Einstieg in Physical Computing

Wir verwenden folgenden Microcontroller mit einem ESP32-Chip.

Posten 0 - Thonny

Installieren Sie Thonny (für Windows und für Mac. Öffnen Sie Thonny und verbinden Sie den Microcontroller per USB-Kabel mit ihrem Notebook. Drücken Sie den «Stop/Start»-Button.

Posten 1 - Hello World!

- Verbinden Sie den ESP32 mit USB-C-Kabel mit ihrem Notebook. Die weisse LED auf dem Mikrocontroller sollte nun dauerhaft leuchten.
- Öffnen Sie Thonny und wählen Sie unter `Tools → Optionen… → Interpreter` den Interpreter `MicroPython (ESP32)` aus.
- Klicken Sie auf `Stop`, nun sollte ihm Shell-Fenster von Thonny `MicroPython v1.22.0…` stehen und >>> erscheinen. Diese Zeichen bedeuten, dass MicroPython-Befehle eingegeben werden können, welche anschliessend an den ESP32 gesendet und auf diesem ausgeführt werden.
- Geben Sie zum Testen die Rechnung 34**4 ein, der ESP32 berechnet nun 34^4. (Befehle müssen mit Enter bestätigt/ausgeführt werden.)
- Probieren Sie noch andere Rechnungen aus.
- Was passiert, wenn Sie 12345^{98765} ausrechnen möchten?
- Sie können auch Text vom ESP32 ausgeben lassen: z.B. mit `print('Hello World!')`.
- Schreiben Sie eine `for`-Schleife, die alle Dreieckszahlen bis n = 15 zusammenzählt.

Posten 2 - LED

- Bauen Sie folgenden Schaltkreis nach.

- Um die LED zu verwenden müssen Sie zuerst die nötigen Module mit dem Befehl `from machine import Pin` laden.
- Die LED ist am Pin 3 angeschlossen. Ordnen Sie die LED diesem Pin in MicroPython zu: `led = Pin(3, Pin.OUT)`
- Die LED können Sie nun mit dem Befehl `led(1)` ein- bzw. mit `led(0)` ausschalten.
- Mit einer for-Schleife kann die LED zum Blinken gebracht werden. Dazu wird noch das time Modul benötigt:

```
import time
led = Pin(2, Pin.OUT)
for i in range(10):
```

```
    led(1)
    time.sleep(0.5)
    led(0)
    time.sleep(0.5)
```

*-Aufgaben:

1. Weshalb kann die LED nicht direkt also ohne Widerstand mit dem Microcontroller verbunden werden?

Posten 3 - Druckknopf

Es kann auch ein Druckknopf verwendet werden, um die LED zu steuern. Ergänzen Sie dazu den Schaltkreis der LED um den Druckknopf. Stecken Sie dazu die LED an Pin 3 (und nicht mehr Pin 2). Stecken Sie den Druckknopf wie folgt auf das Breadboard und verbinden Sie das untere/rote Kabel mit GND und das obere/grüne Kabel mit Pin 2.

```
from machine import Pin
led = Pin(3, Pin.OUT)
button = Pin(2, Pin.IN, Pin.PULL_UP)
while True:
    if not button():
        led(not led())
        while not button():
            pass
```

Hinweis: Um eine while-Schleife (oder auch andere Befehle) zu unterbrechen, drücken Sie die Tastenkombination CTRL+C.

*-Aufgaben:

1. Programmieren Sie den Druckknopf so, dass bei zweimaligen Drücken, die LED zu blinken beginnt.

Posten 4 - Ampel

Bauen Sie folgende Ampelschaltung nach.

(schwarzes Kabel → GND, grünes Kabel → Pin 6, gelbes Kabel → Pin 4, rotes Kabel → Pin 3, violettes Kabel → Pin 2)

Programmieren Sie die Schaltung so, dass die Ampel von rot über orange auf grün schaltet, sobald der Knopf gedrückt wird. Die Ampel soll dann für 2 Sekunden grün sein und danach automatisch wieder auf Rot schalten.

Code-Hilfe:

```
from machine import Pin
import time
led1 = Pin(3, Pin.OUT)
led2 = Pin(4, Pin.OUT)
led3 = Pin(6, Pin.OUT)
button = Pin(2, Pin.IN, Pin.PULL_UP)
led3(1)
while True:
    if not button.value():
        led3(1)
        time.sleep(0.1)
        ...
```

Posten 5 - NeoPixel

Auf dem Mikrocontroller ist auch eine dreifarbige LED (NeoPixel) verbaut. Diese kann über den Pin 7 angesteuert werden (keine Verkabelung nötig).

Laden Sie die NeoPixel-LED mit

```
import machine, neopixel
np = neopixel.NeoPixel(machine.Pin(7),1)
```

Die Farbe kann mit RGB-Farbcode gesetzt werden. Bevor die NeoPixel leuchtet muss der Farbcode gesetzt werden: z.B. rot mit `np[0]=(255,0,0)`. Danach kann mit `np.write()` die LED eingeschaltet werden.

Schreiben Sie einen Code für einen Farbverlauf der NeoPixel-LED. Die RGB-Farbcodes finden Sie z.B. hier.

Posten 6 - Input, Schleifen, Bedingungen

Wie beim normalen Python können Sie auch mit MicroPython `for`- oder `while`-Schleifen programmieren. Und auch die Bedingungen `if` ... `else:` können verwendet werden.

1. Schreiben Sie einen Code, der einen Input verlangt (mit dem Befehl wird die Eingabe in der Variable x gespeichert: `x = input(„Eingabe: “)`) und falls dieser $Licht an!$ ist, soll die LED leuchten bzw. eingeschaltet werden.
2. Erweitern Sie ihren Code mit einer Schleife, der einen Countdown von 10s bis zum Einschalten der LED ausgibt.