskom terminalu, onda je podešavanje monitora uspelo.

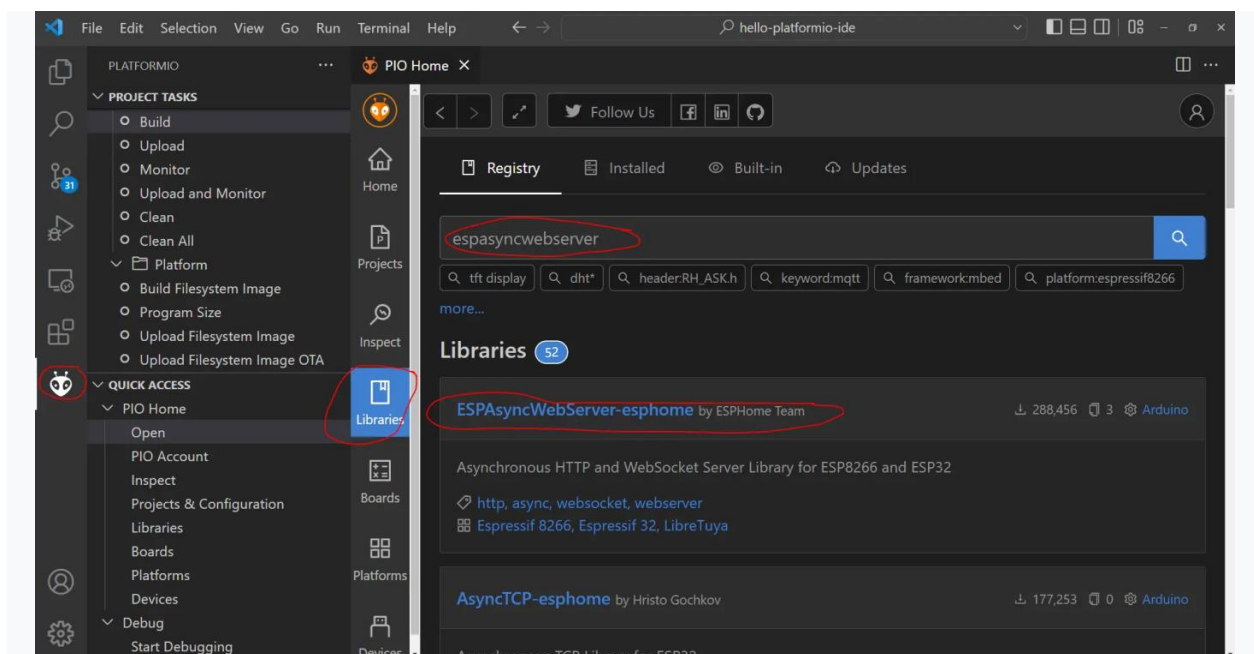
Dodavanje biblioteke ili zavisnosti vašim projektima u PlatformIO

Sledeći koraci koje ćemo uraditi zasnivaju se na ESP32-CAM web serveru koji koristi LittleFS.

Program treperenja diode se može dodatno poboljšati tako što će se treperenje ugrađene LED diode upravljati preko mobilnog telefona ili računara. To će biti urađeno u nastavku.

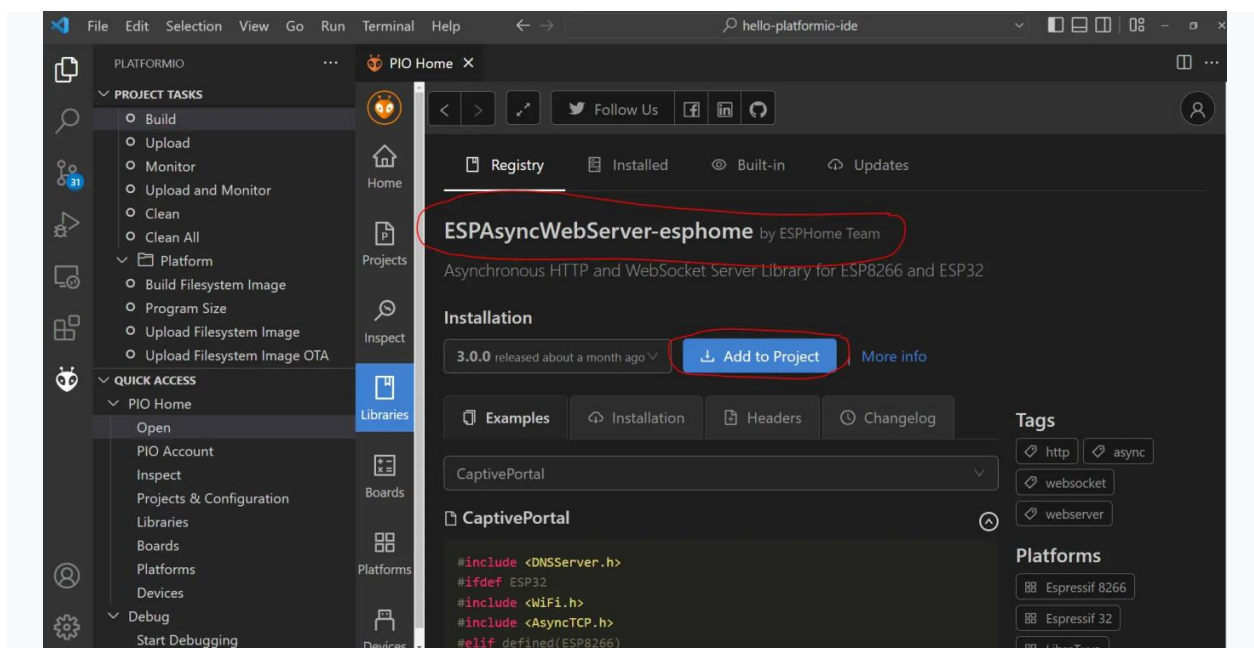
Prvo, dodajmo biblioteku ESPAsyncWebServer u naš projekat. Ovo će nam omogućiti da kreiramo web server koristeći naš mikrokontroler gde možemo da programiramo upravljanje uključivanja ili isključivanja ugrađene LED diode.

Vratite se na svoju PlatformIO Home i ukucajte „espasincwebserver“ u okviru kategorije Libraries (Slika 13.).



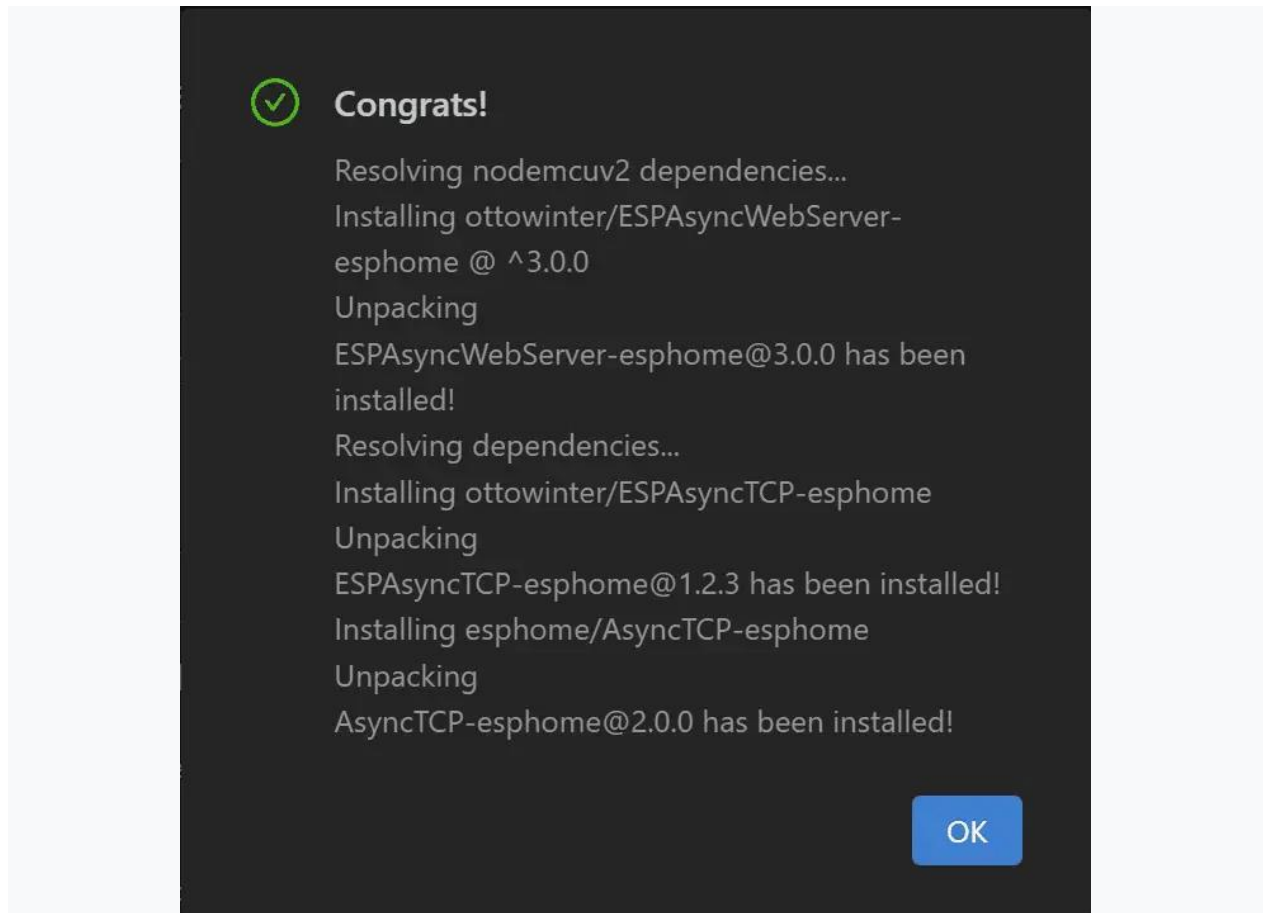
*Slika 13. Dodavanje programskih biblioteka u projekat*

Izaberite biblioteku od ESPHome tima, a zatim kliknite na dugme Dodaj u projekat (Slika 14.).



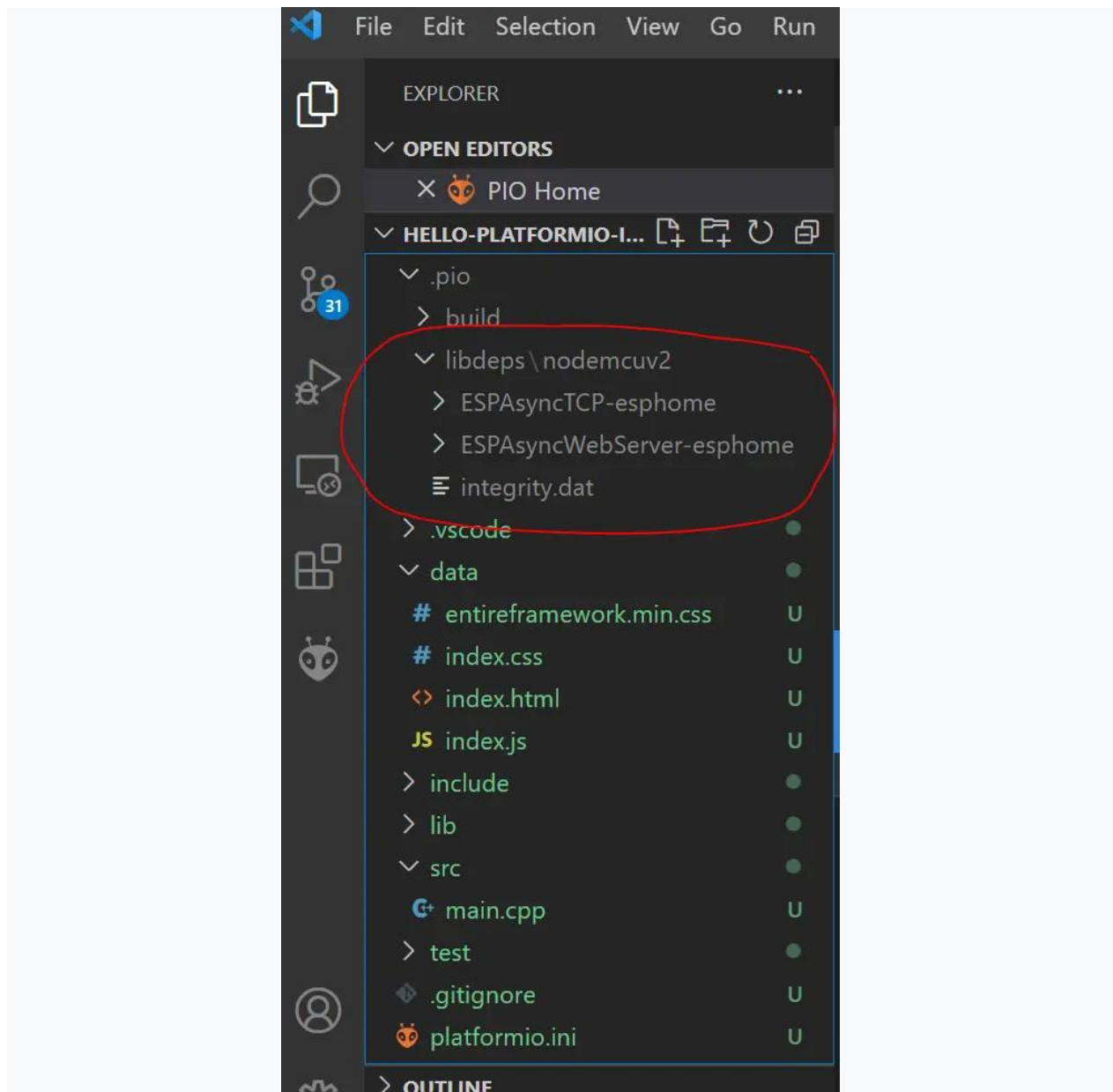
*Slika 14. Završnikorak dodavanja željene biblioteka u projekat*

Izaberite vaš prethodno kreirani projekat u padajućem meniju. Nakon toga se u meniju sa bibliotekama (Libraries) pojavljuje potvrda o uspešnom uključivanju biblioteke u projekat.



*Slika 15. pojavljuje potvrda o uspešnom uključivanju biblioteke u projekat*

Prethodne komande automatski preuzimaju tu biblioteku i sve zavisnosti za vas. Biblioteka će biti smeštena u fasciklu .pio na kartici pretraživača datoteka kao na slici ispod (Slika 16.).



Slika 16. Pronalaženje ubačene biblioteke u okviru glavnog menija

O ostatku podešavanja se pobrinuo PlatformIO IDE. Ako sada proverite datoteku platform.ini onda će izgledati približno ovako:

```
[env:nodemcu2]
platform = espressif8266
board = nodemcu2
framework = arduino
monitor_speed = 115200
board_build.filesystem = littlefs
lib_deps = ottowinter/ESPAsyncWebServer-esphome@^3.0.0
```

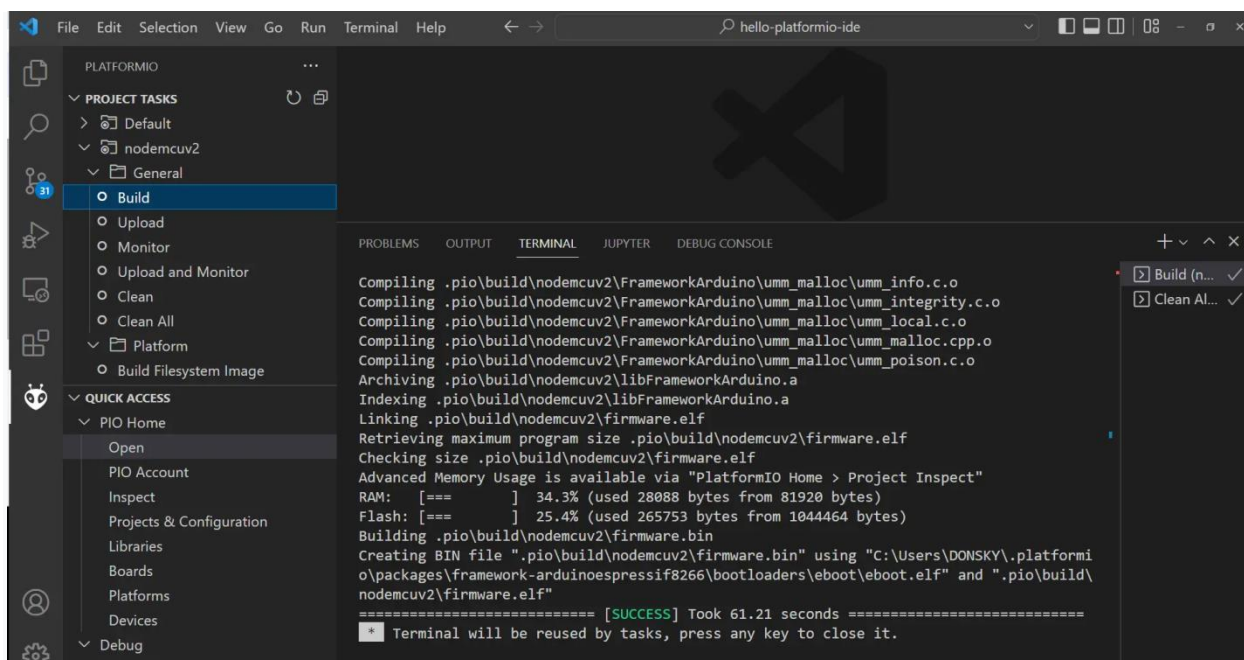
Napomena: Dodata je linija `board_build.filesystem = littlefs` pošto se ovde koristi LittleFS FileSistem.

### Čišćenja i rekonstrukcije u PlatformIO

Kad god uradite nešto na datoteci `platform.ini`, uvek je dobra ideja da očistite i ponovo izgradite svoj projekat. Ovo će omogućiti da se sve ponovo izgradi od nule i ukloni greške u vašem projektu.

Da biste to uradili, otvorite odeljak sa zadacima i kliknite na zadatak očisti sve (Clean i/ili Clean all). Ovo će izbrisati sve preuzete zavisnosti uključujući i konfiguracije koje su postavljene.

Kada je zadatak čišćenja završen, pokrenite opciju Build (Slika 17.). Ovo će ponovo izgraditi sve i preuzeti sve zavisnosti. Ovo može potrajati, pa budite strpljivi i pustite da se završi.



*Slika 17. Opcija Build kojom se vrši ponovna izgradnja projekta i potrebnih datoteka*

### Otpremanje datoteka u fleš memoriju mikrokontrolera u platformi PlatformIO

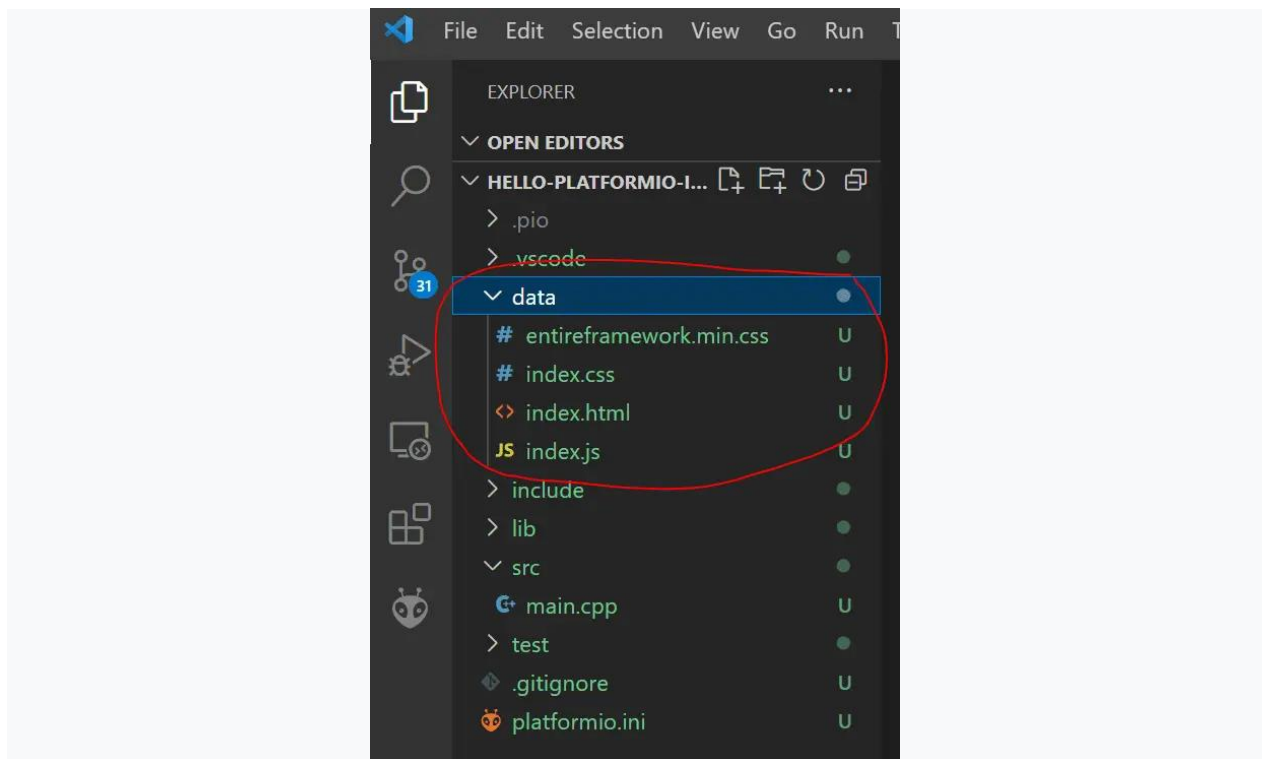
Sada kada smo dodali naše biblioteke, sada počinjemo da programiramo sledeći deo našeg projekta. Potreban nam je korisnički interfejs (tačnije HTML stranica) za prikaz našim korisnicima.

Da biste to uradili, preuzmite dodatni materijal uz vežbu.

PlatformIO – fascikla sa podacima u eksploreru datoteka

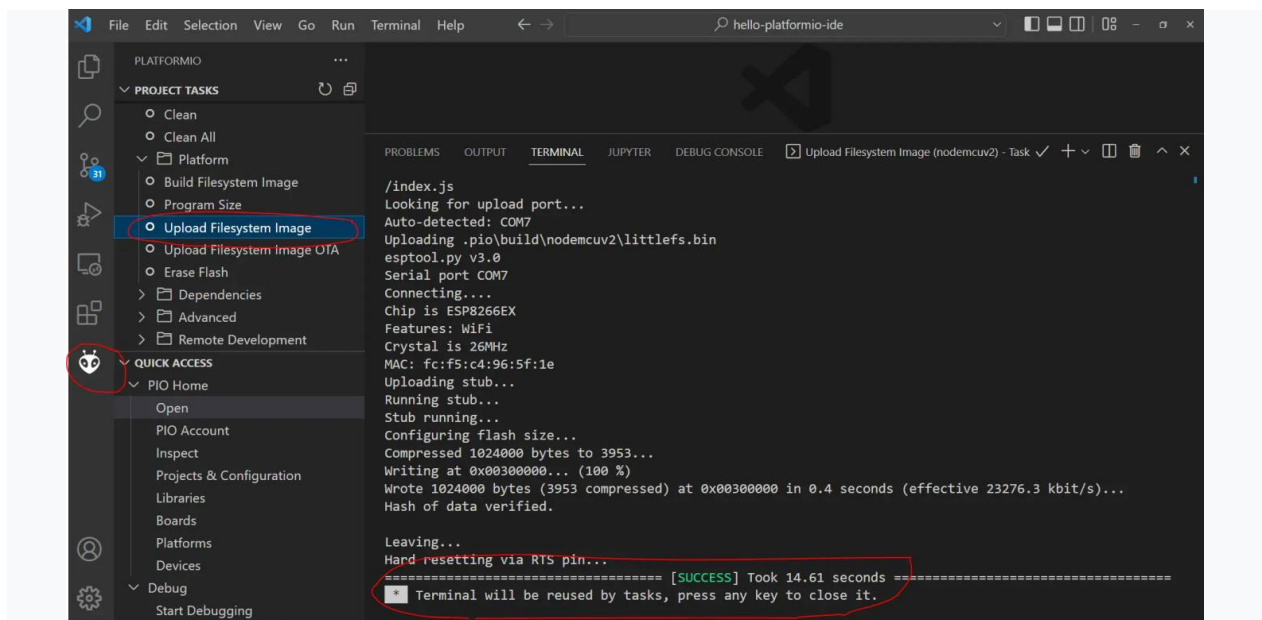
Ako treba da otpremite datoteke u fleš memoriju vašeg mikrokontrolera ili preko SPIFFS-a ili LittleFS-a, trebalo bi da kreirate folder (datoteku) pod nazivom „data“ u korenu vašeg projekta (Slika 18.).





*Slika 18. Datoteka data sa fajlovima potrebnim za kreiranje web stanice*

Zatim ćemo program sa fajlovima web stranice otpremiti u Flash memoriju našeg mikrokontrolera tako što ćemo kliknuti na Upload FileSistem Image na listi zadataka (Slika 19.). Trebalo bi da vidite poruku o uspehu i ako jeste, onda ste uspeali da je otpremite u fleš memoriju vašeg mikrokontrolera.

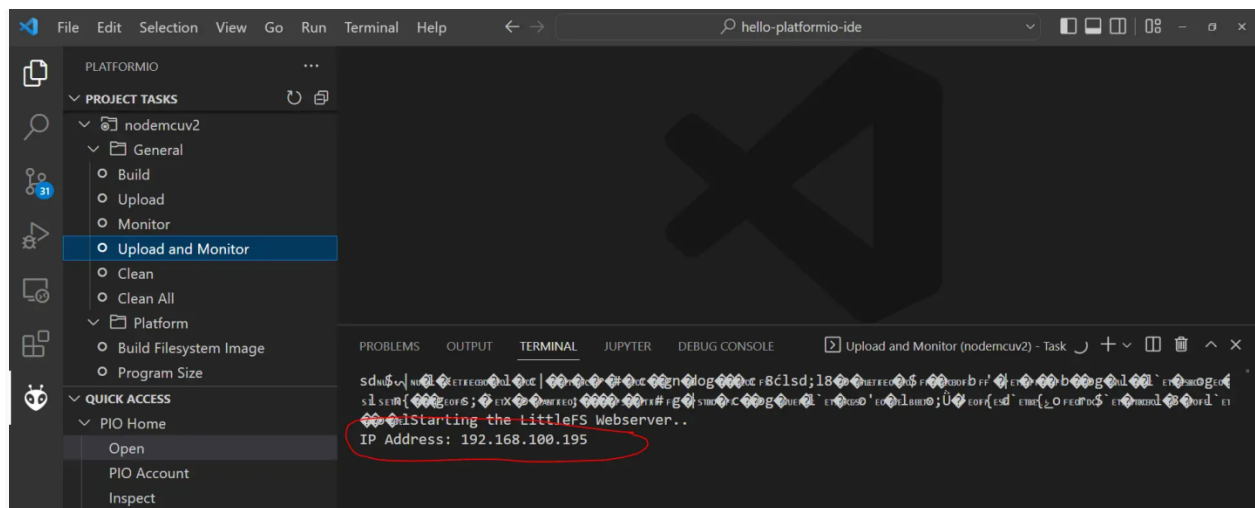


*Slika 19. Potvrda o supešnom otpremanju fajlova potrebnih za realizaciju web stranice*

Potrebno je prilagoditi datoteku main.cpp tako da čita datoteke koje smo dodali u FileSistem našeg mikrokontrolera. Da biste to uradili, preuzmite dodatni materijal za vežbu.

```
/*
    Replace the SSID and Password according to your wifi
*/
const char *ssid = "<REPLACE_WITH_YOUR_WIFI_SSID>";
const char *password = "<REPLACE WITH YOUR WIFI PASSWORD>";
```

Ako sve prođe kako treba, otpremite programski kod, izvršavanjem zadatka Upload and Monitor (Slika 20.). Ako vidite prikazanu IP adresu, to znači da ste uspešno otpremili projekat u mikrokontroler.



Uzmite svoj mobilni telefon ili računar i otvorite pretraživač i unesite IP adresu prikazanu na serijskom monitoru.

Napomena: ispratiti linije programskog koda i utvrditi da li zaista odgovaraju korišćenom kontroleru.