



PlatformIO.org

Embedded Development made easy

Pi and More 9

Nico Maas



Nico Maas

IT Systemelektroniker

Bachelor of Science

mail@nico-maas.de

www.nico-maas.de

@nmaas87

Agenda



- **I. Introduction**
 - What is PlatformIO?
 - Which hardware / frameworks are supported?
 - Which special features do exist?
- **II. Demo**
 - Installation
 - Demo with Arduino Uno / Arduino Due
 - Demo with ESP8266
- **III. End**



PlatformIO is an open source ecosystem for IoT development.

Cross-platform code builder.

Pure Python.

Windows, Linux, Mac.

Continuous Integration and IDE integration.

Arduino and mbed compatible (and a lot more...)



200+ Boards

15+ Platforms:

Atmel AVR, Atmel SAM, Espressif, Freescale Kinetis, Nordic nRF51,
NXP LPC, Silicon Labs EFM32, ST STM32, Teensy, TI MSP430, TI TIVA

10+ Frameworks:

Arduino, CMSIS, WiringPi, libOpenCM3, Energia, SPL, mbed

CLI and IDE (Atom.io)



Easy Development for (nearly) every platform without the need to install additional tools and IDEs.

Crossplatform development.

Build one project for different MCUs.

Library Manager (CLI, Web)

CI Integration (Drone, Travis,...)

IDE Integration (Eclipse, Netbeans, EMACs, VIM, Sublime Text,...)

PlatformIO IDE



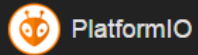
The next generation integrated development environment for IoT

```
1  /**
2   * Blink
3   *
4   * Turns on an LED on for one second,
5   * then off for one second, repeatedly.
6   */
7  #include "Arduino.h"
8
9  void setup()
10 {
11     // initialize LED digital pin as an output.
12     pinMode(LED_BUILTIN, OUTPUT);
13 }
14
15 void loop()
16 {
17     // turn the LED on (HIGH is the voltage level)
18     digitalWrite(LED_BUILTIN, HIGH);
19
20     // wait for a second
21     delay(1000);
22
23     // turn the LED off by making the voltage LOW
24     digitalWrite(LED_BUILTIN, LOW);
25
26     // wait for a second
27     delay(1000);
28 }
29
```




```
(platformio-demo)→ wiring-blink platformio run
[Thu Aug 27 18:29:57 2015] Processing uno (platform: atmelavr, board: uno, framework: arduino)
-----
avr-g++ -o .pioenvs/uno/src/main.o -c -fno-exceptions -fno-threadsafe-statics -g -Os -Wall -ffunction-sections -fdata-sections -MMD -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO_ARCH_AVR -DARDUINO_AVR_UNO -DARDUINO=10605 -DPLATFORMIO=020202 -I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArduinoVariant src/main.cpp
avr-ar rcs .pioenvs/uno/libFrameworkArduinoVariant.a
avr-ranlib .pioenvs/uno/libFrameworkArduinoVariant.a
avr-g++ -o .pioenvs/uno/FrameworkArduino/CDC.o -c -fno-exceptions -fno-threadsafe-statics -g -Os -Wall -ffunction-sections -fdata-sections -MMD -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO_ARCH_AVR -DARDUINO_AVR_UNO -DARDUINO=10605 -I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArduinoVariant .pioenvs/uno/FrameworkArduino/CDC.cpp
avr-g++ -o .pioenvs/uno/FrameworkArduino/HID.o -c -fno-exceptions -fno-threadsafe-statics -g -Os -Wall -ffunction-sections -fdata-sections -MMD -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO_ARCH_AVR -DARDUINO_AVR_UNO -DARDUINO=10605 -I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArduinoVariant .pioenvs/uno/FrameworkArduino/HID.cpp
avr-g++ -o .pioenvs/uno/FrameworkArduino/HardwareSerial.o -c -fno-exceptions -fno-threadsafe-statics -g -Os -Wall -ffunction-sections -fdata-sections -MMD -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO_ARCH_AVR -DARDUINO_AVR_UNO -DARDUINO=10605 -I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArduinoVariant .pioenvs/uno/FrameworkArduino/HardwareSerial.cpp
avr-g++ -o .pioenvs/uno/FrameworkArduino/HardwareSerial0.o -c -fno-exceptions -fno-threadsafe-statics -g -Os -Wall -ffunction-sections -fdata-sections -MMD -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO_ARCH_AVR -DARDUINO_AVR_UNO -DARDUINO=10605 -I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArduinoVariant .pioenvs/uno/FrameworkArduino/HardwareSerial0.cpp
avr-g++ -o .pioenvs/uno/FrameworkArduino/HardwareSerial1.o -c -fno-exceptions -fno-threadsafe-statics -g -Os -Wall -ffunction-sections -fdata-sections -MMD -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO_ARCH_AVR -DARDUINO_AVR_UNO -DARDUINO=10605 -I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArduinoVariant .pioenvs/uno/FrameworkArduino/HardwareSerial1.cpp
avr-g++ -o .pioenvs/uno/FrameworkArduino/HardwareSerial2.o -c -fno-exceptions -fno-threadsafe-statics -g -Os -Wall -ffunction-sections -fdata-sections -MMD -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO_ARCH_AVR -DARDUINO_AVR_UNO -DARDUINO=10605 -I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArduinoVariant .pioenvs/uno/FrameworkArduino/HardwareSerial2.cpp
avr-g++ -o .pioenvs/uno/FrameworkArduino/HardwareSerial3.o -c -fno-exceptions -fno-threadsafe-statics -g -Os -Wall -ffunction-sections -fdata-sections -MMD -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO_ARCH_AVR -DARDUINO_AVR_UNO -DARDUINO=10605 -I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArduinoVariant .pioenvs/uno/FrameworkArduino/HardwareSerial3.cpp
█
```

Library Manager



Get Started

IDE

Development

Libraries

Docs

Donate

Contact

</> Libraries Search

fastled



Examples: [tft display](#) [DS18*](#) [framework:mbed](#) [platform:timsp430](#) [keyword:sensor](#) [author:"Adafruit Industries"](#)

[advanced search syntax](#)

Frameworks

Platforms

FastLED by Daniel Garcia, Mark Kriegsman

17 53 Arduino Atmel AVR, Atmel SAM, Teensy, Freescale Kinetis, ST STM32, Nordic nRF51, NXP LPC

FastLED is a library for programming addressable rgb led strips (APA102/Dotstar, WS2812/Neopixel, LPD8806, and a dozen others) acting both as a driver and as a library for color management and fast math.

led noise rgb math fast

First Previous 1 Next Last

Demo - Installation



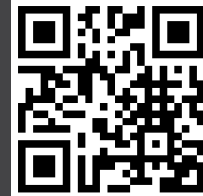
Goto: <http://platformio.org/#!/get-started>

Download the version for your system


Install

Done

Demo - Usage



👉 Step 1

Initialize new *PlatformIO* based project using  button on the Toolbar or `Menu: PlatformIO > Initialize new PlatformIO Project or update existing`.

👉 Step 2

Put your source code `*.h, *.c, *.cpp, *.S, *.ino, etc.` files to `src` directory.

👉 Step 3

Process the project environments using `Menu: PlatformIO > Build / Upload / Clean` or buttons on the Toolbar:

- ✓ Build
- ➔ Upload
- 🗑 Clean

Demo - Demonstration



Demo Time 😊!

Questions?



Thanks a lot for your time!

Sources



- Logos, Icons and Screenshots
 - <http://www.platformio.org/>