

# Plattform IO Einführung

Neue Trends und ihre Umsetzung im EL-Lehrplan

Handouts für online Seminar am 3.12.2020

Thomas Benetik

# Übersicht

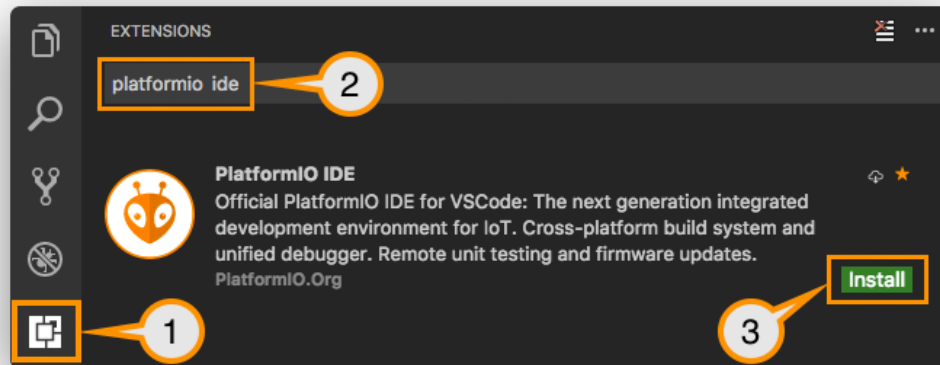
- Installation von VS Code
- Installation von PlatformIO
- neues Projekt anlegen
- Bibliotheken einbinden
- Bibliotheken anlegen
- Features

# Installation von VS Code

- VS Code herunterladen und installieren  
<https://platformio.org/install/ide?install=vscode>

Thank you for choosing PlatformIO IDE for VSCode

- 👉 [Download](#) and install official Microsoft's Visual Studio Code, PlatformIO IDE is built on top of it
- 👉
  1. **Open** VSCode Extension Manager
  2. **Search** for official [PlatformIO IDE](#) extension
  3. **Install** PlatformIO IDE.



- 👉 Check [Quick Start](#) guide (highly recommended).

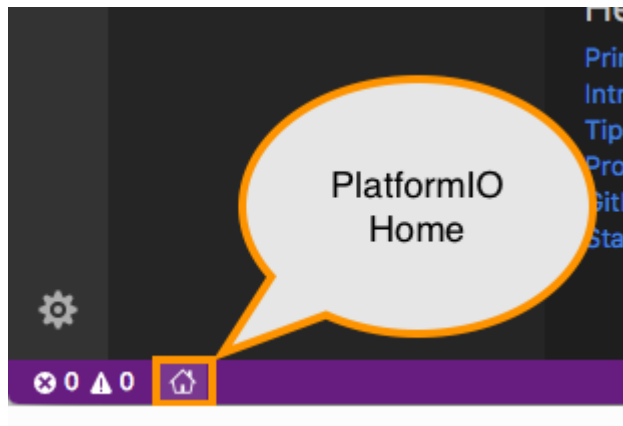
- evtl. muss noch eine aktuelle Version von Python installiert werden

# Bedienelemente

- Quick start guide auf: <https://docs.platformio.org/en/latest/integration/ide/vscode.html#quick-start>
- Platform IO Toolbar:



1. PlatformIO Home
2. PlatformIO: Build
3. PlatformIO: Upload
4. PlatformIO: Clean
5. Serial Port Monitor



# Neues Projekt anlegen

Welcome to PlatformIO

Core 5.0.3 · Home 3.3.1

Quick Access

- + New Project
- Import Arduino Project
- Open Project
- Project Examples

Show at startup

EXPLORER

OPEN EDITORS 1 UNSAVED

- main.cpp Testprojekt · src
- PIO Home

UNTITLED (WORKSPA...)

- Reaktionstest
- Interrupts
- MyShield\_Starter
- Testprojekt
  - .pio
  - .vscode
  - include
  - lib
  - src
    - main.cpp
    - test
  - .gitignore
  - platformio.ini

Project Wizard

This wizard allows you to **create new** PlatformIO project or **update existing**. In the last case, you need to uncheck "Use default location" and specify path to existing project.

Name: Testprojekt

Board: Arduino Uno

Framework: Arduino

Location:  Use default location ?

Cancel Finish

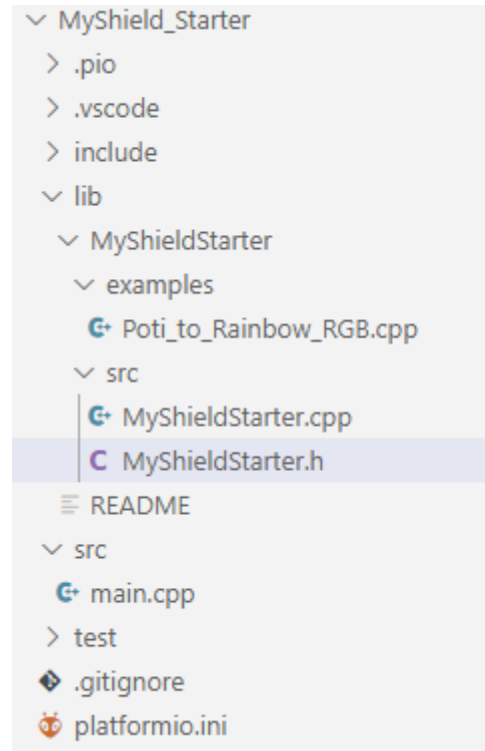
# Installation von Bibliotheken

The image shows a multi-step process for installing a library in PlatformIO. It consists of three main parts:

- Registry Search:** The top part shows the PlatformIO Registry website. A search bar contains the text "lcd". A red box highlights the search button. Below the search bar, the "LiquidCrystal" library by F Malpartida is listed. A red box highlights the "LiquidCrystal" entry in the list.
- Library Details:** The middle part shows the details for the "LiquidCrystal" library. The "Installation" section shows the version "1.5.0" and an "Add to Project" button. A red box highlights the "Installation" tab in the navigation bar.
- IDE Configuration:** The bottom part shows the PlatformIO IDE interface. The "platformio.ini" file is selected in the file explorer (highlighted with a red box). The code in the editor shows the configuration for the "uno" board. A red box highlights the line: `lib_deps = bitbucket-fmalpartida/LiquidCrystal @ ~1.5.0`. The "Library Dependencies" section on the right shows the configuration for the library dependencies, with a red box highlighting the relevant code: `[env:my_build_env] platform = atmelavr framework = arduino lib_deps = # RECOMMENDED # Accept new functionality in a backwards compatible manner and patches bitbucket-fmalpartida/LiquidCrystal @ ^1.5.0 # Accept only backwards compatible bug fixes # (any version with the same major and minor versions, and an equal or greater patch version) bitbucket-fmalpartida/LiquidCrystal @ ~1.5.0`

# Eigene Bibliothek anlegen

- Fertige Ordnerstruktur für eigene Bibliotheken:
  - library
    - src
      - library.cpp
      - library.h
    - examples
      - example1.cpp
      - example2.cpp



```
17
18 #define Dig0_en 13 //active low
19 #define Dig1_en 12 //active low
20
21 class MyShieldStarter
22 {
23 private:
24     /* data */
25 public:
26     MyShieldStarter(/* args */);
27     ~MyShieldStarter();
28
29     void RGB(int r, int g, int b);
30     int POTI();
31     void sevenSeg(int number, int digit);
32     void begin();
33 };
34
35 #endif
```

# Features

- code snippets:

```
switch
```

- switch Code snippet for switch
- switch

```
switch (expression)
{
case /* constant-expression */:
/* code */
break;

default:
break;
}
```

- auto completion:

```
Serial.
```

- available
- availableForWrite
- begin void HardwareSerial::begin(unsigned...
- clearWriteError
- end
- find
- findUntil
- flush
- getTimeout
- getWriteError
- %- operator bool
- \*+ operator bool
- parseFloat



# Features

- hover for defined values:

```
#define DDRB _SFR_IO8(0x04)
Wird erweitert auf:
_SFR_IO8(0x04)
DDRBB |= (1 << DDB3);
```

- view definitions in other libraries:  
(Rechtsklick auf Funktion)

```
void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(1, HIGH);
}
```

Go to Definition	F12
Go to Declaration	
Go to References	Shift+F12