EXPLORER SET

RB-P-XPLR-SET

OBSAH

1. OBECNÉ INFORMACE

Vážený zákazníku, děkujeme, že jste si vybral náš výrobek. V následujících řádcích vás seznámíme s tím, co je třeba mít na paměti při uvádění do provozu a používání.

Pokud se během používání setkáte s neočekávanými problémy, neváhejte se na nás obrátit.

2. PŘEHLED ZAŘÍZENÍ A PŘIŘAZENÍ PINŮ

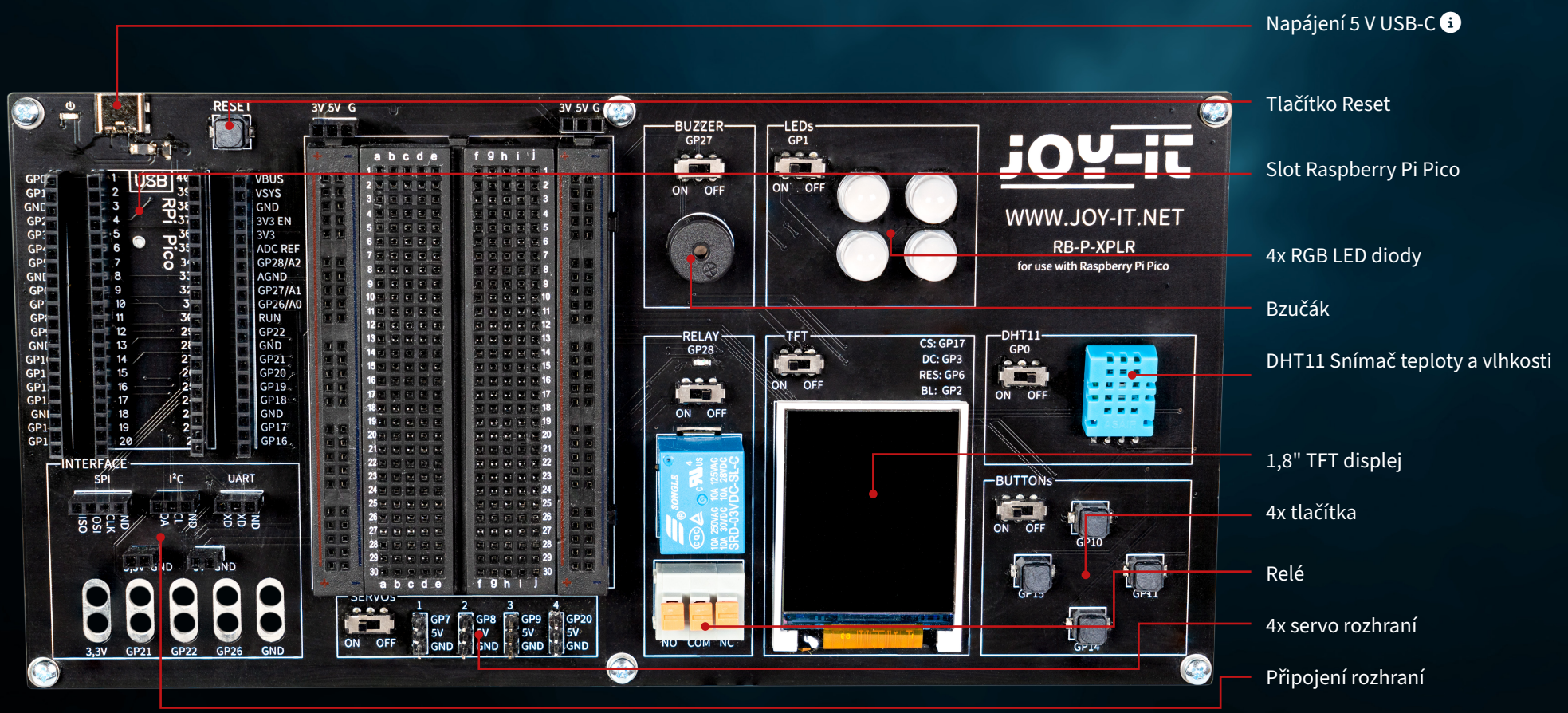
Naše deska Explorer Board představuje jednoduchý a efektivní způsob vývoje projektů Raspberry Pi Pico.

Díky již integrovaným nejdůležitějším komponentám ušetříte čas a námahu při zapojování. Deska Explorer Board má širokou škálu konektorů rozhraní, takže můžete své projekty připojit k různým modulům a zařízením. Díky integrovanému breadboardu můžete rychle sestavovat a realizovat vlastní projekty.

Díky možnosti zapínat nebo vypínat všechny moduly jednotlivě můžete své vývody, které jsou navíc vedeny samostatně ven, kdykoli použít pro jiné projekty nebo experimentovat na integrované desce s chlebem.

Všechny vestavěné komponenty lze vypnout pomocí příslušného přepínače, pokud je nepotřebujete. To znamená, že přidružené piny lze v případě potřeby použít i pro jiné součástky.

Vlevo a vpravo od Raspberry Pi Pico jsou všechny piny dodatečně navrženy. Součástky zde lze připojit přímo nebo je pomocí přídavných kabelů vést do integrovaného breadboardu.

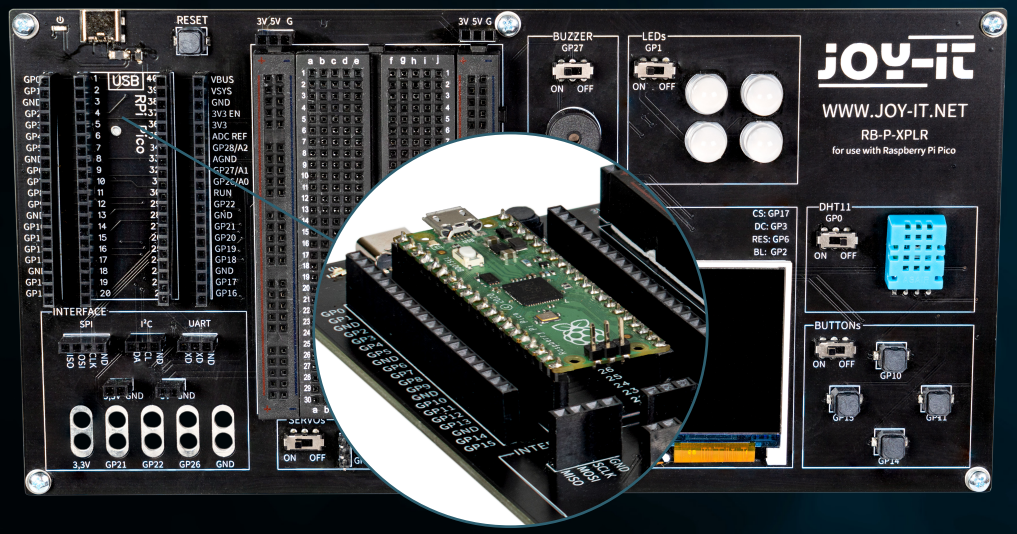


Vezměte prosím na vědomí, že připojení USB-C musí být vždy připojeno, aby bylo možné jej používat. Napájení přes připojení micro USB počítače Raspberry Pi Pico není možné.

PŘIŘAZENÍ PINŮ	
Bzučák	GP27
LED diody	GP1
Relé	GP28
1,8" TFT displej	CS: GP17, DC: GP3, RES: GP6, BL: GP2
DHT11	GP0
Tlačítka	GP10, GP11, GP14 & GP15
Serva	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

Nejprve zapojte počítač Raspberry Pi Pico do slotu na desce.



Nyní připojte kabel micro USB k počítači a k Raspberry Pi Pico pro programování.

POZOR! Port USB-C na desce Explorer slouží výhradně k napájení. Nepoužívá se k přenosu dat do počítače Raspberry Pi. K přenosu našeho ukázkového programu můžete použít vhodný vývojový program podle vlastního výběru. Doporučujeme prostředí **Thonny Python IDE**.

POZOR! Pokud jste ve světě mikrokontrolérů a elektroniky nováčky, nezoufejte! Připravili jsme pro vás speciálního průvodce pro začátečníky. Tato příručka je speciálně přizpůsobena potřebám začátečníků a vysvětluje, jak používat Raspberry Pi Pico krok za krokem.

V tomto průvodci vás provedeme celým procesem od základní konfigurace až po spuštění projektů. Náš průvodce obsahuje srozumitelná vysvětlení a užitečné tipy, které vám pomohou rychle a efektivně rozvíjet své dovednosti s Raspberry Pi Pico. Naši příručku si můžete stáhnout **zde**.

4. MODULY PODROBNĚ

V následujícím textu jsou všechny moduly dostupné na desce Explorer Board vysvětleny jednotlivě s ukázkovými kódy. Zde si můžete stáhnout všechny ukázkové kódy a knihovny a také ukázkový kód, který všechny moduly propojuje.

Pro použití některých modulů se používají externí knihovny a soubor s písmem. Stáhněte si tyto knihovny a nahrajte je do složky lib počítače Raspberry Pi Pico. Soubor s písmem umístěte do kořenového adresáře počítače Raspberry Pi Pico.

4.1 BZUČÁK

Bzučák vydává signální tón podobně jako reproduktor. Na rozdíl od reproduktoru je však vhodný pouze pro omezený frekvenční rozsah, takže nevytváří dobrý zvuk pro reprodukci hudby nebo řeči. Je však ideální pro generování hlasitých výstražných tónů v podobě pípnutí. Kdykoli elektrické zařízení generuje výstražný tón, jedná se téměř vždy o bzučák. Například v budících, detektorech kouře nebo připomínači nezapnutých bezpečnostních pásů v automobilech.

Bzučák je připojen na pin GPIO GP27.

```
# 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 LED DIODY RGB

LED diody RGB jsou typem světelných diod, které kombinují červenou, zelenou a modrou barvu a vytvářejí tak různé barvy. Podobně jako bzučák vydává pouze jednoduché tóny, RGB LED diody nemohou zobrazovat složité obrazy, ale jsou vynikající v míchání a střídání barev. Každá dioda LED v jednotce RGB může mít různou intenzitu a vytvářet různé odstíny, od jemných pastelových až po jasné, syté barvy. Díky tomu jsou ideální pro náladové osvětlení, dekorativní osvětlení a v aplikacích, kde jsou vyžadovány vizuální signály, například v herních sestavách nebo jako indikátory stavu v elektronických zařízeních. Díky své všestrannosti a energetické účinnosti se staly oblíbenou volbou v moderních osvětlovacích systémech, i když stejně jako bzučák nemohou díky svému jednoduchému ovládání vytvářet složité obrazy nebo vzory bez dalších řídicích jednotek.

LED diody GPIO jsou připojeny na pin GPIO GP1.

```
# 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 RELÉ

Relé patří k nejstarším elektromechanickým součástkám a fungují jako elektricky ovládané spínače. S malým vstupním napětím a malým proudem lze na výstupu zapínat a vypínat velkou elektrickou zátěž. Když relé sepne, rozsvítí se také červená LED dioda. Do zásuvky svorkovnice můžete zasunout odizolované konce kabelů (stisknutím oranžové páčky) a použít tři přípojky.

Relé je připojeno na pin GPIO GP28.

```
# 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

Displej z tekutých krystalů (LCD TFT) s přibližně 65 000 barvami a úhlopříčkou 1,8 palce má rozlišení 128 × 160 pixelů a lze jej ovládat prostřednictvím rozhraní SPI. Je vhodný pro zobrazování barevné grafiky a obrázků. Písmena a další znaky se zobrazují jako grafika složená z mnoha jednotlivých bodů.

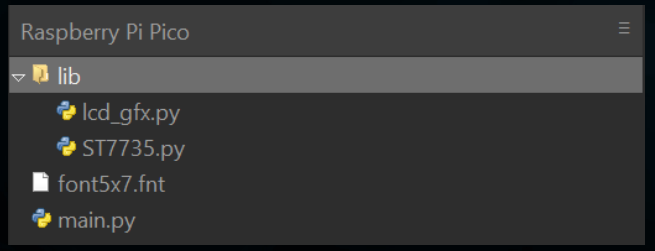
TFT je připojen k pinům GPIO GP17 (CS), GP3 (DC), GP6 (RES) a GP2 (BL).

```
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!‘)
```

Kromě textů lze zobrazit také obdélníky, například:

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

POZOR! Pro TFT displej jsou vyžadovány dva samostatné soubory knihovny a soubor písma; požadované soubory si můžete stáhnout zde. Poté přeneste všechny soubory ze složky Libraries do kořenového adresáře počítače Raspberry Pi Pico tak, aby struktura složek vypadala takto:



4.5 DHT 11

Snímač DHT11 dokáže detekovat teplotu od 0 °C do 50 °C (přesnost ±2 °C) a relativní vlhkost od 20 % do 80 % (±5 %) (maximálně jednou za sekundu). Meteorologické stanice jsou pravděpodobně hlavní oblastí použití pro čidlo, jako je DHT11. K otestování funkčnosti stačí držet ústa v blízkosti snímače a pomalu vydechovat. Vdechovaný vzduch se od okolního prostředí liší teplotou a vlhkostí, což by mělo vést k výrazné změně hodnot.

DHT11 je připojen na pin GPIO GP0.

```
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 TLAČÍTKA

Tlačítka jsou interaktivní prvky v uživatelských rozhraních, které plní jednoduchou, ale zásadní funkci: vstup uživatele. Podobně jako LED diody RGB mohou zobrazovat různé barvy, slouží tlačítka v digitálním prostředí k iniciaci široké škály příkazů a akcí.

Tlačítka jsou připojena k pinům GPIO GP10 (nahoře), GP11 (vpravo), GP14 (dole) a GP15 (vlevo).

```
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 SERVA

Servo se skládá z elektromotoru s převodovkou a řídicí elektroniky. Na výstupní straně převodovky je umístěno ozubené kolo, na kterém je namontován roh serva. Serva se používají v modelářství, například k ovládání polohy křídla nebo kormidla letadla nebo lodi. Stále více servopohonů se používá také v automobilové technice k automatickému zavírání dveří, k ovládání regulátorů oken, zrcátek a dalších nastavitelných prvků.

Připojení servopohonů jsou piny GPIO GP7, GP8, GP9 a 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 ROZHRANÍ

Připojení rozhraní hrají ve světě elektroniky klíčovou roli, podobně jako tlačítka v uživatelských rozhraních. Umožňují komunikaci a napájení mezi různými elektronickými součástmi. V oblasti rozhraní na naší desce Explorer Board proto najdete následující připojení:

SPI (Serial Perhipheral Interface): Toto připojení se používá pro rychlý sériový přenos dat. Obvykle se skládá ze čtyř linek: MISO (Master In, Slave Out), MOSI (Master Out, Slave In), SCK (Serial Clock) a SS (Slave Select). SPI je ideální pro situace, kdy je vyžadována vysoká rychlost přenosu dat, například při ovládání LCD displejů nebo SD karet.

I2C (Inter-Integrated Circuit): I2C je dvouvodičové rozhraní, které se skládá z datové linky (SDA) a hodinové linky (SCL). Běžně se používá v aplikacích mikrokontrolérů pro komunikaci mezi různými integrovanými obvody. Díky své jednoduchosti je ideální pro aplikace, kde není k dispozici mnoho pinů GPIO.

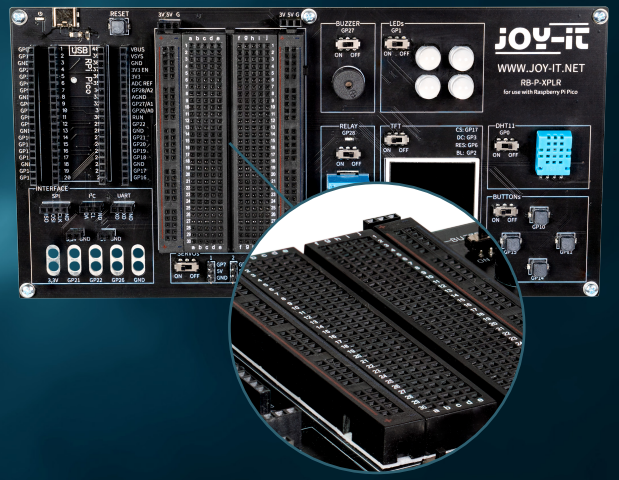
UART (Universal Asynchronous Receiver/Transmitter): Toto rozhraní umožňuje asynchronní sériovou komunikaci prostřednictvím dvou linek: TX (Transmit) a RX (Receive). UART se často používá pro komunikaci mezi mikrokontroléry a počítači nebo pro připojení modulů, jako jsou přijímače GPS nebo moduly Bluetooth.

připojení 3,3 V a 5 V: Tyto přípojky zajišťují napájení elektronických součástek. Napětí 3,3 V se často používá pro moderní mikrokontroléry a senzory, zatímco napětí 5 V se často vyskytuje ve starších nebo energeticky náročnějších zařízeních.

Přípojky pro krokosvorky: Tyto konektory jsou ideální pro dočasná připojení nebo pro testovací účely. Umožňují rychlé a snadné připojení k různým součástkám nebo měřicím zařízením bez nutnosti pájení. Na desce Explorer Board je celkem pět takových konektorů, které lze flexibilně použít pro různé aplikace.

Každé z těchto spojení má v elektronice své specifické použití a význam, podobně jako různé typy tlačítek v uživatelském rozhraní mají různé funkce. Poskytují potřebnou flexibilitu a funkčnost pro nastavení a rozšíření elektronických systémů.

4.9 BREADBOARD



Breadboardy jsou ve světě elektroniky nepostradatelným nástrojem, podobně jako jsou pro propojení různých součástek klíčové konektory rozhraní. Umožňují rychlé sestavení a testování elektronických obvodů bez pájení, a jsou tak oblíbené zejména pro prototypování a vzdělávací účely.

Chlebová deska se obvykle skládá z obdélníkového plastového bloku s velkým počtem vložených otvorů uspořádaných v řadách. Tyto otvory jsou uvnitř propojeny kovovými stopami, které umožňují snadné zapojení a připojení součástek a vodičů. Standardní uspořádání breadboardu zahrnuje dvě hlavní oblasti:

Hlavní oblasti: Ty se skládají z řady rovnoběžných řad otvorů, obvykle oddělených středovou drážkou. Otvory v jedné řadě jsou vzájemně elektricky propojeny. Toto uspořádání je ideální pro vkládání integrovaných obvodů (IC) a dalších součástek.

Napájecí lišty: Na okraji desky jsou obvykle jedna nebo dvě řady otvorů, které slouží jako napájecí lišty. Ty jsou připojeny svisle po celé délce desky a nabízejí pohodlný způsob, jak zajistit napájení a uzemnění na různých místech obvodu.

Flexibilita desky s plošnými spoji spočívá v její opakované použitelnosti a možnosti sestavovat obvody bez trvalých změn. Díky tomu je ideální pro experimentování, protože lze snadno opravovat chyby a odstraňovat součástky bez poškození. Je také vynikajícím nástrojem pro výuku, protože podporuje pochopení logiky obvodů a funkcí součástek praktickým a názorným způsobem.

Kromě toho jsou k dispozici desky s plošnými spoji různých velikostí a s různým počtem připojovacích bodů, které vyhovují různým požadavkům. Menší breadboardy jsou vhodné pro jednoduché projekty a experimenty, zatímco větší jsou vhodné pro složitější obvody.

Navzdory své univerzálnosti mají breadboardy také svá omezení. Nejsou vhodné pro velmi vysoké frekvence nebo pro obvody vyžadující vysoký výkon. Také spoje mohou být někdy méně spolehlivé než pájené spoje, zejména pokud se breadboard časem opotřebuje.

Celkově lze říci, že breadboardy jsou nezbytným nástrojem pro každého, kdo pracuje s elektronikou - od začátečníků, kteří se učí základy, až po zkušené vývojáře, kteří chtějí rychle a efektivně vytvářet prototypy. Jsou elektronickým ekvivalentem skicáku umělce: místem pro zkoumání nápadů a experimentování před vytvořením konečného díla.

5. PROJEKTY

Vítejte v kapitole o inovativních projektech elektroniky s Raspberry Pi Pico! V této kapitole se seznámíte s širokou škálou aplikací, od jednoduchého ovládání LED diod až po vývoj složitějších systémů, jako jsou automatické meteorologické stanice a dynamické osvětlovací systémy. Každý projekt je pečlivě navržen tak, abyste získali praktické zkušenosti s různými hardwarovými součástkami.

Svou cestu za poznáním začněte základními projekty, které vás naučí používat GPIO (General Purpose Input/Output) na Raspberry Pi Pico, a rozšiřte své dovednosti o pokročilejší témata, jako je ovládání servomotorů nebo použití senzorů pro monitorování životního prostředí. Pomocí komponent, jako jsou rotační snímače, ultrazvukové senzory, bzučáky a LED diody Neopixel, se naučíte navrhovat interaktivní a reaktivní systémy.

Každý projekt obsahuje podrobné seznámení s potřebnými součástmi, pokyny ke konfiguraci hardwaru krok za krokem a přehledné příklady programového kódu, které vám pomohou pochopit principy elektroniky a počítačového programování. Vysvětluje také, jak integrovat externí snímače a akční členy pro sběr a reakci na data v reálném čase.

Tyto projekty jsou nejen vzdělávací, ale také zábavné, s mnoha možnostmi přizpůsobení a rozšíření, takže můžete vyvinout vlastní kreativní řešení. Ať už jste začátečník, který právě začíná objevovat svět digitální elektroniky, nebo zkušený vývojář, který chce rozšířit své dovednosti, tato kapitola vám poskytne zdroje a inspiraci, které potřebujete ke zlepšení svých technických dovedností a zábavnému učení. Připravte se na rozšiřování svých programátorských a elektronických dovedností, protože budete provedeni jednotlivými projekty a zároveň se budete bavit při tvorbě a experimentování.

Na konci každého projektu najdete příslušné příklady kódů. Soubory si můžete stáhnout také **zde**.

5.1 ZOBRAZENÍ VZDÁLENOSTI

V našem prvním projektu je naším cílem sestrojit ultrazvukový dálkoměr, který vizualizuje vzdálenosti na našem TFT displeji. Tento projekt je skvělým úvodem do snímání a vizualizace dat pomocí počítače Raspberry Pi Pico.

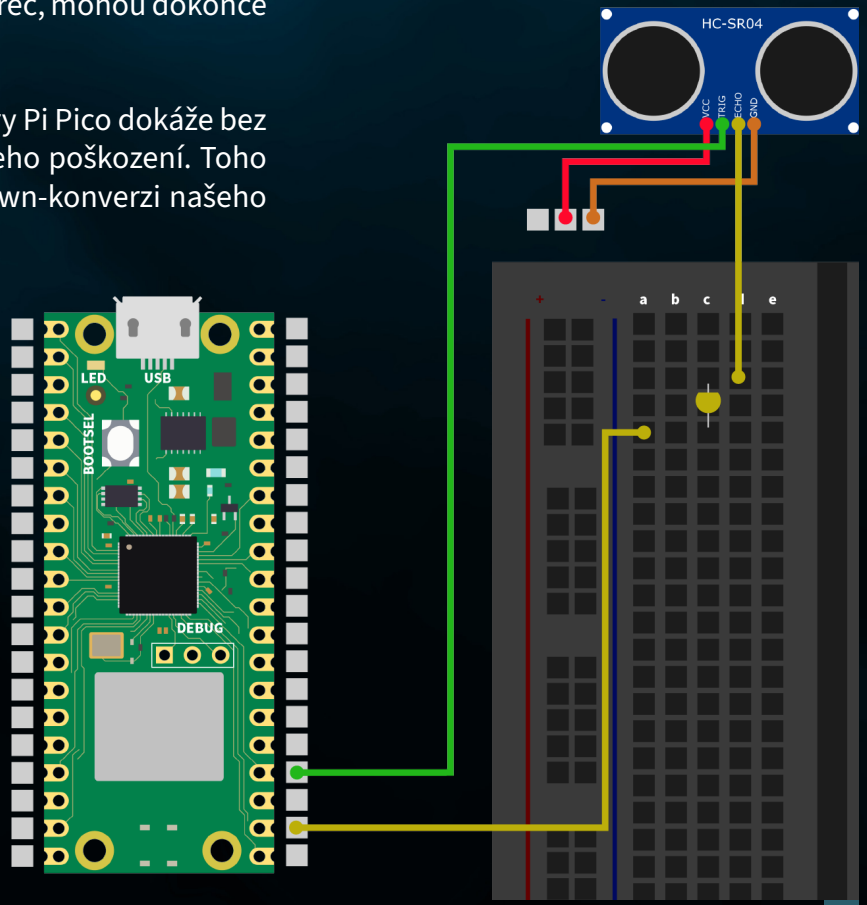
ULTRAZVUKOVÝ SENZOR: Vysílač vysílá ultrazvukovou vlnu a měří dobu, než se vlna odrazí a dorazí zpět k vysílači. Protože je známa rychlost zvuku v různých prostředích, jako je vzduch (343 m/s při 20 °C) a voda (1 484 m/s), lze vypočítat vzdálenost k odrazové ploše (poloviční doba průchodu, protože byla měřena vzdálenost tam a zpět). Impuls z mikrokontroléru na spouštěcím vstupu spustí sekvenci osmi krátkých ultrazvukových impulsů. Jakmile je signál opět přijat, výstup echa se krátce zvýší. Doba mezi spuštěním a signálem ozvěny odpovídá době průchodu. Měření vzdálenosti je možné v rozsahu přibližně od 2 cm do 400 cm a je poměrně přesné, pokud je odrazný povrch co nejtvrdší a nejrovnější. Měkké materiály, jako je například koberec, mohou dokonce měření znemožnit.

Vzhledem k tomu, že ultrazvukový senzor je čidlo, které vyžaduje napájení 5 V, ale Raspberry Pi Pico dokáže bez problémů zpracovávat pouze signály 3,3 V, je nutné snížit napětí signálu, aby nedošlo k jeho poškození. Toho dosáhneme tak, že na desce zapojíme do série naši žlutou LED diodu a použijeme ji k down-konverzi našeho signálu. LED dioda funguje také jako indikátor signálu pro aktivní echo signál.

Další podrobnosti o LED diodách najdete také v kapitole 5.5.

RASPBERRY PI PICO	ULTRAZVUKOVÝ SENZOR
3V3	VCC
GP17	Trig
GP16	Echo
GND	GND

POZOR! Pro tento projekt je nutné nastavit přepínače pro relé a DHT11 na OFF a přepínač pro TFT displej na ON.



SHRNUTÍ: V našem prvním projektu měříme vzdálenosti pomocí ultrazvukového senzoru a naměřenou vzdálenost vizualizujeme pomocí většího či menšího vyplnění grafiky na TFT displeji. V našem příkladu zcela vyplníme displej z naměřené vzdálenosti 100 cm.

```
# Load libraries
from machine import Pin, SPI
import ST7735
import time
import lcd_gfx

# Initialization of GPIO16 as input and GPIO17 as output
trig = Pin(17, Pin.OUT)
echo = Pin(16, Pin.IN, Pin.PULL_DOWN)

# 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)
backlight.high()
lcd.reset()
lcd.begin()
lcd.fill_screen(lcd.rgb_to_565(255, 255, 255))

def translate(value, leftMin, leftMax, rightMin, rightMax):
    # Figure out how 'wide' each range is
    leftSpan = leftMax - leftMin
    rightSpan = rightMax - rightMin

    # Convert the left range into a 0-1 range (float)
    valueScaled = float(value - leftMin) / float(leftSpan)

    # Convert the 0-1 range into a value in the right range.
    return rightMin + (valueScaled * rightSpan)

# Endless loop for measuring the distance
while True:
     # Distance measurement is started using the 10us trigger signal
     trig.value(0)
     time.sleep(0.1)
     trig.value(1)

     # Now wait at the echo input until the signal has been activated
     # Then the time is measured for how long it remains activated
     time.sleep_us(2)
     trig.value(0)
     while echo.value()==0:
          pulse_start = time.ticks_us()
     while echo.value()==1:
          pulse_end = time.ticks_us()
     pulse_duration = pulse_end - pulse_start
```

Inicializace ultrazvukového senzoru a TFT displeje

Pomocná funkce pro nastavení rozsahu hodnot

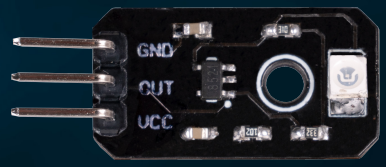
Měření vzdálenosti

```
# Now the distance is calculated using the recorded time
distance = pulse_duration * 17165 / 1000000
distance = round(distance, 0)

# Serial output
print ('Distance:',"{:.0f}".format(distance),'cm')
time.sleep(1)

# Adjust measured value to LCD height
if(distance > 100):
    distance = 100
drawHeight = round(translate(distance, 0, 100, 0, 160))

# Fill the TFT display
lcd.fill_screen(lcd.rgb_to_565(255, 255, 255))
lcd.draw_block(0, 0, 128, drawHeight, lcd.rgb_to_565(0, 255, 0))
```

Výpočet rozsahu hodnot a popis displeje TFT

5.2 METEOROLOGICKÁ STANICE

Ponořte se do druhého projektu našeho elektronického dobrodružství, ve kterém si vytvoříte vlastní meteorologickou stanici! Tento projekt kombinuje použití UV senzoru s čidlem teploty a vlhkosti DHT11, které vám poskytne nejen přehled o aktuálním počasí, ale také změří UV záření na vašem místě. Všechny tyto informace se přehledně zobrazují na barevném TFT displeji, takže počasí a UV záření vidíte na první pohled.



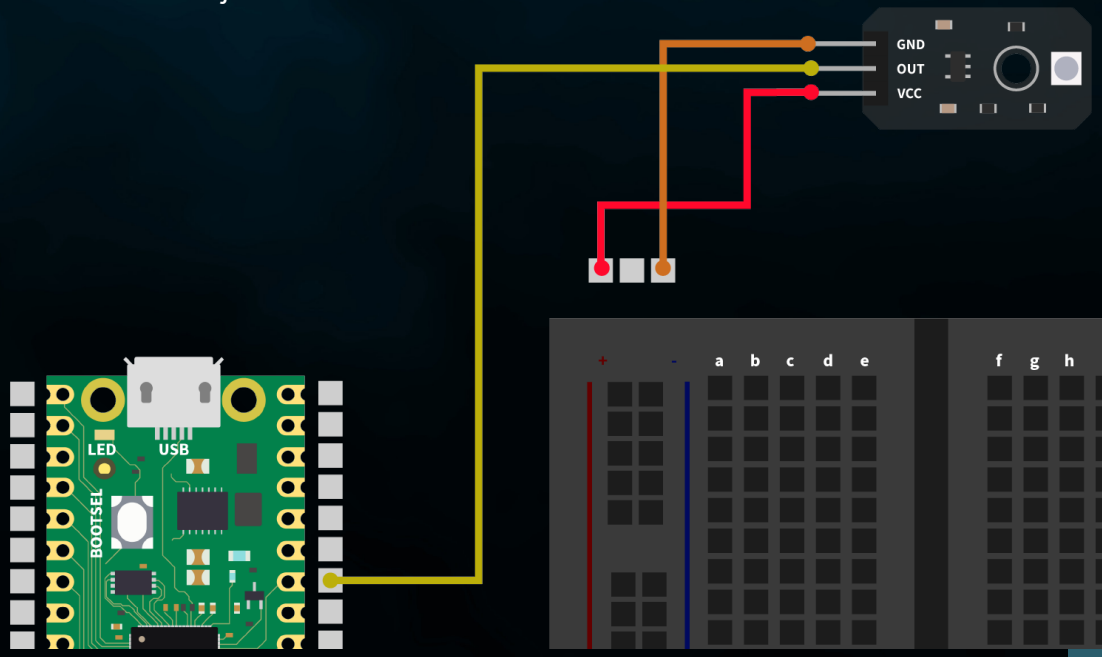
UV SENZOR: UV senzor je malá součástka, která nám pomáhá měřit neviditelné ultrafialové (UV) sluneční záření. UV záření je část slunečního záření, která je neviditelná, ale může mít vliv na naši pokožku a zdraví. Vzpomeňte si na spálení nebo opálení kůže - obojí je způsobeno UV zářením.

Jádro snímače obsahuje materiál, který reaguje na UV záření. Když UV paprsky na tento materiál dopadnou, senzor změní svůj elektrický odpor. Tuto změnu senzor převede na signál, který můžeme změřit a odečíst. Pomocí počítače Raspberry Pi Pico pak můžeme tento signál převést na hodnotu, která udává, jak silné je aktuálně UV záření.

Nejprve připojte UV senzor k počítači Raspberry Pi Pico pomocí přiložených kabelů. Ačkoli jej samozřejmě můžete připojit i k chlebové destičce, přímé kabelové připojení je zde mnohem flexibilnější.

RASPBERRY PI PICO	UV SENZOR
GND	GND
GP28	OUT
3V3	VCC

POZOR! Pro tento projekt je nutné nastavit přepínač pro relé na OFF a přepínače pro TFT displej a senzor DHT11 na ON.



SHRNUTÍ: Naše meteorologická stanice snímá údaje ze senzorů DHT11 a UV a zobrazuje je na TFT displeji.

```
from machine import ADC, Pin, SPI
import utime
import dht
import ST7735  # Assuming this is the library for your TFT display

# Initialize DHT11 sensor
sensor_dht11 = dht.DHT11(Pin(0))

# Initialize UV sensor
uv_sensor = ADC(2)  # Assuming GP28 is ADC pin number 1 in your configuration

# 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=Pin(6), ce=Pin(17), dc=Pin(3))
backlight = Pin(2, Pin.OUT)
backlight.high()
lcd.reset()
lcd.begin()
lcd.fill_screen(lcd.rgb_to_565(255, 255, 255))

while True:
    lcd.fill_screen(lcd.rgb_to_565(255, 255, 255))

    # Read UV value
    uv_value = uv_sensor.read_u16()
    # Conversion in percent
    uv_percent = (uv_value / 65000) * 100
    print("UV Intensity (percent):", uv_percent)

    # DHT11 Read values
    sensor_dht11.measure()
    temp = sensor_dht11.temperature()
    humid = sensor_dht11.humidity()

    # Display values on LCD
    lcd.p_string(20, 20, "Temp: {}C".format(temp))
    lcd.p_string(20, 40, "Humid: {}%".format(humid))
    lcd.p_string(20, 60, "UV: {:.2f}%".format(uv_percent))  # Display of UV
intensity in percent with two decimal places

    utime.sleep(10)
```

Inicializace displeje TFT

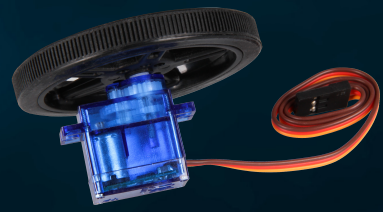
Měření hodnot snímačů

Výstup na displeji

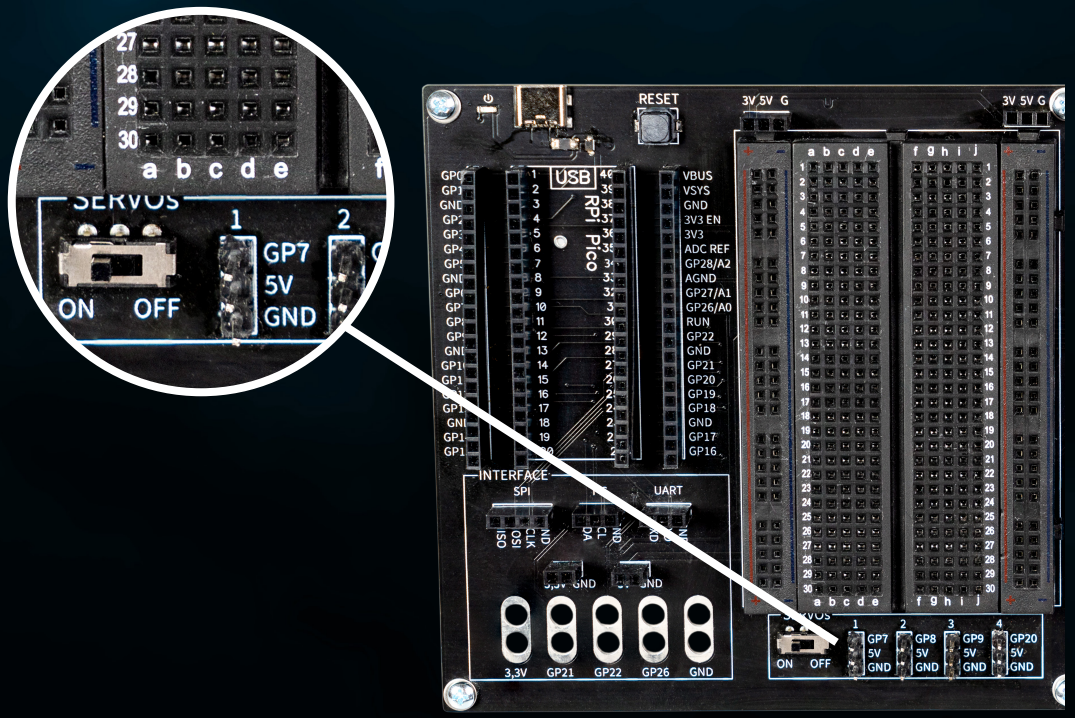
5.3 SERVOŘÍZENÍ

Vítejte u třetího projektu z naší série vzrušujících dobrodružství s elektronikou pomocí sady Explorer! Tentokrát je vše o pohybu a ovládání. Naším cílem je naprogramovat a ovládat servomotor tak, aby bylo možné řídit směr jeho otáčení pouhým stisknutím tlačítka. Tento projekt je nejen vynikajícím úvodem do světa ovládání motorů, ale také ukazuje, jak posílit interakci pomocí vizuální zpětné vazby na TFT displeji.

SERVOMOTOR: Servo se skládá z elektromotoru s převodovkou a řídicí elektroniky. Na výstupní straně převodovky je umístěno ozubené kolo, na kterém je namontováno kolo serva. Serva se používají v modelářství, například k ovládání polohy křídla nebo kormidla letadla nebo lodi. Stále více servopohonů se používá také v automobilové technice k automatickému zavírání dveří, k ovládání regulátorů oken, zrcátek a dalších nastavitelných prvků.



Nejprve připojte servomotor k servorozhraní s číslem 1 na desce Explorer.



POZOR! Pro tento projekt je nutné nastavit přepínač pro TFT displej, tlačítka a serva na ON.

SHRNUTÍ: Ovládáme náš servomotor a necháme ho přepínat mezi levou a pravou rotací, ovládanou našimi tlačítky.

Inicializace servomotoru a tlačítek

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

# Servo pin numbers
servoOnePin = 7

# Key pin numbers
buttonLeftPin = 15
buttonRightPin = 11

# Initialization of the servo
servoOne = PWM(Pin(servoOnePin))

# Initialize the buttons with PULL_UP to use the default HIGH state
buttonLeft = Pin(buttonLeftPin, Pin.IN, Pin.PULL_UP)
buttonRight = Pin(buttonRightPin, Pin.IN, Pin.PULL_UP)

# Servo speeds in nanoseconds
leftSpeed = 1300000  # Moves the servo to the left
rightSpeed = 1700000  # Moves the servo to the right

# Servo frequency
servoOne.freq(50)  # Typical servo frequency of 50Hz

# Status of the servo
servoState = 'left'  # Starts with counterclockwise rotation

# Last state of the buttons to recognize edges
lastButtonLeft = buttonLeft.value()
lastButtonRight = buttonRight.value()

while True:
    # Read out the current status of the buttons
    currentButtonLeft = buttonLeft.value()
    currentButtonRight = buttonRight.value()

    # Check whether an edge from HIGH to LOW was detected (button was pressed)
    if lastButtonLeft == 1 and currentButtonLeft == 0:
        servoState = 'right'  # Changes the direction to the right when the left
button is pressed
    elif lastButtonRight == 1 and currentButtonRight == 0:
        servoState = 'left'  # Changes the direction to the left when the right
button is pressed
```

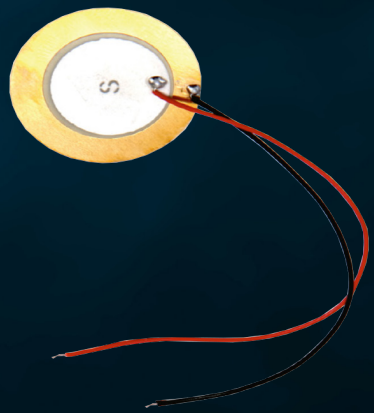
Kontrola tlačítek

```
    # Update the states for the next run
    lastButtonLeft = currentButtonLeft
    lastButtonRight = currentButtonRight

    # Controls the servo based on the current status
    if servoState == 'left':
        servoOne.duty_ns(leftSpeed)  # Moves the servo to the left
    else:
        servoOne.duty_ns(rightSpeed)  # Moves the servo to the right

    sleep(0.1)  # Short break for the control cycle
```

Servořízení

5.4 VLASTNORUČNĚ VYROBENÝ BZUČÁK

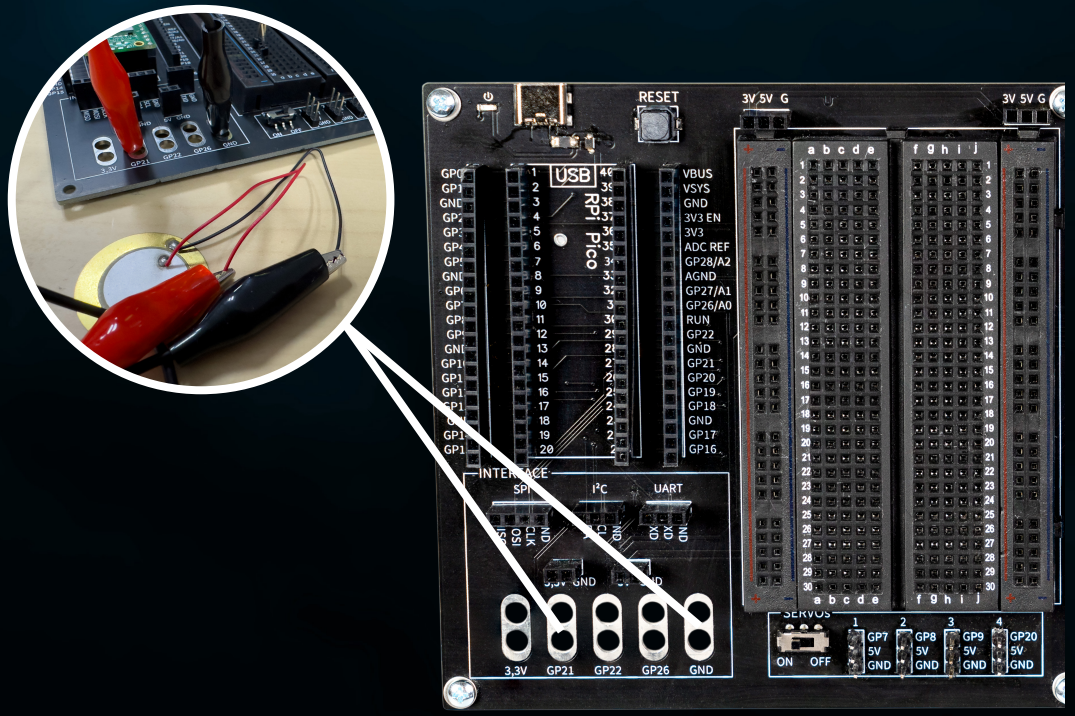
Čtvrtý projekt našeho dobrodružství s elektronikou pomocí sady Explorer je o zvuku! Ponoříme se do světa akustických signálů a vytvoříme si vlastní obvod bzučáku. Tento projekt vám nejen umožní pochopit základy tvorby obvodů, ale také se naučit vytvářet zvukové signály pomocí jednoduchých nástrojů. Díky použití aligátorových svorek je sestavení obzvláště uživatelsky přívětivé a přístupné i pro ty, kteří jsou teprve na začátku své cesty za elektronikou.



Nejprve bzučák opatrně připojíme k desce Explorer Board pomocí krokosvorek. Tyto svorky jsou ideální pro rychlé a flexibilní připojení bez nutnosti pájení. Bzučák, malá součástka schopná vydávat zvuk, se stane ústředním prvkem našeho projektu. Po připojení proudu bzučák vibruje a vydává zvuk.

Nejprve připojte aligátorovou svorku ke konektoru GP21 aligátorové svorky na desce Explorer. Druhý konec krokosvorky připojte k červenému kabelu bzučáku. Připojte další krokosvorku ke krokosvorkovému konektoru GND na desce Explorer. Druhý konec svorky připojte k černému kabelu bzučáku.

RASPBERRY PI PICO	BZUČÁK
GP21	Červený kabel
GND	Černý kabel



POZOR! Pro tento projekt je nutné nastavit přepínač tlačítek do polohy ON.

SHRNUTÍ: K počítači Raspberry Pi Pico připojíme externí bzučák a přehrajeme jednoduchou zábavnou melodii.

```
from machine import Pin, PWM
import utime
# Note frequencies (in Hz)
notes = {
    'C4': 262,
    'D4': 294,
    'E4': 330,
    'F4': 349,
    'G4': 392,
    'A4': 440,
    'B4': 494,
    'C5': 523
}
# Melody and duration (in ms)
melody = ['C4', 'D4', 'E4', 'C4', 'C4', 'D4', 'E4', 'C4', 'E4', 'F4', 'G4', 'E4',
'F4', 'G4']
durations = [500, 500, 500, 500, 500, 500, 500, 500, 500, 500, 1000, 500, 500,
1000]
# Initialization of the buzzer
buzzer = PWM(Pin(21))
buzzer.freq(440) # Set a start frequency
# Function to play a note
def play_note(note, duration):
    if note in notes:
        buzzer.freq(notes[note]) # Set the frequency based on the note
        buzzer.duty_u16(32767) # Start the PWM signal
        utime.sleep_ms(duration) # Hold the note for the duration
    buzzer.duty_u16(0) # Stop the PWM signal (switch off note)
    utime.sleep_ms(50) # Short pause between the notes
# Play the melody
for note, duration in zip(melody, durations):
    play_note(note, duration)
buzzer.deinit() # Deactivate the PWM channel when finished
```

Seznam poznámek

Paměť Melody

Funkce pro přehrávání not

5.5 VÁŠ VLASTNÍ OKRUH

V pátém projektu našeho elektronického dobrodružství se sadou Explorer prozkoumáme fascinující svět ovládání světla. Tentokrát sestavíme obvod, který můžete použít k ovládání LED diod. To poskytuje fantastickou příležitost pochopit principy elektronických obvodů a zároveň získat kontrolu nad světelnými efekty.

LEDS: LED diody jsou malé, ale výkonné zdroje světla, které se používají v mnoha elektronických projektech. Oproti klasickým žárovkám mají mnoho výhod, například delší životnost, nižší spotřebu energie a možnost svítit různými barvami. LED dioda se skládá z polovodičového materiálu, který vyzařuje světlo, když jím protéká elektrický proud.

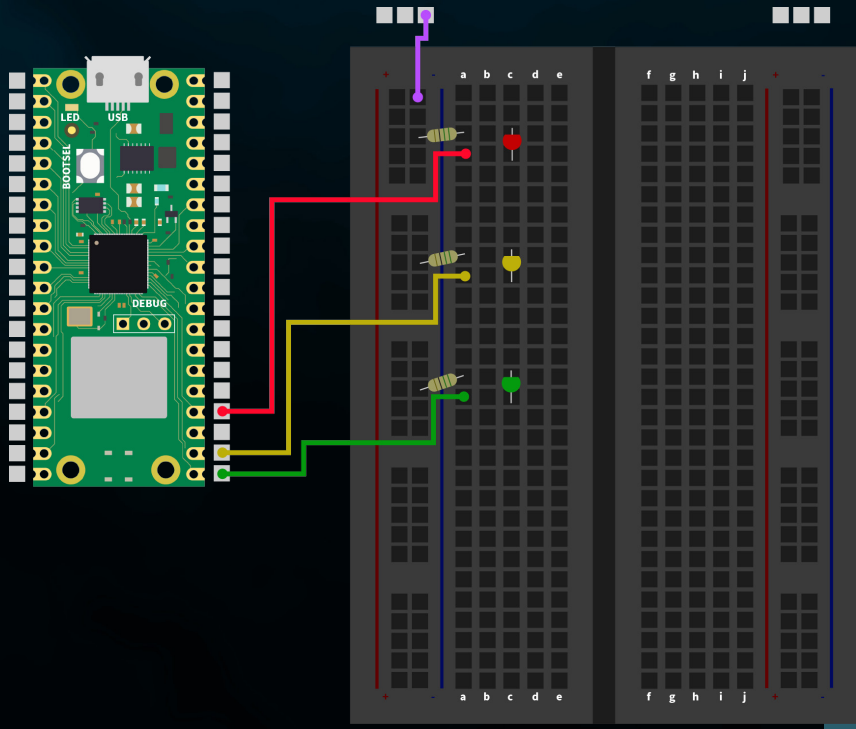
Je důležité rozpoznat správnou polaritu LED diod, protože fungují pouze tehdy, pokud jimi protéká proud správným směrem. To znamená, že kladný pól zdroje napájení musí být připojen ke kladnému konci LED a záporný pól zdroje napájení musí být připojen k zápornému konci LED.

Takto se pozná polarita LED diody:

U většiny LED je delší noha anoda (+), tj. kladné připojení. Kratší noha je katoda (-), tj. záporné připojení.

Plochý okraj: Na straně krytu LED může být plochý okraj. Tato strana obvykle označuje katodu, tj. záporný pól.

Nejprve zrekonstruujte obvod podle následujícího schématu. Ujistěte se však, že polarita LED diod je správná. Pro každou LED použijte také sériový rezistor 56 Ω (zeleno-modro-černý).



RASPBERRY PI PICO	LEDS
GP18	Červená LED dioda
GP17	Žlutá LED dioda
GP16	Zelená LED dioda

POZOR! Pro tento projekt je nutné nastavit přepínač pro TFT displej do polohy OFF.

SHRNUTÍ: Prostřednictvím pinů počítače Raspberry Pi Pico ovládáme tři různé LED diody, přičemž každá z nich střídavě bliká.

Inicializace LED diod

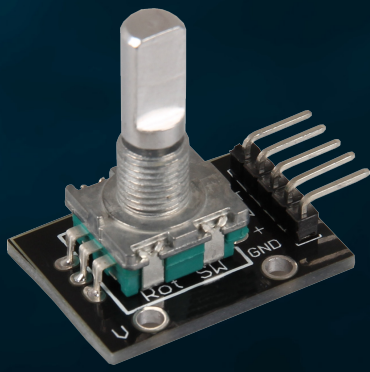
Funkce blikání LED

Hlavní smyčka

```
from machine import Pin
import utime

# Initialize the LEDs
red_led = Pin(18, Pin.OUT)
yellow_led = Pin(17, Pin.OUT)
green_led = Pin(16, Pin.OUT)

# Flashing function for one LED
def blink_led(led, duration):
    led.value(1)  # Switch on LED
    utime.sleep(duration)
    led.value(0)  # Switch off LED
    utime.sleep(duration)

# Hauptschleife
while True:
    blink_led(red_led, 0.5)    # Red LED flashes for 0.5 seconds
    blink_led(yellow_led, 0.5) # Yellow LED flashes for 0.5 seconds
    blink_led(green_led, 0.5)  # Green LED flashes for 0.5 seconds
```

5.6 OVLÁDÁNÍ LED

V šestém projektu našeho dobrodružství s elektronikou používáme otočný snímač k ovládání jasu a barvy LED diod - je to jednoduchý, ale fascinující způsob, jak se ponořit do elektroniky. Otočný enkodér, náš ústřední ovládací prvek, umožňuje díky své dvojí funkci hravou interakci. Změny jasu se ovládají otáčením, zatímco stisknutím enkodéru se cyklicky mění barvy LED diod - ideální pro ilustraci základů elektroniky a míchání barev.



ROTAČNÍ SNÍMAČ: Rotační snímač je chytré malé zařízení, které převádí rotační pohyby na elektronické signály. Představte si otočný knoflík, jaký znáte z rádia. Když tímto knoflíkem otáčíte, rotační snímač dokáže změřit, jak daleko a jakým směrem jste jím otočili. Tuto informaci pak můžete použít například ke změně hlasitosti, procházení nabídek nebo v našich projektech k nastavení jasu LED diod.

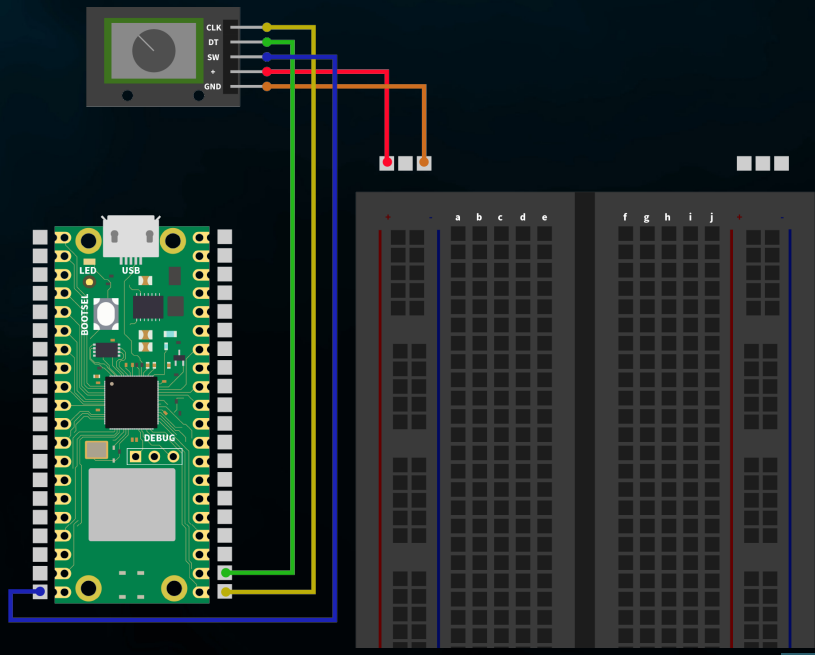
Otočné snímače mají často také zabudované tlačítko - to znamená, že mohou fungovat také jako tlakový spínač. Když stisknete otočný knoflík, rotační snímač rozpozná tento tlak jako samostatný signál. Toho lze využít k různým funkcím, například k zapnutí a vypnutí zařízení nebo ke změně provozních režimů.

SHRNUTÍ: Otočný snímač umožňuje posílat různé příkazy projektům elektroniky otáčením a stisknutím. Jedná se o intuitivní a všestranný nástroj, díky kterému je interakce s vašimi projekty snadná a zábavná.

Nejprve připojte otočný snímač k počítači Raspberry Pi Pico následujícím způsobem:

RASPBERRY PI PICO	ROTAČNÍ SNÍMAČ
GP16	CLK
GP17	DT
GP15	SW
3V3	+
GND	GND

POZOR! Pro tento projekt je nutné nastavit přepínač pro TFT displej na OFF a přepínač pro LED diody na ON.



SHRNUTÍ: K ovládání barvy a jasu našich čtyř diod LED používáme otočný snímač. Otáčením enkodéru měníme jas, zatímco stisknutím enkodéru nastavujeme barvu diod LED.

Inicializace LED diod a otočného ovladače

```
from machine import Pin
import utime
import neopixel

# Neopixel setup
NUM_LEDS = 4
PIXEL_PIN = 1
np = neopixel.NeoPixel(Pin(PIXEL_PIN), NUM_LEDS)

# Rotate encoder setup
PIN_CLK = Pin(16, Pin.IN, Pin.PULL_UP)
PIN_DT = Pin(17, Pin.IN, Pin.PULL_UP)
BUTTON_PIN = Pin(15, Pin.IN, Pin.PULL_UP)

# Global variables
counter = 0
PIN_CLK_LAST = PIN_CLK.value()
delayTime = 0.001
debounce_time_encoder = 0
debounce_time_button = 0

# Initialize colors
colors = [(255, 0, 0), (0, 255, 0), (0, 0, 255), (255, 255, 255)]  # Rot, Grün,
Blau, Weiß
color_index = 0

# Initialize brightness
brightness_levels = [0.2, 0.4, 0.6, 0.8, 1.0]
brightness_index = 0

def update_leds(color, brightness):
    dimmed_color = tuple([int(c * brightness) for c in color])
    for i in range(NUM_LEDS):
        np[i] = dimmed_color
    np.write()

def rotaryFunction(null):
    global counter, brightness_index, debounce_time_encoder
    PIN_CLK_CURRENT = PIN_CLK.value()
    if PIN_CLK_CURRENT != PIN_CLK_LAST and (utime.ticks_ms() - debounce_time_
encoder) > 300:
        if PIN_DT.value() != PIN_CLK_CURRENT:
            brightness_index = (brightness_index + 1) % len(brightness_levels)
        else:
            brightness_index = (brightness_index - 1) % len(brightness_levels)
        update_leds(colors[color_index], brightness_levels[brightness_index])
        debounce_time_encoder = utime.ticks_ms()
```

Funkce pro rotační snímač

```
def counterReset(null):
    global color_index, debounce_time_button
    if (utime.ticks_ms() - debounce_time_button) > 300:
        color_index = (color_index + 1) % len(colors)
        update_leds(colors[color_index], brightness_levels[brightness_index])
        debounce_time_button = utime.ticks_ms()



PIN_CLK.irq(trigger=Pin.IRQ_FALLING | Pin.IRQ_RISING, handler=rotaryFunction)
BUTTON_PIN.irq(trigger=Pin.IRQ_FALLING, handler=counterReset)

update_leds(colors[color_index], brightness_levels[brightness_index])

while True:
    utime.sleep(delayTime)
```

Seznam poznámek

5.7 AUTOMATICKÉ ŘÍZENÍ JASU

V sedmém projektu našeho dobrodružství s elektronikou používáme fotodiodu k automatickému řízení jasu LED diod. Fotodioda převádí světlo na elektrický signál, takže LED diody svítí jasněji, když je tma, a tlumeněji, když je více okolního světla.

Připojením fotodiody k desce Explorer Board a jejím naprogramováním na Raspberry Pi Pico se LED diody inteligentně přizpůsobí okolnímu jasu. Tento projekt ukazuje, jak lze pomocí jednoduchých součástek vytvořit reaktivní a energeticky úsporné elektronické systémy.

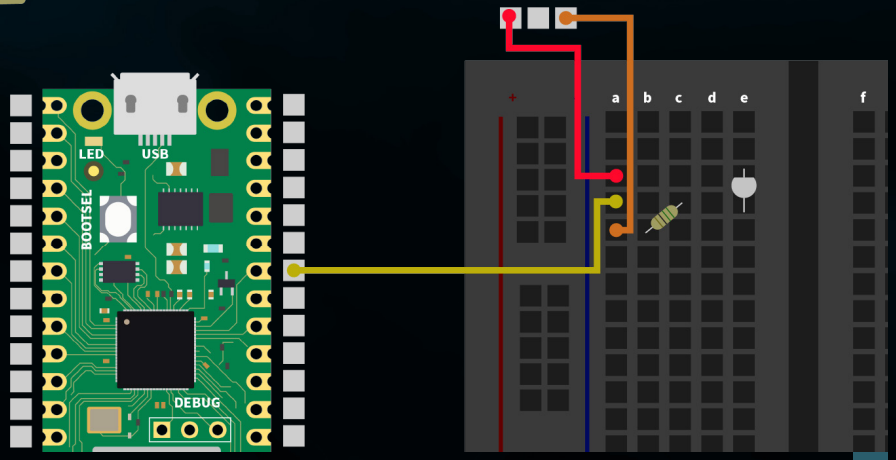
FOTODIODA: Fotodioda je speciální typ polovodiče, který reaguje na dopadající světlo generováním elektrického proudu. Představte si fotodiodu jako malý solární panel: když na ni dopadá světlo, přemění ho na elektrický signál. Čím více světla na fotodiodu dopadá, tím je signál silnější.

Fotodiody jsou velmi citlivé a dokáží detekovat i malé množství světla, což je předurčuje k projektům, kde je důležité měřit jas nebo přítomnost světla. Mohou být například použity v automatických regulátorech jasu, světelných senzorech nebo jako součást systému řízení osvětlení.

Fotodiody jsou zkrátka účinné detektory světla, které nám umožňují, aby elektronická zařízení inteligentně reagovala na změny okolního osvětlení.

Nejprve připojte fotodiodu přes destičku následujícím způsobem. Upozorňujeme, že i zde je nutné použít rezistor. Zde použijte rezistor 100 kΩ (hnědo-černo-žlutý).

RASPBERRY PI PICO	FOTODIODA
3V3	Kladná katoda
GP28/A2	Záporná anoda



POZOR! Pro tento projekt je nutné nastavit přepínač pro relé na OFF a přepínač pro LED diody na ON.

SHRNUTÍ: Pomocí fotodiody měříme jas okolí a nastavujeme jas čtyř LED diod. Intenzita LED diod se mění podle světla detekovaného fotodiodou, přičemž tmavší prostředí vede k jasnějším LED diodám a naopak.
Pro dosažení nejlepšího možného výsledku je nejlepší použít svítilnu.

```
from machine import Pin, ADC
import neopixel
import utime

# Neopixel setup
NUM_LEDS = 4
PIXEL_PIN = 1
np = neopixel.NeoPixel(Pin(PIXEL_PIN), NUM_LEDS)

# Photodiode setup on ADC pin GP28 (A2)
fotodiode = ADC(2)

# Conversion function for brightness values of the photodiode into a suitable
brightness for the LEDs
def brightness_from_light(sensor_value):
    # Minimum and maximum sensor value
    min_sensor_value = 400
    max_sensor_value = 10000

    # Invert the sensor value within the actual range
    normalized_value = max_sensor_value - sensor_value + min_sensor_value

    # Scale the inverted value to a brightness range (0.05 to 0.5)
    # Adjust the scaling: Divide by (max_sensor_value - min_sensor_value)
    return max(0.05, min(0.5, normalized_value / (max_sensor_value - min_sensor_
value) * 0.45 + 0.05))

def update_leds(brightness):
    color = (255, 255, 255)  # White
    dimmed_color = tuple([int(c * brightness) for c in color])
    for i in range(NUM_LEDS):
        np[i] = dimmed_color
    np.write()


while True:
    # Read the sensor value from the photodiode
    light_value = fotodiode.read_u16()
    print(light_value)

    # Calculate the brightness based on the sensor value
    brightness = brightness_from_light(light_value)

    # Update the LEDs with the new brightness
    update_leds(brightness)

    # Waiting time to reduce the load on the CPU and for smoother brightness
transitions
    utime.sleep(0.5)
```

Inicializace LED a fotodiody

Měření fotodiody a řízení jasu

5.8 OVLÁDÁNÍ RGB LED

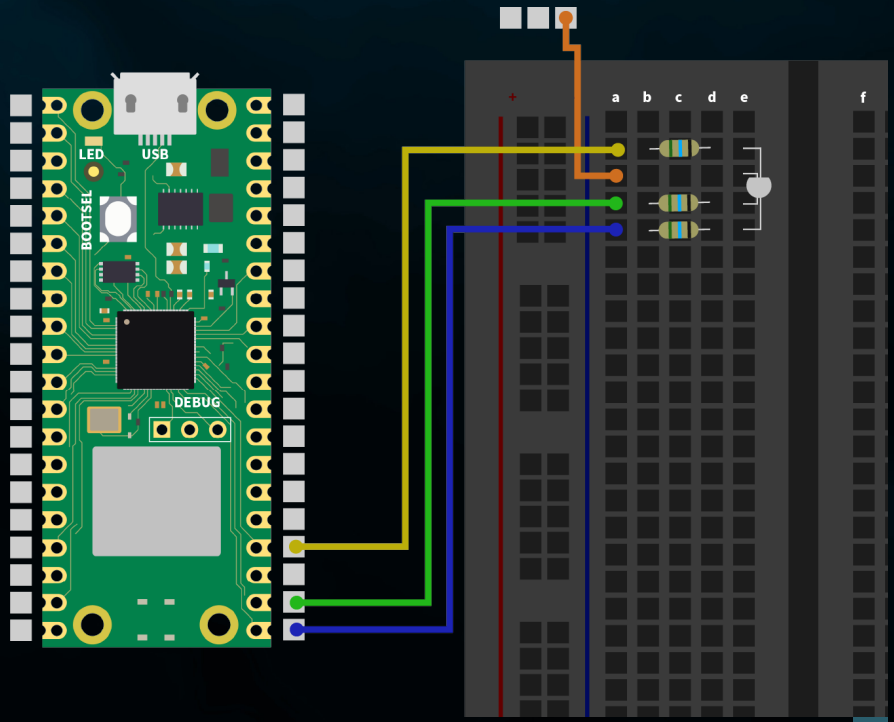
V osmém a posledním projektu našeho seriálu o elektronice se zaměříme na ovládání barev RGB LED pomocí integrovaných tlačítek na desce Explorer Board. LED diody RGB jsou speciální světelné diody, které kombinují červené, zelené a modré (RGB) světlo a zobrazují širokou škálu barev. Individuálním nastavením intenzity každé barevné složky můžeme vytvořit téměř jakoukoli barvu.

V tomto projektu připojíme RGB LED diodu k desce plošných spojů a pomocí stávajících tlačítek budeme ovládat barvy LED diody. Každému tlačítku je přiřazena jedna barva (červená, zelená, modrá).

RGB LED: LED dioda RGB kombinuje červenou, zelenou a modrou barvu v jednom světelném bodě. Změnou jasu každé z těchto tří barev lze vytvořit téměř jakoukoli barvu. K tomu slouží pulzně šířková modulace (PWM), která řídí intenzitu jednotlivých barev. LED diody RGB s pouhými třemi barvami tak umožňují široké barevné spektrum, které je ideální pro projekty barevného osvětlení.

Nejprve připojte LED diodu RGB k desce s plošnými spoji následujícím způsobem. Všimněte si, že každý ze tří barevných kanálů zde také vyžaduje sériový rezistor. Zde byste měli použít rezistor 56 Ω (zeleno-modro-černý).

RASPBERRY PI PICO	RGB-LED
GP18	První kolík
GND	Druhý kolík
GP17	Třetí kolík
GP16	Čtvrtý kolík



POZOR! Pro tento projekt je nutné nastavit přepínač pro TFT na OFF a pro tlačítka na ON.

SHRNUTÍ: Tři barevné kanály RGB LED (červená, zelená a modrá) se zapínají a vypínají pomocí tlačítek (vlevo, nahoře a vpravo).

```
from machine import Pin
import utime

# Initialize the LED pins
red_led = Pin(18, Pin.OUT)
green_led = Pin(17, Pin.OUT)
blue_led = Pin(16, Pin.OUT)

# Initialize the button pins
button_red = Pin(15, Pin.IN, Pin.PULL_UP)
button_green = Pin(10, Pin.IN, Pin.PULL_UP)
button_blue = Pin(11, Pin.IN, Pin.PULL_UP)

# Save states of the LEDs
red_state = False
green_state = False
blue_state = False

def toggle_led(led, state):
    led.value(state)

while True:
    # Check the status of the red button
    if button_red.value() == 0:
        red_state = not red_state
        toggle_led(red_led, red_state)
        utime.sleep(0.2)  # Entprellung

    # Check the status of the green button
    if button_green.value() == 0:
        green_state = not green_state
        toggle_led(green_led, green_state)
        utime.sleep(0.2)  # Entprellung

    # Check the status of the blue button
    if button_blue.value() == 0:
        blue_state = not blue_state
        toggle_led(blue_led, blue_state)
        utime.sleep(0.2)  # Debouncing
```

Inicializace LED a tlačítek

Testování tlačítek a ovládání LED diody

6. INFORMAČNÍ POVINNOSTI A POVINNOSTI ZPĚTNÉHO ODBĚRU

NAŠE INFORMAČNÍ POVINNOSTI A POVINNOSTI ZPĚTNÉHO ODBĚRU PODLE NĚMECKÉHO ZÁKONA O ELEKTRICKÝCH A ELEKTRONICKÝCH ZAŘÍZENÍCH (ELEKTROG)

SYMBOL NA ELEKTRICKÝCH A ELEKTRONICKÝCH ZAŘÍZENÍCH: 
Tento přeškrtnutý odpadkový koš znamená, že elektrické a elektronické spotřebiče nepatří do domovního odpadu. Staré spotřebiče musíte odevzdat na sběrném místě. Před odevzdáním musíte oddělit použité baterie a akumulátory, které nejsou přiloženy ke starému spotřebiči.

MOŽNOSTI VRÁCENÍ:
Jako koncový uživatel můžete při nákupu nového spotřebiče bezplatně odevzdat svůj starý spotřebič (který v podstatě plní stejnou funkci jako nový spotřebič zakoupený u nás) k likvidaci. Malé spotřebiče, jejichž vnější rozměry nejsou větší než 25 cm, můžete likvidovat v běžném množství v domácnosti bez ohledu na to, zda jste si zakoupili nový spotřebič.

MOŽNOST VRÁCENÍ V SÍDLE NAŠÍ SPOLEČNOSTI BĚHEM OTEVÍRACÍ DOBY:
SIMAC Electronics GmbH, Pascalstr. 8, D-47506 Neukirchen-Vluyn

MOŽNOST VRÁCENÍ VE VAŠÍ OBLASTI:
Zašleme vám známku na balík, s níž nám můžete zařízení bezplatně vrátit. Za tímto účelem nás prosím kontaktujte e-mailem na adrese service@joy-it.net nebo telefonicky.

INFORMACE O BALENÍ:
Starý spotřebič pro přepravu bezpečně zabalte. Pokud nemáte vhodný obalový materiál nebo nechcete použít svůj vlastní, kontaktujte nás a my vám zašleme vhodný obal.

7. PODPORA

Jsme tu pro vás i po nákupu. V případě nezodpovězených otázek nebo problémů jsme vám k dispozici také prostřednictvím e-mailu, telefonu a systému ticketové podpory.

E-Mail: service@joy-it.net
Ticket-System: http://support.joy-it.net
Telefon: +49 (0)2845 9360 – 50 (Po - Čt: 09:00 - 17:00 nebo hodiny, Pá: 09:00 - 14:30 nebo hodiny)

Další informace naleznete na našich webových stránkách:

WWW.JOY-IT.NET