

PM2

Workshop 1

Mbed Studio / Nucleo F446RE

M. Peter

E-Mail: pmic@zhaw.ch

Office : TE 309

C. Huber

E-Mail: hurc@zhaw.ch

Office : TE 673

Ablauf

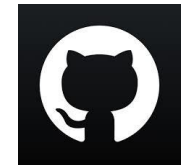
Workshop 1

- Kennenlernen von Arm Mbed und Nucleo F446RE
- Mbed Studio / Keil Studio Cloud
- Arm Mbed Code Repositories
- Herunterladen und Aufsetzen eines ersten Programms
- Inbetriebnahme einfacher Hardware
- IR-Sensor kalibrieren

Software / Accounts

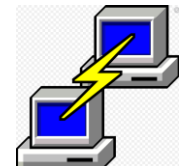
Arm Mbed

- Open Source Plattform und Betriebssystem für 32-Bit-ARM Cortex-M-Mikrocontroller
- Erstellen Sie sich einen Account mit Ihrer privaten E-Mail für Mbed und github:
 - <https://os.mbed.com/>
 - <https://github.com/>



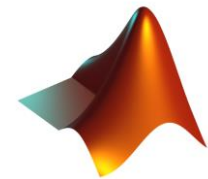
PuTTY (optional)

- Freeware SSH an telnet client: <https://www.putty.org/>
- Datenausgabe via serieller Schnittstelle und Daten loggen

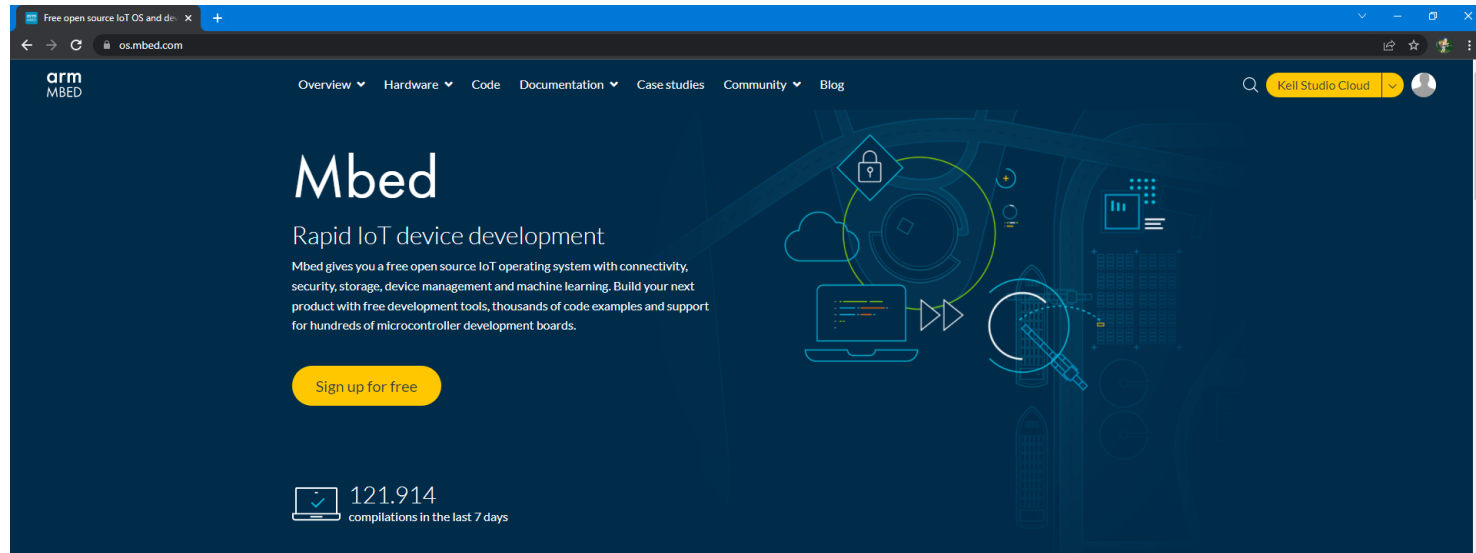


MATLAB

- Studenten Account: <https://ch.mathworks.com/>



Arm Mbed: <https://os.mbed.com>



What is Mbed?

Mbed makes device development quicker

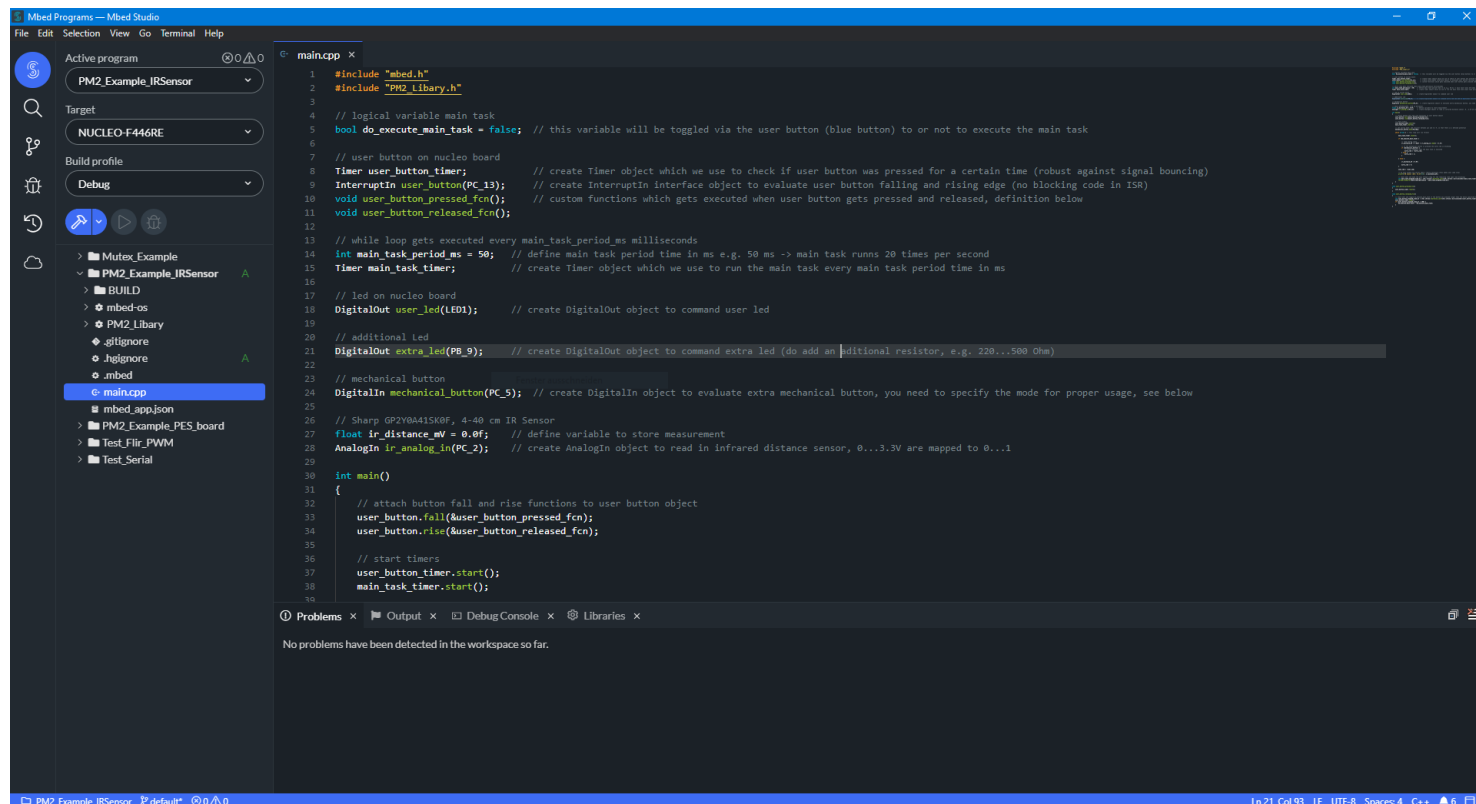


Operating system

Voll integrierte Lösung

- Mbed OS 6 Real-Time Operating System
- Mbed Studio

Mbed Studio

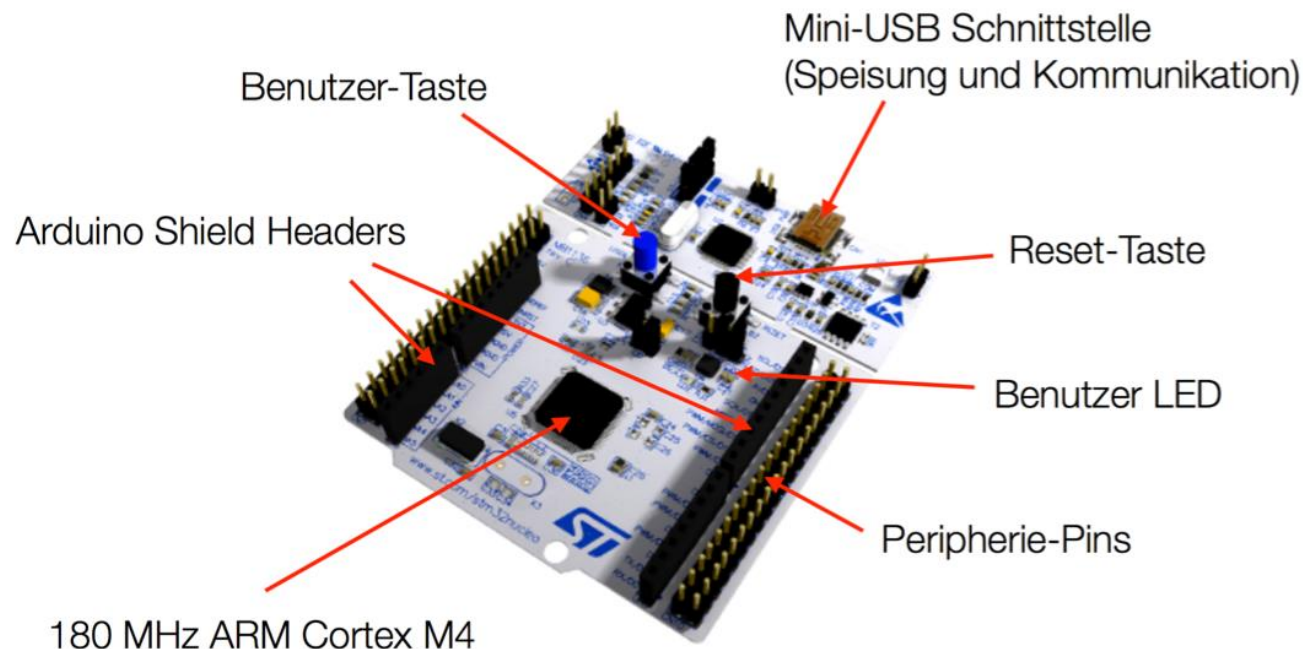


Mbed Studio is a free IDE for Mbed OS application and library development, including all the dependencies and tools you need in a single package so that you can create, compile and debug your Mbed programs on the desktop.

Mbed Version und Nucleo F446RE

Arm Mbed OS 6

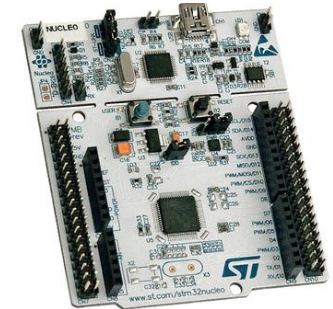
- Wir arbeiten ausschliesslich mit Mbed OS 6
- Es gibt auch ältere Versionen. Die Funktionalität und die Syntax sind versionsabhängig → sind Sie sich dessen bewusst, auch bei der Suche von öffentlichen Treibern und Beispielprogrammen



Hardware

Nucleo F446RE

- Specs, Interfaces und Pinout: <https://os.mbed.com/platforms/ST-Nucleo-F446RE/>
- Zusammen mit Mbed: Plug and Play



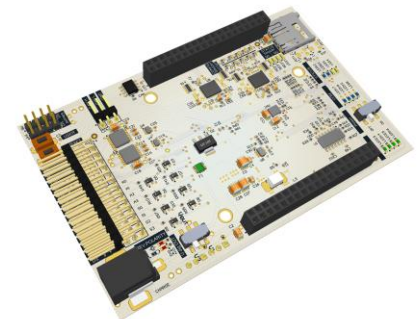
PES-Board

Hardwaretreiber

- 3 DC-Motoren (brushed)
- 4 Servos (besetzen die 4 DI/O falls verwendet)
- 4 DI/O, 3.3V (5V tolerant)
- 4 AI/O, 3.3V (5V tolerant)

Sensoren

- 3 Encoder-Counter (für Geschwindigkeitsregelung der DC-Motoren)



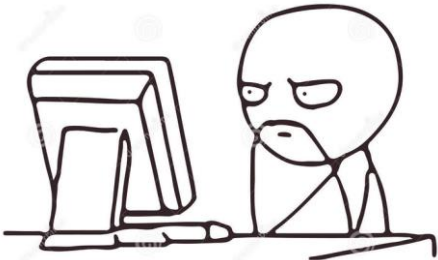
Sensoren

Sharp GP2Y0A41SK0F analog IR Distanz Sensor

- Nichtlineare Sensorkennlinie → muss kalibriert werden

Workflow Firmware Entwicklung

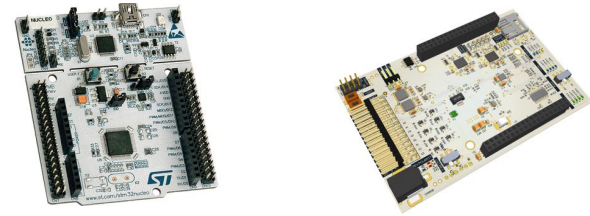
Devs



Arm Mbed /
Mbed Studio

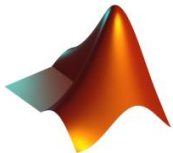


Nucleo F446RE & PES-Board

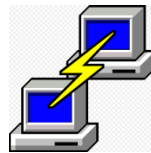


- Erstellen Sie ein Flussdiagramm
- Testen und Debuggen Sie Schritt für Schritt

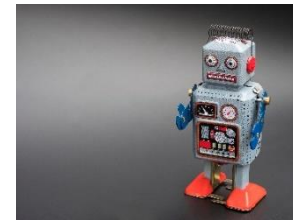
Auswertung Daten /
Algorithmen
entwickeln



Datenausgabe /
Debugging



Produkt



Einlesen analoger IR Distanz Sensor

Aufgabenstellung

1. Stecken Sie die nachfolgende Hardware ein und testen Sie das Beispielprogramm. Sie benötigen:
<https://os.mbed.com/platforms/ST-Nucleo-F446RE/> und [PES_board_peripherie.pdf](#)
 - Zusätzliche LEDs auf PB_9 gegenüber Ground direkt am Nucleo F446RE
 - Mechanischer Schalter PC_5 gegenüber Ground am PES-Board
2. Erweitern Sie das Beispielprogramm bei allen auskommentierten Stellen `//???` so dass Sie den analogen IR Sensor in eine Variable einlesen, skalieren und über die serielle Schnittstelle ausgeben können. Sie benötigen:
[GP2Y0A41SK0F_datasheet.pdf](#)
 - Erzeugen des AnalogIn Objekt: `AnalogIn ir_analog_in(PC_2);`
 - Einlesen des Spannungssignals: `ir_distance_mV = 1.0e3f * ir_analog_in.read() * 3.3f;`
 - **Anmerkung:** AnalogIn mappen (0-3.3 V) → (0-1), AnalogOut mappen (0-1) → (0-3.3 V)
 - Stecken Sie den IR Sensor an PC_2 direkt am PES-Board ein
3. Messen Sie die Sensorkennlinie und stellen Sie diese in MATLAB dar. Als Beispiel steht Ihnen [eval_ir_sensor.m](#) zur Verfügung. Darin ist bereits Beispielcode um die inverse Kennlinie über einen analytischen Funktionsansatz abzubilden. Die Koeffizienten werden durch einen numerischen Abgleich / Fit bestimmt (optimiert).
4. Implementieren Sie die gefundene inverse Funktion (float), so dass Sie direkt den Abstand in cm ausgeben.

Das Beispielprogramm finden Sie unter dem branch `workshop_1` in Ihrem Fork auf github.