GPIOs kurz und bündig

GPIOs (**G**eneral **P**urpose **I**nput **O**utput) sind Pins am Pico, die zur Ein- und Ausgabe von digitalen (Ein/Aus bzw. 1/0) Signalen geeignet sind. Damit können direkt angeschlossene Bauteile geschaltet werden (z.B. LED an/aus) oder auch eingehende Signale (Schalter gedrückt ja/nein) ausgelesen werden.

Auf der Übersichtsgrafik der Anschlüsse sind diese Pins mit grün und der Beschriftung **GPxx** (z.B. *GP12*) gekennzeichnet. Der Pico hat 29 GPIOs, wobei nur 26 als volle Pins rausgeführt sind. Die GPIOs sind von 0 bis 28 durchnummeriert, wobei GP23 und GP24 nicht existieren (bzw. nicht verfügbar sind) und GP25 die onboard-LED ist.

GPIOs schalten

Wie immer muss zuerst die Bibliothek des Pico geladen werden, um die Hardware direkt ansteuern zu können:

```
import machine
```

Lädt die notwendige Bibliothek und sollte am Beginn des Programms stehen.

Nun muss der GPIO als Objekt initialisiert und einer Variablen zugewiesen werden:

```
led_intern = machine.Pin(25, machine.Pin.OUT)
```

Weist der Variablen **led_intern** den **GPIO25** (die onboard LED) als **Output** zu. Damit kann der Anschlus jetzt geschaltet werden. Dabei steht **1** für AN und **0** für AUS.

```
led_intern.value(1)   # schaltet den GPIO an
led_intern.value(0)   # schaltet den GPIO aus
led_intern.toggle()   # schaltet den GPIO um
```

GPIOs auslesen

Auch hier wird zu erst die Bibliothek geladen, dann kann der GPIO initialisiert und zugewiesen werden:

```
taster = machine.Pin(0, machine.Pin.IN, machine.Pin.PULL_DOWN)
```

Weist der Variablen **taster** den **GPIO0** als Input mit **PullDown**-Widerstand zu. Letzteres in notwendig, weil sonst ein Kurzschluss entstehen würde, der die Elektronik beschädigen könnte!

Mit der Funktion **value()** kann nun der aktuelle Wert des GPIO ausgelesen werden:

```
aktueller_wert = taster.value()
```

Beispiel LED blinken

```
# ------------------------------------------------------
# LED per Knopf schalten
# ------------------------------------------------------

# Zusatzfunktionen laden
import machine   # Raspi Pico Funktionen
import time      # Funktionen rund um Zeit

# Weist GPIO25 als Ausgabe der Variablen ledIntern zu
ledIntern = machine.Pin(25, machine.Pin.OUT)

# Weist GPIO15 als Ausgabe der Variablen ledStatus zu
ledStatus = machine.Pin(15, machine.Pin.OUT)

# Weist GPIO0 als Eingang der Variablen taster zu
taster = machine.Pin(16, machine.Pin.IN, machine.Pin.PULL_DOWN)

alterTasterWert = 0

# Endlosschleife
while True:
    # Aktuellen Wert des Tasters auslesen
    tasterWert = taster.value()
    # Wenn sich der Wert geändert hat, dann...
    if (tasterWert != alterTasterWert):
        # In Kommandozeile ausgeben
        print("Taster jetzt {}".format(tasterWert))
        # Status LED entsprechend schalten
        ledStatus.value(tasterWert)
        # den neuen Status merken
        alterTasterWert = tasterWert
    # halbe Sekunde warten
    time.sleep(0.5)
    # onboard LED toggeln - zeigt Programmaktivität
    ledIntern.toggle()
```

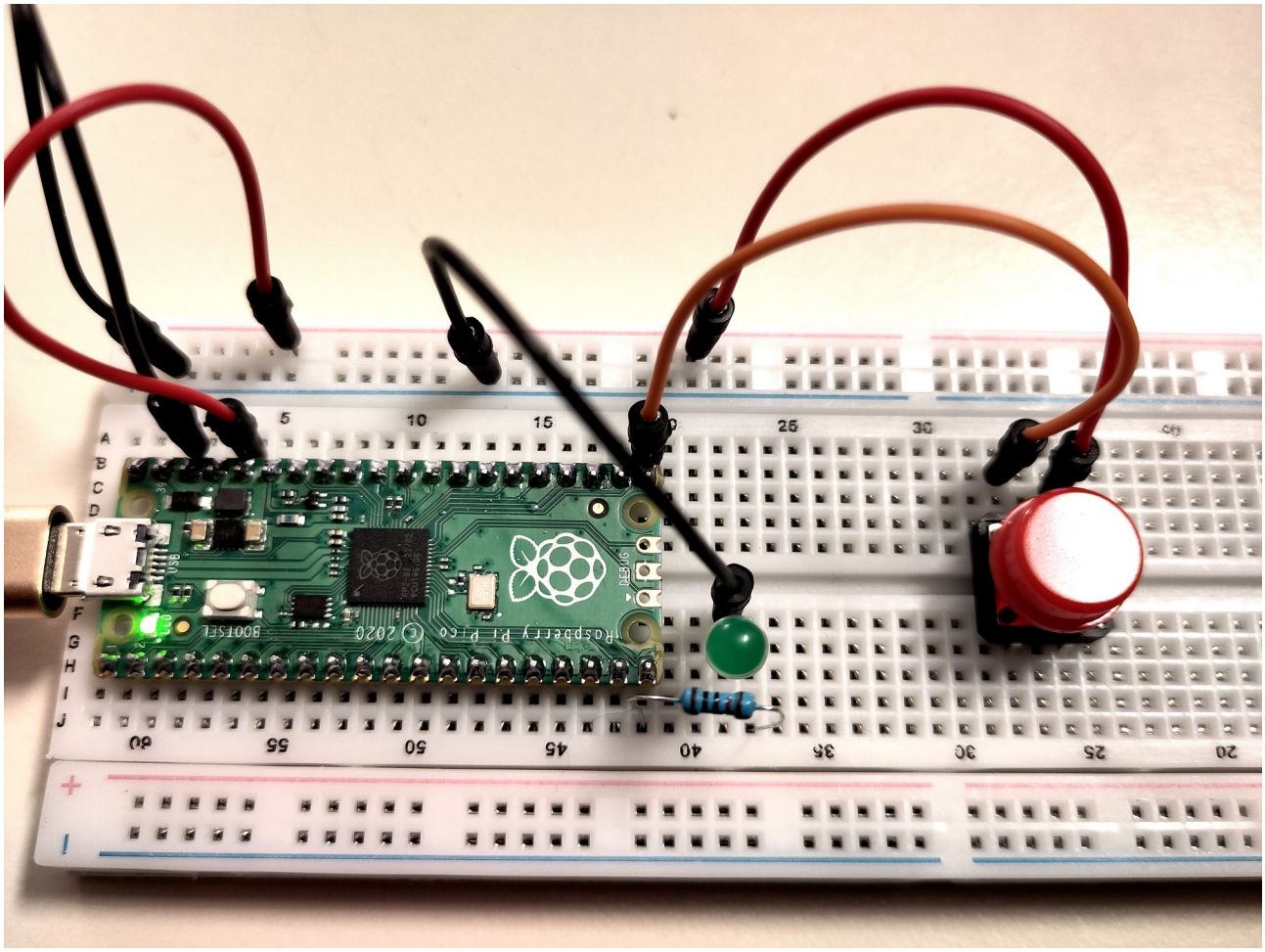

Bild 1: Aufbau der Schaltung zu "LED per Knopf schalten"

Der Aufbau

- Pin 36 (3V3 OUT) mit der roten PLUS-Zeile verbinden
- Pin 38 (GND) mit der blauen MINUS-Zeile verbinden
- Wiederstand von Pin 20 (GP15) zu einer leeren Spalte des Breadboard verbinden. ⚠ Der Widerstand ist wichtig, da die LED kaum einen elektrischen Widerstand hat und es daher ansonsten zu einem Kurzschluss kommen würde, der die Elektronik beschädigen könnte!
- LED mit dem langen Fuß in die gleiche Spalte wie den Wiederstand, mit dem kurzen in eine freie daneben
- Die Spalte mit dem kurzen Fuß der LED mit der blauen MINUS-Zeile verbinden (GND)
- Den Knopf mittig auf das Board setzen, so dass die Beinchen jeweils in der zweiten Zeile von der Mitte sitzen (siehe Foto)
- Eines der Beinchen mit Pin 21 (GP16) verbinden
- Das andere auf der gleichen Seite mit der roten MINUS-Zeile (3V3 OUT)