

A detailed illustration of an ESP32 development board, showing its various components and pin headers. The board is oriented vertically. At the top, there is a USB-C port, a USB-A port, and a 5V power input. Below these are several pin headers labeled with pin numbers and functions. The central part of the board features the ESP32 microcontroller chip, surrounded by various passive components like resistors and capacitors. The bottom of the board has a large pin header with pins numbered 1 through 40.

IoT Crashcourse for Begyndere

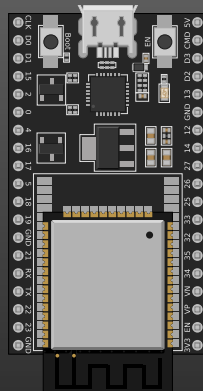
Med ESP32

Jacob Bechmann Pedersen

January 15, 2024

Indhold

1. Hvem er jeg?
2. Formål
3. Ressourcer
4. Setup af VSCode og PlatformIO
 - Setup VSCode
 - Setup PlatformIO
5. Setup af ESP32 projekt i PlatformIO
 - Tilføj libraries til ESP32 projekt
 - Indlæsning af ESP32 Projekt i PlatformIO
6. IoT basics
7. Byg en simpel ESP32 webserver
 - Eksempel: Simpel Server
8. WebSockets på ESP32
 - Eksempel: WebSocket Server
9. MQTT på ESP32
 - Eksempel: MQTT Client



Figur 1: ESP32 DevkitC v4, boardet vi skal arbejde med

Hvem er jeg?



Jacob Bechmann Pedersen

- Kursus-/Foredragsholder om embedded elektronik, programmering og Arduino
- Tidl. Embedded electronics engineer hos DTU Elektro, Automation and Control
 - Robotter, embedded Linux, autonome systemer
- Tidl. Embedded software developer hos Oticon
 - Applikationer til høreapparaternes OS, unit- og device testing
- Underviser på MakerCamp
 - "Inventors" linje - 12-16 årige
- Frivillig i Coding Pirates 2016-2018
- Electronic Design Engineer (AU, 2019)
- Startede med Arduino i 2014



Formål

- At forstå grundprincipperne bag IoT
 - Topologier
 - Protokoller, herunder:
 - HTTP
 - Websockets
 - MQTT
- Programmere simple implementationer
 - På ESP32
 - Med Arduino platformen

Ressourcer

Nogle nyttige links:

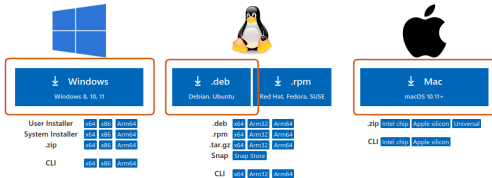
- <https://github.com/iakop/IoT-Crashcourse>
 - Præsentation og kode til denne workshop
- <https://code.visualstudio.com/>
 - Download af Visual Studio Code
- <https://platformio.org/>
 - Download af PlatformIO
- <https://www.arduino.cc/en/reference>
 - Reference for keywords i Arduino
- <http://mqtt-explorer.com/>
 - MQTT client til at udforske topics på en broker
- <https://nodered.org/>
 - Editorbaseret løsning til flowbaseret IoT programmering

Setup af VSCode og PlatformIO

Setup VSCode

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

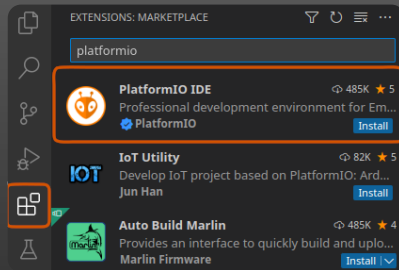


Figur 2: Visual Studio Code downloadsiden har versioner til mange forskellige arkitekturer, typisk vil default knappen hente den korrekte installer

- Download Visual Studio Code fra linket:
 - <https://code.visualstudio.com/Download>
- Tryk på den store knap for dit OS
- Kør installationsprocessen, dette skulle ske uden stort besvær
- **HVIS** det ikke virker, prøv en af disse:
 - Windows:
 - Hvis du ikke har admin-rettigheder, prøv at hente **User Installer** (typisk x64-versionen)
 - Linux:
 - Hvis du ikke bruger Ubuntu, tjek din package managers repositorier, eller prøv **CLI** installeren
 - Mac OS X:
 - Prøv **Universal .zip**, eller måske App Store? 🙄

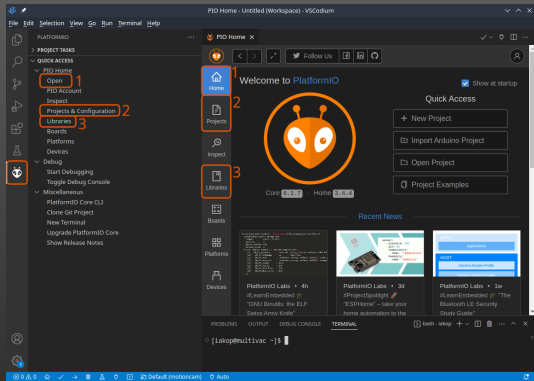
Setup PlatformIO

- Når VSCode er installeret og startet op, navigér til **Extensions** fanen
 - Her, søg efter PlatformIO
 - Vælg extensionen på figuren, og tryk **install**
 - VSCode installerer og indstiller automatisk PlatformIO



Figur 3: Installation af **PlatformIO** i VSCode

Setup PlatformIO



Figur 4: PlatformIO standard view, med Home , Projects og Libraries , samt Quick Access

- PlatformIO extensionen kan åbnes ved at klikke på fanen PlatformIO
- Herunder findes en række muligheder, de vigtigste:

1. Open / Home

- Hovedsiden til PlatformIO, med quick access og faner til resten af funktionerne

2. Projects & Configuration / Projects

- Projektside, til at oprette og indstille softwareprojekter

3. Libraries

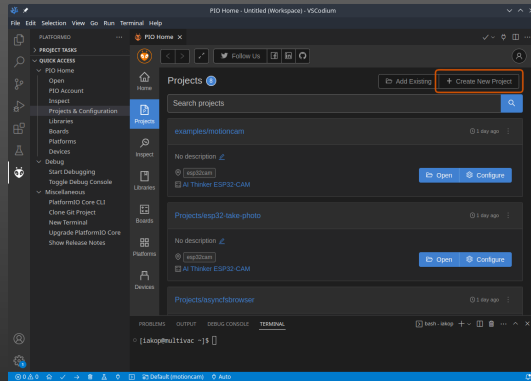
- Til søgning og tilføjelse af Libraries til PlatformIO projekter



Setup af ESP32 projekt i PlatformIO

Setup af ESP32 projekt i PlatformIO

- Klik på fanen **Projects** for at få visningen af projekter frem
- For at skabe et nyt projekt, klik på **Create New Project**



Figur 5: **Projects** fanen i PlatformIO. Her kan projekter oprettes og redigeres i GUI'en

Setup af ESP32 projekt i PlatformIO

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.

1 Name: coolnewproject

2 Board: DOIT ESP32 DEVKIT V1

3 Framework: Arduino

Location: ☒ Use default location ⓘ

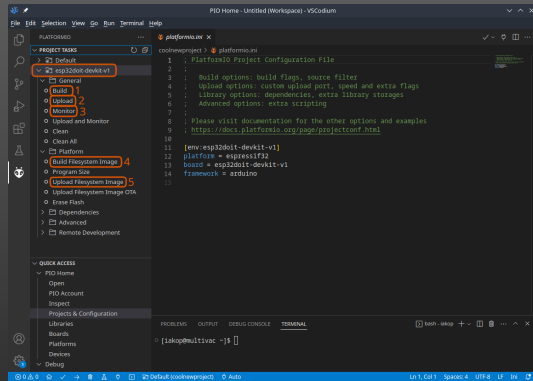
Cancel Finish

Figur 6: Project Wizard dialogen i PlatformIO, med indstillinger for navn, board og framework

- Der vil komme en Project Wizard dialog
- Den har 3 felter, som i vores tilfælde udfyldes således:
 - Name
 - Et passende navn til projektet, eks. "coolnewproject"
 - Board
 - DOIT ESP32 DEVKIT V1
 - Framework
 - Arduino
- Afslut ved at trykke Finish

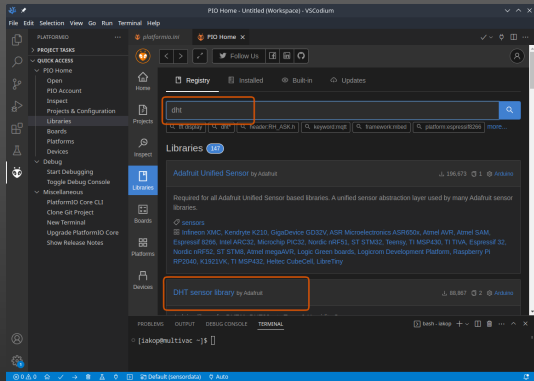
Setup af ESP32 projekt i PlatformIO

- PlatformIO vil generere et projekt og automatisk sætte værktøjerne op
 - Dette kræver en internetforbindelse
- Når det er færdigt, indlæses `platformio.ini`
 - Denne fil indeholder indstillingerne for projektet og kan redigeres i hånden
- På siden ses nogle **Project Tasks** :
 1. Build
 - Bygger et image til at programmere hardwaren med
 2. Upload
 - Uploader imaget til hardwaren gennem en automatisk detekteret USB/UART forbindelse
 3. Monitor
 - Overvåger UART forbindelsen til hardwaren (Baud raten kan indstilles i `platformio.ini`)
 4. Build Filesystem Image
 - Bygger image af et filsystem til hardwaren (ligger i `data` mappen under projektet)
 - `data` mappen skal oprettes manuelt
 - Filsystemet kan specificeres i `platformio.ini`
 5. Upload Filesystem Image
 - Uploader det byggede filsystem til boardet
 - **VIGTIGT** : Monitor kan ikke være aktivt under upload



Figur 7: Åbent projekt i PlatformIO, viser `platformio.ini` samt hvilke **Project Tasks** der er mulige for projektet

Tilføj libraries til ESP32 projekt

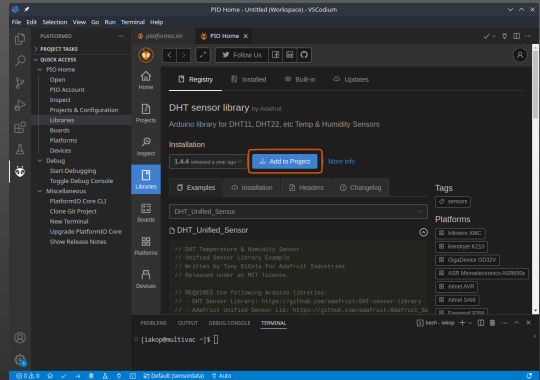


Figur 8: Libraries fanen i PlatformIO. Her kan libraries søges frem og tilføjes til projekter

- For at tilføje eksterne libraries til ens projekt, kan man under Libraries fanen søge efter libraries
- De kan fremsøges under Registry
- Installerede libraries kan vises under Installed
- Herefter klik på det relevante library

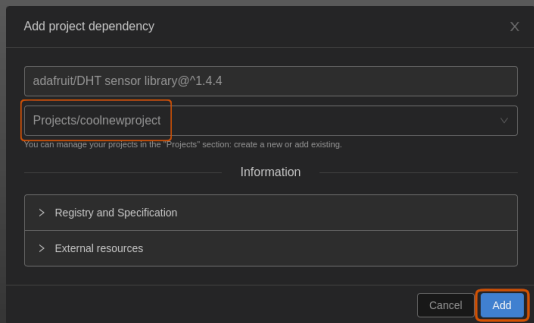
Tilføj libraries til ESP32 projekt

- Inde under det pågældende library kan der findes:
 - Eksempler
 - Headers
 - mm.
- Klik **Add to Project** for at tilføje den til et projekt



Figur 9: DHT sensor library i PlatformIO. Kan tilføjes til projekter, og understøtter Arduino frameworket

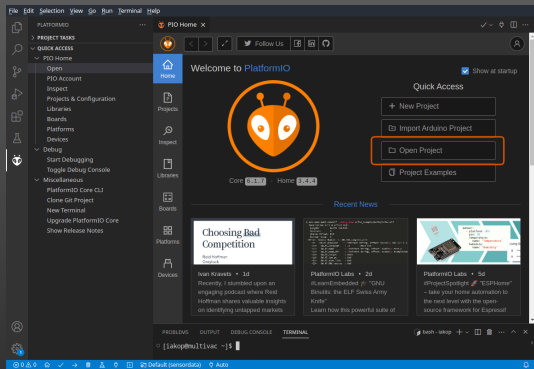
Tilføj libraries til ESP32 projekt



Figur 10: Add project dependency dialogen i PlatformIO. Her vælges hvilket projekt der skal tilføjes et library

- Der åbnes en Add project dependency dialog
- Vælg under Select a project , hvilket projekt der skal bruge libraryet
- Klik Add
- PlatformIO tilføjer automatisk en lib_deps dependency i platformio.ini , og sætter libraryet op

Indlæsning af ESP32 Projekt i PlatformIO

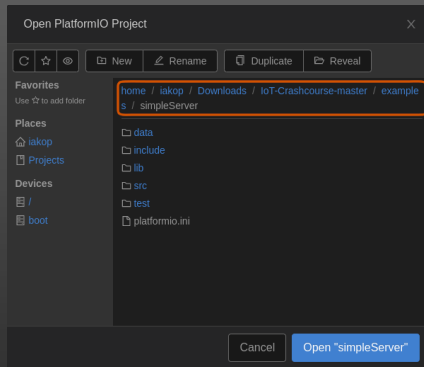


Figur 11: Home fanen i PlatformIO, med markeret indstilling for at indlæse et projekt fra disk

- Projekterne i denne workshop bruger specifikke libraries og indstillinger
- For at have dem sat op hurtigt, kan hele projekter indlæses fra Github repoet
- Download hele workshopkens materialer her:
 - <https://github.com/iakop/IoT-Crashcourse/archive/refs/heads/master.zip>
- Pak dem ud hvor de kan findes
- Under Home fanen, klik **Open Project**

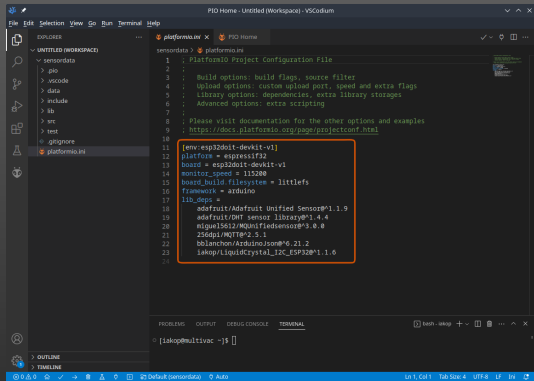
Indlæsning af ESP32 Projekt i PlatformIO

- I **Open PlatformIO Project** dialogen, åbn mappen hvor eksemplerne ligger
- Hvis **Open** knappen f.eks. siger **Open "simpleServer"** står dialogen det rigtige sted
- Klik **Open** knappen



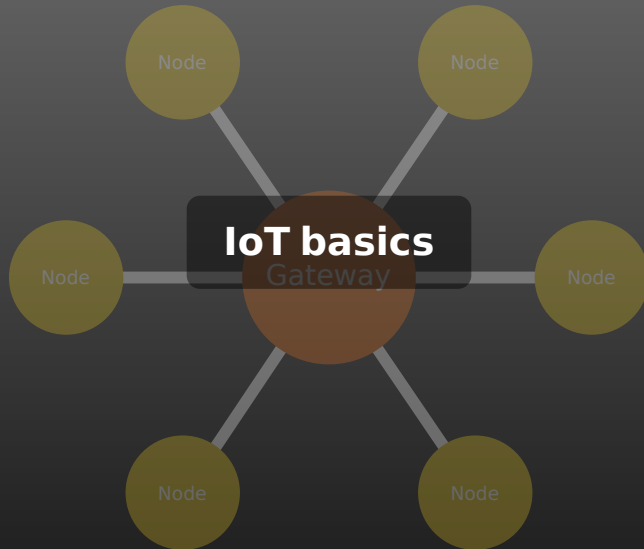
Figur 12: **Open PlatformIO Project** dialogen i PlatformIO. For at åbne et projekt, skal man finde den på disken, og stå i mappen e.g.: `Downloads/IoT-Crashcourse-master/examples/simpleServer`

Indlæsning af ESP32 Projekt i PlatformIO



- Når projektet er indlæst, åben **platformio.ini** for projektet
- Den specificerer alle libraries og komponenter projektet afhænger af
- Værktøjerne og indstillingerne for projektet sættes automatisk op af PlatformIO

Figur 13: Eksempel på et **platformio.ini** for et projekt. Bemærk **monitor-speed**, **filesystem** og **lib_deps** er defineret



IoT basics

- IoT (Internet of Things), er en fællesbetegnelse for netværksopkoblede genstande
- En sådan genstand indeholder typisk:
 - En microprocessor eller -computer
 - Sensorer
 - Aktuatorer
 - Trådet eller trådløs opkobling



Figur 14: Nedis SmartLife adskilt for at komme til indmaden. Indeholder bl.a. et TYWE3S WiFi modul og en HLW8012 effektsensor
Kilde: <https://callaa.github.io/2021/01/26/liberating-nedis-smartplug.html>

IoT basics



Figur 15: Star-topologi, hvor hver enhed kommunikerer med en central gateway til resten af internettet



Figur 16: Tree-topologi, hvor enhederne er forbundet i forgreninger, og videre-giver informationer herigennem til gateway



Figur 17: Mesh-topologi, hvor alle enheder kommunikerer internt, og videre giver information gennem hinanden til gateway

- Kommunikation mellem enheder foregår på mange forskellige måder
- Nogle gængse netværkstopologier er bla.:
 - Star
 - Tree
 - Mesh

IoT basics

- Der findes en række protokoller for enheder at kommunikere gennem
- Dem vi vil fokusere på:
 - HTTP
 - Den gængse Hypertext Transfer Protocol, der bruges til at overføre webindhold, bla. mellem servere og browsere
 - WebSocket
 - En fuld duplex (tovejs kommunikation) protokol til hurtig samtidig kommunikation mellem klient og server - lav overhead
 - MQTT
 - (Oprindeligt forkortelse for MQ (Message Queue) Telemetry Transport)
Publish-subscribe baseret protokol mellem enheder og central broker - lav overhead



Figur 18: HTTP logo
Kilde: https://en.wikipedia.org/wiki/File:HTTP_logo.svg
Licens: Public Domain



Figur 19: WebSocket logo
Kilde: <https://logodix.com/logos/1825947>
Licens: Non-Commercial



Figur 20: MQTT logo
Kilde: <https://en.wikipedia.org/wiki/File:Mqtt-hor.svg>
Licens: Public Domain

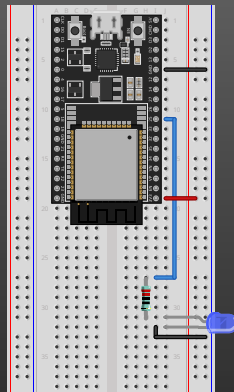


A screenshot of a web browser window. The browser's address bar shows 'http://simpleserver.local/'. The page content includes the title 'Simple Server', a text label 'LED state: 1', a green button labeled 'ON', and a red button labeled 'OFF'. A dark semi-transparent banner is overlaid on the center of the image, containing the title text.

Byg en simpel ESP32 webserver

Simpel Server

- Til vores eksempel skal vi bruge en breadboard opstilling
 - Et ESP32 board
 - En LED
 - En 220Ω modstand
- HTML og Arduino programmet gennemgår vi i fællesskab
- Kildefiler kan hentes på:
 - <https://github.com/iakop/IoT-Crashcourse/tree/master/examples/simpleServer>



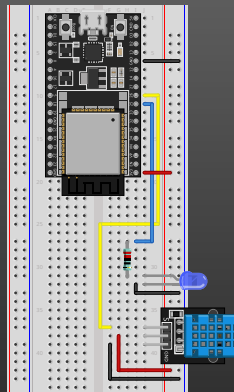
Figur 21: Breadboard setup med ESP32 og LED



WebSockets på ESP32

WebSocket Server

- Til dette eksempel tilføjes en sensor til vores opstilling
 - Et ESP32 board
 - En LED
 - En 220Ω modstand
 - Et DHT11 temperatur-/fugtighedssensor modul
- Der skal tilføjes Javascript og WebSocket forbindelse, som vi gennemgår i fællesskab
- Kildefiler kan hentes på:
 - <https://github.com/iakop/IoT-Crashcourse/tree/master/examples/websocketServer>

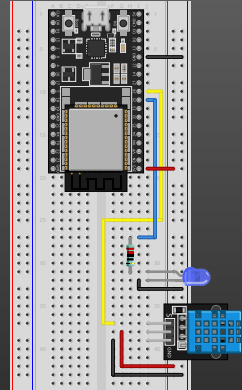


Figur 22: Breadboard setup med ESP32, LED og DHT11 sensor

MQTT på ESP32

MQTT Client

- Samme opstilling som sidst
 - Et ESP32 board
 - En LED
 - En 220Ω modstand
 - Et DHT11 temperatur-/fugtighedssensor modul
- Al server kode skiftes ud med client kode, som forbinder via SSL til en hosted MQTT broker
- Kildefiler kan hentes på:
 - <https://github.com/iakop/IoT-Crashcourse/tree/master/examples/mqttClient>



Figur 23: Breadboard setup med ESP32, LED og DHT11 sensor

MQTT Client

The screenshot shows the MQTT Explorer interface. On the left, a sidebar lists connections, with 'Bechmann' selected. The main area displays the configuration for this connection. The title is 'MQTT Connection' with the URL 'mqtt://mqtt.bechmann.xyz:8883/'. Below this, there are two toggle switches: 'Validate certificate' (off) and 'Encryption (tls)' (on). The 'Name' field contains 'Bechmann'. Further down, there are three input fields: 'Protocol' (set to 'mqtt://'), 'Host' (set to 'mqtt.bechmann.xyz'), and 'Port' (set to '8883'). Below these are 'Username' and 'Password' fields, both of which are empty. At the bottom of the dialog, there are four buttons: 'DELETE' (with a trash icon), 'ADVANCED' (with a gear icon), 'SAVE' (yellow), and 'CONNECT' (blue).

Figur 24: MQTT Explorer Connection dialogvindue, her med forbindelsen til `mqtt.bechmann.xyz` defineret

- MQTT Explorer kan bruges til at tjekke topics:
 - <http://mqtt-explorer.com/>
- Indstillinger for forbindelse til workshopens offentlige server:
 - Name: `Bechmann` (Valgfri)
 - Validate certificate: `off`
 - Bug i MQTT Explorers cert storage forhindrer validering
 - Encryption (tls): `on`
 - Protocol: `mqtt://`
 - Host: `mqtt.bechmann.xyz`
 - Port: `8883`
 - Username: `blank`
 - Password: `blank`