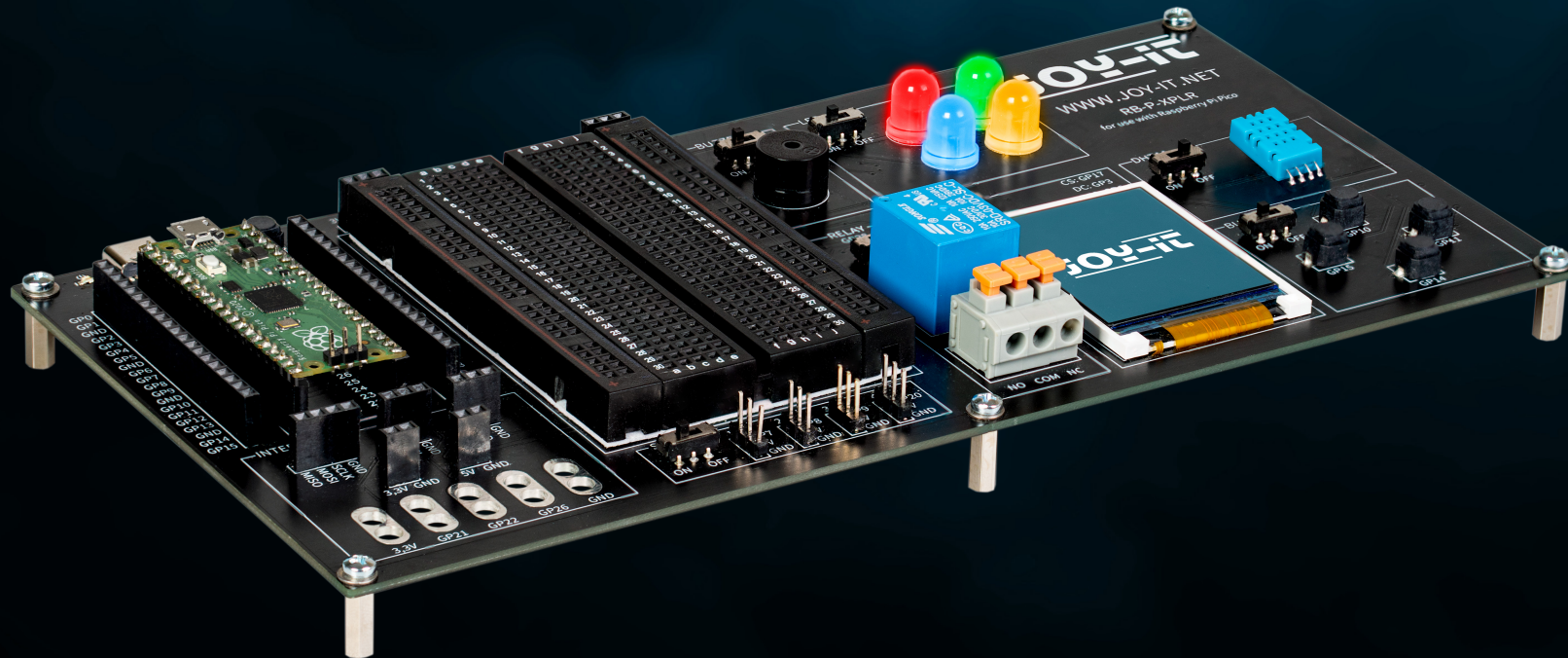


# EXPLORER BOARD

RB-P-XPLR

JOY-IT



# INHALTSVERZEICHNIS

1. Allgemeine Informationen .....	3
2. Geräteübersicht & Pin-Belegung .....	3
3. Raspberry Pi Pico .....	5
4. Module im Detail.....	7
4.1 Buzzer .....	7
4.2 RGB LEDs.....	8
4.3 Relais.....	9
4.4 TFT .....	10
4.5 DHT11 .....	11
4.6 Buttons.....	12
4.7 Servos .....	13
4.8 Interfaces .....	14
4.9 Breadboard .....	15
5. Informations- & Rücknahmepflichten .....	16
6. Support .....	17

# 1. ALLGEMEINE INFORMATIONEN

Sehr geehrte\*r Kunde\*in,  
vielen Dank, dass Sie sich für unser Produkt entschieden haben. Im Folgenden zeigen wir Ihnen, was bei der Inbetriebnahme und der Verwendung zu beachten ist.

Sollten Sie während der Verwendung unerwartet auf Probleme stoßen, so können Sie uns selbstverständlich gerne kontaktieren.

# 2. GERÄTEÜBERSICHT & PIN-BELEGUNG

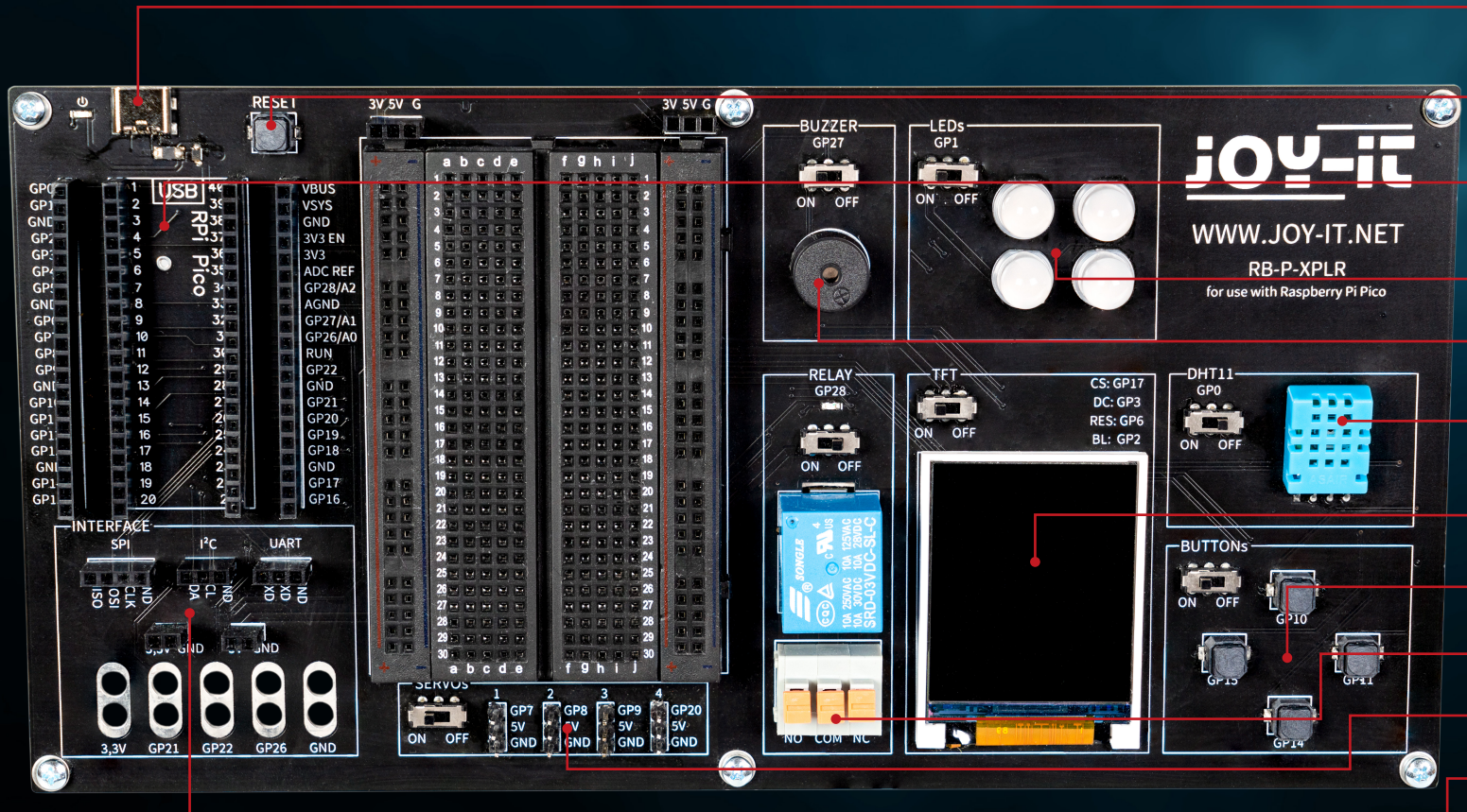
Unser Explorer Board ist die einfache und effiziente Möglichkeit, Ihre Raspberry Pi Pico Projekte zu entwickeln.

Da die wichtigsten Komponenten bereits integriert sind, sparen Sie Zeit und Mühe beim Verkabeln. Das Explorer Board verfügt über eine breite Palette an Interface-Anschlüssen, sodass Sie Ihre Projekte mit einer Vielzahl von Modulen und Geräten verbinden können. Mit dem integrierten Breadboard lassen sich eigene Projekte schnell aufbauen und realisieren.

Dank der Möglichkeit, alle Module einzeln zu- oder abzuschalten, können Sie Ihre Pins, welche zusätzlich separat nach außen geführt sind, jederzeit für andere Projekte nutzen oder auf dem integrierten Breadboard experimentieren.

Alle verbauten Komponenten lassen sich über den jeweiligen Schalter ausschalten, falls diese nicht benötigt werden. So können die zugehörigen Pins, falls nötig, auch für andere Komponenten verwendet werden.

Links und Rechts des Raspberry Pi Picos sind alle Pins noch einmal zusätzlich ausgeführt. Hier können Komponenten direkt angeschlossen oder über zusätzliche Kabel zum integrierten Breadboard geführt werden.



5 V USB-C  
Spannungsversorgung

Reset Button

Steckplatz  
Raspberry Pi Pico

4x RGB LED

Buzzer

DHT11 Temperatur-  
& Feuchtigkeitssensor

1,8" TFT Display

4x Buttons

Relais

4x Servo-Interface

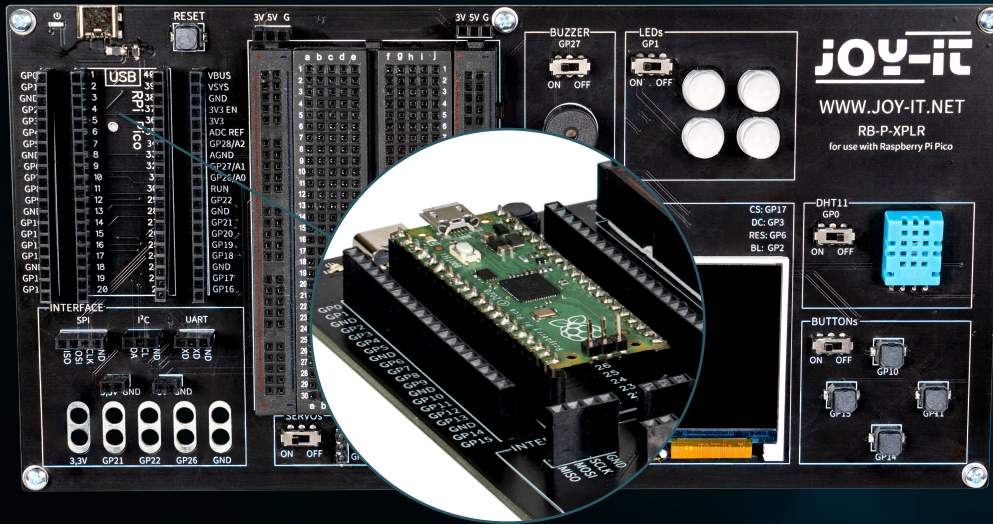
Interface-Anschlüsse

## PIN-BELEGUNG

Buzzer	GP27
LEDs	GP1
Relais	GP28
1,8" TFT Display	CS: GP17, DC: GP3, RES: GP6, BL: GP2
DHT11	GP0
Buttons	GP10, GP11, GP14 & GP15
Servos	GP7, GP8, GP9 & GP20
UART	RXD: GP13, TXD: GP12
I2C	SDA: GP4, SCL: GP5
SPI	MISO: GP16, MOSI: GP19, SCLK: GP18

### 3. RASPBERRY PI PICO

Stecken Sie zunächst Ihren Raspberry Pi Pico auf dem Steckplatz Ihres Boards ein.



Schließen Sie zur Programmierung nun ein Micro-USB Kabel an Ihren Computer und an den Raspberry Pi Pico an.

**ACHTUNG!** Der USB-C Anschluss auf dem Explorer-Board dient ausschließlich zur Stromversorgung. Hierüber werden keine Daten an den Raspberry Pi übertragen.

Zur Übertragung unseres Beispielprogramms können Sie ein geeignetes Entwicklungsprogramm Ihrer Wahl verwenden. Wir empfehlen hier die **Thonny Python IDE**.

**ACHTUNG!** Wenn Sie neu in der Welt der Mikrocontroller und Elektronik sind, keine Sorge! Wir haben eine spezielle Anfängeranleitung für Sie vorbereitet. Diese Anleitung ist speziell auf die Bedürfnisse von Anfängern zugeschnitten und erklärt Ihnen die Verwendung des Raspberry Pi Picos Schritt für Schritt.

Von der Grundkonfiguration bis hin zur Ausführung von Projekten - in dieser Anleitung begleiten wir Sie durch den gesamten Prozess. Unsere Anleitung umfasst leicht verständliche Erklärungen und nützliche Tipps um Ihnen zu helfen, Ihre Fähigkeiten im Umfang mit dem Raspberry Pi Pico schnell und effektiv zu entwickeln. Unsere Anleitung können Sie **hier** herunterladen.

## 4. MODULE IM DETAIL

Im Folgenden werden alle Module, die auf dem Explorer Board vorhanden sind, einzeln mit Beispielcodes erläutert. **Hier** können Sie alle Beispielcodes und Bibliotheken herunterladen, sowie einen Beispielcode, der alle Module miteinander verbindet.

Für die Verwendung einiger Module werden externe Bibliotheken, sowie eine Schrift-Datei verwendet. Laden Sie sich die Bibliotheken herunter und laden Sie diese in den lib-Ordner Ihres Raspberry Pi Picos. Platzieren Sie die Schrift-Datei im Hauptverzeichnis Ihres Raspberry Pi Picos.

### 4.1 BUZZER

Ein Buzzer oder Summer erzeugt einen Signalton, ähnlich wie ein Lautsprecher. Im Gegensatz zum Lautsprecher eignet er sich jedoch nur für einen eingeschränkten Frequenzbereich, sodass er keinen guten Klang zur Wiedergabe von Musik oder Sprache erzeugt. Dafür eignet er sich hervorragend, um laute Warntöne in Form von Piepsern zu erzeugen. Wann immer ein elektrisches Gerät einen Warnton erzeugt, übernimmt das fast immer ein Buzzer. Zum

```
# Load libraries
from machine import Pin, PWM

buzzerPin = Pin(27)
buzzer = PWM(buzzerPin)

while True:
    # Activate buzzer for 1 sec
    buzzer.freq(1000)
    buzzer.duty_u16(1000)
    sleep(1)
    buzzer.duty_u16(0)
    sleep(1)
```



## 4.2 RGB LEDs

RGB-LEDs sind eine Art Leuchtdiode, die Rot, Grün und Blau kombiniert, um eine Vielzahl von Farben zu erzeugen. Ähnlich wie ein Buzzer nur einfache Töne erzeugt, können RGB-LEDs keine komplexen Bilder darstellen, aber sie sind hervorragend darin, Farben zu mischen und zu variieren. Jede LED in einer RGB-Einheit kann in ihrer Intensität variiert werden, um unterschiedliche Farbtöne zu erzeugen, von sanften Pastellfarben bis hin zu leuchtenden, satten Farben. Dies macht sie ideal für Stimmungsbeleuchtung, dekorative Beleuchtung und in Anwendungen, wo visuelle Signale erforderlich sind, wie in Gaming-Setups oder als Statusanzeigen in elektronischen Geräten. Ihre Vielseitigkeit und Energieeffizienz haben sie zu einer beliebten Wahl in modernen Beleuchtungssystemen gemacht, obwohl sie, ähnlich wie der Buzzer, aufgrund ihrer einfachen Funktionsweise keine komplexen Bilder oder Muster ohne zusätzliche Steuerungseinheiten erzeugen können.

**Der GPIO LEDs sind an den GPIO Pin GP1 angeschlossen.**

```
# Load libraries
from machine import Pin, PWM
from utime import sleep
from neopixel import NeoPixel

ledPin = 1
ledCount = 4

# Initialize GPIOs
led = Pin(ledPin, Pin.OUT)
led = NeoPixel(Pin(ledPin, Pin.OUT), ledCount)

while True:
    # Turn LEDs white
    for i in range (ledCount):
        led[i] = (255, 255, 255)
    led.write()
    sleep(1)
    # Turn LEDs red
    for i in range (ledCount):
        led[i] = (255, 0, 0)
    led.write()
    sleep(1)
    # Turn LEDs blue
    for i in range (ledCount):
        led[i] = (0, 0, 255)
    led.write()
    sleep(1)
    # Turn LEDs green
    for i in range (ledCount):
        led[i] = (0, 255, 0)
    led.write()
    sleep(1)
```





## 4.3 RELAIS

Relais sind einige der ältesten elektromechanischen Komponenten und funktionieren als elektrisch gesteuerte Schalter. Mit einer kleinen Eingangsspannung und geringem Strom kann eine große elektrische Last am Ausgang ein- und ausgeschaltet werden. Wenn das Relais durchschaltet, leuchtet auch die rote LED. In die Klemmbuchse können Sie abisolierte Kabelenden einstecken (durch Niederdrücken des orangenen Hebels), um die drei Anschlüsse zu nutzen.

**Das Relais ist an den GPIO Pin GP28 angeschlossen.**

```
# Load libraries
from machine import Pin, PWM
from utime import sleep

relayPin = 28
# Initialize GPIOs
relay = Pin(relayPin, Pin.OUT)

while True:
    # Toggle Relay
    relay.on()
    sleep(1)
    relay.off()
    sleep(1)
```



## 4.4 TFT

Das Flüssigkristalldisplay (LCD TFT) mit rund 65.000 Farben und einer Diagonale von 1,8 Zoll hat eine Auflösung von 128×160 Pixeln und kann über SPI angesteuert werden. Es eignet sich für die Darstellung von bunten Grafiken und Bildern. Buchstaben und andere Zeichen werden dabei als Grafiken dargestellt, die aus vielen einzelnen Punkten zusammengesetzt sind.

**Das TFT ist an die GPIO Pins GP17 (CS), GP3 (DC), GP6 (RES) und GP2 (BL) angeschlossen.**

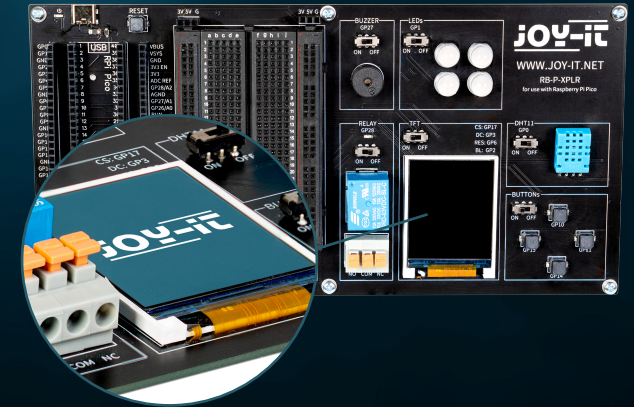
```
from machine import Pin, SPI
import ST7735

# Initialize LCD
spi = SPI(0, baudrate=8000000, polarity=0, phase=0, sck=Pin(18), mosi=Pin(19),
miso=Pin(16))
lcd = ST7735.ST7735(spi, rst=6, ce=17, dc=3)
backlight = Pin(2, Pin.OUT)

# Turn backlight on
backlight.high()
lcd.reset()
lcd.begin()

# Display content on the LCD
lcd.fill_screen(lcd.rgb_to_565(0, 255, 0)) # Fills the screen with a green color

# Display text
lcd.p_string(20, 50, 'Hello, World!')
```



Neben Texten können beispielsweise auch Rechtecke dargestellt werden:

```
# Draw red rectangle
lcd.draw_block(10, 10, 50, 50, lcd.rgb_to_565(255, 0, 0))
```

## 4.5 DHT 11

Der DHT11-Sensor kann Temperaturen von 0 °C bis 50 °C ( $\pm 2$  °C Genauigkeit) und eine relative Luftfeuchtigkeit von 20 % bis 80 % ( $\pm 5$  %) erfassen (höchstens einmal pro Sekunde). Wetterstationen sind vermutlich das primäre Einsatzgebiet für einen Sensor wie den DHT11. Um die Funktionalität zu testen, genügt es, den Mund nahe über den Sensor zu halten und langsam auszuatmen. Die Atemluft unterscheidet sich hinsichtlich Temperatur und Feuchtigkeit von der Umgebung, was zu einer deutlichen Änderung der Werte führen sollte.

**Der DHT11 ist an den GPIO Pin GP0 angeschlossen.**

```
from machine import Pin
from dht import DHT11
from utime import sleep

# Initialize DHT11 Sensor
dhtPin = 0
dht = DHT11(Pin(dhtPin, Pin.IN))

while True:
    # Measure DHT11 values
    dht.measure()
    temp = dht.temperature() # Temperature in Celsius
    humid = dht.humidity()   # Relative Humidity in %

    # Print the measurements
    print('Temperature:', temp, '°C')
    print('Humidity:', humid, '%')

    sleep(2) # Wait for 2 seconds before the next reading
```



## 4.6 BUTTONS

Buttons sind interaktive Elemente in Benutzeroberflächen, die eine einfache, aber essenzielle Funktion erfüllen: die Benutzereingabe. Ähnlich wie RGB-LEDs eine Vielzahl von Farben anzeigen können, dienen Buttons dazu, eine breite Palette von Befehlen und Aktionen in digitalen Umgebungen zu initiieren.

**Die Buttons sind an die GPIO Pins GP10 (Oben), GP11 (Rechts), GP14 (Unten) und GP15 (Links) angeschlossen.**

```
from machine import Pin

# Define button pins
buttons = [10, 11, 14, 15]

# Initialize buttons
buttonOne = Pin(buttons[0], Pin.IN, Pin.PULL_DOWN)
buttonTwo = Pin(buttons[1], Pin.IN, Pin.PULL_DOWN)
buttonThree = Pin(buttons[2], Pin.IN, Pin.PULL_DOWN)
buttonFour = Pin(buttons[3], Pin.IN, Pin.PULL_DOWN)

# Define button handler functions
def buttonUp(pin):
    print("Button Up Pressed")

def buttonRight(pin):
    print("Button Right Pressed")

def buttonDown(pin):
    print("Button Down Pressed")

def buttonLeft(pin):
    print("Button Left Pressed")

# Attach interrupt handlers to buttons
buttonOne.irq(trigger=Pin.IRQ_RISING, handler=buttonUp)
buttonTwo.irq(trigger=Pin.IRQ_RISING, handler=buttonRight)
buttonThree.irq(trigger=Pin.IRQ_RISING, handler=buttonDown)
buttonFour.irq(trigger=Pin.IRQ_RISING, handler=buttonLeft)
```



## 4.7 SERVOS

Ein Servo besteht aus einem Elektromotor mit Getriebe und Steuerelektronik. Auf der Ausgangsseite des Getriebes befindet sich ein Zahnrad, auf das das Servohorn montiert wird. Der Servo kann die Achse in einem Bereich von etwa 180° bewegen. Einsatzgebiete für Servos sind zum einen der Modellbau, um bei einem Flugzeug oder Schiff beispielsweise die Flügel- oder Ruderstellung zu kontrollieren. Auch im Automobilbau werden immer mehr Servos verbaut, um Türen automatisch zu schließen, für Fensterheber, Spiegel und andere verstellbare Elemente.

**Die Servo-Anschlüsse sind die GPIO Pins GP7, GP8, GP9 und GP20.**

```
from machine import Pin, PWM
from utime import sleep

# Servo pin numbers
servoOnePin = 7
servoTwoPin = 8
servoThreePin = 9
servoFourPin = 20

# Initialize servos
servoOne = PWM(Pin(servoOnePin))
servoTwo = PWM(Pin(servoTwoPin))
servoThree = PWM(Pin(servoThreePin))
servoFour = PWM(Pin(servoFourPin))

# Servo degree positions in nanoseconds
deg0 = 500000
deg45 = 1000000
deg90 = 1500000
deg135 = 2000000
deg180 = 2500000

while True:
    # Move each servo through a range of angles
    for servo in [servoOne, servoTwo, servoThree, servoFour]:
        servo.duty_ns(deg0)
        sleep(1)
        servo.duty_ns(deg45)
        sleep(1)
        servo.duty_ns(deg90)
        sleep(1)
        servo.duty_ns(deg135)
        sleep(1)
        servo.duty_ns(deg180)
        sleep(1)
```



## 4.8 INTERFACES

Interface-Anschlüsse spielen eine entscheidende Rolle in der Welt der Elektronik, ähnlich wie Buttons in Benutzeroberflächen. Sie ermöglichen die Kommunikation und Stromversorgung zwischen verschiedenen elektronischen Bauteilen. Auf unserem Explorer Board sind daher im Interface-Bereich die folgenden Anschlüsse zu finden:

**SPI (Serial Peripheral Interface):** Dieser Anschluss wird für die schnelle serielle Datenübertragung verwendet. Er besteht typischerweise aus vier Leitungen: MISO (Master In, Slave Out), MOSI (Master Out, Slave In), SCK (Serial Clock) und SS (Slave Select). SPI eignet sich hervorragend für Situationen, in denen eine hohe Datenübertragungsrate erforderlich ist, wie bei der Ansteuerung von LCD-Displays oder SD-Karten.

**I2C (Inter-Integrated Circuit):** I2C ist ein Zwei-Draht-Interface, bestehend aus einer Datenleitung (SDA) und einer Taktleitung (SCL). Es wird häufig in Mikrocontroller-Anwendungen für die Kommunikation zwischen verschiedenen integrierten Schaltungen verwendet. Seine Einfachheit macht es ideal für Anwendungen, in denen nicht viele GPIO-Pins zur Verfügung stehen.

**UART (Universal Asynchronous Receiver/Transmitter):** Dieses Interface ermöglicht eine asynchrone serielle Kommunikation über zwei Leitungen: TX (Transmit) und RX (Receive). UART wird häufig für die Kommunikation zwischen Mikrocontrollern und Computern oder für die Anbindung von Modulen wie GPS-Empfängern oder Bluetooth-Modulen genutzt.

**3,3 V und 5 V Anschlüsse:** Diese Anschlüsse stellen die Stromversorgung für elektronische Komponenten bereit. 3,3 V wird oft für moderne Mikrocontroller und Sensoren verwendet, während 5 V häufig bei älteren oder leistungshungrigeren Geräten anzutreffen ist.

**Anschlüsse für Krokodilklemmen:** Diese Anschlüsse sind ideal für temporäre Verbindungen oder für Testzwecke. Sie ermöglichen eine einfache und schnelle Verbindung zu verschiedenen Komponenten oder Messgeräten ohne Löten. Insgesamt gibt es auf dem Explorer Board fünf solcher Anschlüsse, die für eine Vielzahl von Anwendungen flexibel eingesetzt werden können.

Jeder dieser Anschlüsse hat seine spezifische Anwendung und Bedeutung in der Elektronik, ähnlich wie verschiedene Arten von Buttons in einer Benutzeroberfläche unterschiedliche Funktionen haben. Sie bieten die notwendige Flexibilität und Funktionalität für den Aufbau und die Erweiterung elektronischer Systeme.



## 4.9 BREADBOARD

Breadboards sind ein unverzichtbares Werkzeug in der Welt der Elektronik, ähnlich wie Interface-Anschlüsse für die Verbindung verschiedener Komponenten entscheidend sind. Sie ermöglichen es, elektronische Schaltungen schnell und ohne Löten aufzubauen und zu testen, was sie besonders bei Prototyping und Bildungszwecken beliebt macht.

Ein Breadboard besteht typischerweise aus einem rechteckigen Kunststoffblock mit einer Vielzahl von eingebetteten Löchern, die in Reihen angeordnet sind. Diese Löcher sind intern durch metallische Leiterbahnen verbunden, die es ermöglichen, Komponenten und Drähte einfach einzustecken und zu verbinden. Die Standardanordnung eines Breadboards umfasst zwei Hauptbereiche:

**Die Hauptbereiche:** Diese bestehen aus einer Reihe paralleler Reihen von Löchern, üblicherweise durch eine zentrale Rinne getrennt. Die Löcher innerhalb einer Reihe sind elektrisch miteinander verbunden. Diese Anordnung eignet sich hervorragend für das Einsetzen von integrierten Schaltungen (ICs) und anderen Komponenten.

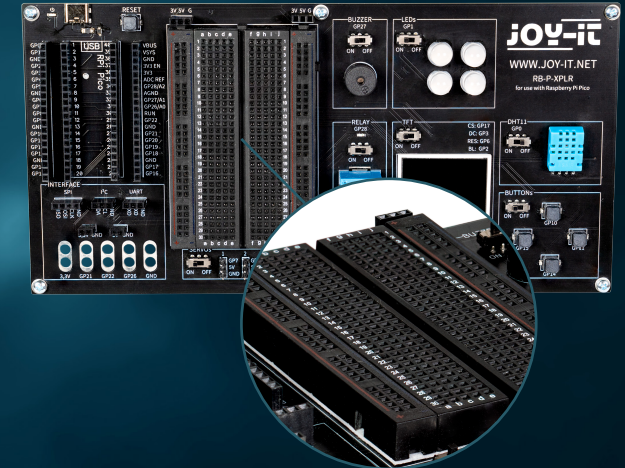
**Die Stromleisten:** Am Rand des Breadboards befinden sich in der Regel ein oder zwei Reihen von Löchern, die als Stromleisten dienen. Diese sind vertikal über die gesamte Länge des Breadboards miteinander verbunden und bieten eine bequeme Möglichkeit, Strom und Masse an verschiedenen Punkten der Schaltung zur Verfügung zu stellen.

Die Flexibilität eines Breadboards liegt in seiner Wiederverwendbarkeit und der Fähigkeit, Schaltungen ohne permanente Veränderungen aufzubauen. Dies macht es ideal für Experimente, da Fehler leicht korrigiert und Komponenten ohne Beschädigung entfernt werden können. Es ist auch ein ausgezeichnetes Lernwerkzeug, da es das Verständnis der Schaltungslogik und der Komponentenfunktionen in einer praktischen, visuellen Weise fördert.

Darüber hinaus sind Breadboards in verschiedenen Größen und mit unterschiedlichen Anzahlen von Verbindungspunkten erhältlich, um verschiedenen Anforderungen gerecht zu werden. Kleinere Breadboards eignen sich gut für einfache Projekte und Experimente, während größere für komplexere Schaltungen geeignet sind.

Trotz ihrer Vielseitigkeit haben Breadboards auch Einschränkungen. Sie eignen sich nicht gut für sehr hohe Frequenzen oder für Schaltungen, die eine hohe Leistung erfordern. Auch die Verbindungen können manchmal weniger zuverlässig sein als gelötete Verbindungen, besonders wenn das Breadboard mit der Zeit abgenutzt wird.

Insgesamt sind Breadboards ein unverzichtbares Werkzeug für jeden, der mit Elektronik arbeitet – von Anfängern, die die Grundlagen lernen, bis hin zu erfahrenen Entwicklern, die schnell und effizient Prototypen erstellen möchten. Sie sind das elektronische Äquivalent zu einem Skizzenbuch für einen Künstler: ein Ort, um Ideen zu erforschen und zu experimentieren, bevor das endgültige Werk entsteht.



# 5. INFORMATIONS- & RÜCKNAHMEPFLICHTEN

## UNSERE INFORMATIONS- UND RÜCKNAHMEPFLICHTEN NACH DEM ELEKTROGESETZ (ELEKTROG)

### **SYMBOL AUF ELEKTRO- UND ELEKTRONIKGERÄTEN:**

Diese durchgestrichene Mülltonne bedeutet, dass Elektro- und Elektronikgeräte nicht in den Hausmüll gehören. Sie müssen die Altgeräte an einer Erfassungsstelle abgeben. Vor der Abgabe haben Sie Altbatterien und Altakkumulatoren, die nicht vom Altgerät umschlossen sind, von diesem zu trennen.

### **RÜCKGABEMÖGLICHKEITEN:**

Als Endnutzer können Sie beim Kauf eines neuen Gerätes, Ihr Altgerät (das im Wesentlichen die gleiche Funktion wie das bei uns erworbene neue erfüllt) kostenlos zur Entsorgung abgeben. Kleingeräte bei denen keine äußere Abmessung größer als 25 cm sind, können unabhängig vom Kauf eines Neugerätes in haushaltsüblichen Mengen abgeben werden.

### **MÖGLICHKEIT RÜCKGABE AN UNSEREM FIRKENSTANDORT WÄHREND DER ÖFFNUNGSZEITEN:**

SIMAC Electronics GmbH, Pascalstr. 8, D-47506 Neukirchen-Vluyn

### **MÖGLICHKEIT RÜCKGABE IN IHRER NÄHE:**

Wir senden Ihnen eine Paketmarke zu, mit der Sie das Gerät kostenlos an uns zurücksenden können. Hierzu wenden Sie sich bitte per E-Mail an [service@joy-it.net](mailto:service@joy-it.net) oder per Telefon an uns.

### **INFORMATIONEN ZUR VERPACKUNG:**

Verpacken Sie Ihr Altgerät bitte transportsicher, sollten Sie kein geeignetes Verpackungsmaterial haben oder kein eigenes nutzen möchten kontaktieren Sie uns, wir lassen Ihnen dann eine geeignete Verpackung zukommen.



## 6. SUPPORT

Wir sind auch nach dem Kauf für Sie da. Sollten noch Fragen offen bleiben oder Probleme auftauchen, stehen wir Ihnen auch per E-Mail, Telefon und Ticket-Supportsystem zur Seite.

E-Mail: [service@joy-it.net](mailto:service@joy-it.net)

Ticket-System: <http://support.joy-it.net>

Telefon: +49 (0)2845 9360 – 50 (Mo. - Do.: 09:00 - 17:00 Uhr, Fr: 09:00 - 14:30 Uhr)

Für weitere Informationen besuchen Sie unsere Website:

**[WWW.JOY-IT.NET](http://www.joy-it.net)**