

HAPPY PLANTY

PC-INTERFACETECHNIK

DI CHRISTIAN HARTINGER

FH CAMPUS 02 Automatisierungstechnik

Ernad Zulum

Severin Gruber

Florian Strauß

Ivan Pozgaj

Vorstellung – Thema (Problemstellung)

Aufbau :

→ *Hardware*

→ *Software*

→ *Prozess*

Funktion

AGENDA

VORSTELLUNG (THEMA)



#?\$&!



OIDA!?

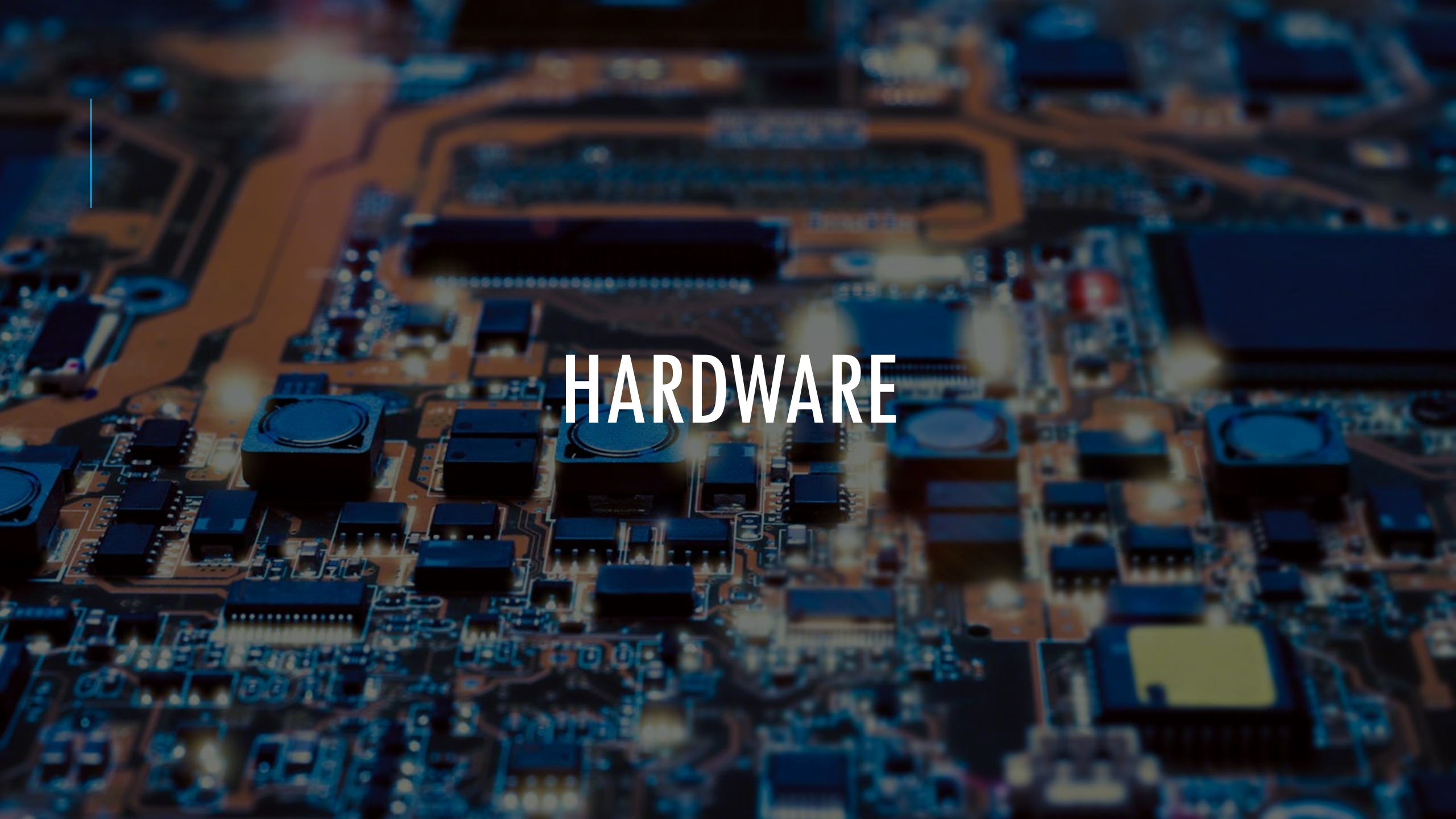


HAPPY PLANTY ©

DAS TEAM



AUFBAU



HARDWARE



MIKROCONTROLLER

WEMOS Lolin 32 ESP32 v1.0.0

Microcontroller	ESP-32
Betriebsspannung	3.3V
Eingangsspannung (USB)	5V
Digital E/A Pins	26
Analog Eingangs Pins	12(Max input: 3.2V)
Flash Memory	4M bytes
Clock Speed	240MHz
CPU	32-bit
Länge	58mm
Breite	24.5mm
Gewicht	5.8g
Eigenschaft	Bluetooth, 10 Touch Sensor Pins, Batterie Anschluß
Batterie Anschluss	PH-2 2.0mm
Stromverbrauch Normalbetrieb	45,4 mA
Stromverbrauch Schlafmodus	1,28 mA

FEUCHTIGKEITSSENSOR

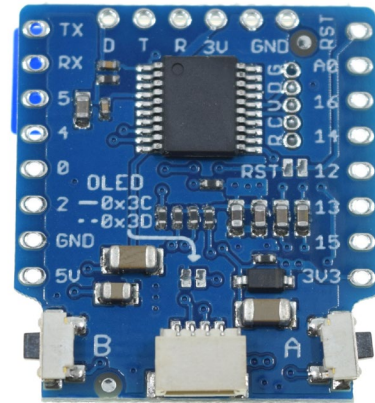
Capacitive Soil Moisture Sensor v1.2



Technical specifications

Dimensions	22 x 97 x 9mm with a 3-cable cable of 190mm
Output voltage	Analogue
Sensor	Capacitive
Operating voltage	5V DC
Interface	PH2.0-3P

OLED SHIELD



WEMOS D1 Mini Shield, OLED 0.66 Shield

Features

- Screen Size: 64x48 pixels (0.66" Across)
- Operating Voltage: 3.3V
- Driver IC: SSD1306 (I2C Address: 0x3C or 0x3D)
- 2x I2C Button (customizable I2C Address, default:0x31)

Pins

D1 mini	GPIO	Shield
D1	5	SCL
D2	4	SDA

AKKU



Mechanical Characteristics

Cell	▶ LP5551106
PCM	▶ Yes
NTC	▶ 10K Ω 1% +25C B3435
Configuration	▶ 1S1P
Weight	▶ appr. 74g
Connector	▶ JST ADHR-03V-H

Electrical Specification

Rated Capacity	▶ 3700mAh min, 3700mAh typ.
Nominal Voltage	▶ 3.7V
Wat-Hour Rating	▶ 13.69Wh
Max. Operating Voltage Range	▶ 2.75V to 4.20V
Max. Charge Voltage	▶ 4.20V $\pm$ 50mV
Max. Charge Current	▶ 2000mA
Standard Charge Current	▶ 1000mA
Max. Discharge Current	▶ 3000mA
Standard Discharge Current	▶ 1500mA
Discharge Cut Off	▶ 2.75V
Internal Impedance	▶ <200m Ω
Expected Cycle Life @ (0.5C/0.5C) @ 23 $\pm$ 5°C	▶ 500 cycles $\geq$ 80%

Cell Protection

Overcharge Detection	▶ 4.275 $\pm$ 50mV (0.7 to 1.3sec. delay, release 4.075V $\pm$ 50mV)
Overdischarge Detection	▶ 2.75V $\pm$ 50mV (14 to 26msec. delay, resume 3.05V $\pm$ 50mV)
Overcurrent Detection	▶ 4A to 4.5A (8 to 16msec. delay)

Ambient Conditions

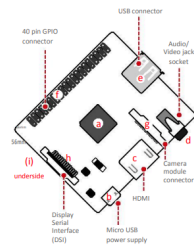
Charge Temp. Range	▶ +10°C to +45°C
Discharge Temp. Range	▶ -20°C to +60°C
Storage Temp. Range	▶ 1 year at -20°C to +30°C >70%
	▶ 3 months at -20°C to +45°C >70%
	▶ 1 month at -20°C to +60°C >70%

Environmental and Safety

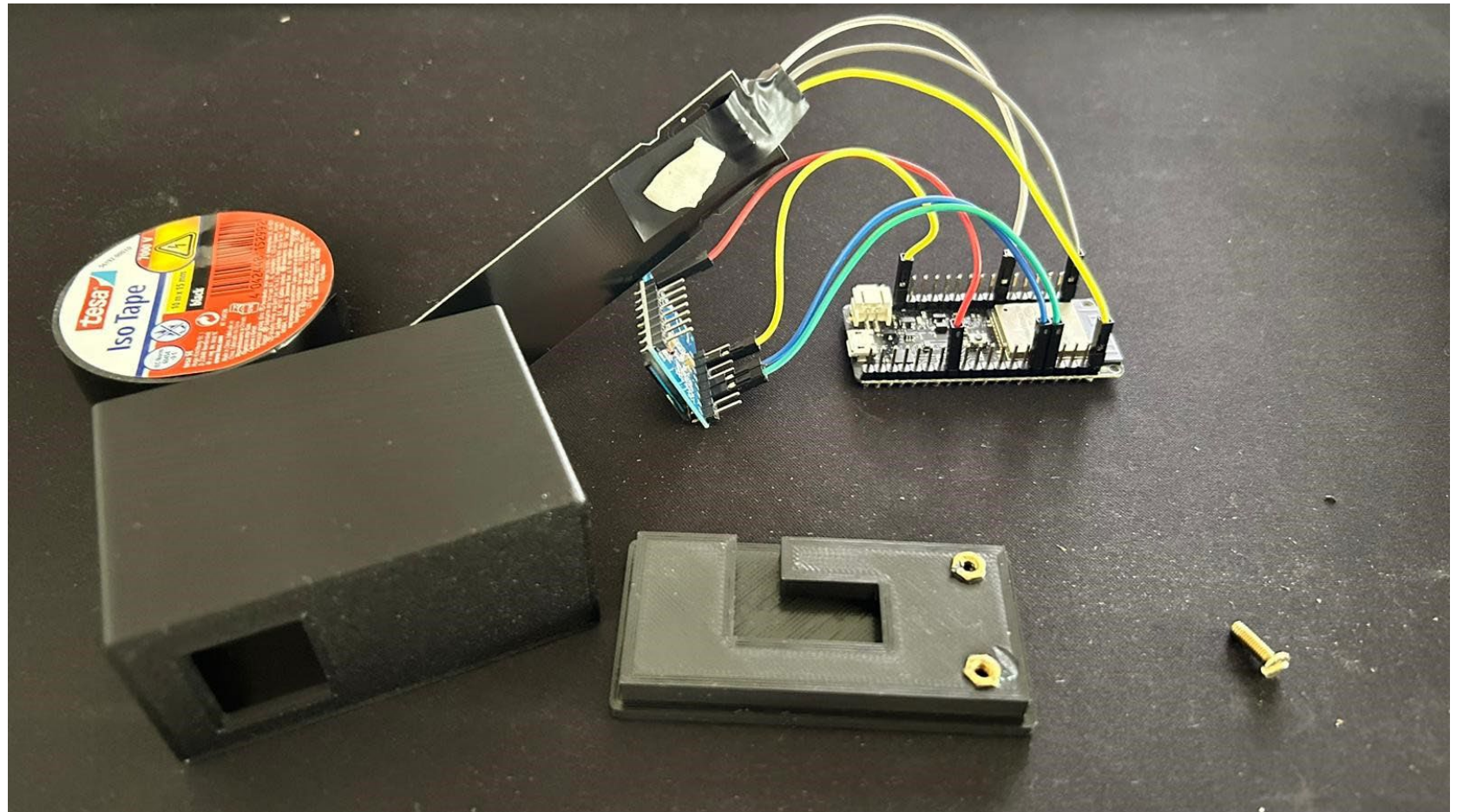
Please follow LiPol Handling and Safety Precautions for Lithium Polymer Battery. This battery meets the requirements of Battery Directives, and the battery parts are IEC62133-2:2017 & RoHS-Compliant. For more safety precautions and performance standards, please go to www.lipolbattery.com/support.html

RASPBERRY PI

Product Name	Raspberry Pi Model A+
Product Description	The Raspberry Pi model A+ features lower power consumption, better audio performance and a 40-pin GPIO connector in an even smaller package.
RS Part Number	833-2699
Specifications	
Chip	Broadcom BCM2835 (a)
Core architecture	ARM11
CPU	700 MHz Low Power ARM1176JZF5 Applications Processor
GPU	Dual Core VideoCore IV™ Multimedia Co-Processor Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA Infrastructure
Memory	512MB SDRAM
Operating System	Boots from micro SD card, running a version of the Linux operating system
Dimensions	66 x 56 x 14mm
Power	Micro USB socket 5V, 2A (b)
Connectors:	
Digital AV Output	HDMI (rev 1.3 & 1.4) (c)
Analogue AV Output	3.5mm jack (d), Stereo audio, Composite video (PAL, NTSC)
USB	USB 2.0 Connector (e)
GPIO Connector	40-pin 0.1in header compatible with Model A/B 26-pin add-on boards (f)
Camera Connector	15-pin MIPI Camera Serial Interface (CSI-2) (g)
Display Connector	15-pin Display Serial Interface (DSI) (h)
Memory Card Slot	Micro SD (i)



ZUSAMMENBAU



SOFTWARE

```
from datetime import datetime
import calendar
import math
import pytz
import io, base64

class AdmissionExtensionOnlineController(http.Controller):

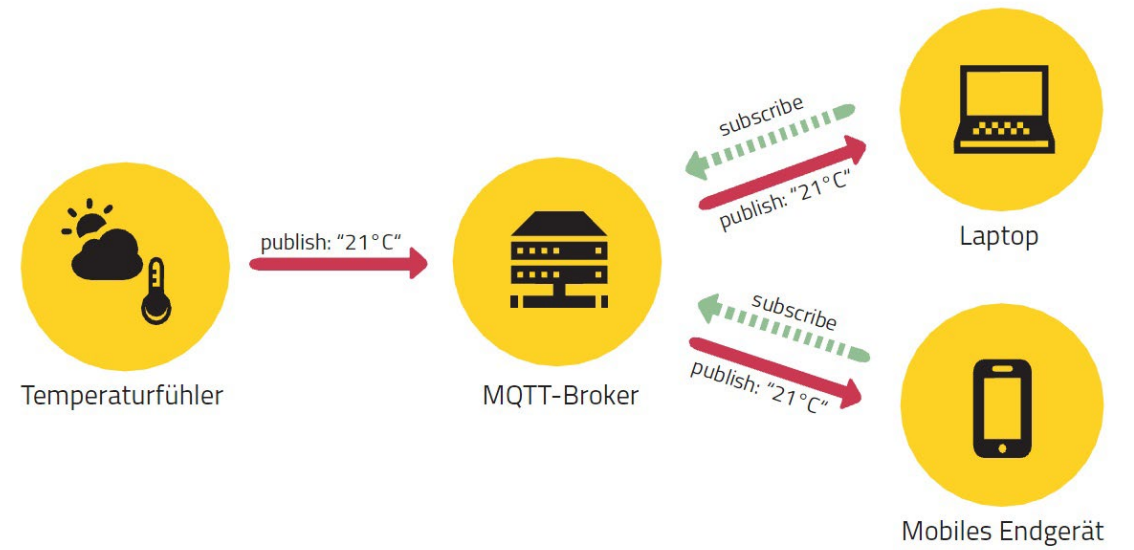
    @http.route('/get/type_wise_program', website=True, auth='none')
    def type_wise_program(self, **kwargs):
        if len(kwargs['types']) <= 0:
            return "None"

        types = kwargs['types']
        program_list = []
        domain = []

        if types == 'local_bachelor_program_hsc':
            domain = [('course_id.is_local_bachelor_program_hsc', '=')]
        elif types == 'local_bachelor_program_a_level':
            domain = [('course_id.is_local_bachelor_program_a_level', '=')]
        elif types == 'local_bachelor_program_diploma':
            domain = [('course_id.is_local_bachelor_program_diploma', '=')]
        elif types == 'local_masters_program_bachelor':
            domain = [('course_id.is_local_masters_program_bachelor', '=')]
        elif types == 'international_bachelor_program':
            domain = [('course_id.is_international_bachelor_program', '=')]
        elif types == 'international_masters_program':
            domain = [('course_id.is_international_masters_program', '=')]
        domain = []
```

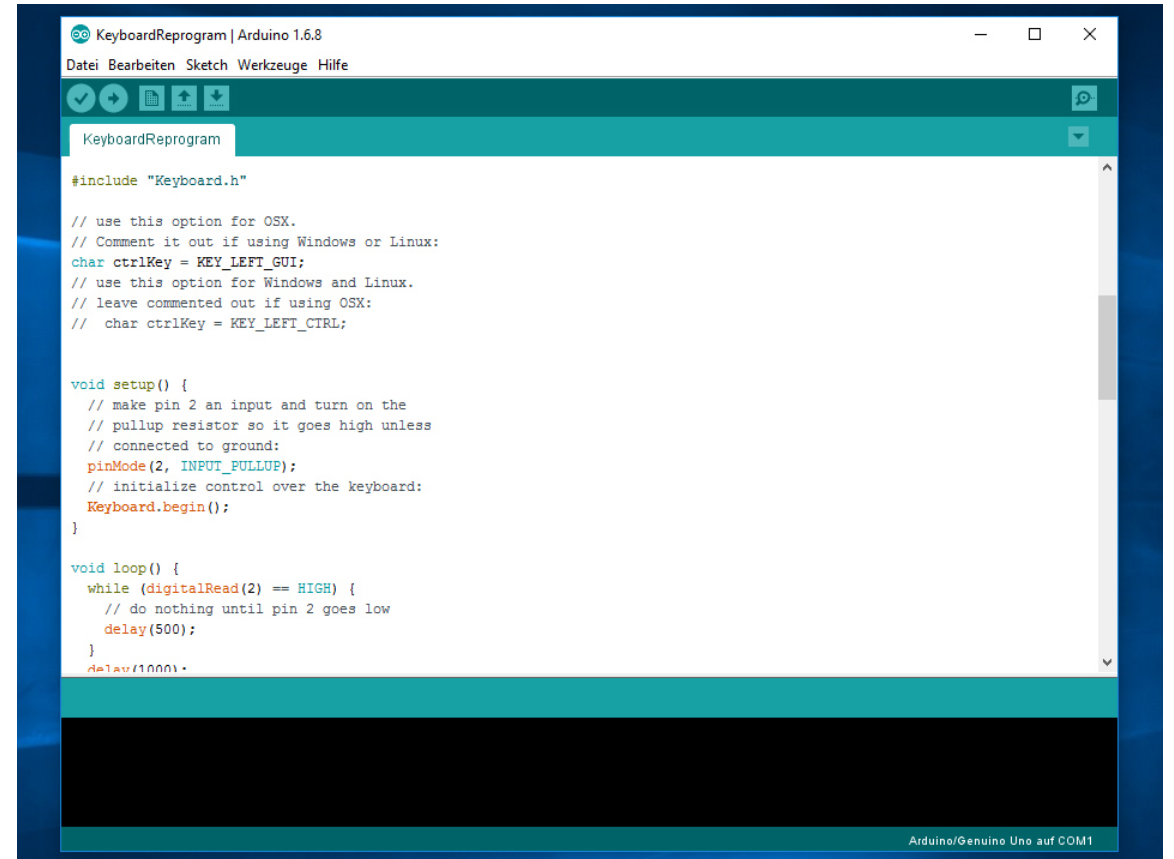
KOMMUNIKATIONSprotokoll

MQTT



PROGRAMIERSPRACHE

ARDUINO IDE



The screenshot shows the Arduino IDE interface with a sketch named "KeyboardReprogram" open. The code is written in C++ and includes the `Keyboard.h` library. It configures pin 2 as an input with a pullup resistor and initializes the `Keyboard` library. The `loop()` function checks if pin 2 is high and delays 500ms, then delays 1000ms at the end of each loop iteration.

```
KeyboardReprogram | Arduino 1.6.8
Datei Bearbeiten Sketch Werkzeuge Hilfe

KeyboardReprogram

#include "Keyboard.h"

// use this option for OSX.
// Comment it out if using Windows or Linux:
char ctrlKey = KEY_LEFT_GUI;
// use this option for Windows and Linux.
// leave commented out if using OSX:
// char ctrlKey = KEY_LEFT_CTRL;

void setup() {
  // make pin 2 an input and turn on the
  // pullup resistor so it goes high unless
  // connected to ground:
  pinMode(2, INPUT_PULLUP);
  // initialize control over the keyboard:
  Keyboard.begin();
}

void loop() {
  while (digitalRead(2) == HIGH) {
    // do nothing until pin 2 goes low
    delay(500);
  }
  delay(1000);
}
```

Arduino/Genuino Uno auf COM1

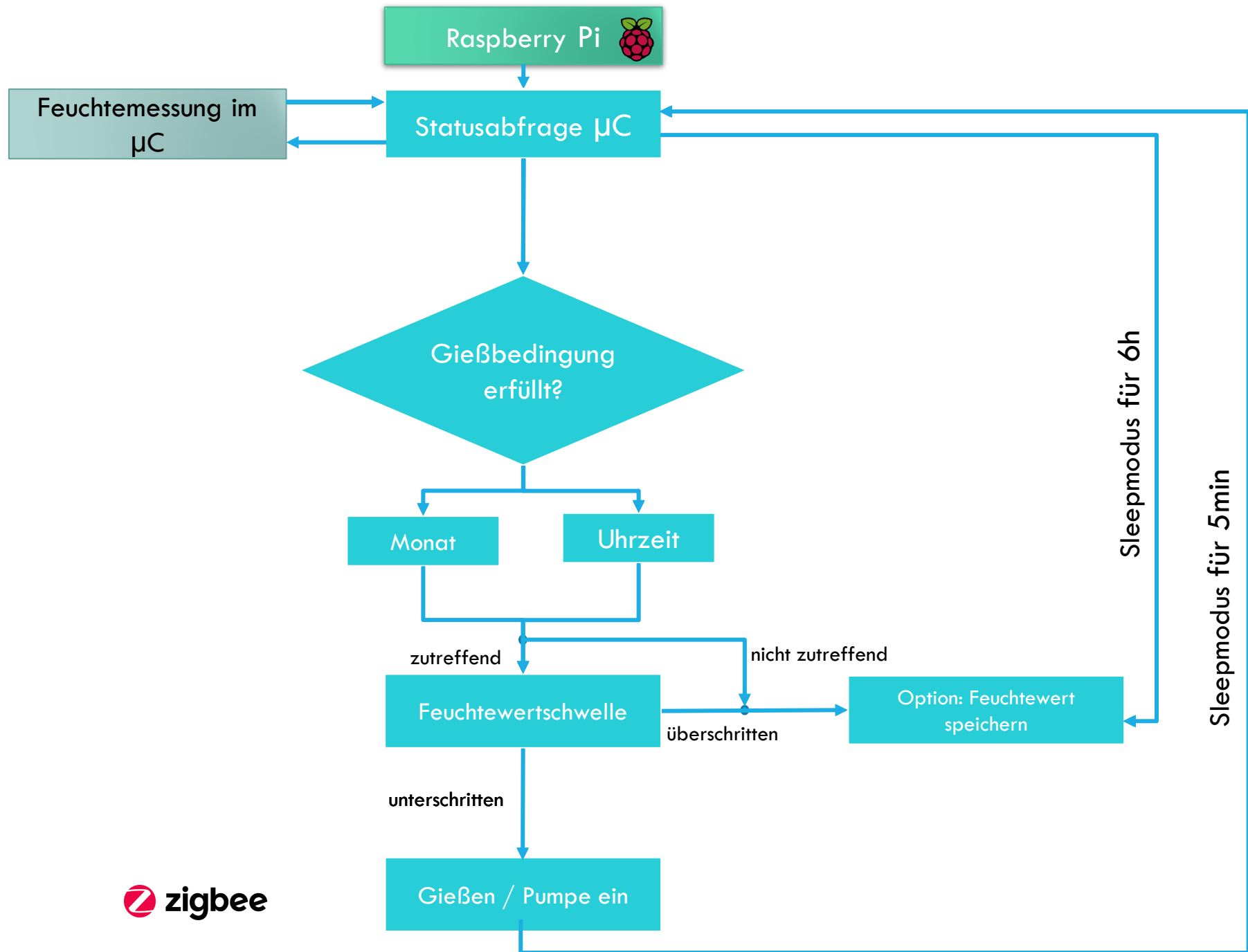
PROGRAMMIERSPRACHE

PYTHON



The background features a blurred industrial setting with a person in the foreground. Overlaid on this are several semi-transparent gears of different sizes. Each gear contains a white icon: a globe, a robotic arm, a bar chart, a line graph, a circuit board, and a set of three horizontal bars. The word "PROZESS" is centered in white, bold, sans-serif capital letters.

PROZESS





FUNKTION

```
225     sleeptime = 300;
226     delay(5000);
227
228 }
229
230 if (sleeptime > 0)
231 {
```

Output Serial Monitor ✕

Message (Enter to send message to 'WEMOS LOLIN32' on 'COM3')

```
-----
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0030,len:1184
load:0x40078000,len:13160
load:0x40080400,len:3036
entry 0x400805e4
```

Connecting to A1-3B8989

...

WiFi connected

IP address:

10.0.0.1

9%

Publish message: 9



A small, tree-like character (Baby Groot) stands in the center of the frame. He has a brown, textured body and a small head with two large, dark eyes. He is waving his right hand. The background is dark and blurry, showing some industrial structures and lights. The text "DANKE FÜR IHRE ZEIT!" is overlaid in the center in a bold, white, sans-serif font.

DANKE FÜR IHRE ZEIT!