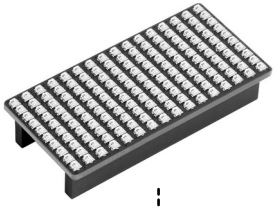
HARD- UND SOFTWARESETUP

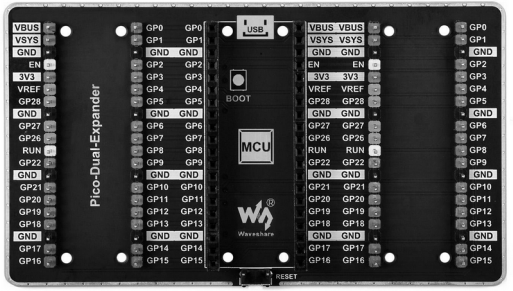
Der **Waveshare Dual GPIO Expander** (SKU 19343) wird um das **Waveshare 1.14 Pico LCD Display** (SKU 19340), die **Waveshare 16x10 LED Matrix** (SKU 20170) und einen **Raspberry Pi Pico / Raspberry Pi Pico 2** (jeweils mit Headern) erweitert. Die Ausrichtung der Komponenten erfolgt über den Micro-USB Anschluss am **Raspberry Pi Pico / Raspberry Pi Pico 2**. Zu diesem Zweck weisen alle Komponenten einschlägige Markierungen auf. Die Energieversorgung erfolgt über die **ISY Powerbank** (IPP-5000-CBK) und dem beigefügten **USB-A auf Micro-USB Kabel**, welches direkt an den **Raspberry Pi Pico / Raspberry Pi Pico 2** angeschlossen wird. Die **Software** (main.py) zur ersten Inbetriebnahme steht über den QR-Code, der gleichermaßen als Link verwendet werden kann, zur Verfügung und kann schließlich mit **Thonny** auf den **Raspberry Pi Pico / Raspberry Pi Pico 2** übertragen werden.

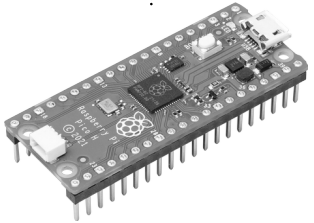
SOFTWARE

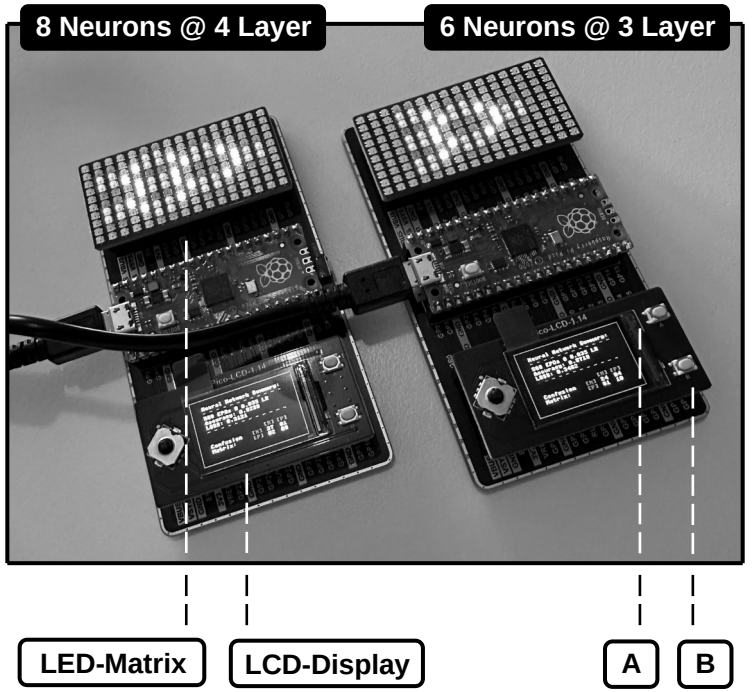


KI-ENNA

(E)IN (N)EURONALES (N)ETZ ZUM (A)USPROBIEREN








```python
# Sigmoid function
def sigmoid(x):
    import math
    z = [1 / (1 + math.exp(-x[kk])) for kk in range(len(x))]
    return z
```

1

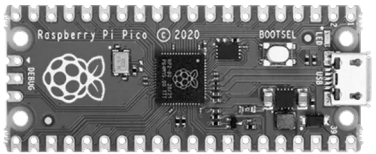
PROGRAMMIERBEISPIELE

Die Software von KI-ENNA ermöglicht die Auswahl von zwei Neuronalen Netzen, jeweils mit **8 Neuronen in 4 Schichten** bzw. **6 Neuronen in 3 Schichten**. Die Auswahl erfolgt über **Button A** bzw. **Button B** und wird über das LCD Display angeleitet. Als **Aktivierungsfunktionen** sind die **Sigmoid-Funktion** (1) und **ReLU-Funktion** (2) vorprogrammiert. In MicroPython können auch alternative Aktivierungsfunktionen programmiert werden. Darüber hinaus sind Beispiele zum Betrieb der **LED-Matrix** und des **LCD-Displays** vorprogrammiert. Um KI-ENNA mit anderen Datensätzen zu trainieren, sind auf GitHub beispielhaft das in der **Anaconda Cloud** vortrainierte Modell, dessen **Parameter und Hyperparameter** und deren Umsetzung für den Microcontroller hinterlegt. Das Ergebnis wird als **Confusion-Matrix** auf dem LCD-Display ausgewiesen.

```python
# ReLU function
def relu(x):
    y = []
    for i in range(len(x)):
        if x[i] >= 0:
            y.append(x[i])
        else:
            y.append(0)

    return y
```

2



python