

IoT Crashcourse 2

Sensordata og Prototyper

Jacob Bechmann Pedersen

January 27, 2024

Indhold

1. Hvem er jeg?

2. Formål

3. Ressourcer

4. Setup af VSCode og PlatformIO

Nødvendig software

Setup VSCode

Setup PlatformIO

5. Setup af ESP32 projekt i PlatformIO

Tilføj libraries til ESP32 projekt

6. Sensordata - opsamling og visning

Forklaring

Fksemnel

Indlæsning af ESP32 Projekt i PlatformIO

MOTT Client til dataovervågning

7. Adgangskontrol med RFID og elektronisk lås

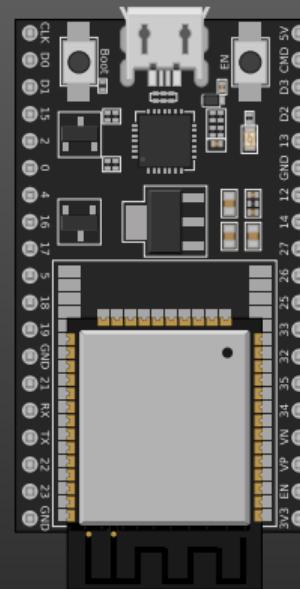
Forklaring

Eksempler

8. Overvågning med PIR sensor og kamera

Forklaring

Eksempler



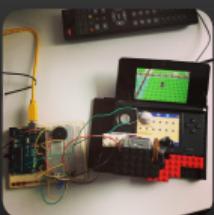
Figur 1: ESP32 DevkitC v4, boardet vi skal arbeide med

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 dykke ned i diverse komponenter
 - Deres relevans for IoT løsninger
 - Hvordan de fungerer:
 - Eksempelkredsløb - på breadboard
 - Eksempelkode - med MQTT, HTTP og Websockets
- Bygge og programmere simple prototyper på IoT løsninger
 - På ESP32 med Arduino framework
 - Vha. PlatformIO og VSCode

Ressourcer

Nogle nyttige links:

- <https://github.com/iakop/IoT-Crashcourse-2>
 - 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

Nødvendig software

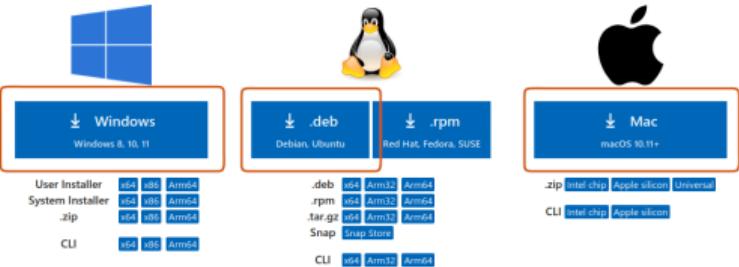
For at Platformio og VSCode kan køre kræves følgende software afhængigt af platform:

- Windows
 - Du skal så vidt muligt have fulde administratorrettigheder
 - Git (Download):
 - <https://git-scm.com/download/win> (64-bit Git for Windows Setup, gælder for de fleste)
 - CP210x chip driver til ESP32 boardet:
 - <https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers?tab=downloads> (CP210x Universal Windows Driver)
 - Download, udpak alle, højreklik `silabser.inf` filen, klik **Install /Installér**
- Linux
 - Vær sikker på du har din distibutions `python 3` og `python 3 venv`, e.g.:
 - Debian/Ubuntu/Mint: `sudo apt-get install python3 python3-venv`
 - Arch/Manjaro: `sudo pacman -S python python-virtualenv`
- Mac OS X
 - XCode Command Line Tools (guide):
 - <https://www.freecodecamp.org/news/install-xcode-command-line-tools/>

Setup VSCode

Download Visual Studio Code

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

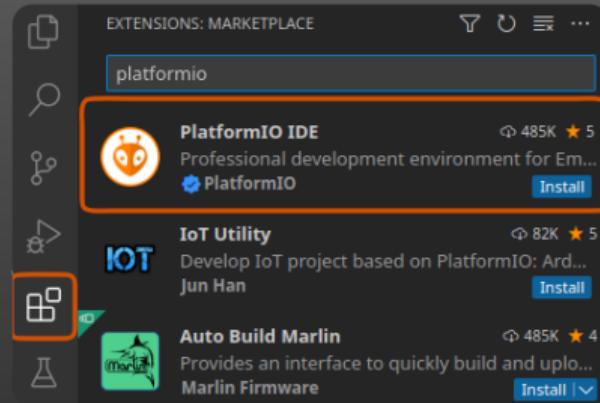


Figur 2: [Visual Studio Code](#) downloadsiden har versioner til mange forskellige arkitekturen, typisk vil default knappen hente den korrekte installer

- 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](#) installereren
 - Mac OS X:
 - Prøv [Universal .zip](#), eller måske App Store? 🤔

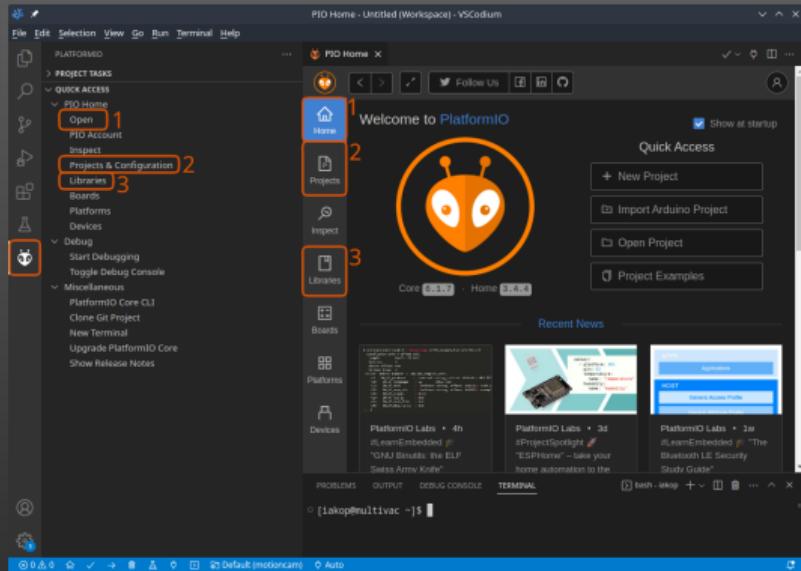
Setup PlatformIO

- 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: Installion 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

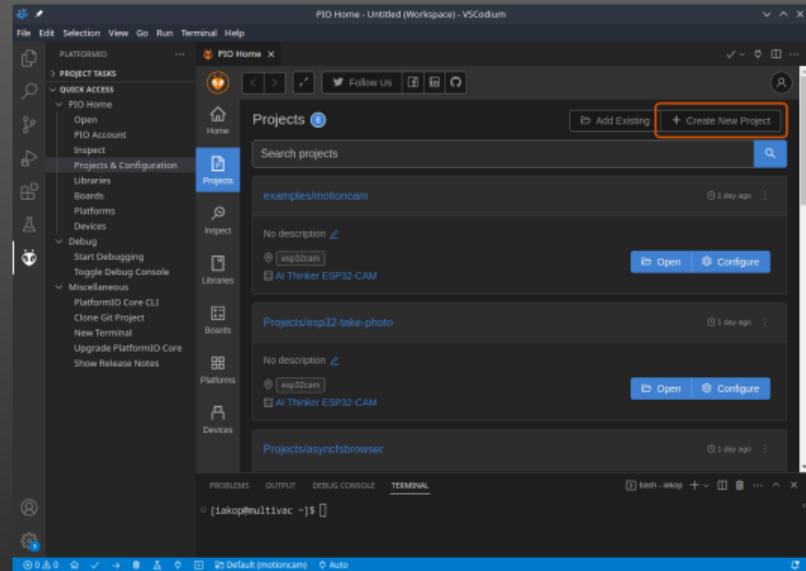
- Til søgning og tilføjelse af Libraries til PlatformIO projekter



Setup af ESP32 projekt i PlatformIO

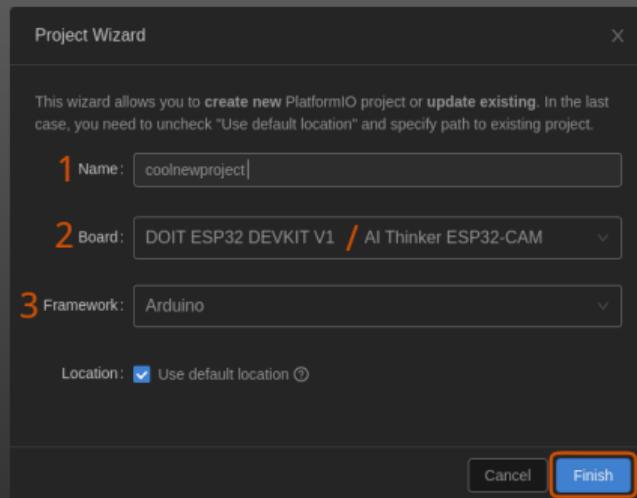
Setup af ESP32 projekt i PlatformIO

- 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



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
 - hhv. DOIT ESP32 DEVKIT V1 eller Ai Thinker ESP32-CAM
 - 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 imagem 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 filesystem 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 filesystem til boardet
 - **VIGTIGT** : Monitor kan ikke være aktivt under upload

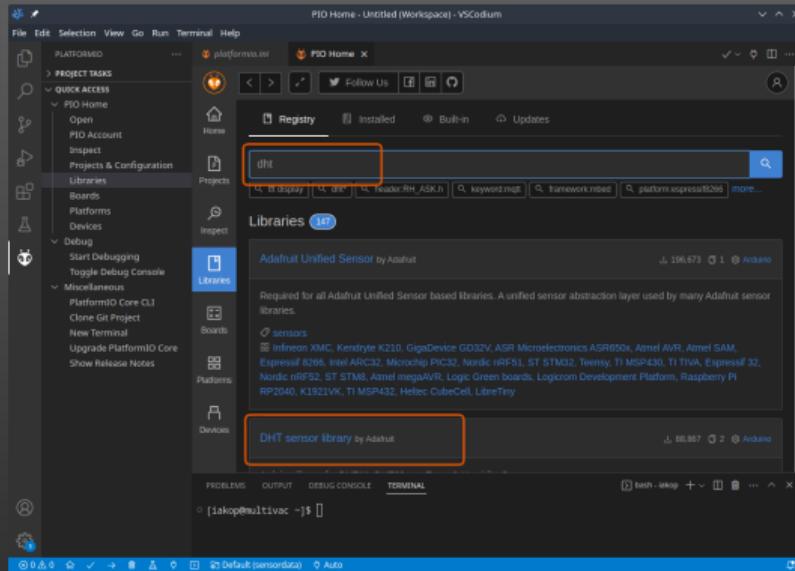
The screenshot shows the PlatformIO IDE interface. On the left, the 'PROJECT TASKS' sidebar lists several options: General (Build 1, Upload 2, Monitor 3), Platform (Build Filesystem Image 4, Program Size, Upload Filesystem Image 5), and Advanced (Upload Filesystem Image GTA, Erase Flash, Dependencies, Remote Development). The 'Build' task is highlighted with a red box. On the right, the main area displays the `platformio.ini` configuration file content:

```
coolnewproject> platformio.ini
1 ; PlatformIO Project Configuration File
2 ;
3 ; Build options: build flags, source filter
4 ; Upload options: custom upload port, speed and extra flags
5 ; Library options: dependencies, extra library storages
6 ; Advanced options: extra scripting
7 ;
8 ; Please visit documentation for the other options and examples
9 ; https://docs.platformio.org/page/projectconf.html
10 ;
11 [env:esp32dool-devkit-v1]
12 platform = espressif32
13 board = esp32dool-devkit-v1
14 framework = arduino
15
```

The bottom of the screen shows the terminal window with the command `[iskop@multivac ~]$`.

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

Tilføj libraries til ESP32 projekt

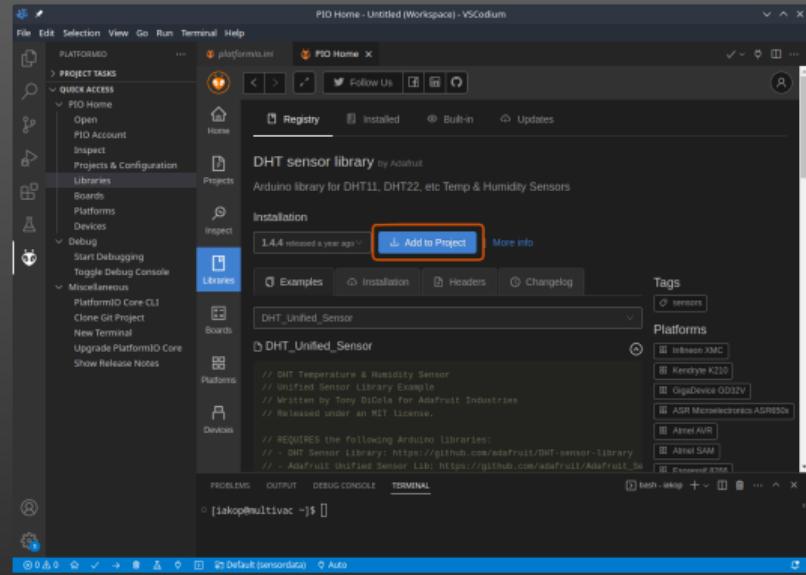


Figur 8: **Libraries** fanen i PlatformIO. Her kan libraries søges frem og tilføjes til projekter

- 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

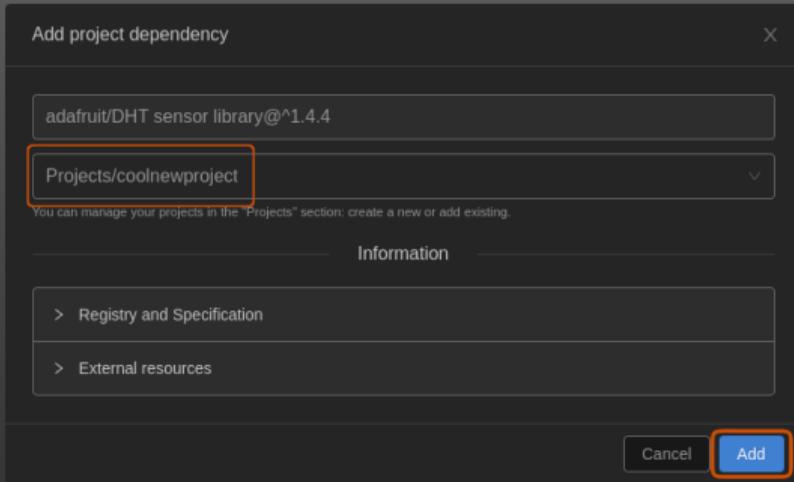
Tilføj libraries til ESP32 projekt

- 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

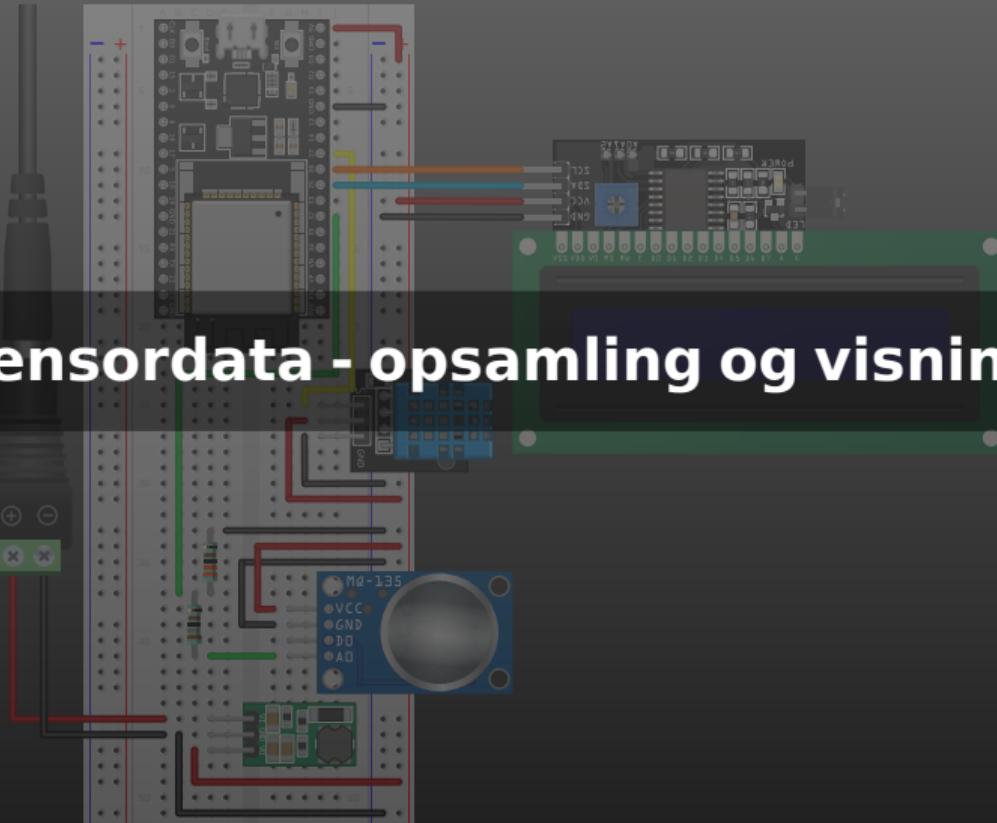
Tilføj libraries til ESP32 projekt



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

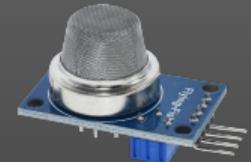
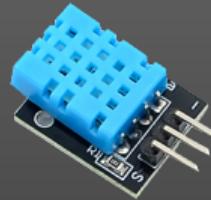
- Der åbnes en [Add project dependency](#) dialog
- Vælg under [Select a project](#), hvilket projekt der skal bruge librariet
- Klik [Add](#)
- PlatformIO tilføjer automatisk en [lib_deps](#) dependency i [platformio.ini](#), og sætter librariet op

Sensordata - opsamling og visning



Sensordata - opsamling og visning

- Det første eksempel handler om opsamling og visning af data
- Formålet er at samle datapunkter, sende dem til fælles broker, og vise dem
- Til dette vil der blive brugt:
 - DHT11
 - En temperatur- og fugtighedssensor
 - Kan læse mellem 0-50°C
 - Mellem 20-90%RH
 - MQ-135
 - CO2
 - CO (kulilte)
 - Alkohol
 - Toluen (opløsningsmiddel i lim og maling)
 - NH4 (Ammonium)
 - Acetone
 - Måler én ad gangen bestemt af applikationsområdet
 - LCD
 - Kan styres over I2C
 - 16x2 karakterer
 - MQTT
 - Publish-subscribe baseret protokol mellem enheder og central broker



Figur 11: DHT11 temperatur- og fugtighedssensor

Figur 12: MQ-135 gassensor

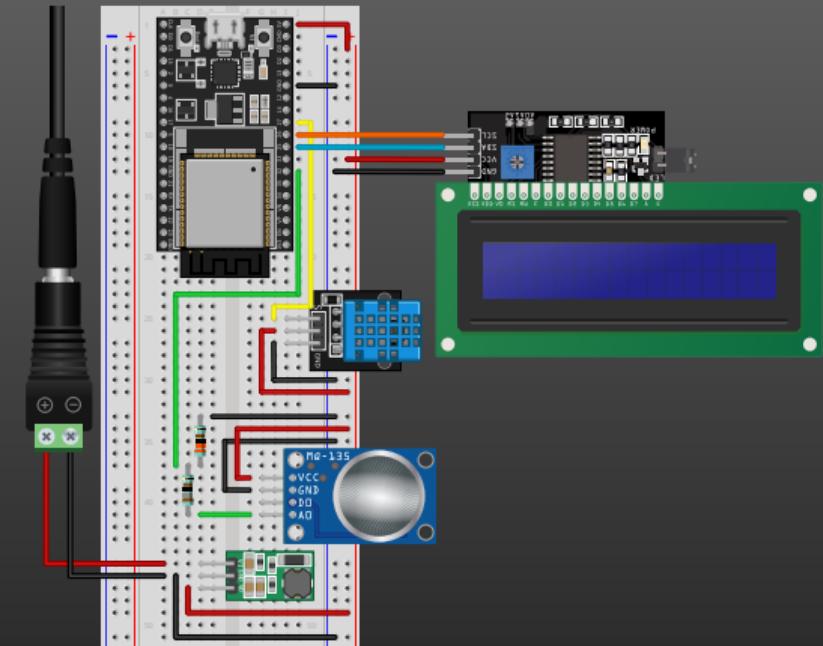


Figur 13: LCD med I2C modul

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

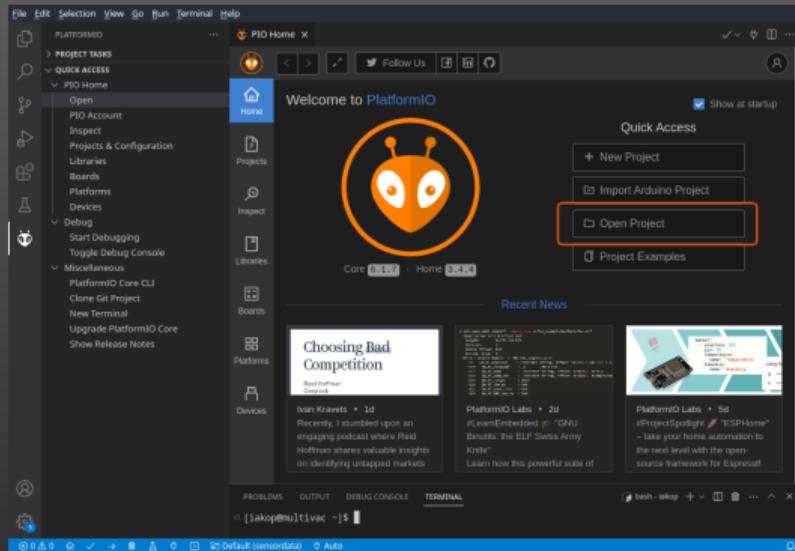
Sensordata - opsamling og visning

- Breadboard opstillingen ser således ud
 - Et ESP32 board
 - Et DHT11 modul
 - Et LCD med I2C modul
 - En MQ-135 sensor
 - En spændingdeler á $1,8\text{k}\Omega$ og $3,3\text{k}\Omega$
 - En 12V-5V DC-DC converter
 - En DC strømforsyning
- Programmet gennemgår vi i fællesskab
- Kildefiler kan hentes på:
 - <https://download-directory.github.io?url=https://github.com/iakop/IoT-Crashcourse-2/tree/master/examples/sensordata>



Figur 15: Breadboard setup med ESP32, DHT11, MQ-135, Buck converter og LCD

Indlæsning af ESP32 Projekt i PlatformIO

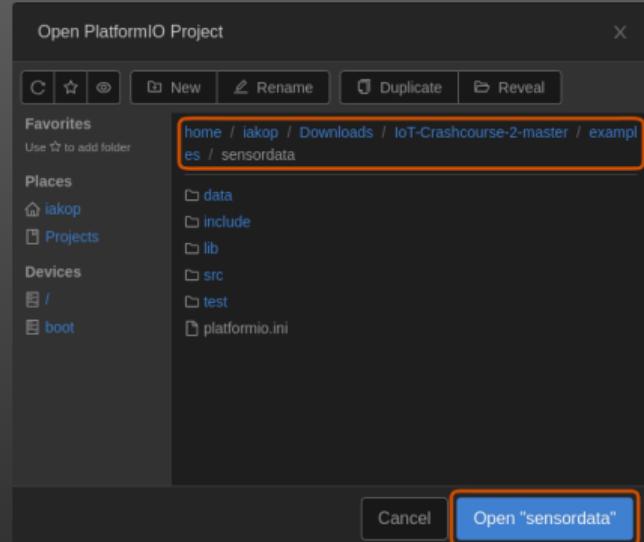


Figur 16: [Home](#) fanen i PlatformIO, med markeret indstilling for at indlæse et projekt fra disk

- 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 workshoppens materialer her:
 - <https://github.com/iakop/IoT-Crashcourse-2/archive/refs/heads/master.zip>
- Pak dem ud hvor de kan findes
- Under [Home](#) fanen, klik [Open Project](#)

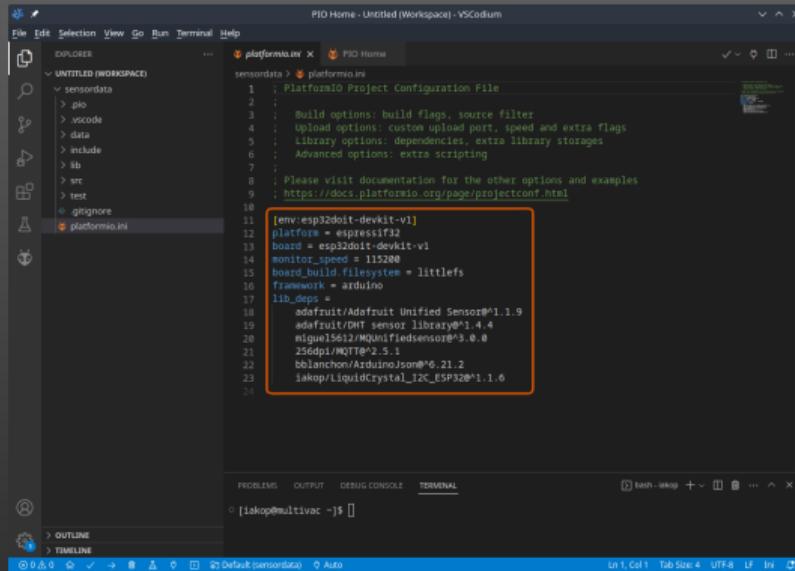
Indlæsning af ESP32 Projekt i PlatformIO

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



Figur 17: 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-2-master/examples/sensordata](#)

Indlæsning af ESP32 Projekt i PlatformIO



```
File Edit Selection View Go Run Terminal Help
EXPLORER UNTITLED (WORKSPACE) platformio.ini PIO Home
sensordata > platformio.ini
1 ; PlatformIO Project Configuration File
2 ;
3 ; Build options: build flags, source filter
4 ; Upload options: custom upload port, speed and extra flags
5 ; Library options: dependencies, extra library storages
6 ; Advanced options: extra scripting
7
8 ; Please visit documentation for the other options and examples
9 ; https://docs.platformio.org/page/projectconf.html
10
11 [env:esp32doit-devkit-v1]
12 platform = esp32v1
13 board = esp32doit-devkit-v1
14 monitor_speed = 115200
15 board_build.filesystem = littlefs
16 framework = arduino
17 lib_deps =
18     adafruit/Adafruit Unified Sensor@1.1.9
19     adafruit/DHT sensor library@1.4.4
20     miquel5612/WiQuiFiedsensor@3.0.0
21     256dpf/NOT@2.5.1
22     bblanchon/ArduinoJson@6.21.2
23     iakop/LiquidCrystal_I2C_ESP32@1.1.6
24
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

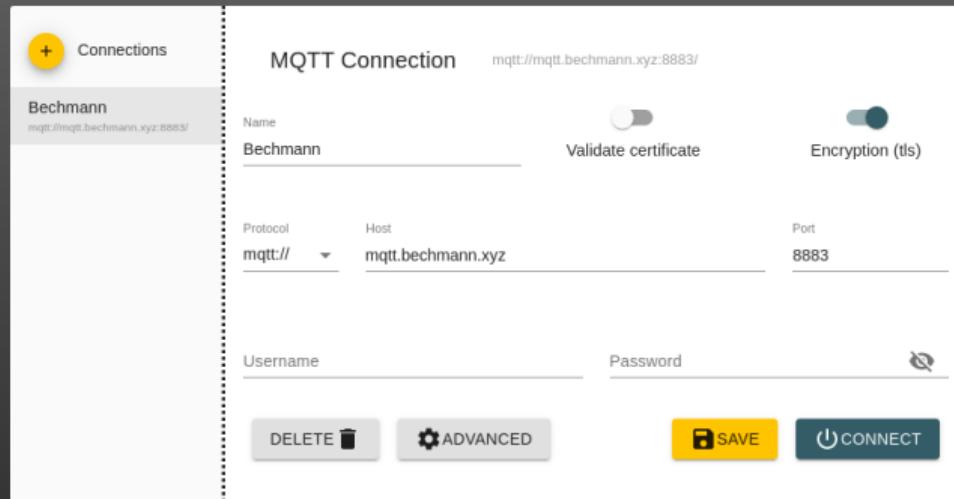
[iakop@multivac ~]\$

Default consoledata Auto

Figur 18: `platformio.ini` for `sensordata` projektet. Bemærk `monitor-speed`, `filesystem` og `lib_deps` er defineret

- 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

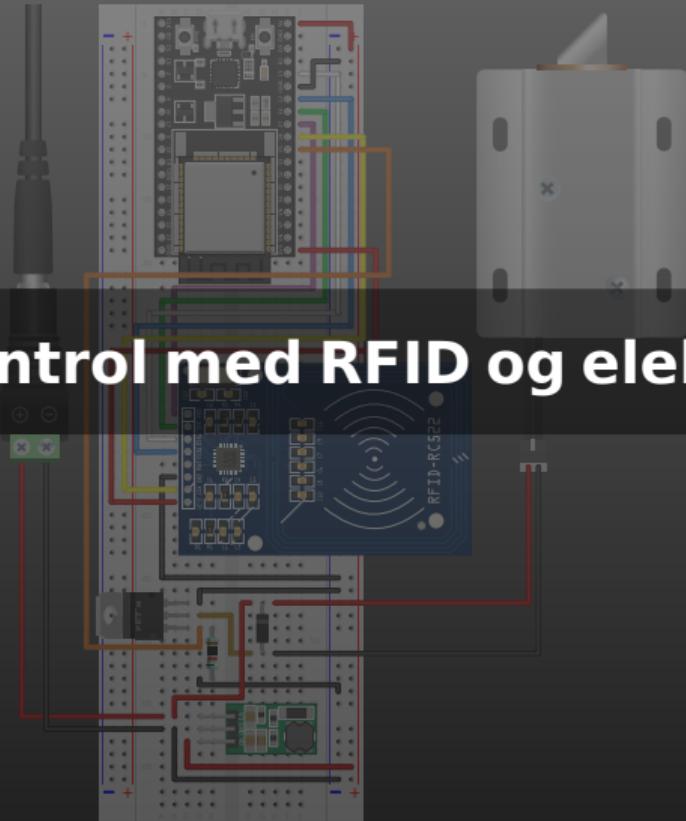
MQTT Client til dataovervågning



Figur 19: MQTT Explorer Connection dialogvindue, her med forbindelsen til `mqtt.bechmann.xyz` defineret

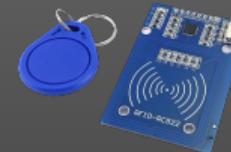
- MQTT Explorer kan bruges til at tjekke topics:
 - <http://mqtt-explorer.com/>
- Indstillinger for forbindelse til workshoppens **offentlige** server:
 - Name: `Bechmann` (Valgfri)
 - Validate certificate: **off**
 - Bug i MQTT Explorers cert storage forhindrer validering
 - Encryption (tls): **on**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

Adgangskontrol med RFID og elektronisk lås



Adgangskontrol med RFID og elektronisk lås

- Det andet eksempel handler om RFID kontrol og operation af lås
- Formålet er at kunne scanne RFID kort og administrere dem i et system
- Dette system bliver sat op på en hjemmeside
- Til dette vil der blive brugt:
 - MFRC522
 - RFID kortlæser
 - ISO/IEC 14443 A/MIFARE og NTAG kort
 - SARY DC12V
 - Elektronisk lås
 - 12V 0,8A
 - Normally closed
 - WebSocket
 - Fuld duplex protokol til hurtig samtidig kommunikation mellem klient og server



Figur 20:
MFRC522 RFID ko-
rtlæser

Figur 21: SARY DC12V
elektronisk lås

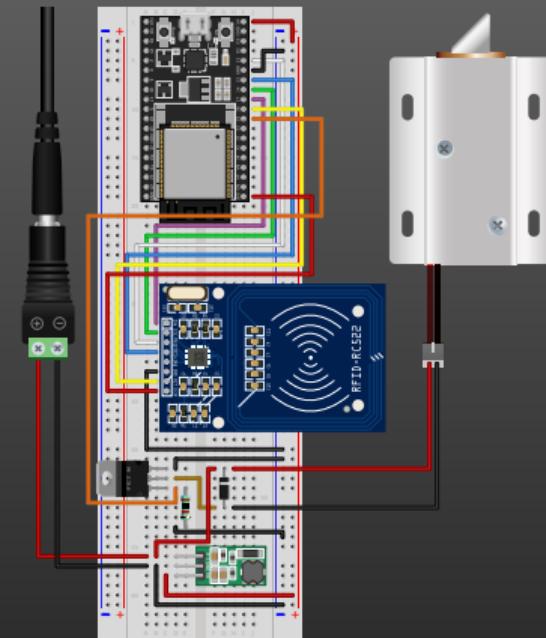


WebSocket

Figur 22: WebSocket logo
Kilde: [https://logodix.com/
logos/1825947](https://logodix.com/logos/1825947)
Licens: Non-Commercial

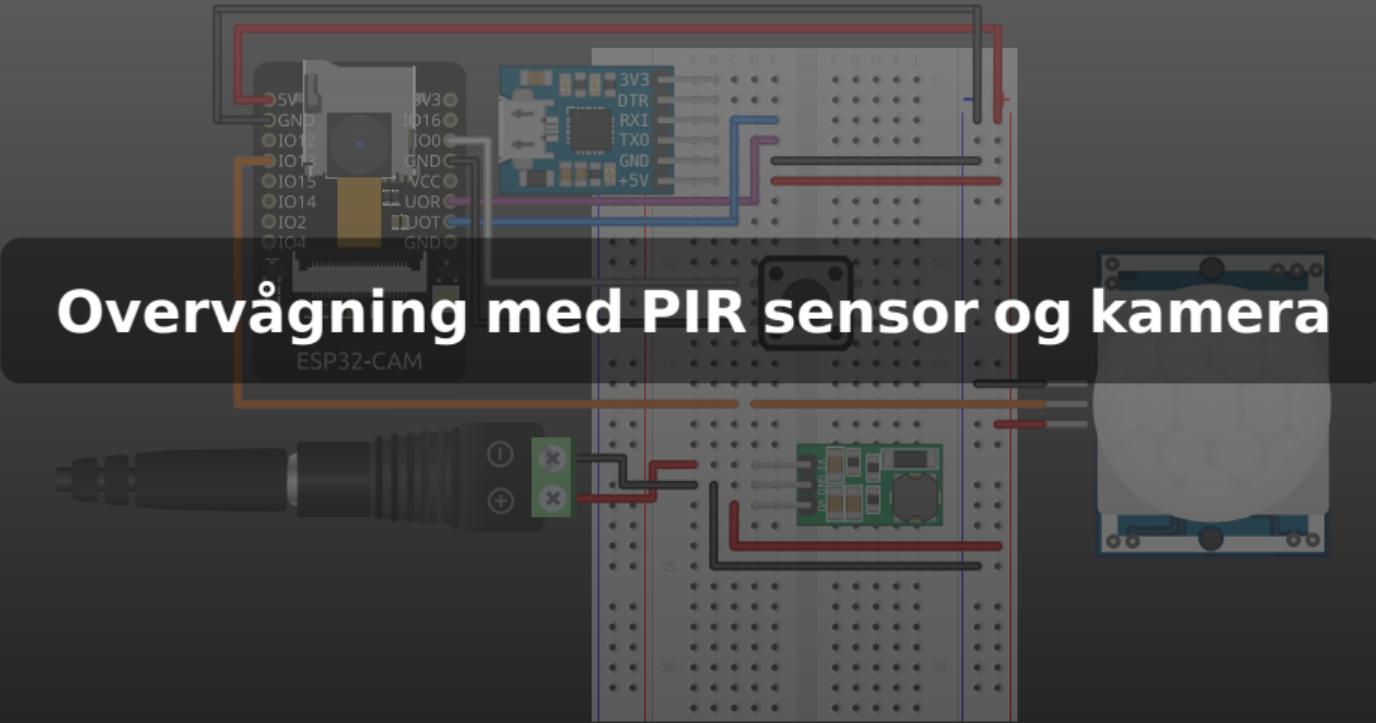
Adgangskontrol med RFID og elektronisk lås

- Breadboard opstillingen ser således ud
 - Et ESP32 board
 - Et MFRC522 modul
 - En N-type MOSFET
 - En flyback diode
 - En SARY 12V Lås
 - En 12V-5V DC-DC converter
 - En DC strømforsyning
 - Programmet gennemgår vi i fællesskab
 - Kildefiler kan hentes på:
 - <https://download-directory.github.io?url=https://github.com/iakop/IoT-Crashcourse-2/tree/master/examples/rfidlock>



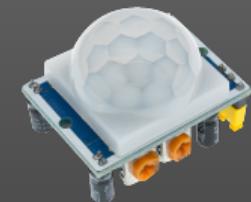
Figur 23: Breadboard setup med ESP32, MFRC522, MOSFET, SARY 12V, flyback diode, Buck converter

Overvågning med PIR sensor og kamera



Overvågning med PIR sensor og kamera

- Det tredje eksempel handler om automatisk overvågning
- Formålet er at ESP32-CAM skal tage billeder hver gang PIR sensoren opfanger bevægelse
- Den vil gemme de seneste billeder i filsystemet
- Dette system bliver sat op på en hjemmeside
- Til dette vil der blive brugt:
 - ESP32-CAM
 - ESP32 med kamera
 - Ekstra PSRAM til billeder
 - Kræver eksternt UART modul
 - HC-SR501
 - PIR sensor (Passive InfraRed)
 - Fanger bevægelse ud fra infrarød belysning
 - 3,3V IO
 - WebSocket
 - Fuld duplex protokol til hurtig samtidig kommunikation mellem klient og server



Figur 24:
ESP32-CAM

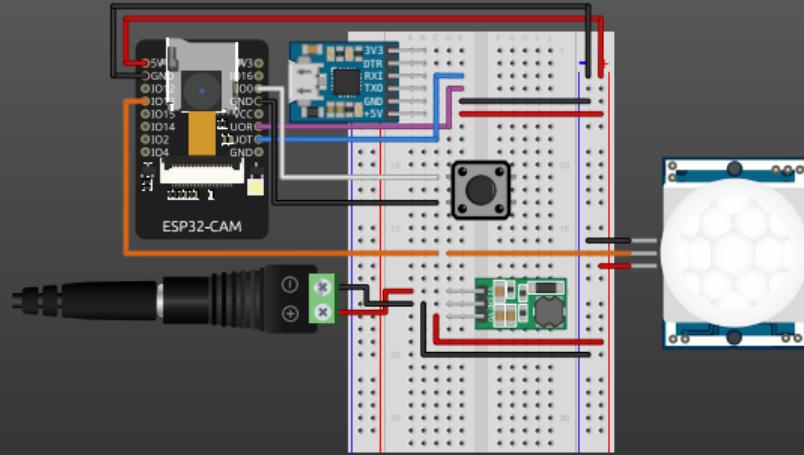
Figur 25: HC-SR501 PIR sensor



Figur 26: WebSocket logo
Kilde: <https://logodix.com/logos/1825947>
Licens: Non-Commercial

Overvågning med PIR sensor og kamera

- Breadboard opstillingen ser således ud
 - Et ESP32-CAM board
 - Et CP2101 UART modul
 - En trykknap til flashing
 - Et HC-SR501 modul
 - En 12V-5V DC-DC converter
 - En DC strømforsyning
- Programmet gennemgår vi i fællesskab
- Kildefiler kan hentes på:
 - <https://github.com/iakop/IoT-Crashcourse-2/tree/master/examples/motioncam>



Figur 27: Breadboard setup med ESP32-CAM, CP2101, tryknap, HC-SR501, og buck converter