

---

# **Dětská univerzita ČVUT**

## **14.7.2023**

### **Mikropočítač Raspberry PI PICO a jeho programování grafickou metodou ČVUT- FEL**

Prezentující: doc. Fischer, Dr. Petrucha, katedra měření ČVUT - FEL

# Náplň

---

**Mikropočítač Raspberry PI PICO –**

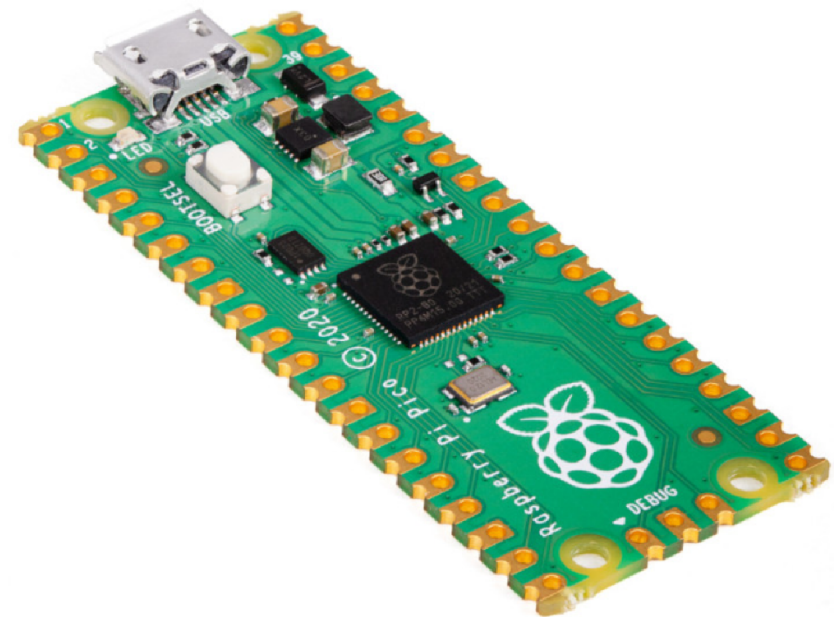
**co to je a jak s ním jednoduše začít pracovat**

**Programování na základě Micropython s využitím nadstavby pro  
grafické programování BIPES**

**První kroky**

**Naše sestava mikropočítače na nepájivém kontaktním poli**

**Budeme si jednotlivé možnosti  
vykládat postupně a také zkoušet**



# Mikropočítač - Raspberry PI PICO

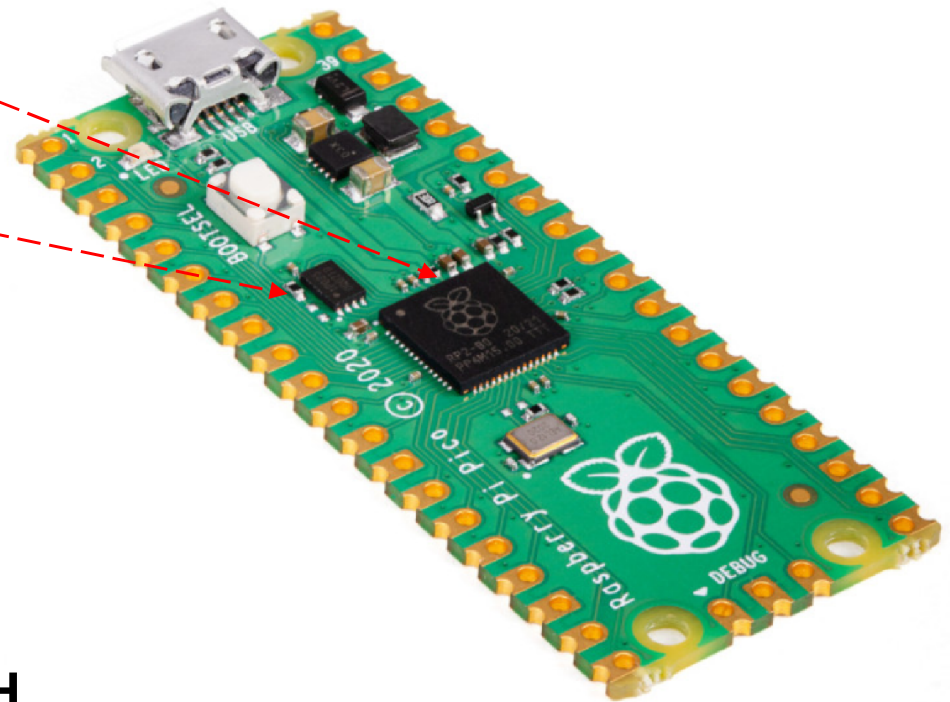
**Mikropočítač Raspberry PI PICO**

**S procesorovým obvodem RP2040**

**2 Mbyte externí FLASH**

**Cena cca 120 Kč s DPH**

**Informace a odkazy**



**Vlastní procesorový obvod RP2040**

**Dvě 32- bitová jádra ARM Cortex M0+**

**264 kByte SRAM, nemá interní FLASH**

**Hodinový signál procesoru- 133 MHz**

**Rozhraní: USB, UART, I<sup>2</sup>C BUS, SPI**

**Převodník ADC 12 bitový. Až 500 000 vzorků za sekundu**

**[https://embedded.fel.cvut.cz/procesory/RP\\_PICO](https://embedded.fel.cvut.cz/procesory/RP_PICO)**

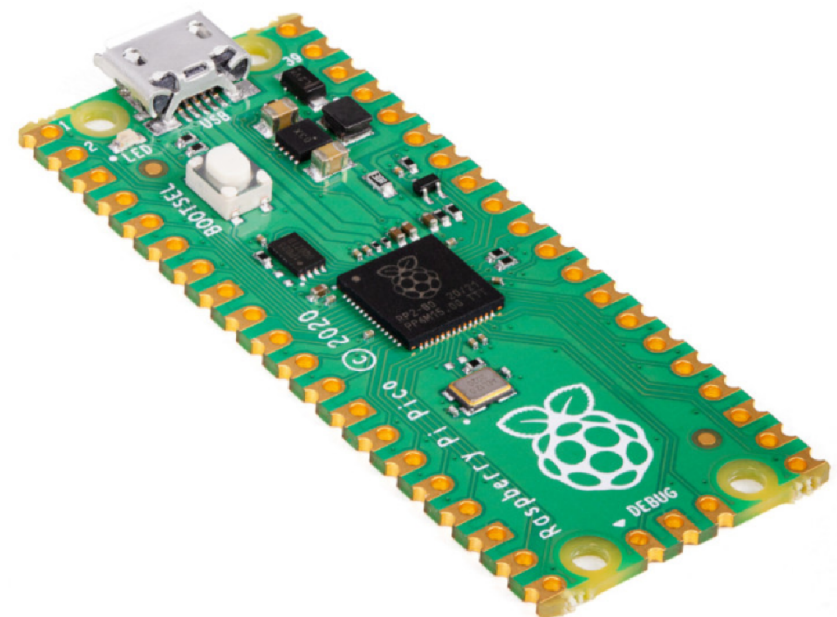
▪  
Informace

[https://embedded.fel.cvut.cz/procesory/RP\\_PICO](https://embedded.fel.cvut.cz/procesory/RP_PICO)

<https://www.raspberrypi.com/products/raspberry-pi-pico/>

Pořízení v ČR

[https://rpishop.cz/raspberry-pi/3352-675-raspberry-pi-pico-0617588405587.html#/188-prislusenstvi-bez\\_prislusenstvi](https://rpishop.cz/raspberry-pi/3352-675-raspberry-pi-pico-0617588405587.html#/188-prislusenstvi-bez_prislusenstvi)





# Mikroprocesor, Mikropočítač

---

Procesor- komunikace pouze pomocí **jedniček** a **nul**

**Náš procesor Raspberry Pi PICO má napájecí napětí +3,3 V, ale může mít i jen +3 V. Toto napětí označíme jako  $V_{DD} = +3\text{ V}$**

**Vůči čemu je toto napětí vztažené - proti společné svorce GND groud, zem.**

**(Analogie – když porovnáváte svou výšku s bratrem, postavíte se oba na zem- to je vztažná úroveň). Bratr nemůže stát v dolíku- to by nebylo srovnání.**

**Podobně v mikroprocesor se všechna napětí vztahují vůči společné zemi GND.**

**Jak náš procesor „řekne“ svému okolí, že signalizuje **jedničku** na daném kontaktu? Přivede tam své napájecí **napětí + 3 V**, jako by se tam připojila baterie (dvě tužkové baterie zapojené sériově -za sebou)**

**(pozn. u Arduina je napájecí napětí + 5 V, tedy Arduino signalizuje jedničku na pinu (výstupu) napětím + 5 V proti zemi - GND)**

# Mikroprocesor, Mikropočítač

---

Jak poznáme, že na pinu je kladné napětí + 3 V ?

**Připojíme tam diodu LED, která bude (přes ochranný rezistor připojena na sledovaný kontakt) . Z kontaktu do LED poteče proud, který bude z LED vytékat do společné elektrody GND Groud, zem.**

**( jako když vody vytéká z hadice a pak dopadá na zem)**

**V naší desce máme čtyři různé LED červená, žlutá, zelená, modrá.**

**Jak procesor řekne svému okolí, že signalizuje **nulu** (false) na daném pinu? Přivede tam **nulové** napětí.**

**Připojená LED nebude svítit.**

**Dále na desce máme bzučák (buzzer) a tlačítko.**

# Grafické programování pomocí Raspberry PI PICO

---

Naši desku Raspberry PI PICO budeme „programovat graficky“

<https://bipes.net.br/ide/> pozor psát https://

Použít programy *Opera*, nebo *Google Chrome*, *Microsoft Edge*, které jsou schopny komunikovat s deskou přes USB.

(Pozor Mozilla nepodporuje připojení desky)

Při ovládání pinů a rozsvícení, nebo zhasnutí LED musíme zadat, který pin chceme ovládat. Jejich označení je **GP** a číslo

## ***Zapojení experimentální desky***

**GP16** Tlačítko k GND

**GP18** 470 R+ Červená **LED** na GND

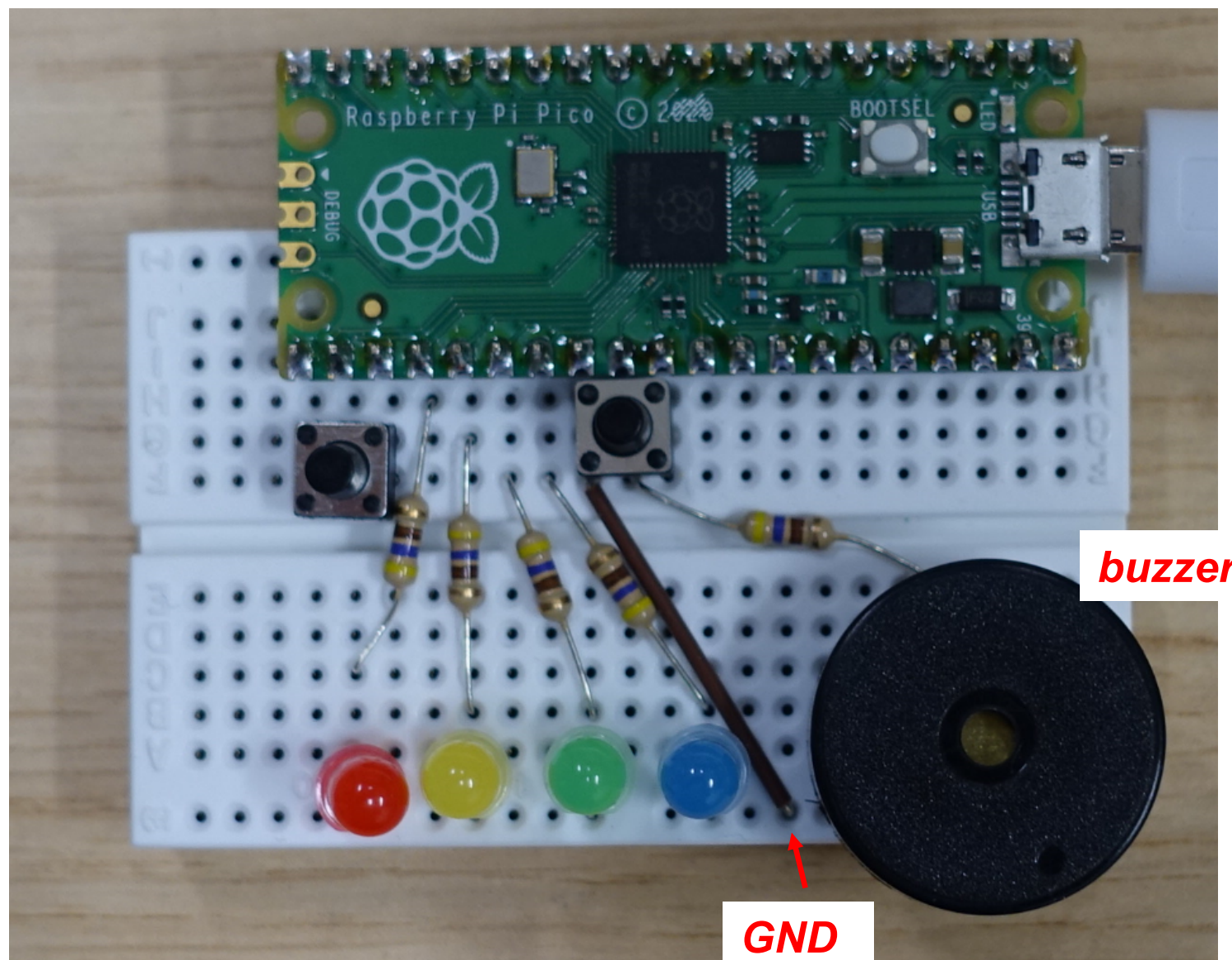
**GP19** 470 R+ Žlutá **LED** na GND

**GP20** 470 R+ Zelená **LED** na GND

**GP21** 470 R+ Modrá **LED** na GND,

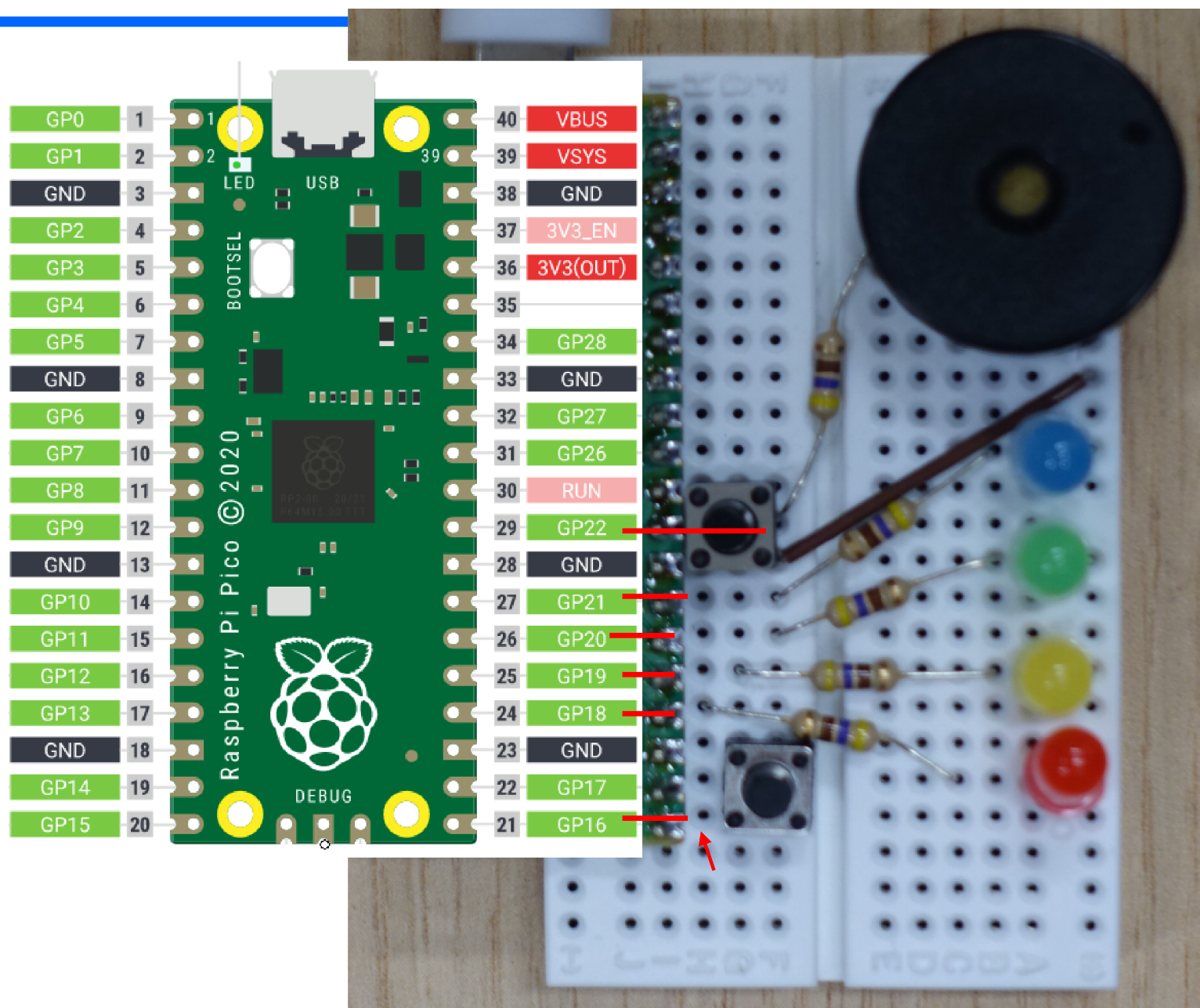
**GP22** 470 R+ Buzzer proti GND

# Osazení experimentálního pole





# Výstupní piny



# Grafické programování pomocí Raspberry PI PICO

---

**Zkusíme rozsvítit a zhasnout červenou LED  
spustit program Google Chrome a otevřít stránku**

**<https://bipes.net.br/ide/>**

**kabelem **Micro USB** připojit desku Raspberry PI PICO**

*Do desky jsme již dříve nahráli soubor pro program Micropython.*

*V nové desce ještě není nahraný. Nahraje se tak, že se při zapnutí – to je připojení na USB – drží stisknuté tlačítko.*

*Deska se v PC ukáže jako Flash disk, do kterého se nakopíruje soubor  
**[rp2-pico-20220117-v1.18.uf2](#)***

*případně jeho novější varianta. Ke stažení je na  
[Micropython.org](https://micropython.org)*

*<https://micropython.org/download/rp2-pico/>*

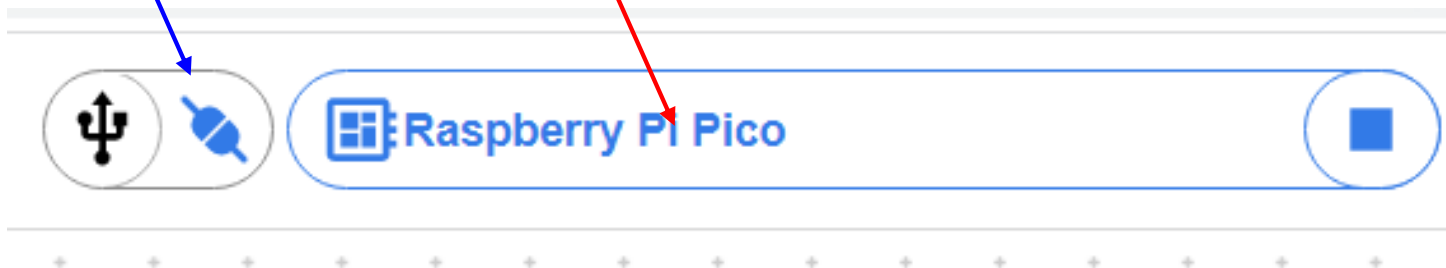
▪

---

**Výběr desky** klik na pole implicitně ta může být ESP8266,  
**zvolit Raspberry Pi PICO**

Připojení / odpojení desky

Doporučujeme připojit, odpojit, připojit – spíše začne korektně fungovat





Logic

Loops

Math

Text

Lists

Variables

Functions

BIPES

Python

Timing

▼ Machine

CPU

In/Out Pins

► Displays

► Sensors

► Actuators

► Communication

Files

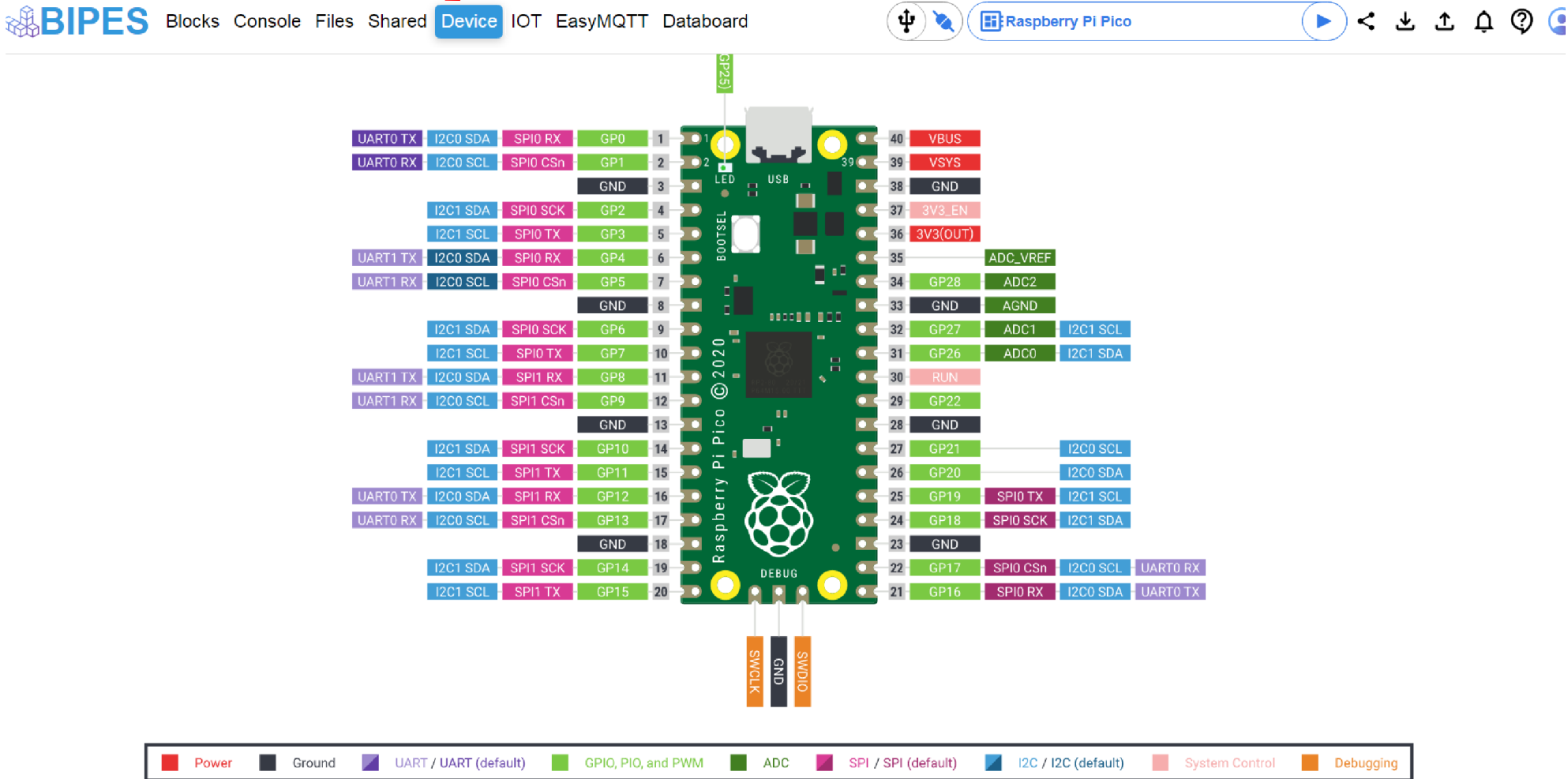
► Network

► micropython

**Připojení** desky, volba rozhraní **USB**  
někdy potřebuje **opakovat dvakrát**  
***připojit, odpojit, připojit***

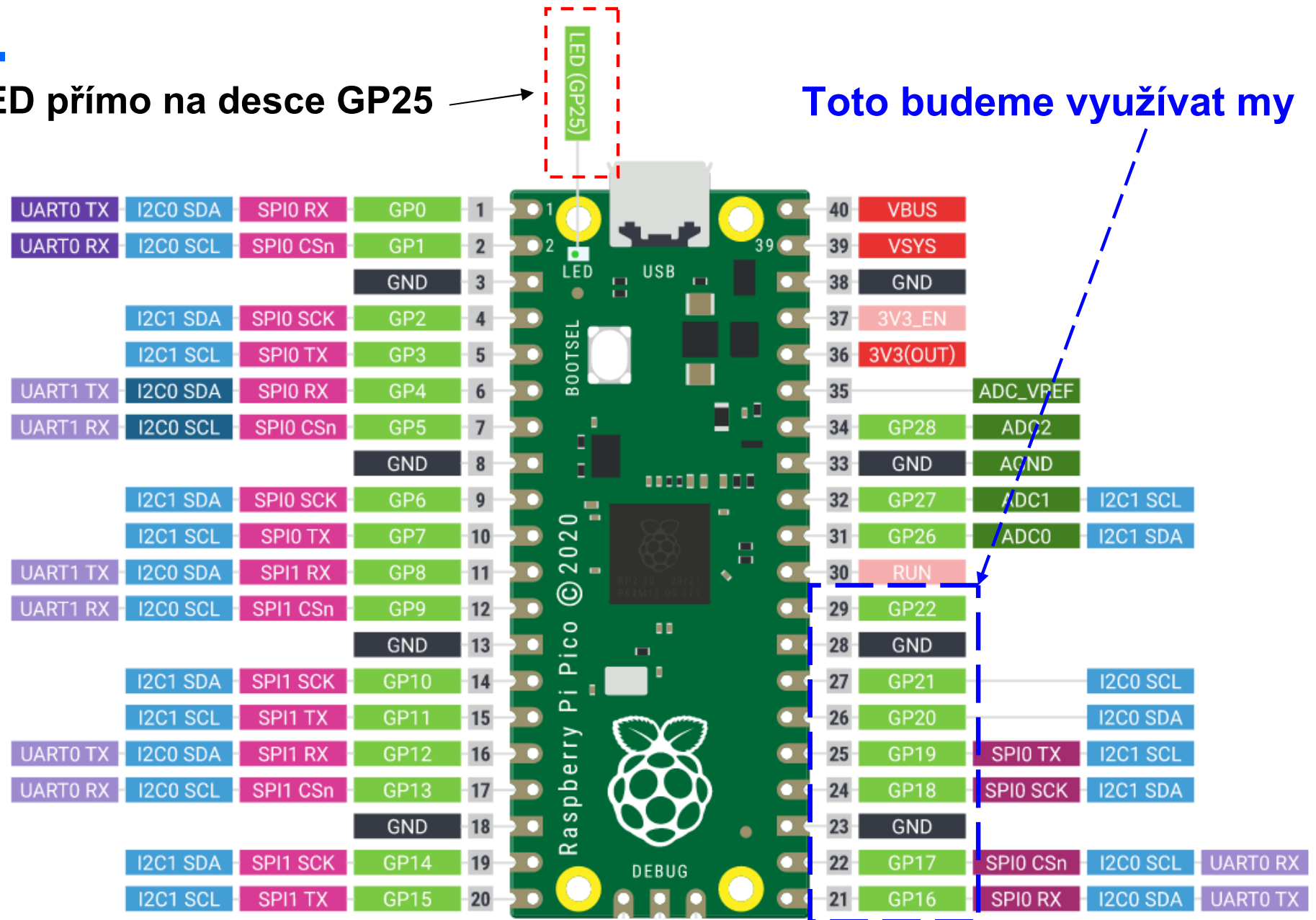
# Nabídky panelů

Device- zobrazí rozložení vývodů desky, to budeme potřebovat při programování



LED přímo na desce GP25

Toto budeme využívat my



# Zapojení experimentální desky

---

Při ovládání pinů a rozsvícení, nebo zhasnutí LED musíme zadat, který pin chceme ovládat. Jejich označení je **GP** a číslo

## *Zapojení experimentální desky*

**GP16** Tlačítko k GND

**GP18** 470 R+ Červená **LED** na GND

**GP19** 470 R+ Žlutá **LED** na GND

**GP20** 470 R+ Zelená **LED** na GND

**GP21** 470 R+ Modrá **LED** na GND,

**GP22** 470 R+ Buzzer proti GND

*470 R značí zapojení rezistoru o odporu 470 Ohmů do série s pinem*

*Úroveň „**true**“ na pinu LED rozsvítí,  
úroveň „**false**“ LED na pinu zhasne*

▪

---

**Vytvoříme program, který proběhne *jen jednou*, rozsvítí červenou LED na dvě sekundy a pak zhasne a pak se ukončí a bude čekat na další pokyny.**

**blok pin a wait**

# Rozsvítit LED- nástroje

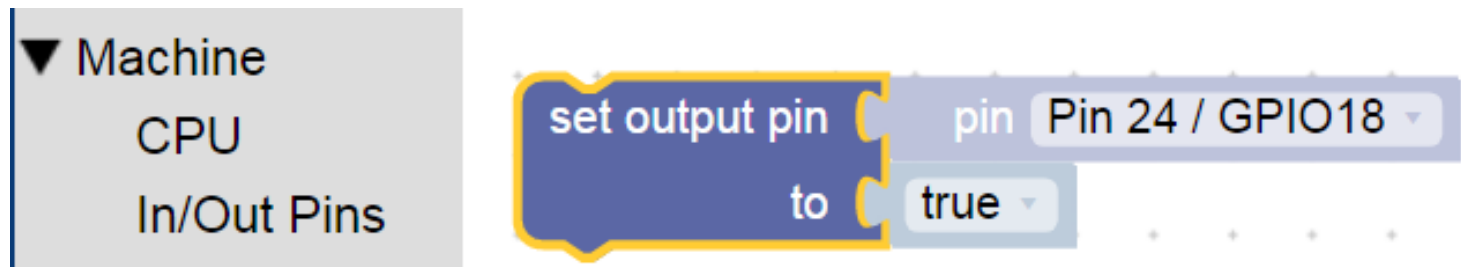
▪

The screenshot displays the BIPES Blocks environment. On the left, a sidebar lists various block categories: Logic, Loops, Math, Text, Lists, Variables, Functions, BIPES, Python, Timing, Machine (highlighted with a red dashed box), CPU, In/Out Pins, and Display. The 'In/Out Pins' category is selected, showing a workspace with three blocks:

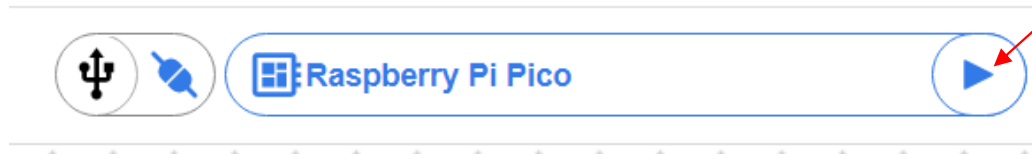
- pin** LED / GPIO25
- set output pin** to **pin** LED / GPIO25 **to** true
- read digital input** Pull-up **pin** LED / GPIO25 **true**

# Jednorázový běh programu ( bez opakování)

Program se provede- rozsvítí se červená LED a opět je vidět trojúhelník



**Klik sem- spustí program**



**Pokud zvolíme menu „ Console“,  
bude vidět, že počítač čeká  
na další pokyn**

```
>>>
paste mode; Ctrl-C to cancel, Ctrl-D to finish
=== from machine import Pin
===
=== def gpio_set(pin,value):
===     if value >= 1:
===         Pin(pin, Pin.OUT).on()
===     else:
===         Pin(pin, Pin.OUT).off()
===
===
=== gpio_set((18), True)
```



# **Další postup, jeden průchod, více průchodů - smyčky**

---

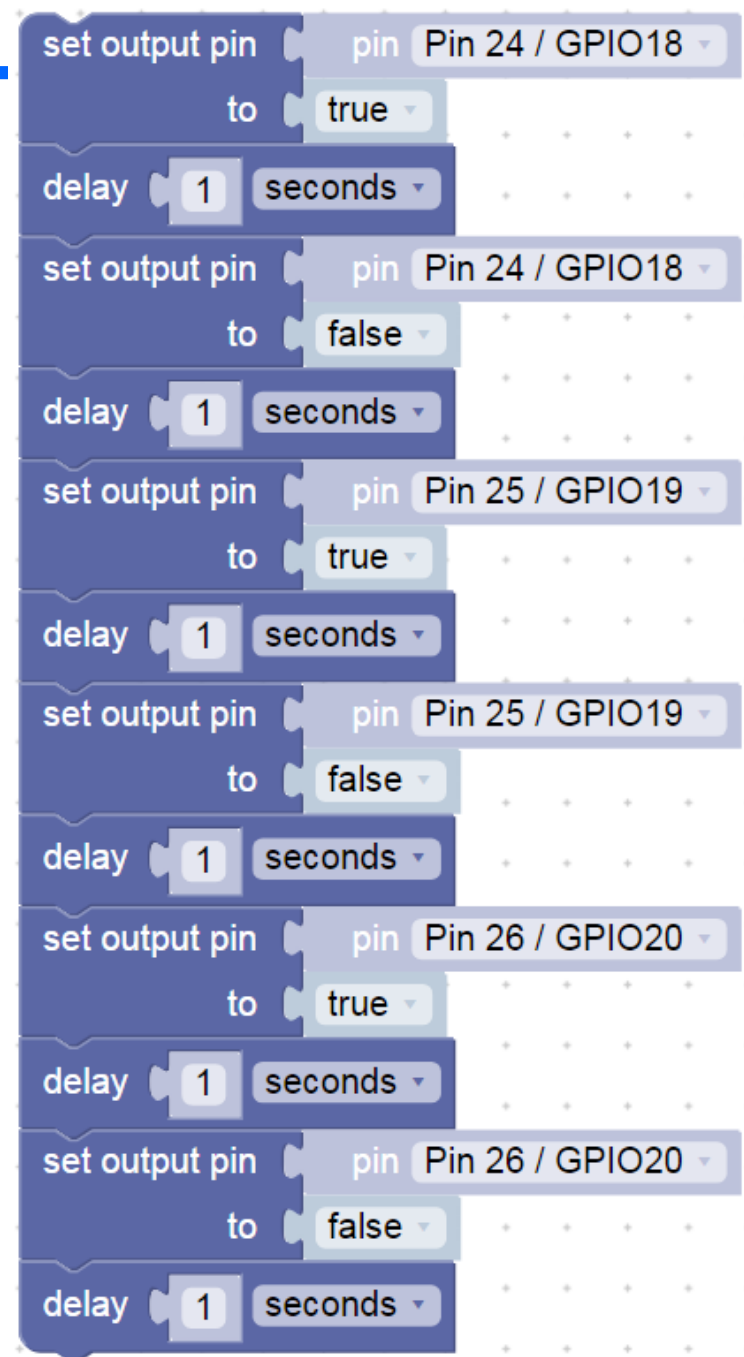
**Program postupně rozsvítí a zhasne různé LED a ukončí se.**

**Program postupně rozsvítí a zhasne různé LED a vše provede několikrát a ukončí se LOOP**

**Program s podmínkou**

**Nekonečný program s podmínkou true**

# Rozsvícení a zahasnutí LED, 1x



# Stálý běh programu, použití nekonečné smyčky



Raspberry Pi Pico



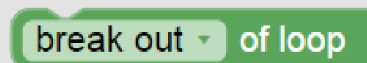
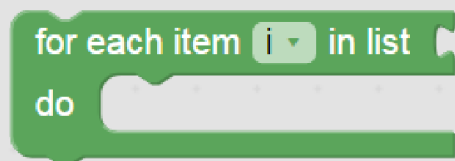
Aktuální běh programu – symbol **čtvereček**

## Menu Loops

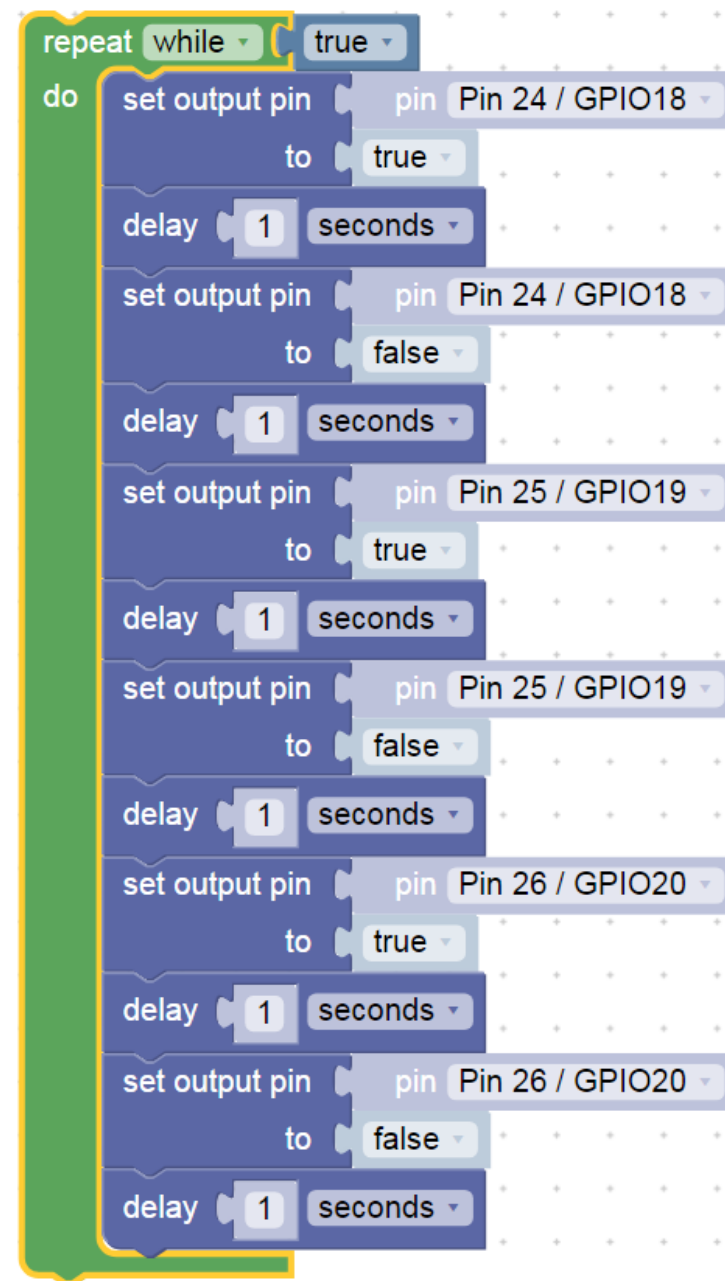
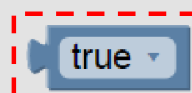
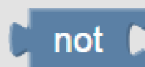
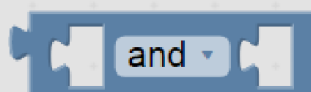
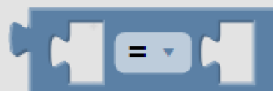
## Menu Logic

# Stálý běh programu, použití nekonečné smyčky

## Menu Loops



## Menu Logic



## Bloky pro smyčky

Logic	repeat 10 times
Loops	do
Math	
Text	
Lists	
Variables	repeat while
Functions	do
BIPES	
Python	count with i from 1 to 10 by 1
Timing	do
Machine	
CPU	
In/Out Pins	for each item i in list
Displays	do
Sensors	
Actuators	
Communication	break out of loop
Files	

## Bloky pro práci s piny

The screenshot displays the MicroPython IDE's block editor. On the left is a sidebar with a category list: Logic, Loops, Math, Text, Lists, Variables, Functions, BIPES, Python, Timing, Machine (expanded), CPU, In/Out Pins (selected), Displays, Sensors, Actuators, Communication, Files, Network, and micropython. The main workspace contains the following blocks:

- In/Out Pins** header block.
- pin** block with dropdown menu set to **LED / GPIO25**.
- set output pin to** block with dropdown menu set to **true**.
- read digital input** block with dropdown menu set to **LED / GPIO25** and **Pull-up** dropdown set to **true**.
- Read RPI Pico ADC Input** block with dropdown menu set to **LED / GPIO25**.
- RPI Pico PWM # 0** block with **Pin** dropdown set to **LED / GPIO25**, **Frequency** set to **1000**, and **Duty** set to **50**.
- PWM # 0 frequency** block with **frequency** set to **1000**.
- PWM # 0 duty** block with **duty** set to **50**.
- PWM # 0 init** block with dropdown menu set to **LED / GPIO25**.
- deinit PWM # 0** block.

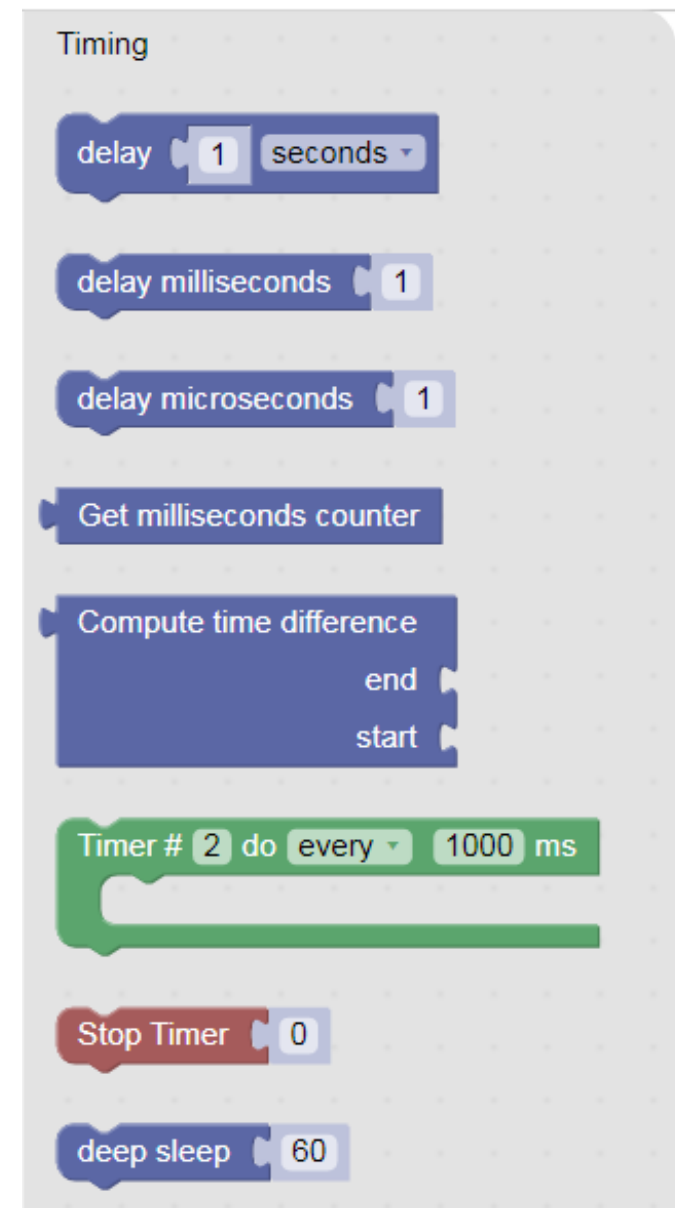
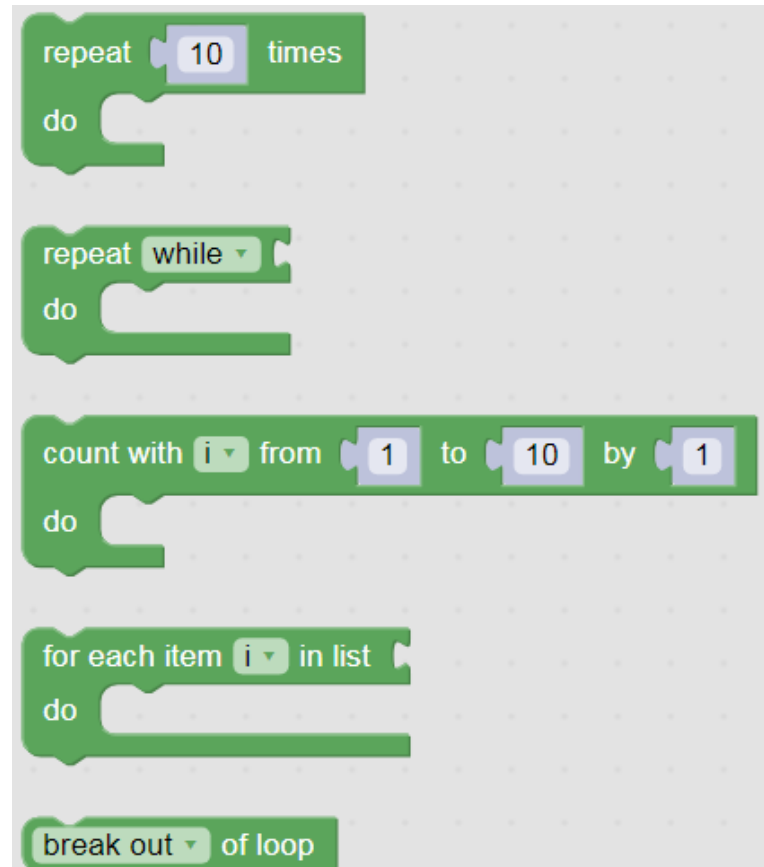
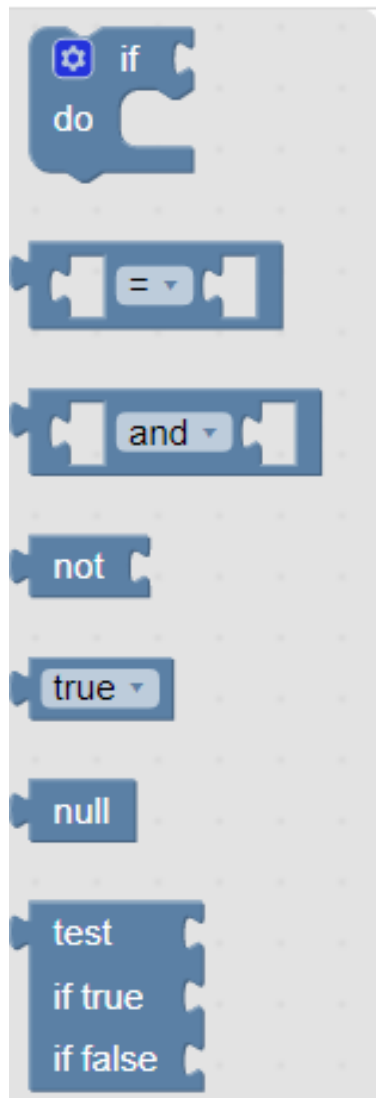
## Bloky pro časování

The image shows the block palette of a MicroPython IDE. The 'Timing' category is selected and expanded, displaying the following blocks:

- delay (1 seconds)
- delay milliseconds (1)
- delay microseconds (1)
- Get milliseconds counter
- Compute time difference (with 'end' and 'start' ports)
- Timer # 2 do every 1000 ms
- Stop Timer (0)
- deep sleep (60)



# Bloky logika, smyčky, timing stroj- pin



# Zrychlit blikání LED

Timing- delay 50 ms. **50 milisekund** – 50 tisícín sekundy

Pak dát **10 ms/10 ms**, zrychlení

Nebude vidět blikání

**Změnit poměr**

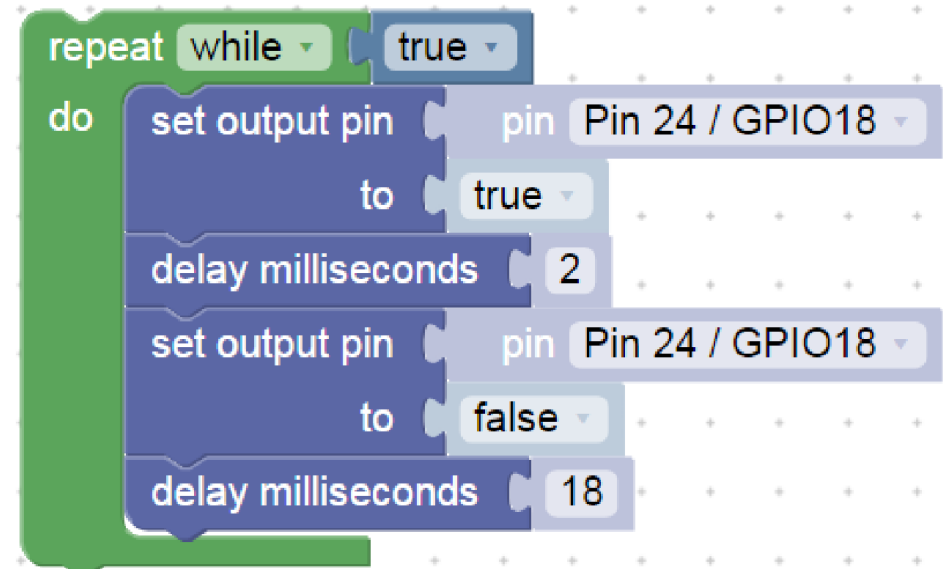
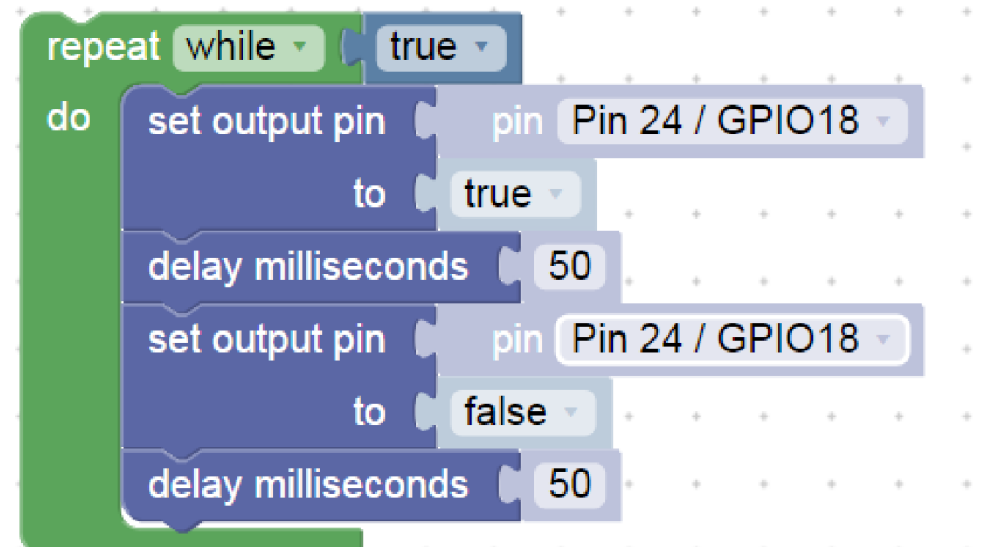
Dále 2 ms /18 ms

Zkusit jinak **měnit poměr časů**

Co se změnilo? Nestačíme sledovat blikání, ale jen vidíme **pokles jasu LED**.

To jsme vytvořili programově řízenou **PWM**

**Pulse Width Modulation**



# PWM

Pokud chceme **řídít jas** více LED bez programového zatěžování procesoru, použijeme hardwarovou PWM, Většina procesorů obsahuje blok pro generaci PWM signálu

V menu Machine In/Out pins je Frequency- frekvence (kmitočet).

Duty – střída, poměr času ,  
kdy to svítí, vůči periodě blikání

Pozor, zde číslo

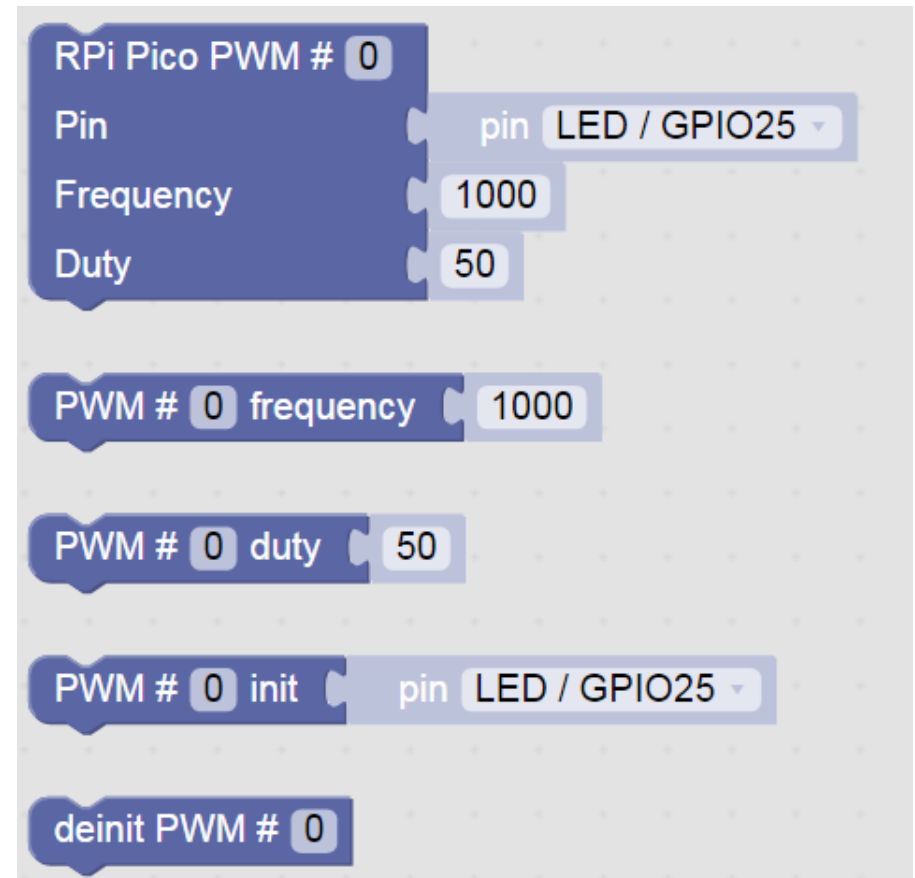
**32768 znamená 50 procent !!**

To souvisí s formou vyjádření  
16- bitového binárního čísla

Pokud dáme 50, tak to skoro  
nebude svítit.

Nula- 0 nesvítí vůbec,

**65535** svítí plně

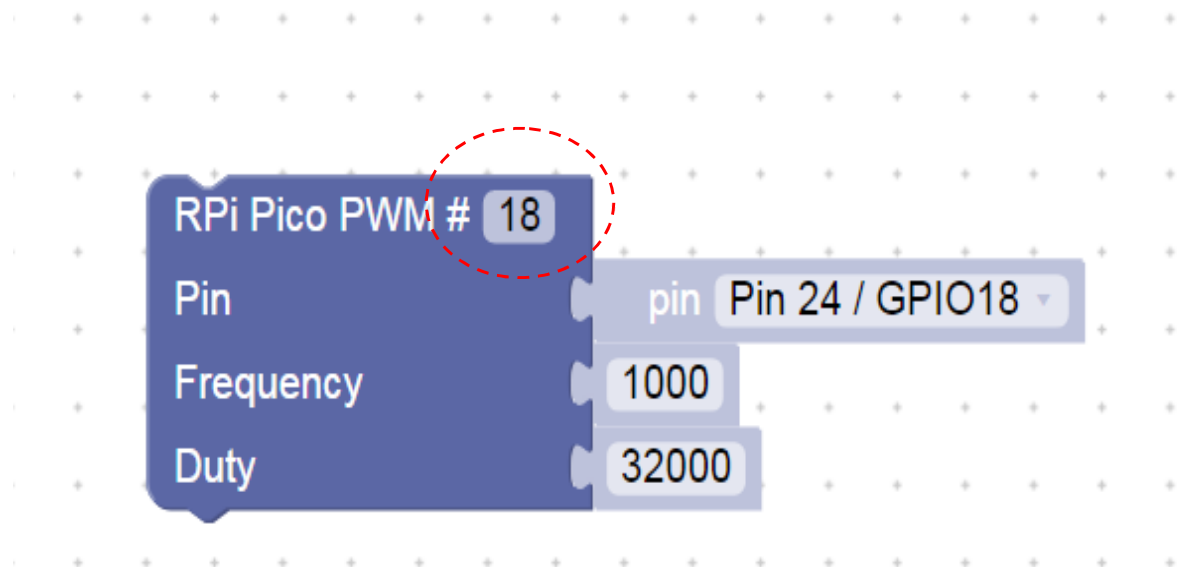


# Jednorázová inicializace PWM

Červená LED bude svítit plovicním jase

Můžeme zkusit i frekvenci 10 Hz – budeme vidět blikání

1 Hz nelze použít (procesor to nepoporuje)

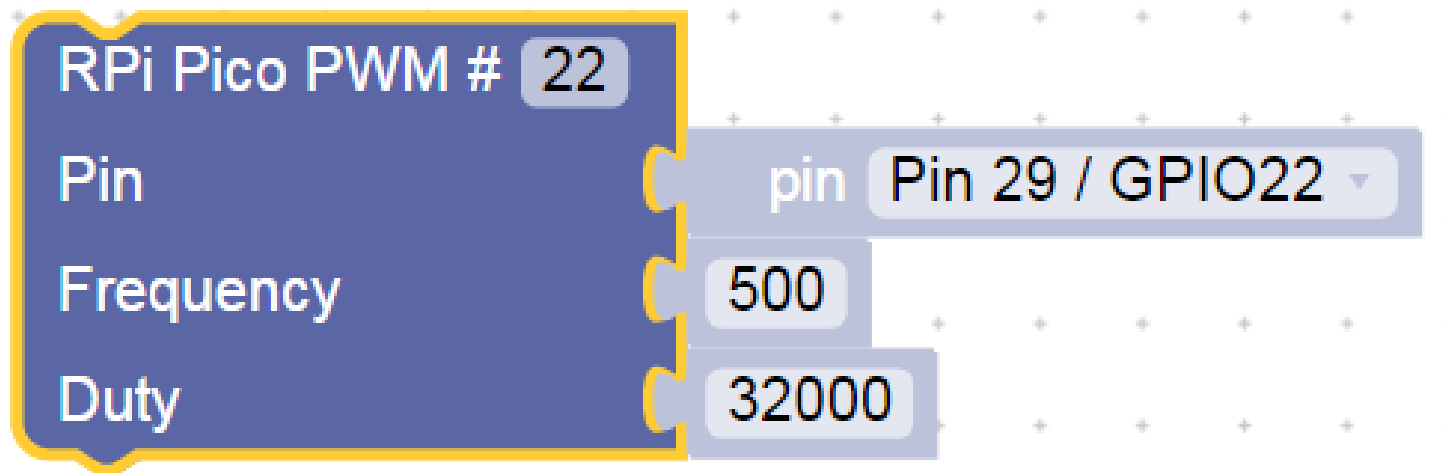


# Jednorázová inicializace PWM pro Buzzer

## Jednorázová inicializace

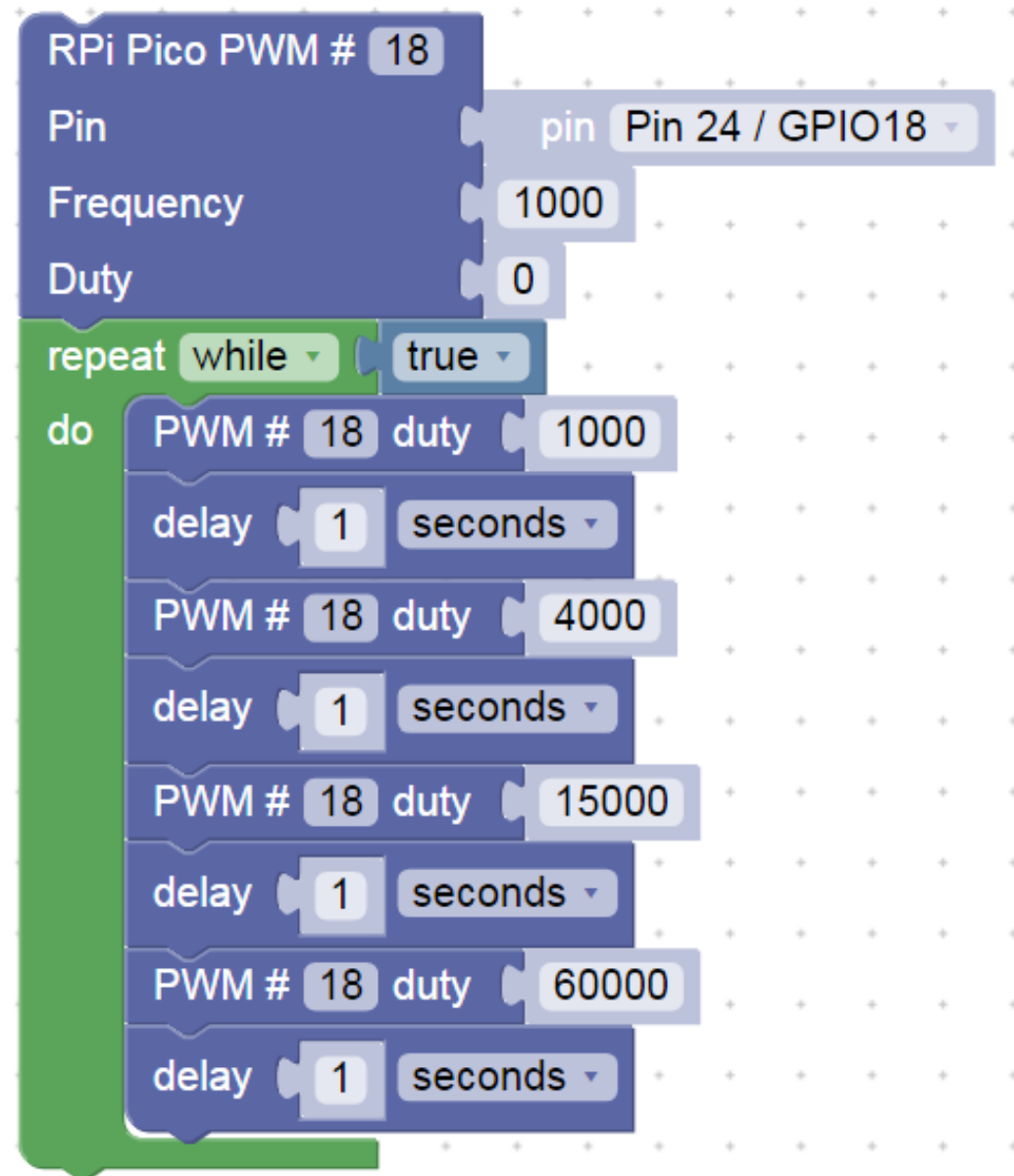
Pokud chceme změnit, je nutno změnit parametr ( frekvenci nebo střidu) a znovu spustit program.

Pokud chceme, aby nás to nerušilo, dát malou střidu  
např. Duty = 0



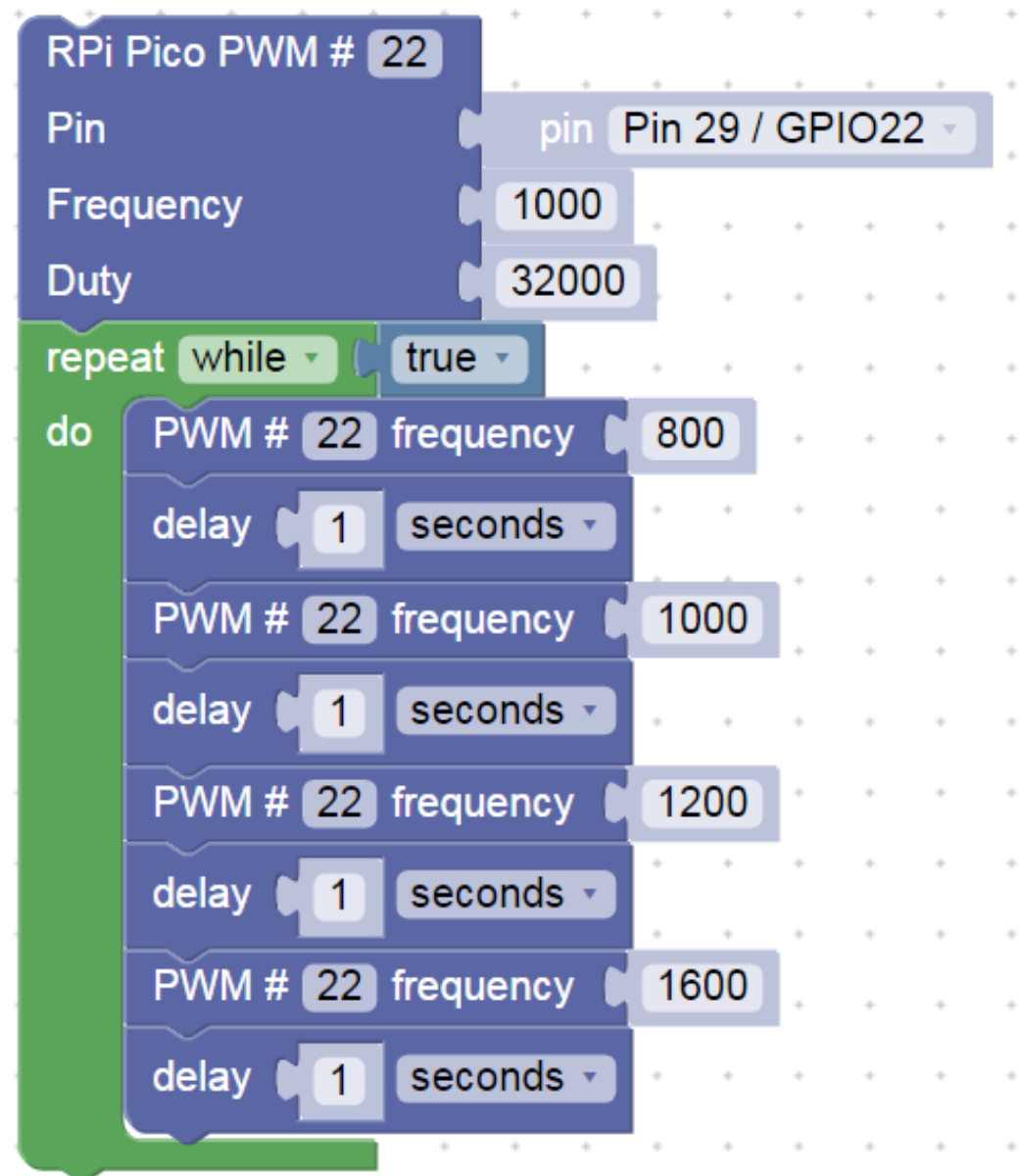
# Nekonečná postupná **změna jasů** LED ve stupních

## Postupné **rozsvícení** a **naráz zhasnutí** LED



# Ovládání bzučáku (buzzer) pomocí PWM

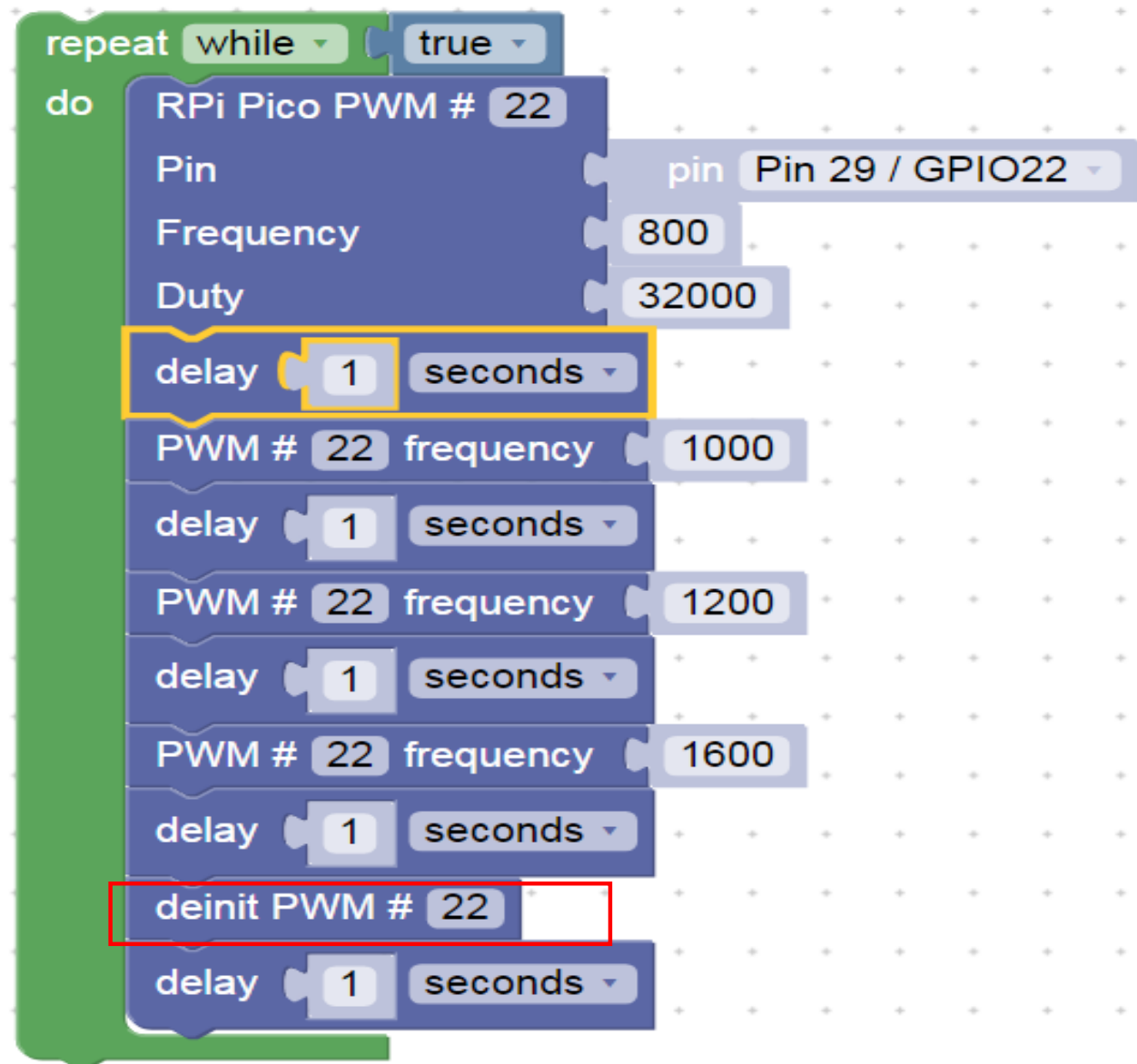
## Změna frekvence PWM





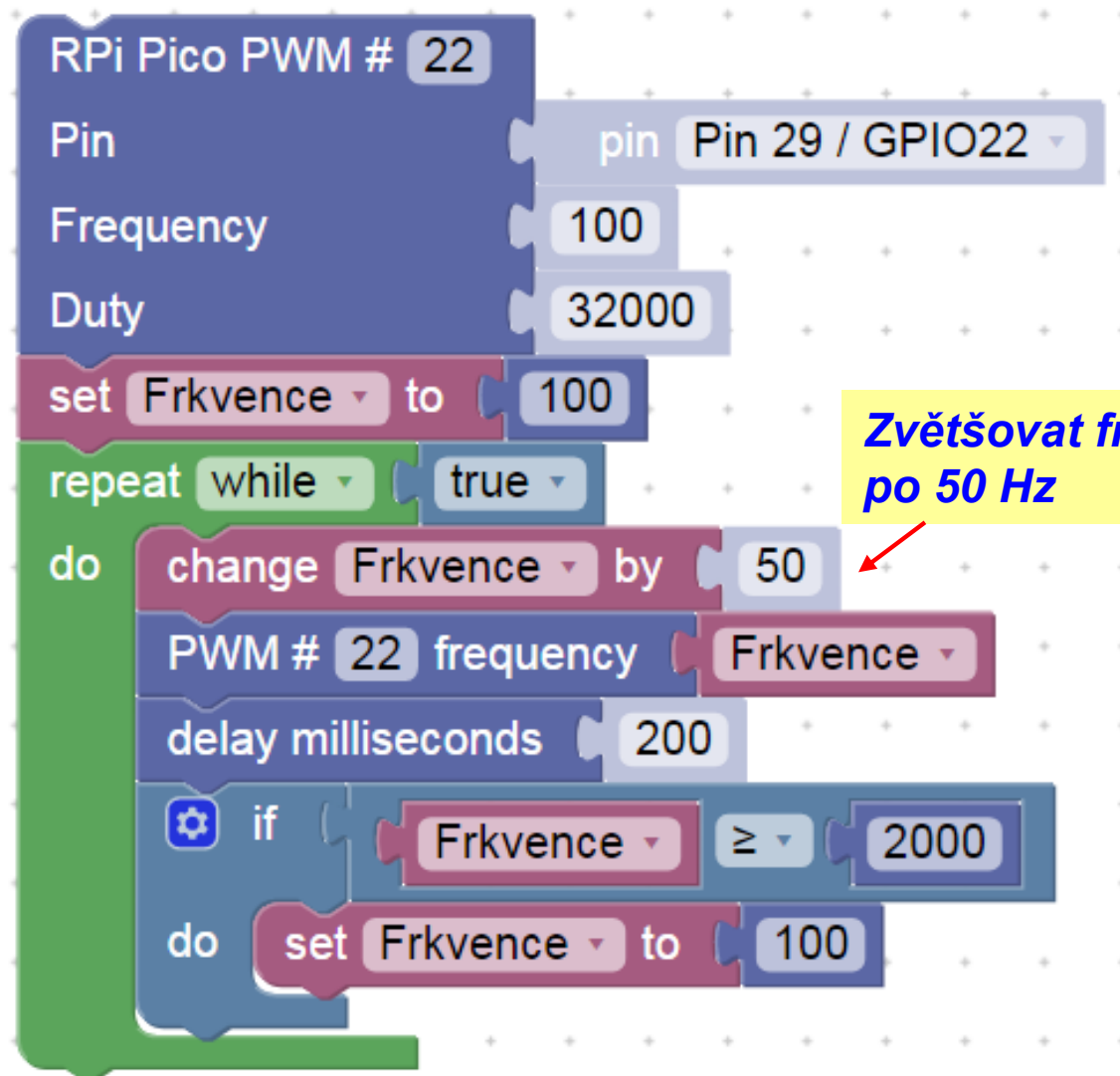
# Bzučák s PWM s deinicializací PWM

Na chvíli úplně vypneme PWM



# Postupná změna frekvence PWM pro bzučák (buzzer)

Frekvence 100 Hz až 2000 Hz



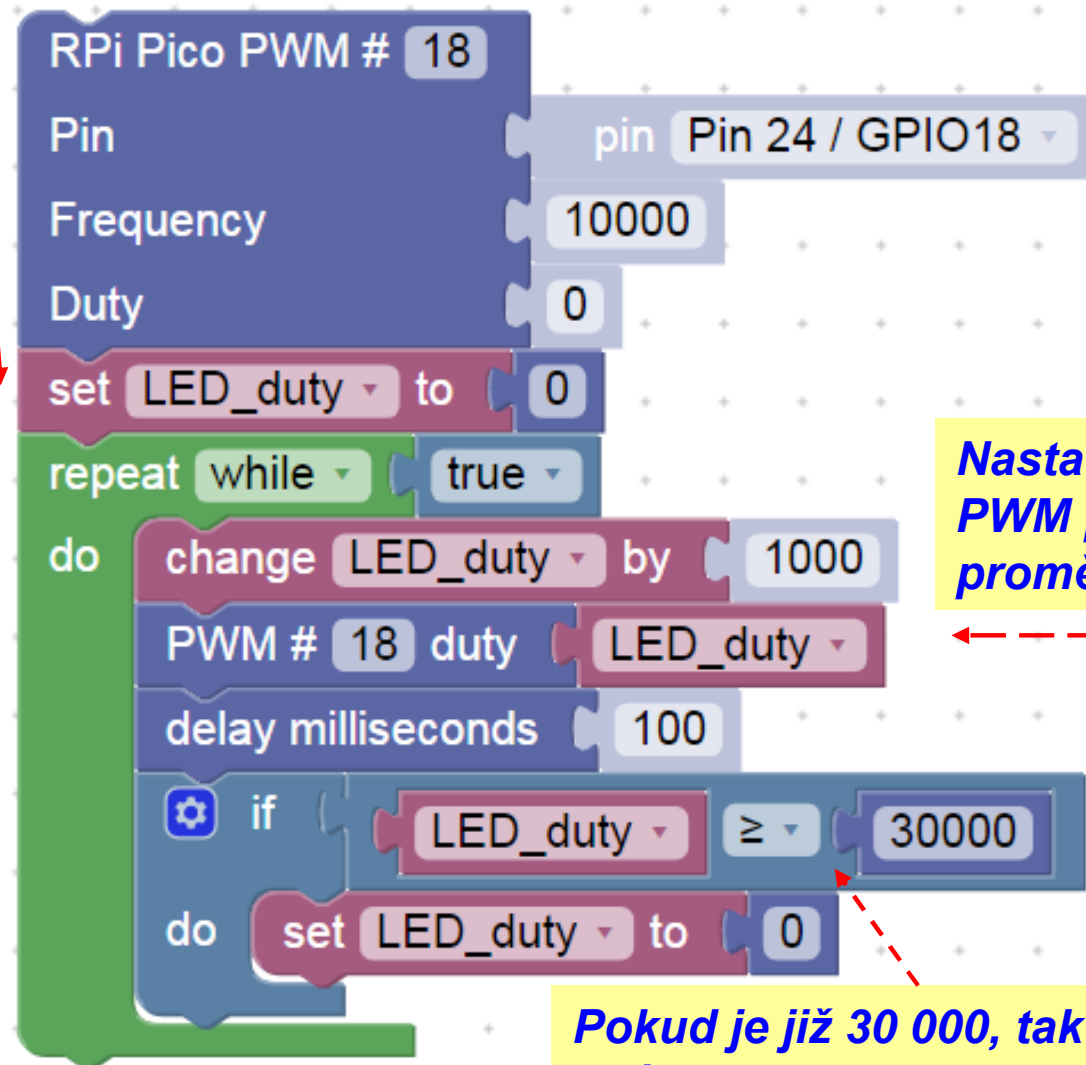
*Zvětšovat frek.  
po 50 Hz*

# Postupná změna jasu LED na pinu GPIO18

Proměnná **LED\_duty** nese informaci o požadované střídě

Postupné rozsvěování

LED a naráz pohasnutí

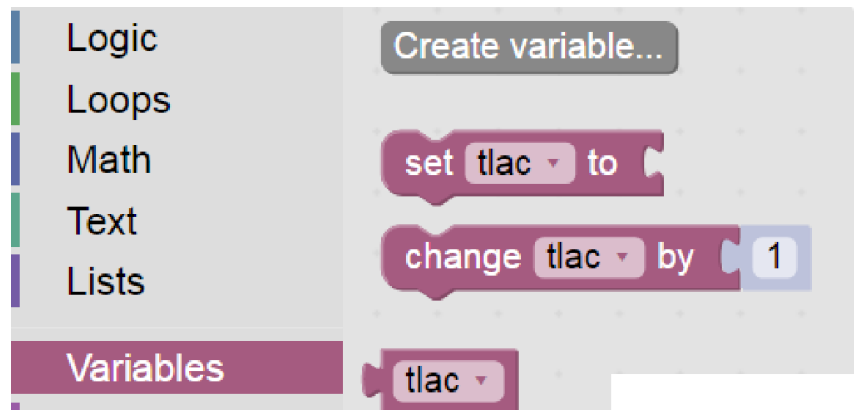
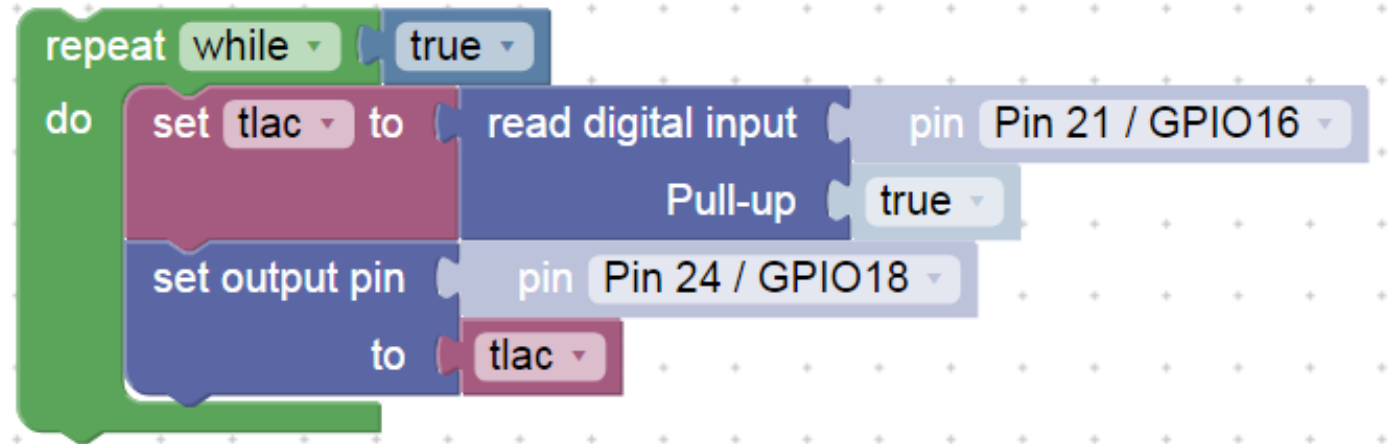


*Nastavení  
PWM podle  
proměnné*

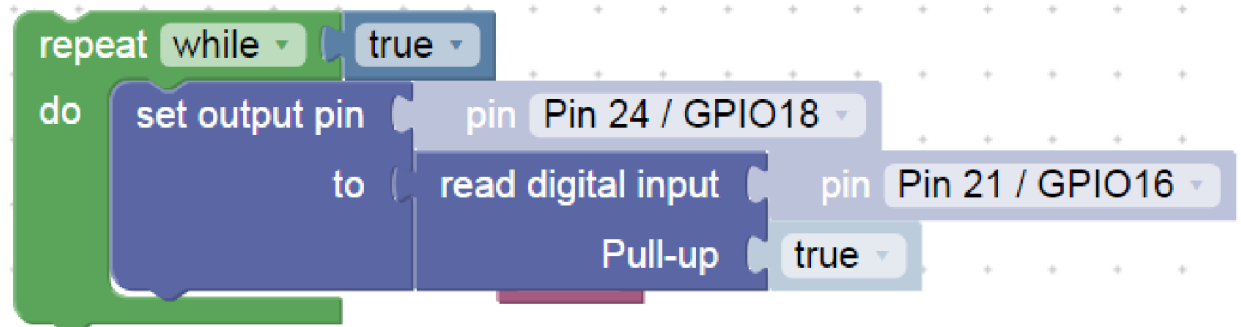
*Pokud je již 30 000, tak  
zpátky na nulu*

# Čtení stavu tlačítka a uložení do **proměnné**

Nekonečná smyčka, stále opakovaně čte stav tlačítka a podle toho nastavuje červenou LED na GPIO18. Reaguje okamžitě



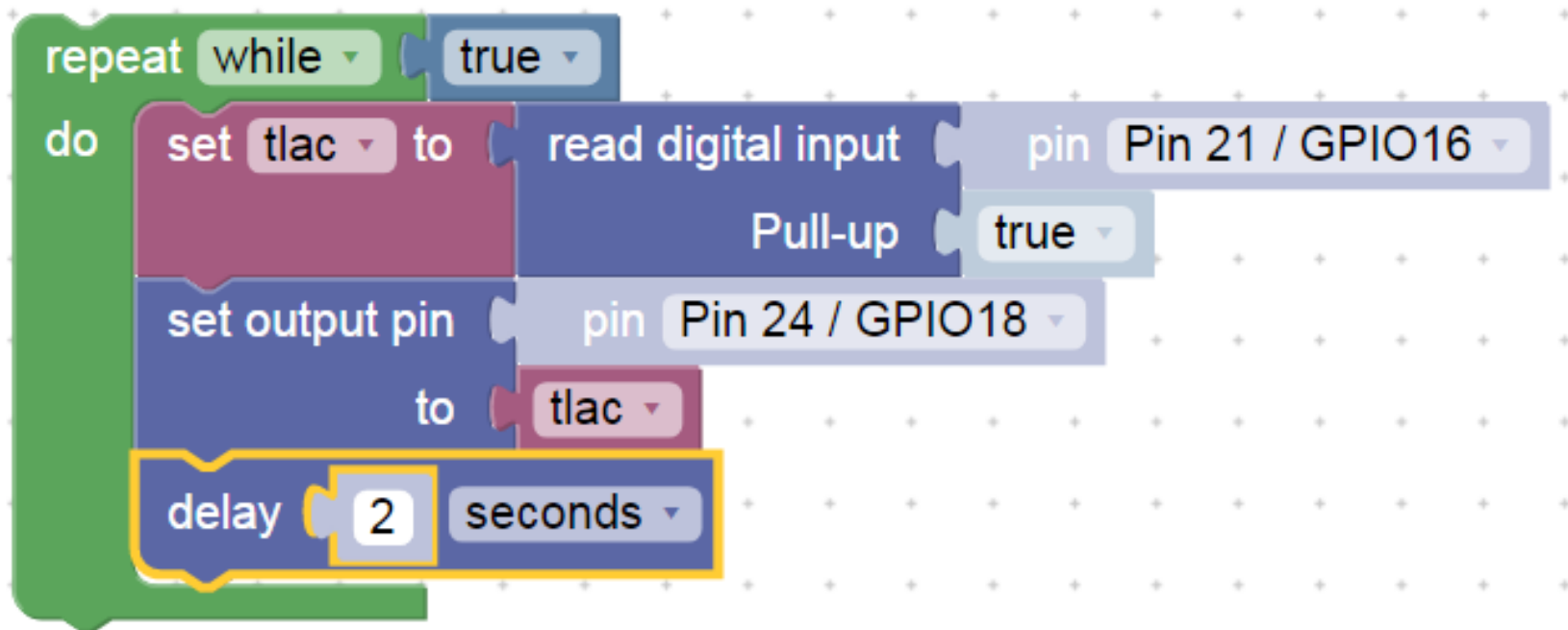
Jednodušší řešení  
**bez proměnné**



# Čtení stavu tlačítka a uložení do proměnné

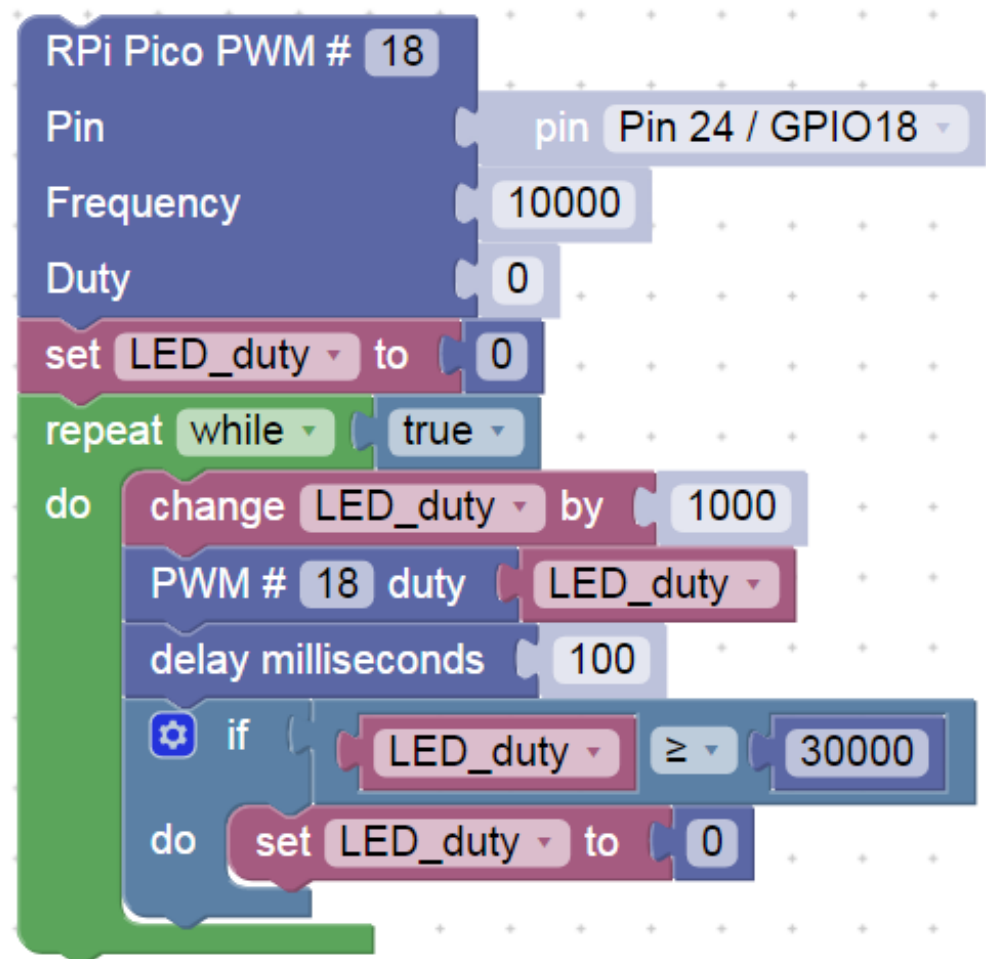
Zpomalení reakce, čteme stav tlačítka jen 1x za 2 sekundy

Důsledek- zpomalení reakce „**pomalé tlačítko**“



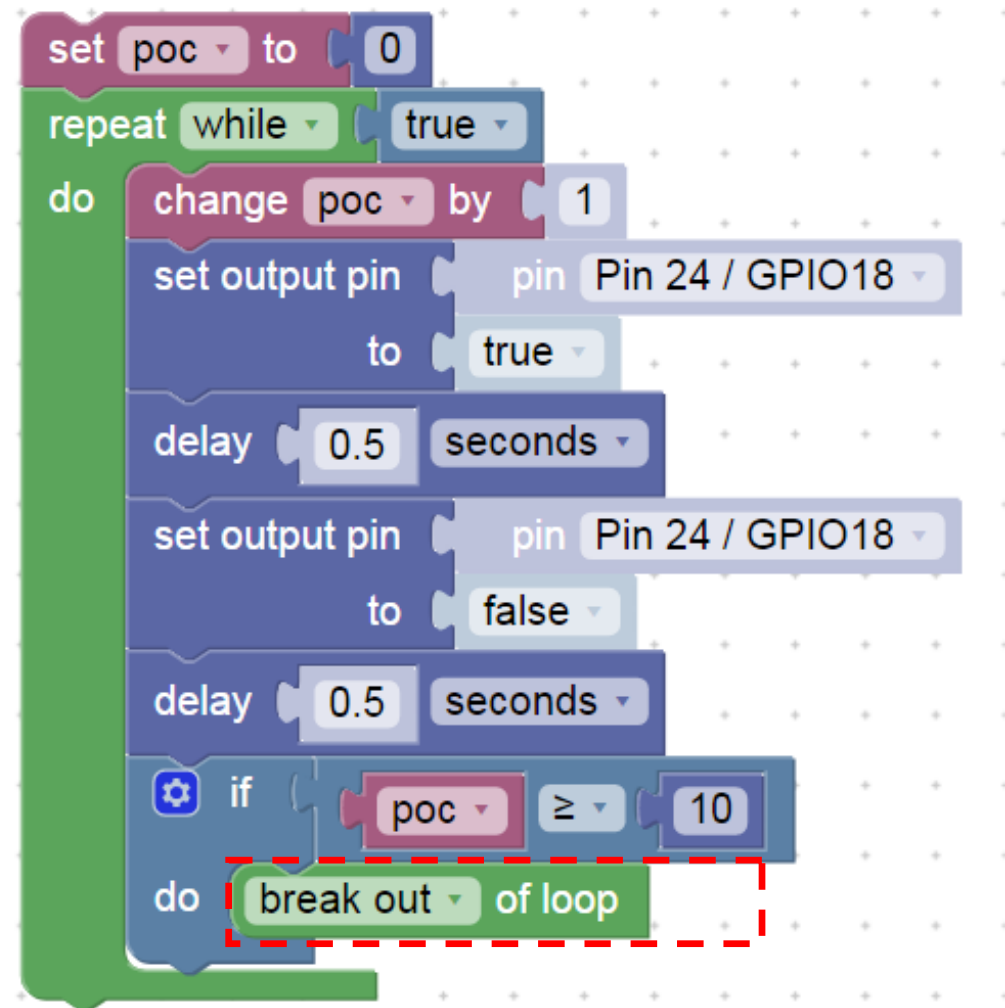
# Postupná změna jasu LED

Proměnná **LED\_duty** nese informaci o požadované střídě



# Ukončení smyčky podle podmínky

Při splnění podmínky se ukončí  
smyčka i celý program



## **Další náplň - elektronika**

---

**Pokud nám programování jde dobře, můžeme si zkusit zapojit jednoduchý elektronický obvod s LED a tranzistorem**

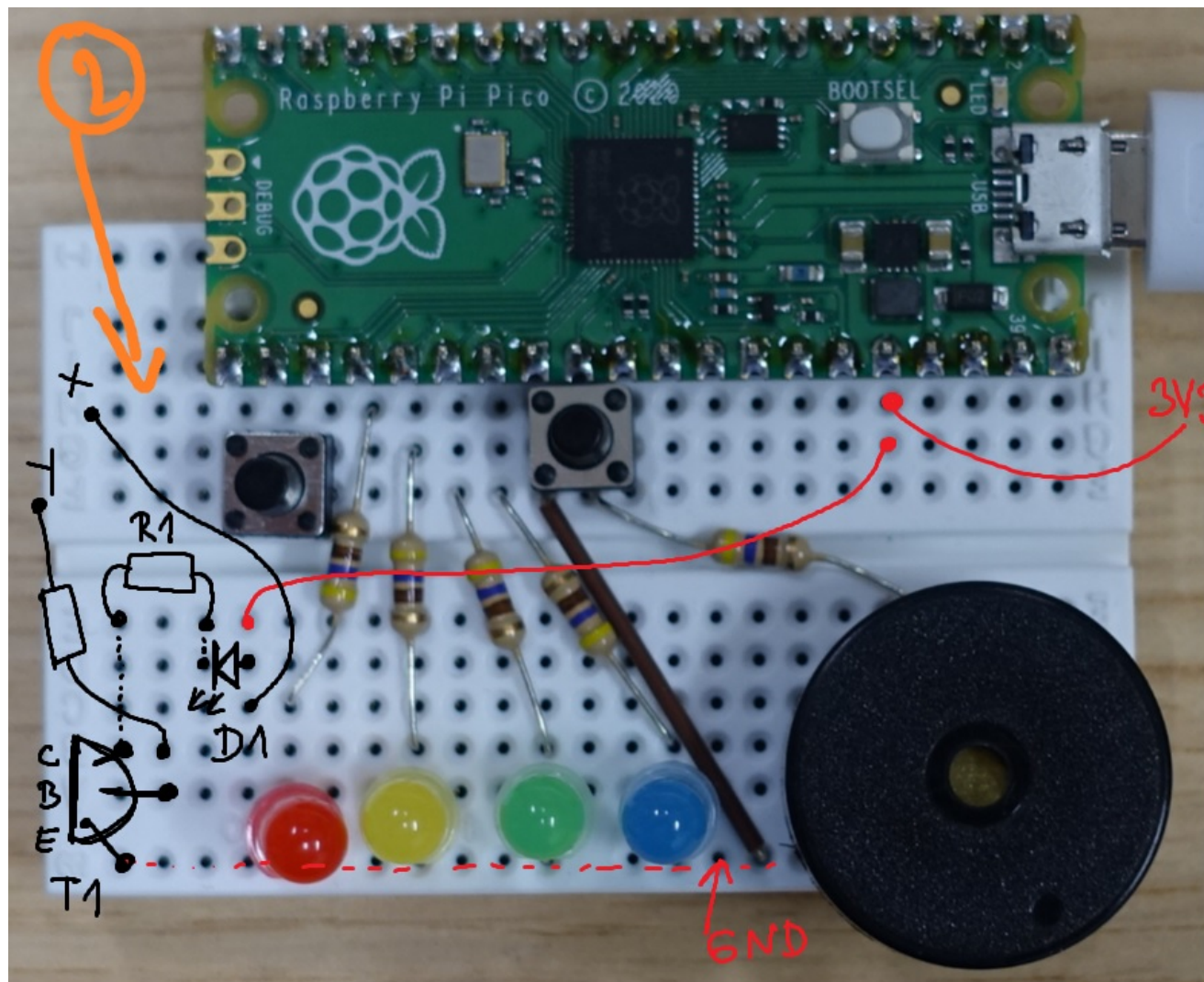
**Máme tranzistor NPN typu BC547C, LED**

**Rezistor 470 Ohmů ( hnědý)**

**Rezistor 10 000 Ohmů ( modrý)**

**LED kratší vývod je katoda, připojit na zem GND**



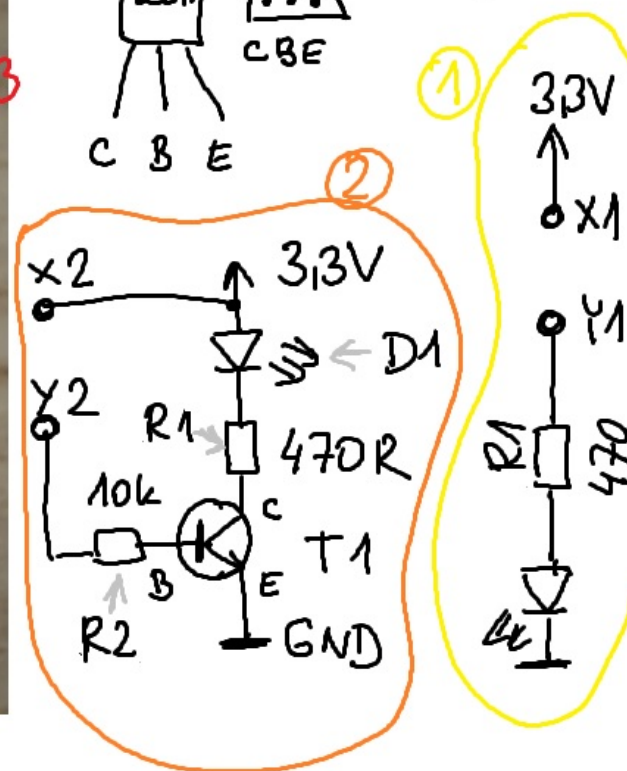


TRANZISTOR

NPN

BC 547

nožičky  
dole



# Osciloskop a sledování signálu

---

**Při programování blikání LED jsme byli schopni sledovat blikání.**

**Pokud budem řídit jas LED pomocí PWM, pak rychlé blikání již nejsme schopni zrakem sledovat.**

**Sledování signálu PWM – přístroj – osciloskop**

**S podobným přístrojem u nás pracují studenti – středoškoláci v klubu ETC22.**

**GND připojení na společný vodič zem, GND – ground**

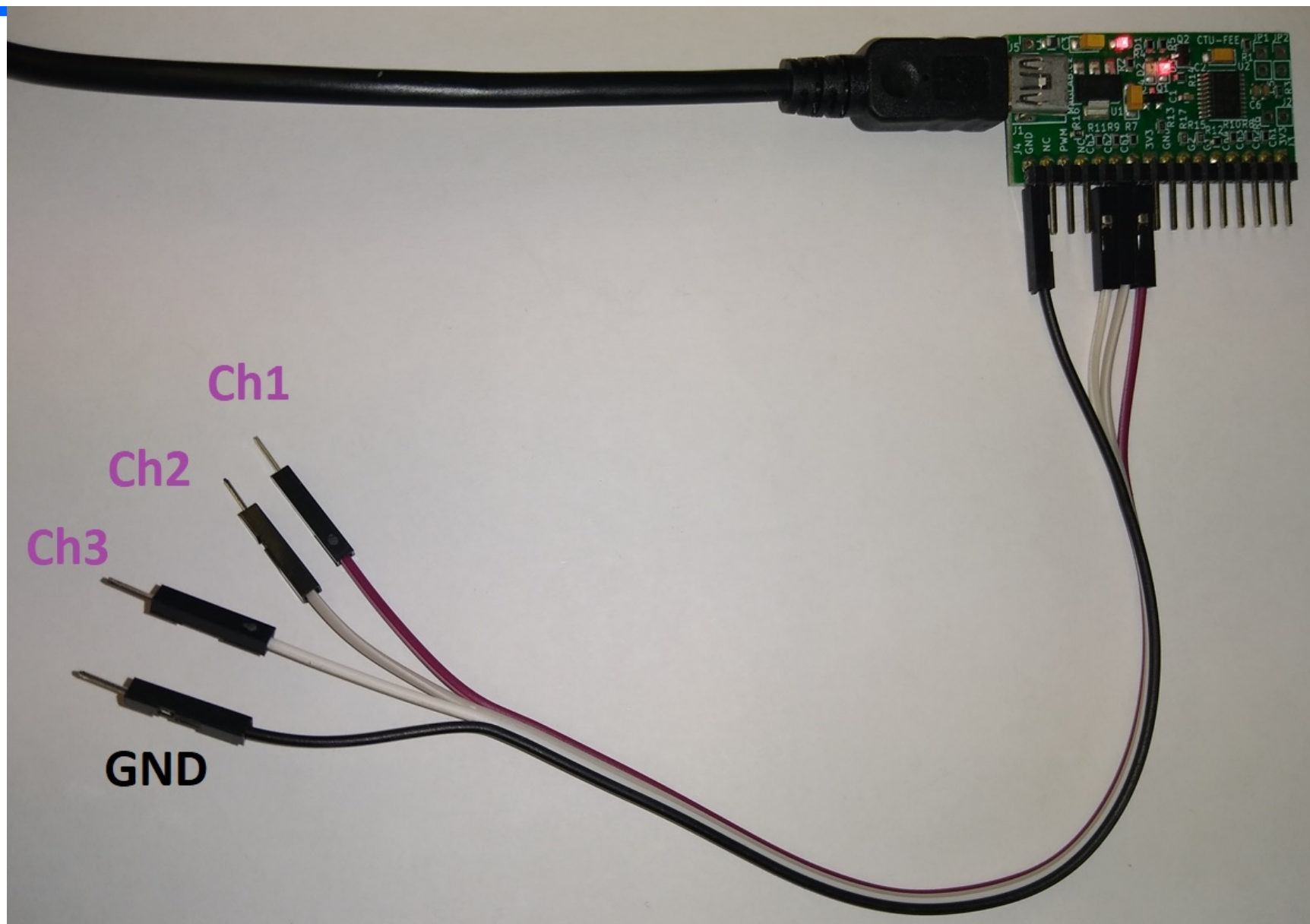
**Ch1, Ch2, Ch3- vstupy osciloskopu**

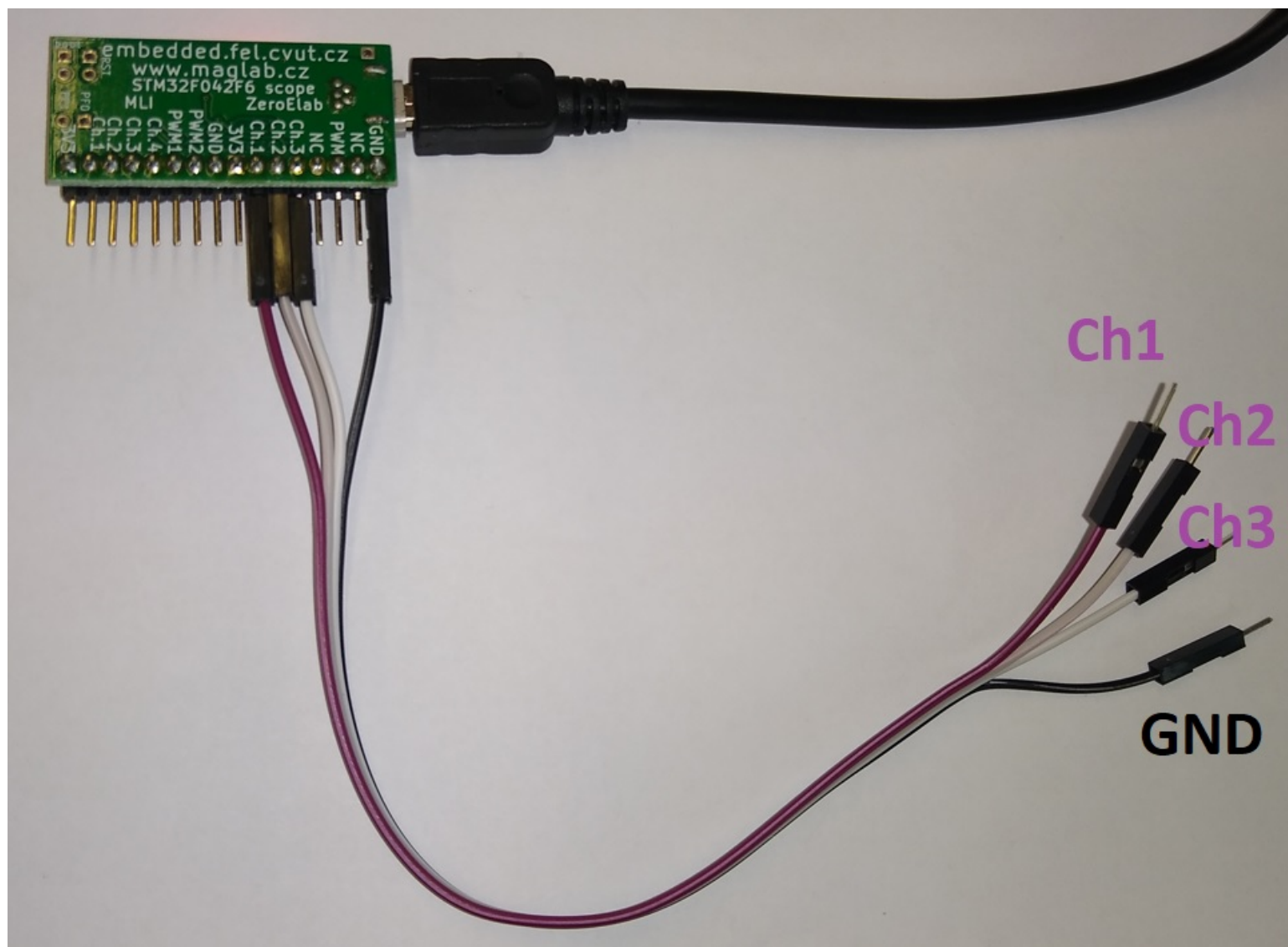
**PC obslužná aplikace Zero elab Viewer**

**Připojení a napájení přes USB.**

**Úkol. Naprogramovat generátor PWM a sledovat osciloskopem signál**

**Naprogramovat generátor PWM, kde se bude pomalu po skocích měnit frekvence a osciloskopem sledovat signál**





# KONEC

---

▪