

Anleitung für ein Praktikum mit dem
HRWizzard Development Board
(Raspberry Pi Pico) und dem
Temperatursensor HTS221 (über I²C)

Praktikums- beispiel HRWizzard

Temperatursensor HTS221

Nikolaj Pauly
Céline Gutt
Shuhrat Okhunov

Inhaltsverzeichnis

Einleitung.....	2
Ziel des Praktikums.....	2
Grundkenntnisse	2
Anleitung	2
Erweiterungen	3

Einleitung

Im Bereich der Mikrocontroller ist die Temperaturerfassung oft eines der ersten Beispiele. Hier soll der Mikrocontroller [Raspberry Pi Pico](#) verwendet werden, welcher auf dem HRWizzard Development Board zur Verfügung steht. Auf diesem Board gibt es einen [I²C](#) Anschluss, an welchem eine Sensorplatine angeschlossen werden kann. Diese Sensorplatine dient als Adapter Platine für eine Vielzahl von Sensorplatinen der Firma Adafruit von der Bauform [STEMMA QT](#). Der hier verwendete Temperatursensor ist der [HTS221](#) von ST.

Als Aktoren könnten folgende Elemente verwendet werden:

- Die LED-Platine des HRWizzard mit den [WS2812](#) LEDs
- Die Debugging Ausgabe

Das Besondere am Raspberry Pi Pico ist, dass dieser mit der Programmiersprache Micro Python programmiert werden kann. Im Gegensatz zur Programmiersprache C/C++ (wie z.B. bei den Arduino Mikrocontrollern) ist die reduzierte Semantik und vereinfachte Syntax, da es sich bei Python um eine Skriptsprache handelt. Dies reduziert die Einstiegshürde erheblich und macht den Code leichter verständlich.

Ziel des Praktikums

Im Fokus dieses Praktikums steht das Bussystem I²C. Für den Temperatursensor ist bereits ein Python Code vorhanden, welcher dazu verwendet werden kann, von diesem die Temperatur (bzw. die Luftfeuchtigkeit) über den I²C Bus auszulesen.

Wenn dies erstmal funktioniert, steht es dem Praktikanten frei daraus eine sinnvolle Ausgabe für die LED-Platine zu kreieren. Für diese LED-Platine steht ebenfalls ein Python-Code zur Verfügung, welcher als Grundlage verwendet werden kann.

Grundkenntnisse

Empfohlen werden folgende Grundkenntnisse:

- Erstellen von Funktionen in Python
- Einbinden von externen Python Dateien
- Grundlagen I²C Bus
- Einfache Programmstrukturen, wie:
 - o If/elif/else
 - o For/while Schleifen
 - o Variablen und deren Manipulationen

Anleitung

1. Verbinden Sie die Hauptplatine, die Sensorplatine sowie die LED-Platine zusammen. Verwenden Sie dafür die Vierfachsteckverbinder. Achten Sie darauf, dass Sie die richtigen Buchsen für die jeweiligen Platinen verwenden (auf der Hauptplatine die Beschriftung beachten „LED“ und „I²C“).
2. Verbinden Sie die Hauptplatine mit dem PC. Verwenden Sie dafür ein Micro-USB Kabel. Nun sollte die „PWRon“ LED leuchten.
3. Starten Sie die IDE Thonny und öffnen Sie die folgenden Dateien:
 - a. HRWizzardLED.py
 - b. HRWizzardI2C.py
 - c. HRWizzardSensorHTS221.py

4. Erstellen Sie eine neue Datei: main.py
5. Speichern Sie die Dateien auf den Raspberry Pi Pico, indem Sie unter Datei → Speichern unter als Speicherort den Pico auswählen

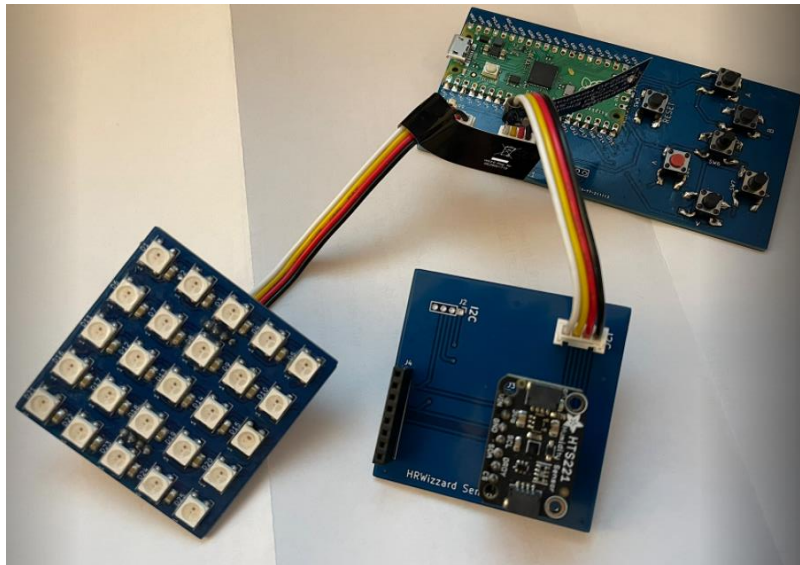


Abbildung 1: Zusammengesteckte Platinen

Die main.py könnte wie folgt aussehen:

```
import HRWizzardButtons
import HRWizzardLED
import HRWizzardI2C
import HRWizzardSensorHTS221
import machine, time

HRWizzardButtons.initButtons()

print("Welcome to HRWizzard!")

if HRWizzardSensorHTS221.checkHTS221isPresent() == True:
    HRWizzardSensorHTS221.initHTS221()
    #HRWizzardSensorHTS221.getTempFromHTS221()

    while True:
        #HRWizzardLED.testLED()
        temp = HRWizzardSensorHTS221.getTempFromHTS221()

        HRWizzardLED.setColorToAllLED(
            HRWizzardSensorHTS221.switchTempColor(temp))
        time.sleep_ms(100)
```

Erweiterungen

- Es könnte eine zweite LED- Platine verwendet werden, um zweistellige Zahlen (also die Temperatur) darstellen zu können.
- Es könnte bei der LED-Platine, die Hälfte der LED-Fläche für die Luftfeuchtigkeit verwendet werden.