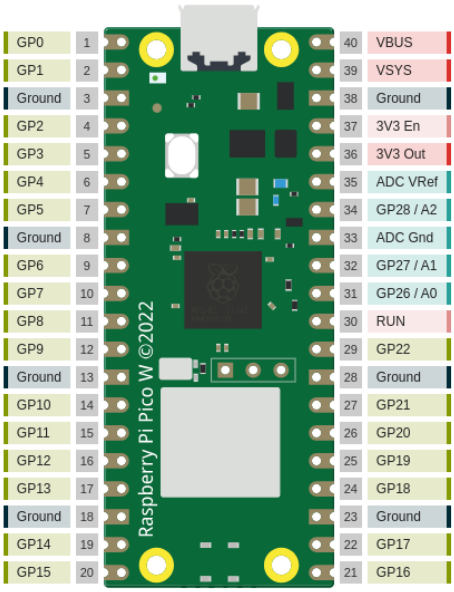
1 GPIO - General purpose Input and Output

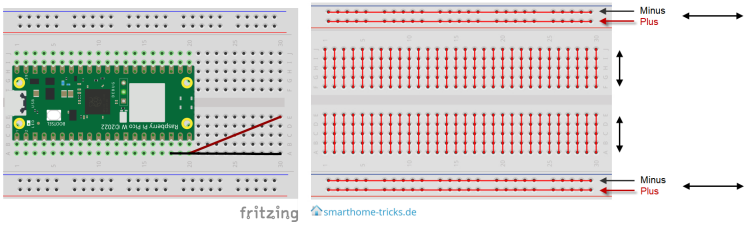
Über die **GPIO Pins** können verschiedene externe Geräte auf dem Steckbrett angeschlossen werden. Der Pico hat die folgenden Anschlüsse:



Uns interessieren heute aber nur die GP und die Ground Anschlüsse. Da wir gesehen haben, dass wir den Pico auch über USB Programmieren werden, wir uns aber nicht ständig selber Hacken wollen, wollen wir jetzt einen sehr primitiven Schalter einbauen. Sodass der Pico erkennt, ob er einen Angriff durchführen soll oder nicht.

1.1 attackMode

Unser Schalter besteht dabei aus zwei „ Jumper-Kabeln“, die wir mithilfe des Steckbretts verbinden. Beachte daher den folgenden Aufbau, sowie die Verbindungen auf einem Steckbrett:



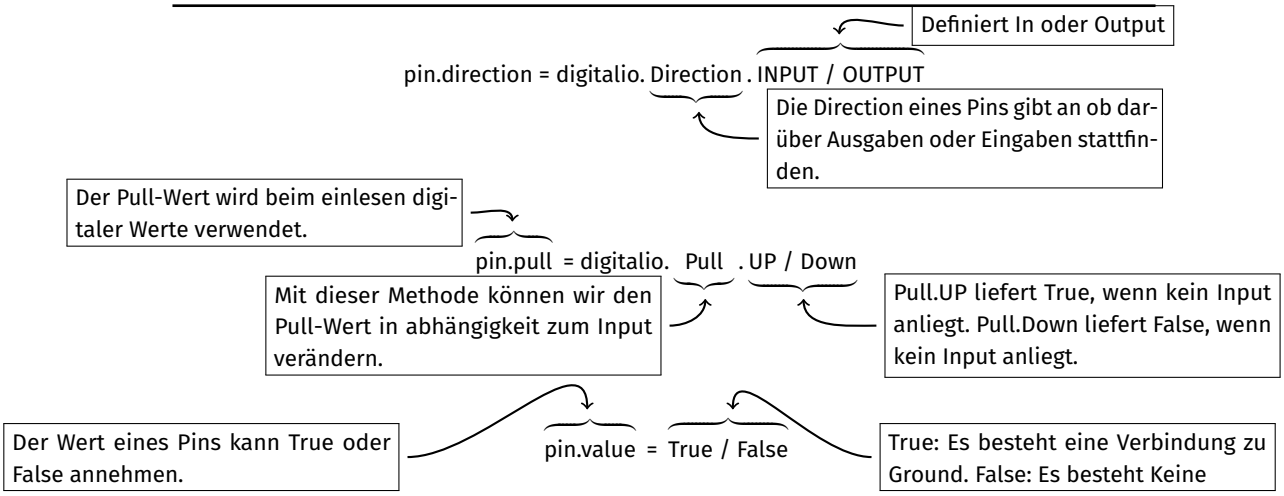
Damit der Pico prüfen kann, ob diese Verbindung besteht oder nicht brauchen wir die folgenden Befehle:

digitalio ist ein Modul, dass uns ermöglicht GPIO Pins anzusteuern und auszulesen.

```
pin = digitalio.DigitalInOut(GPXX)
```

GPXX ist ein Pin aus der obigen Abbildung. Z.B. GP9

Diese Methode teilt dem Pico mit, dass wir diesen Pin in Zukunft nutzen wollen.



Weitere Infos: https://docs.circuitpython.org/en/latest/shared-bindings/digitalio
In euren Dokumenten findet ihr bereits den folgenden Quellcode:

```
attackMode = False
attackPin = digitalio.DigitalInOut(???)
attackPin.direction = digitalio.Direction.???
attackPin.pull = digitalio.Pull.UP
attackMode = attackPin.???
```

Aufgabe: Vervollständige

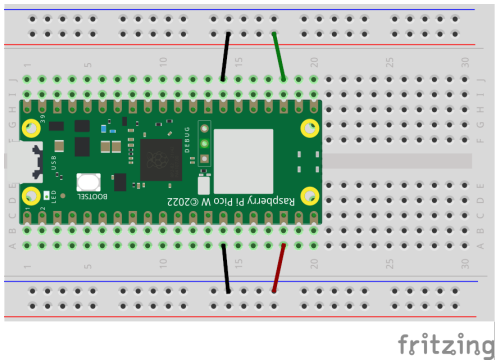
a) Ersetze im obigen Quellcode die Fragezeichen, sodass der Pico erkennt, ob der GP15 mit dem Ground verbunden ist.

b) Verwende die Variable attackMode, sodass die Codezeile Payload.run() nur dann ausgeführt wird, wenn attackMode auf True steht. Bedenke dabei die Einrückungen! Die mit # versehen Zeilen 18 - 21 kannst du fürs erste ignorieren.

Jetzt haben wir schon einiges über GPIO-Pins gelernt. Im nächsten Schritt wollen wir durch blinkende LEDs signalisieren, ob der attackMode aktiv ist oder nicht.

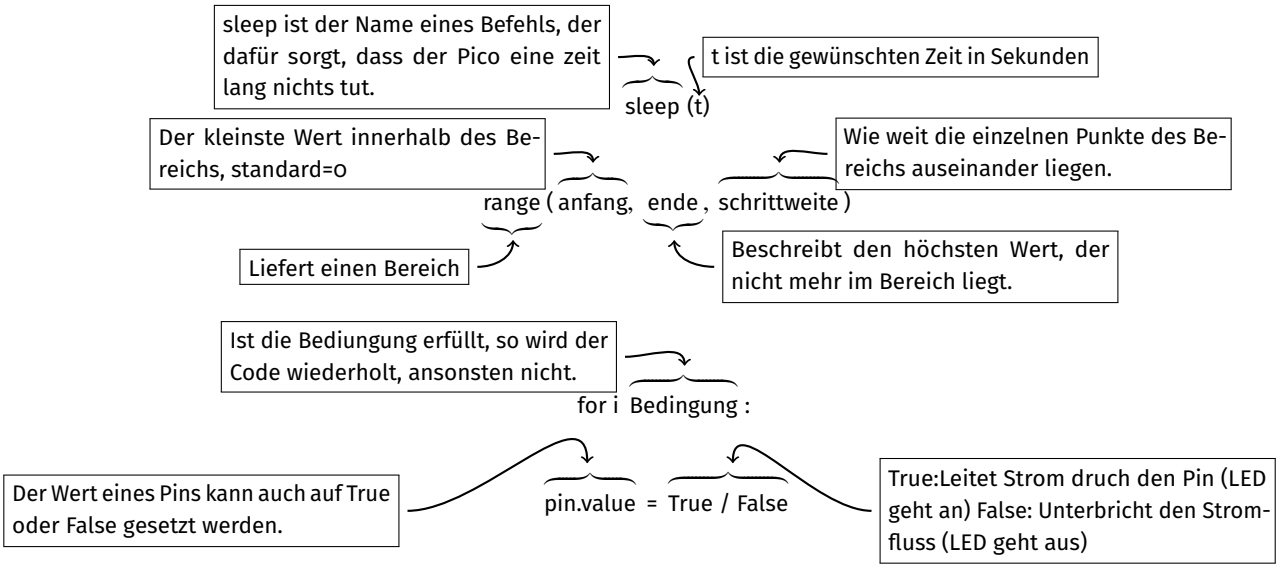
1.2 GPIOs und LEDs

Dafür verwenden wir den folgenden Aufbau:



Die schwarzen Linien stellen dabei zwei 180 Ohm Widerstände, die grüne Linie die grüne LED und die rote Linie die rote LED dar.

Die rote LED ist also über den **GP10** als **OUTPUT** Pin angeschlossen. Die grüne LED ist über den **GP21** als **OUTPUT** Pin angeschlossen. Weiter brauchen wir eine **For - Schleife**, den **sleep** und den **range** Befehl.



Aufgabe: Blinkende LEDs

a) In eurem Quellcode findet ihr passende Platzhalter, die mit: #<— und #—> gekennzeichnet sind. Schreibe an deren Stelle mit der oben kennen gelernten Syntax einen Code, der die rote LED 10 mal blinken lässt, wenn attackMode auf True steht. Die LED soll dabei 0.25 Sekunden an und dann wieder 0.25 Sekunden aus sein.

b) In eurem Quellcode findet ihr passende Platzhalter, die mit: #<— und #—> gekennzeichnet sind. Schreibe an deren Stelle mit der oben kennen gelernten Syntax einen Code, der die grüne LED 10 mal blinken lässt, wenn attackMode auf False steht. Die LED soll dabei 0.25 Sekunden an und dann wieder 0.25 Sekunden aus sein.

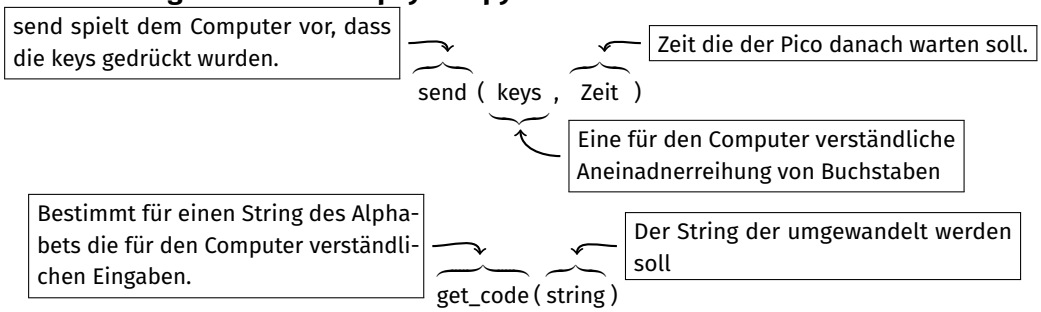
Tipp: https://www.youtube.com/watch?v=fTeiPwYLw8w?t=234

Nachdem wir unseren Pico jetzt abgesichert haben können wir nun anfangen ein paar harmlose Angriffe durchzuführen.

2 HID - Human Interface Devices und Bad USBs

Alle in dieser Woche gezeigten Angriffe machen sich eine grundlegende Schwachstelle zunutze. Computer vertrauen bei HID - Geräten wie Tastaturen und Mäuse noch immer darauf, dass an der Tastatur ein Mensch sitzt. Das muss aber nicht sein. Heute wollen wir uns genau diese Schwachstelle zunutze machen und ebenfalls auf diesem Weg Angriffe starten.

2.1 Wichtige Methoden für payload.py



2.2 Cheat Sheet - spezieller Tasten

Da nicht jede Taste einer Tastatur sich in einem Alphabet befindet, gibt es diese Liste solcher Tasten. Ihre Bedeutung ergibt sich häufig selbst.

- BACKSPACE
- DEL
- DOWNARROW
- END
- ENTER
- ESC
- HOME
- INSERT
- LALT
- LCONTROL
- LEFTARROW
- LGUI
- LSHIFT
- PAGEDOWN
- PAGEUP
- PAUSE
- PRTSCR
- RALT
- RCONTROL
- RGUI
- RIGHTARROW
- RSHIFT
- SPACE
- TAB
- UPARROW
- WIN
- SHIFT
- ALT
- CTRL
- COMMAND
- OPTION
- CTRLALTDEL
- RUN
- SPOTLIGHT
- CLOSE

Aufgabe: Attack on Python

Ergänze mit den oben genannten Befehlen und den Sondertasten, die Datei payload.py die du im CIRCUITPY Ordner findest, sodass beim Ausführen „notepad“ (der standard Texteditor von Windows) geöffnet wird und „Hallo Welt“ in den Editor eingegeben wird.

Möglicherweise interessantes Projekt:
https://www.youtube.com/watch?v=JON76zbiL1o

Weitere interessante Quellen:
https://www.github.com/dsymbol/ducky-payloads/tree/main/payloads
https://www.github.com/hak5/usbrubberducky-payloads/tree/master/payloads