

---

# **Seminář a workshop při akci „Podzimní škola fyziky“**

**Mikropočítač Raspberry PI PICO, jeho grafické programování  
a využití jako softwarově definovaného přístroje pro  
laboratorní experimenty**

**Prezentující: doc. Fischer, katedra měření ČVUT – FEL, 2.12.2022**

# Náplň

---

**Mikropočítač Raspberry PI PICO (dále PICO) a možnosti jeho využití v laboratorní výuce a ve výuce programování**

**Softwarově definované přístroje SDI - Software Defined Instruments**

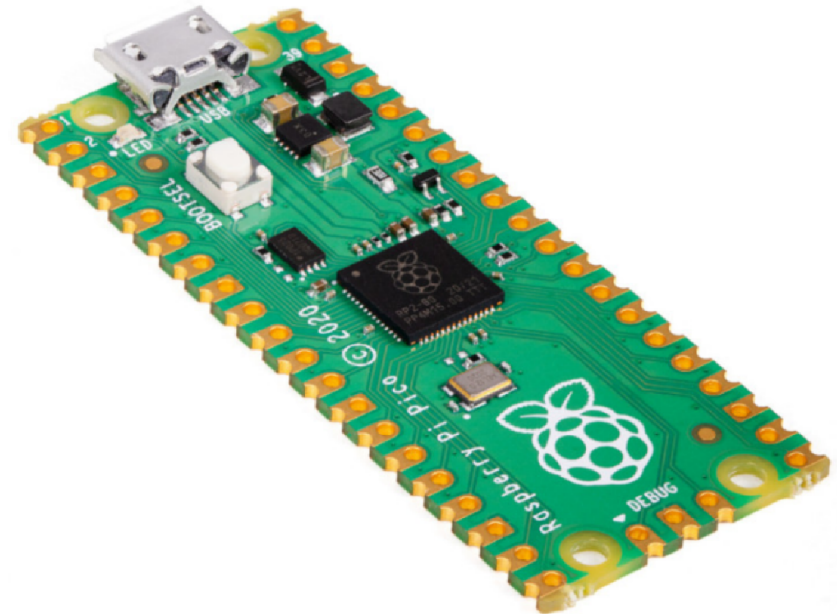
**Možnosti realizace SDI**

**SDI , které jsou k dispozici**

**PICO jako SDI (osciloskop, ...)**

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

**Workshop s PICO**



# Mikropočítač - Raspberry PI PICO

---

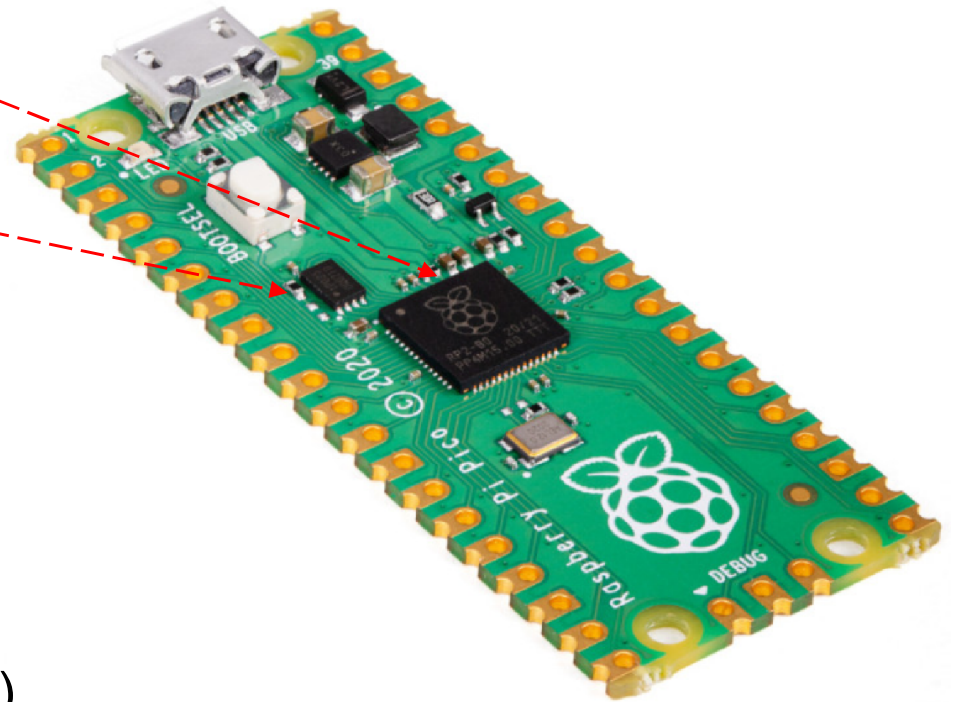
**Mikropočítač Raspberry PI PICO**

**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)

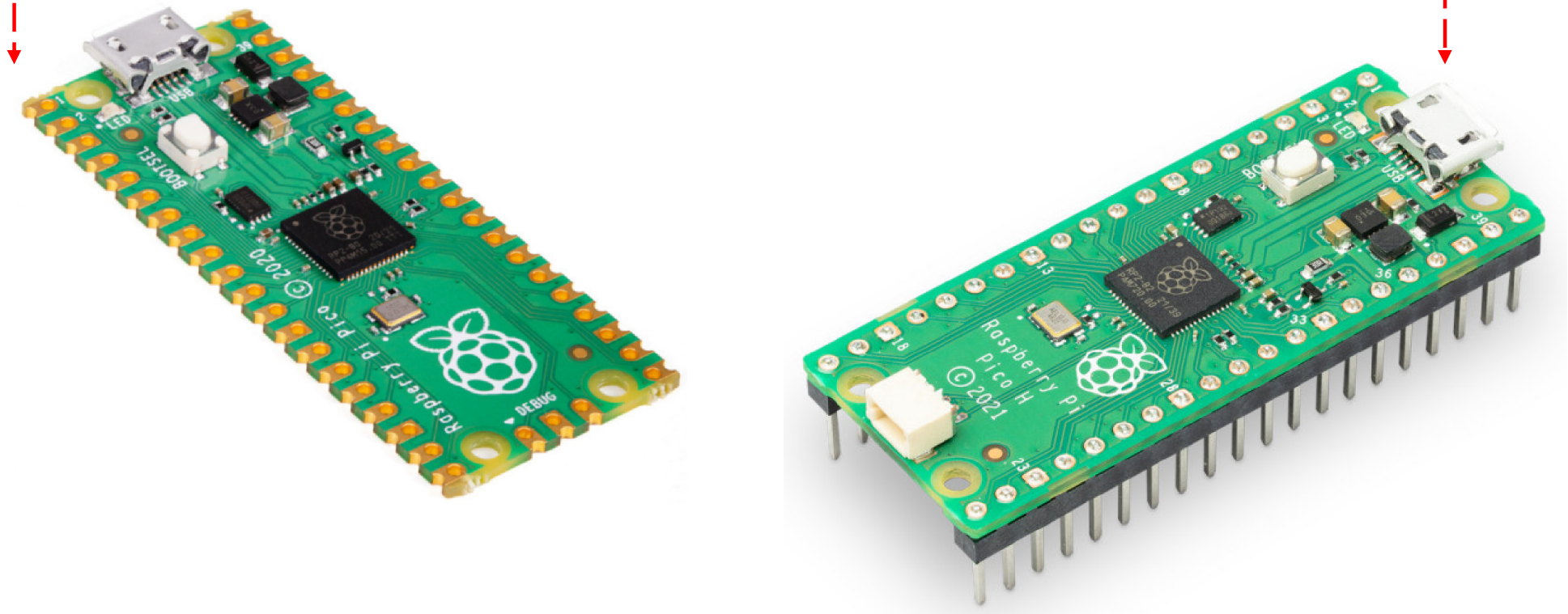
<https://www.raspberrypi.com/products/raspberry-pi-pico/>

– Raspberry PI PICO - pořízení v ČR, RPI Shop, Č.Budějovice

<https://rpishop.cz/pico/5117-raspberry-pi-pico-0617588405587.html>

Pokud nemáme možnost pájet headery, koupit již verzi **PICO H**

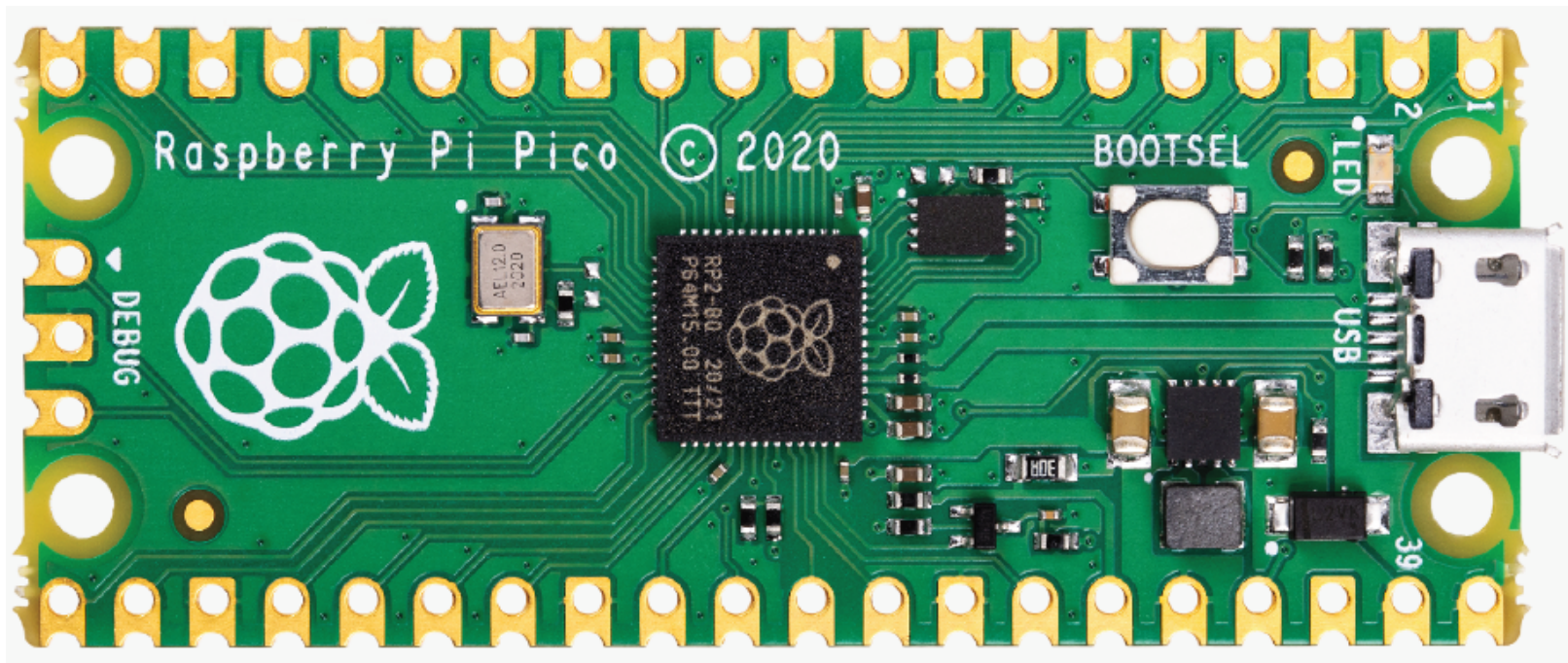
<https://rpishop.cz/raspberry-pi/5069-raspberry-pi-pico-h-5056561803180.html>



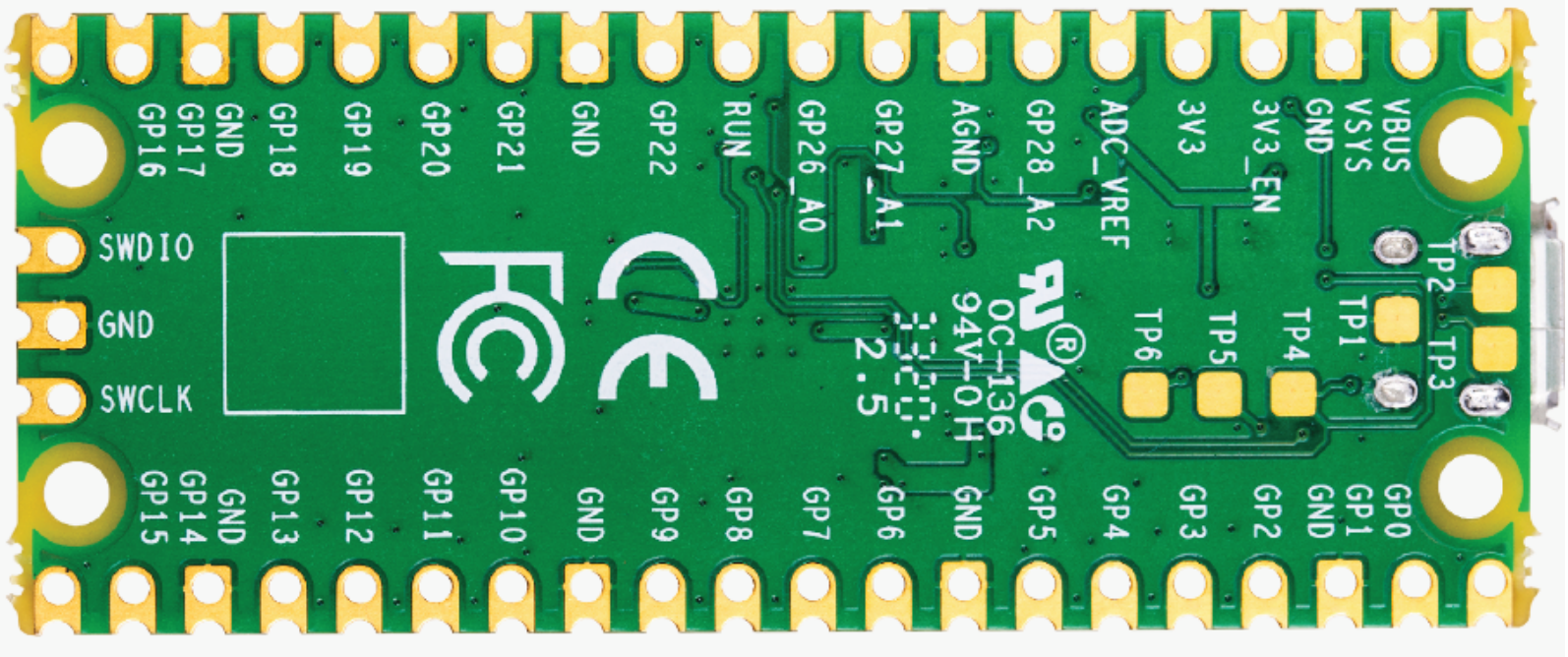
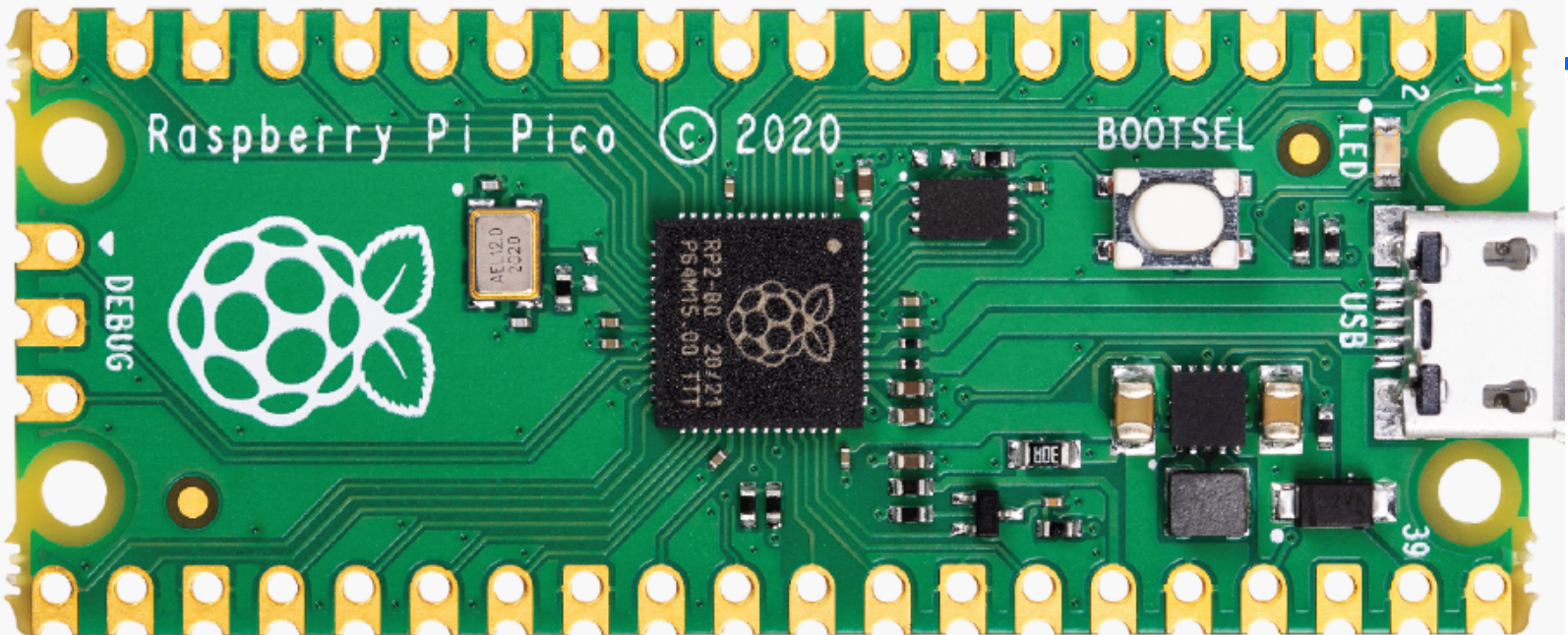


# Mikropočítač - PI PICO

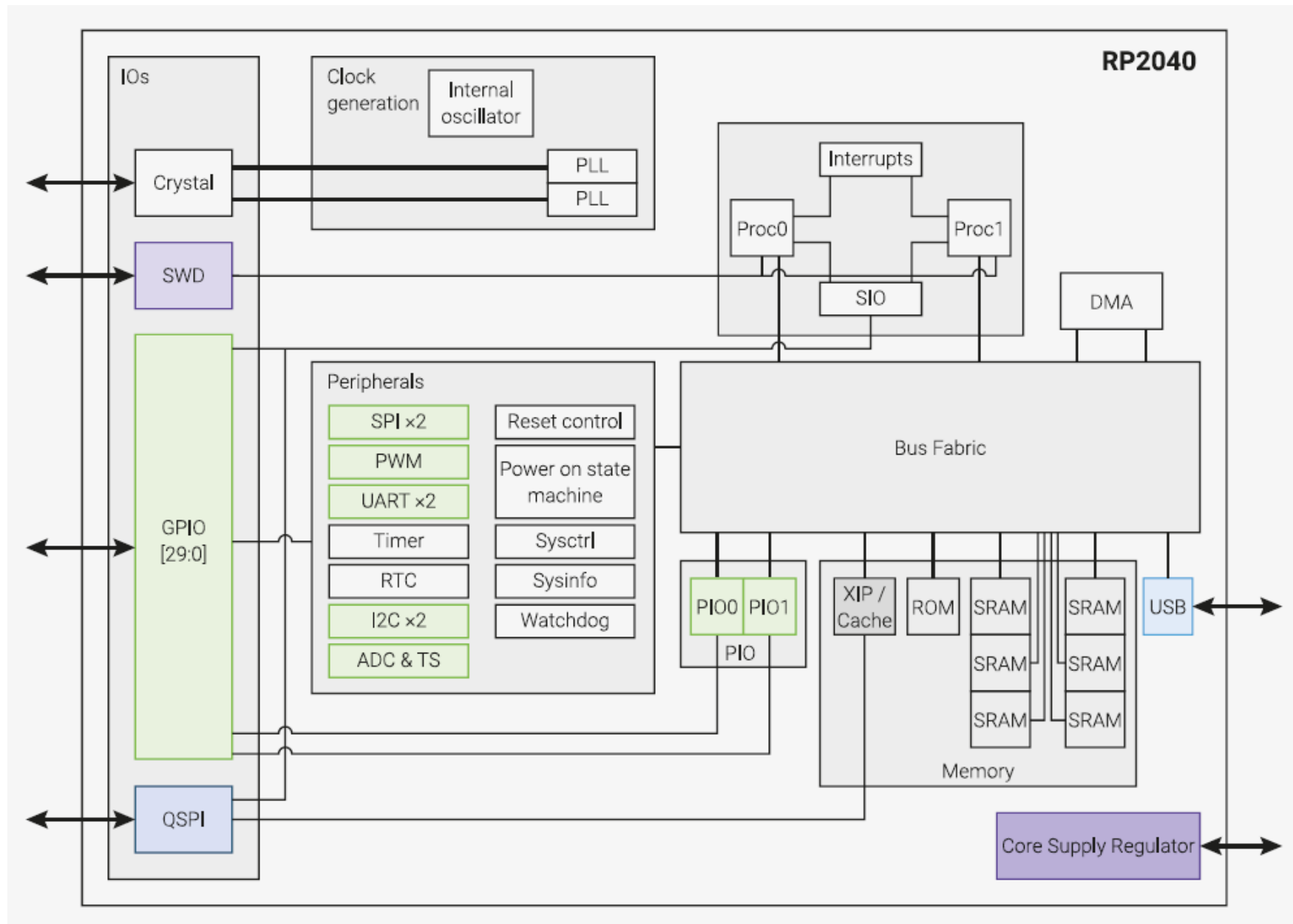
---



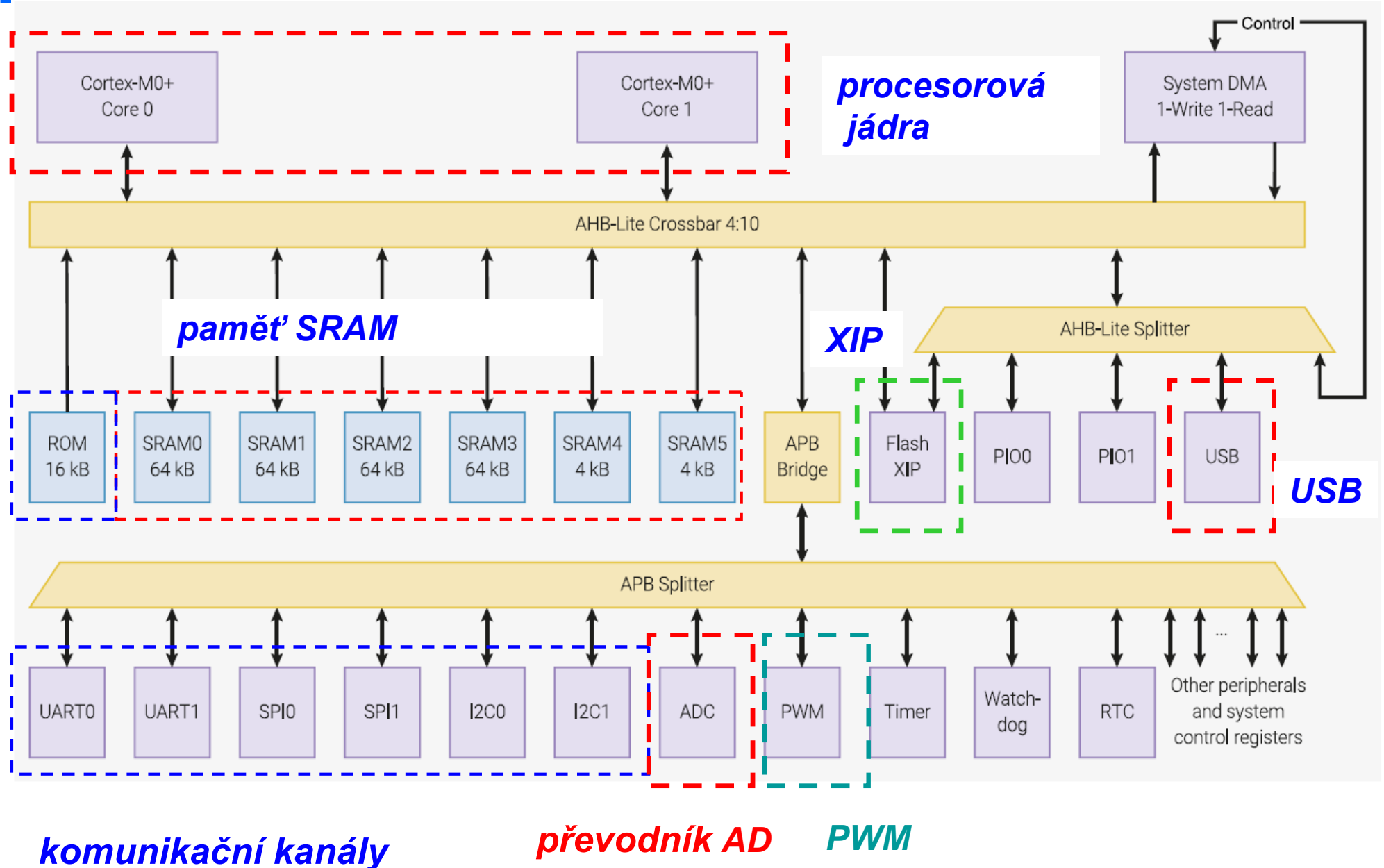




# PI PICO struktura



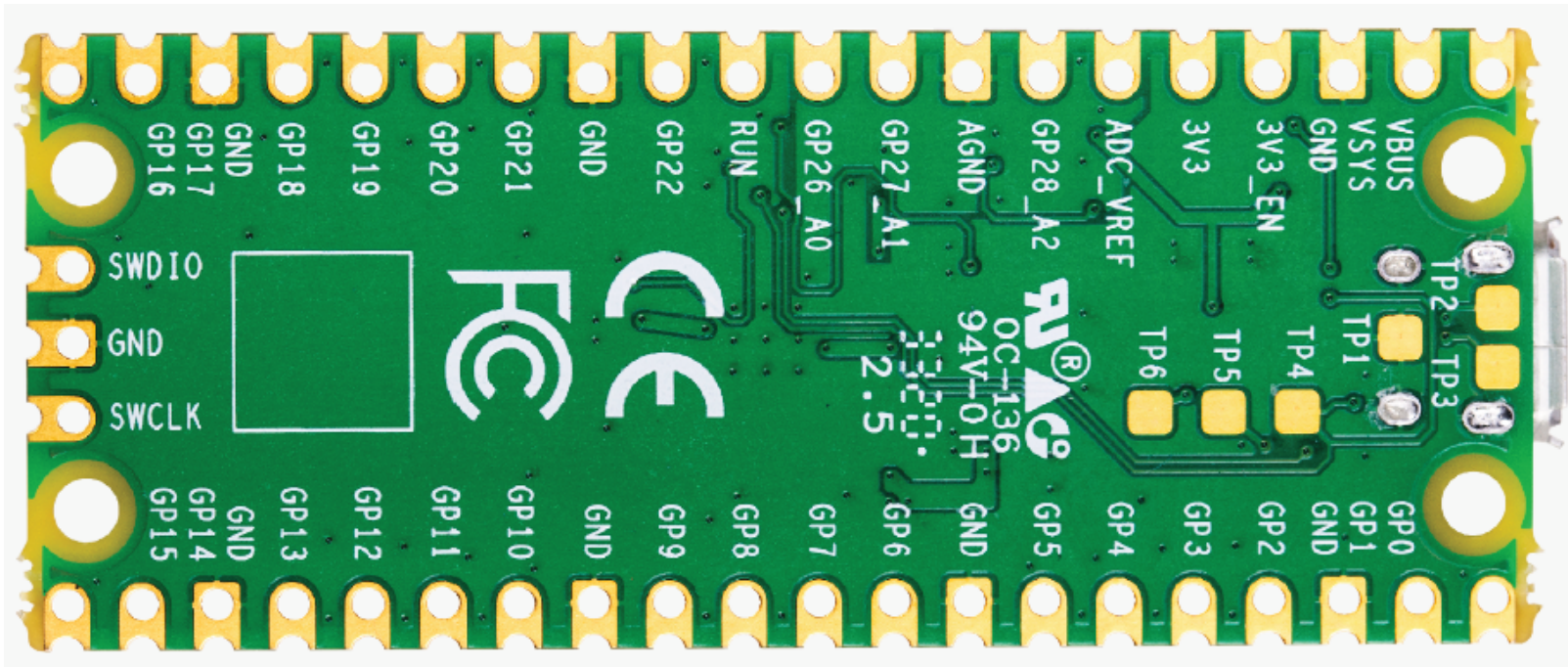
# PI PICO Paměťové bloky a periferie





# Mikropočítač - Raspberry Pi PICO

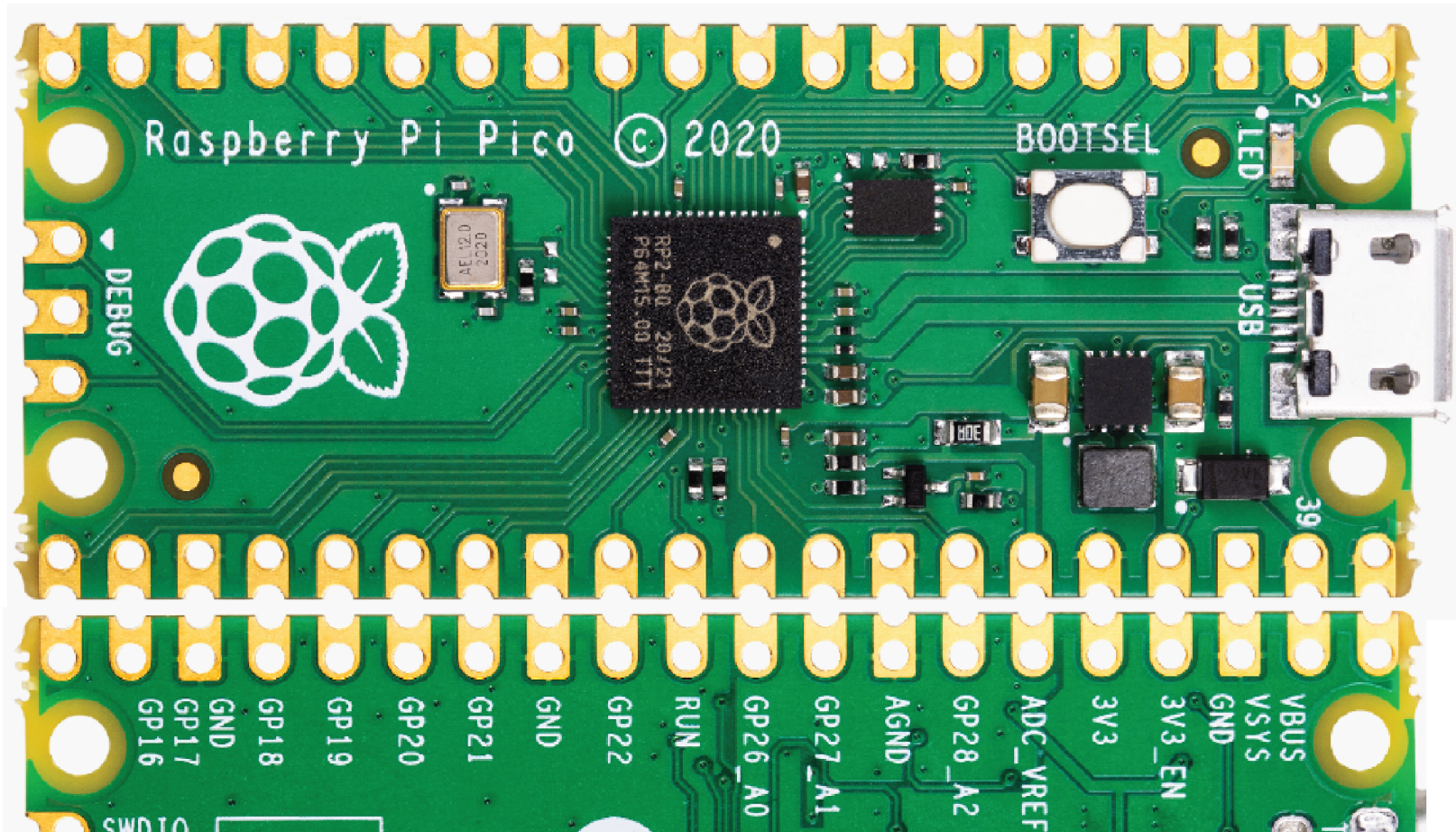
---





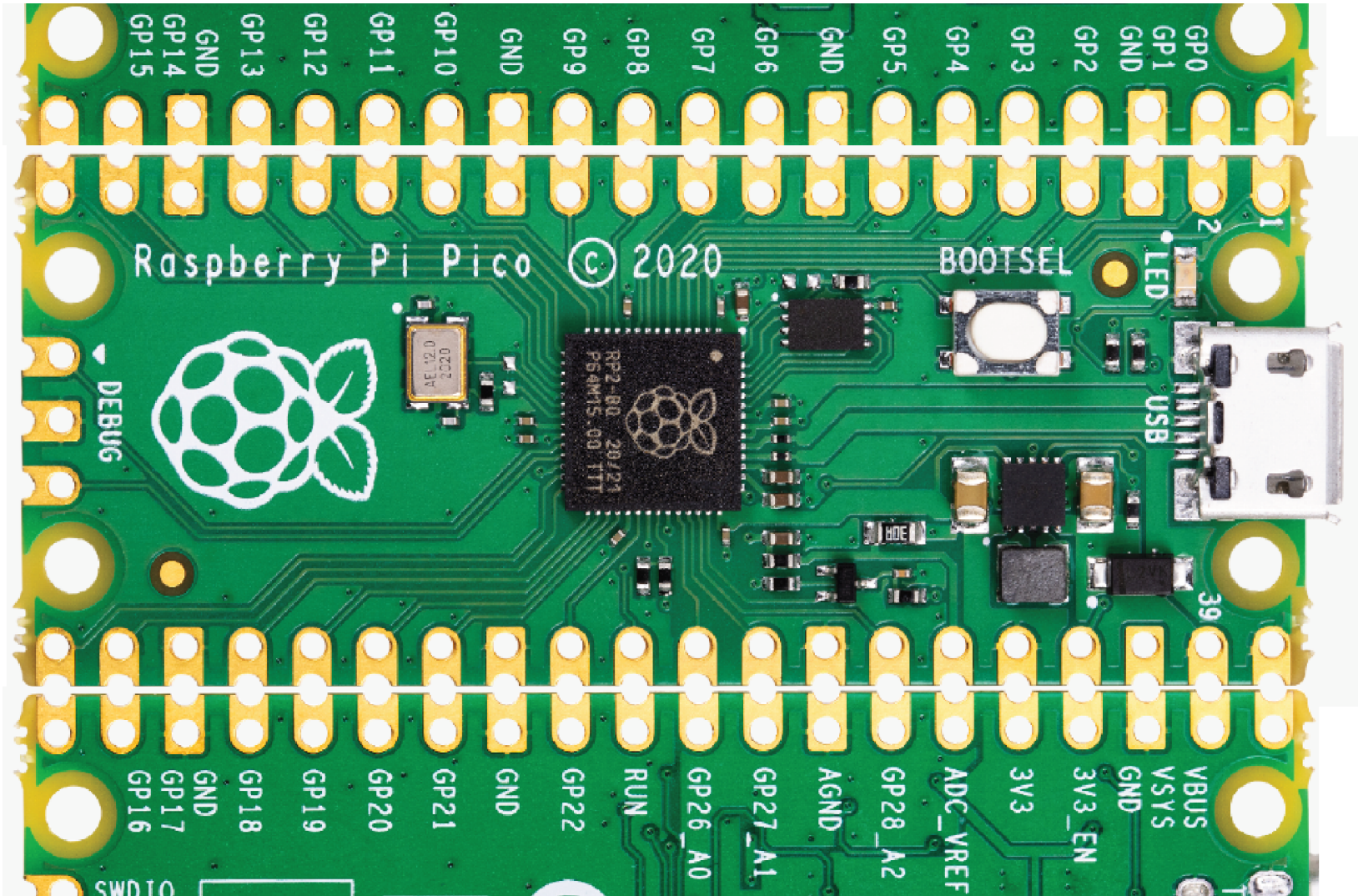
# PI PICO - strana s vývody pro naše experimenty

---



# PI PICO vývody

---



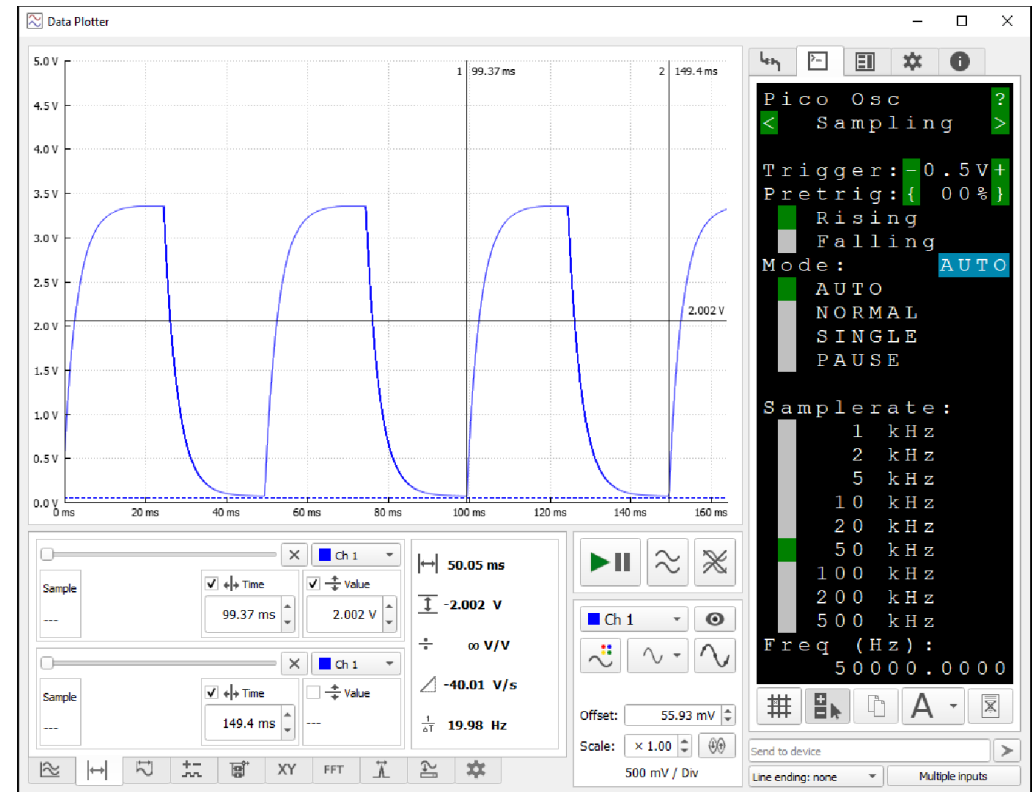
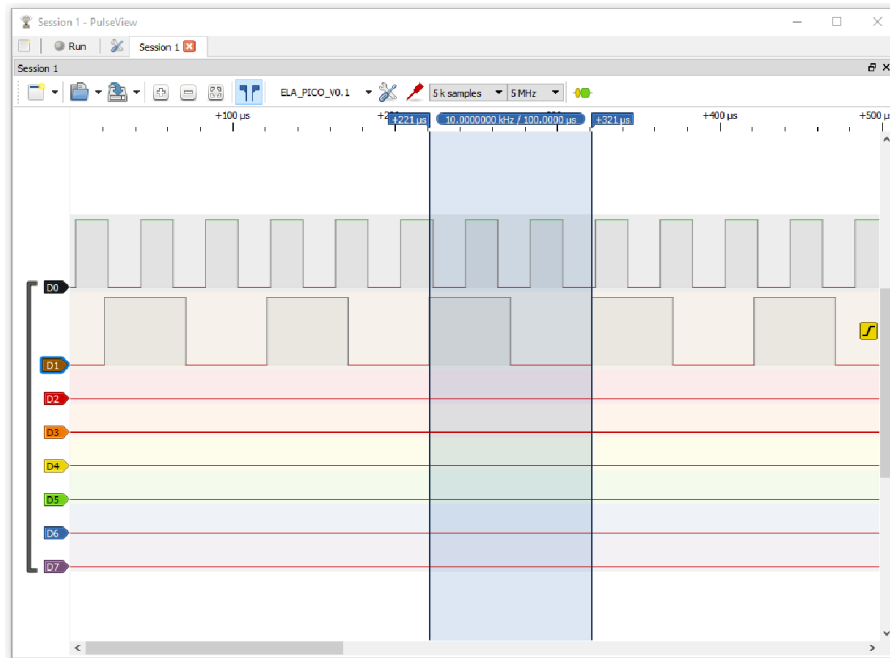


# Přístroje s RP PICO pro experimenty

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

Přístroje – osciloskop, logický analyzátor

Stačí jen nahrát náš SW (volné)



Další přístroje - např. osciloskop pro Arduino na

<https://embedded.fel.cvut.cz/SDI>

# Přístroje SDI k dispozici pro laboratoře

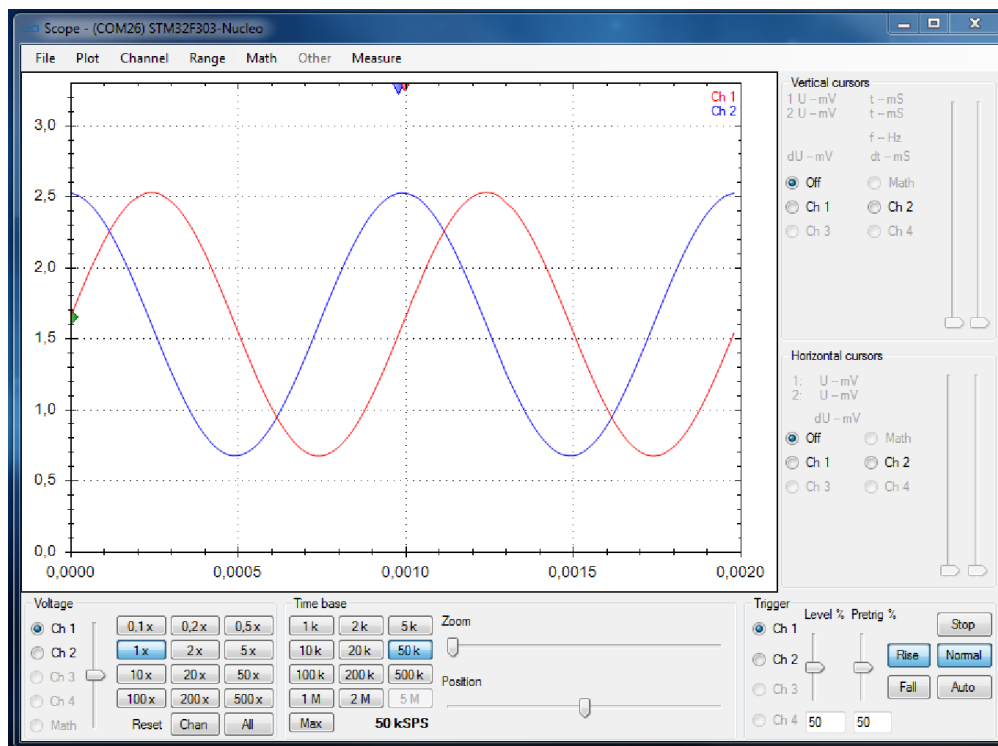
SDI na bázi STM32, STM32F303RE, STN32F103, STM32F042

Více – del

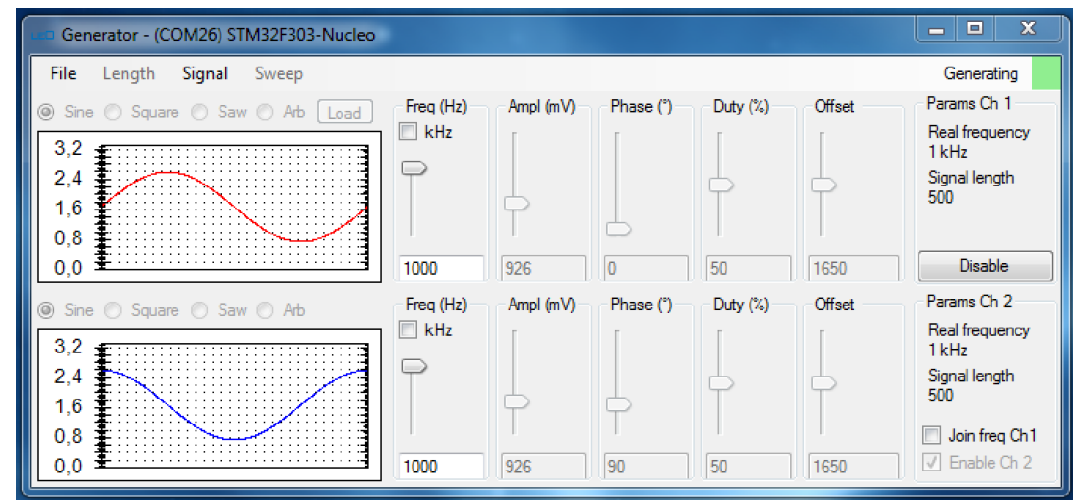
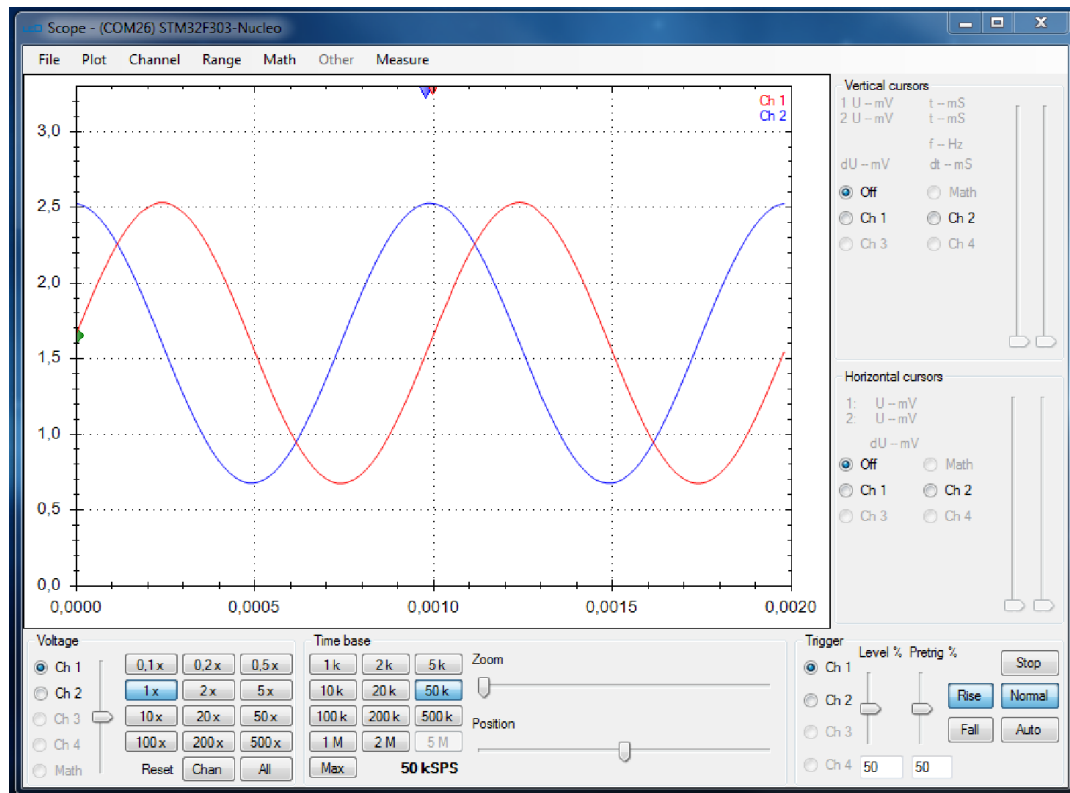
Raspberry PI PICO

Arduino

<https://embedded.fel.cvut.cz/SDI>

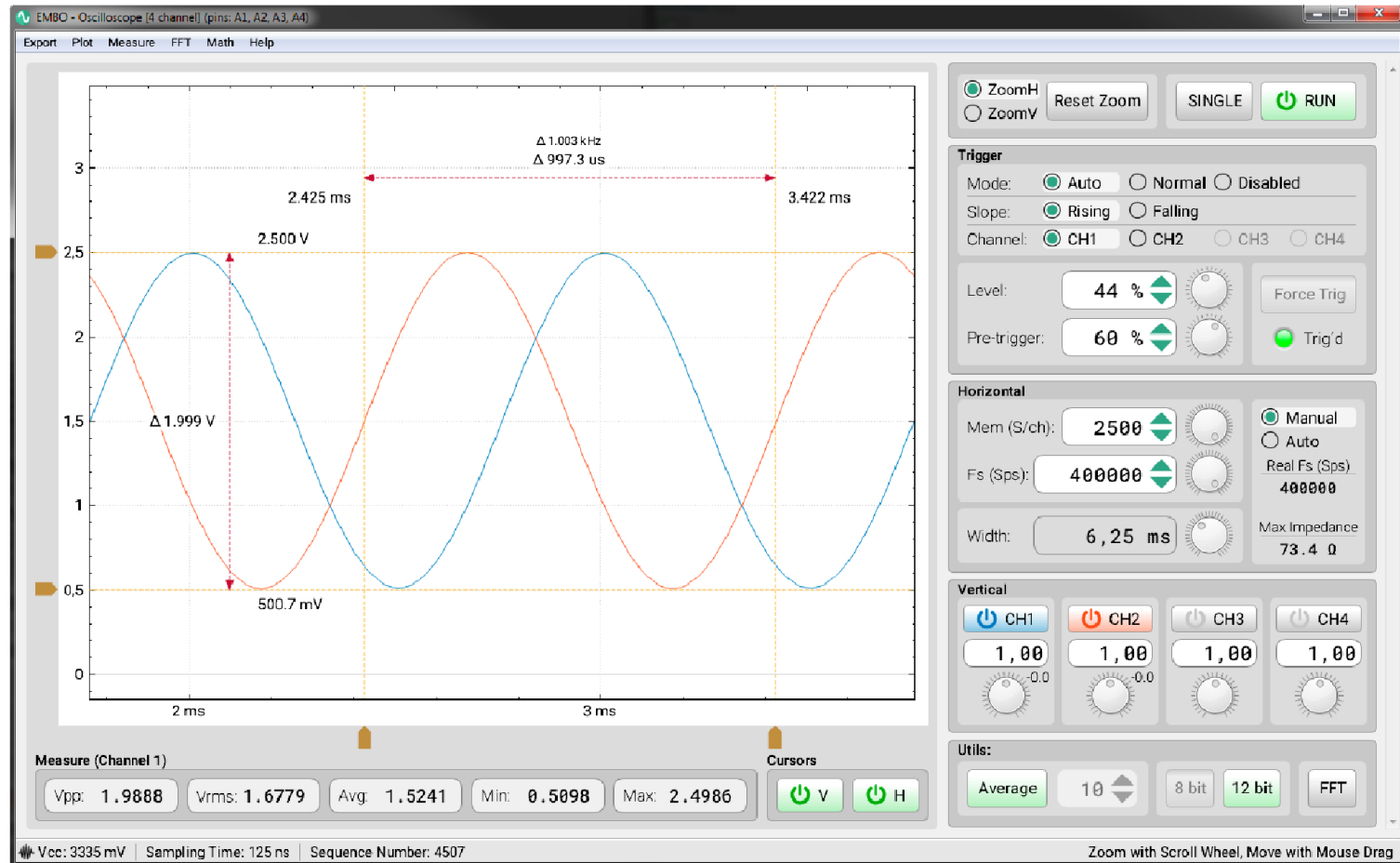


<https://embedded.fel.cvut.cz/platformy/leo>

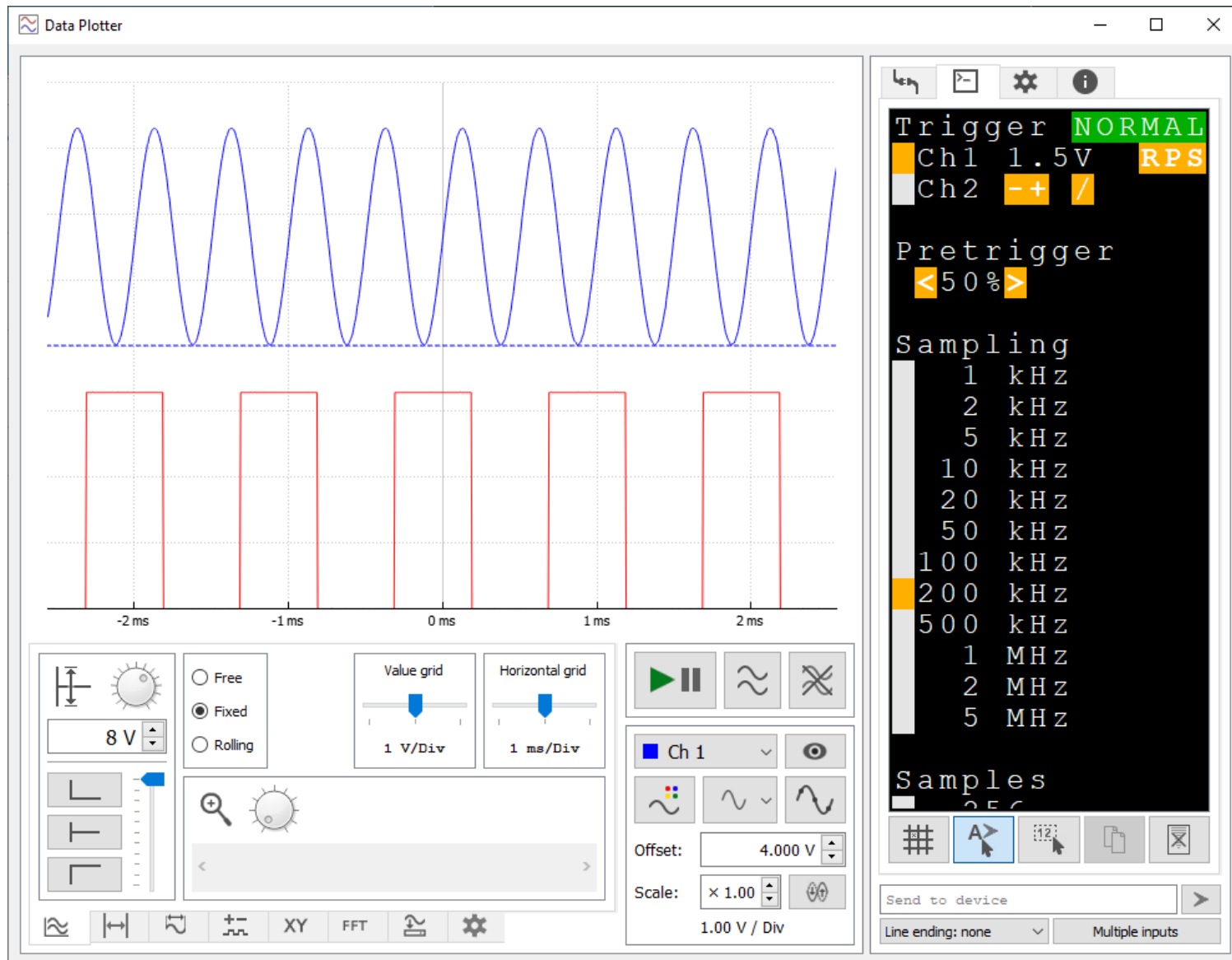




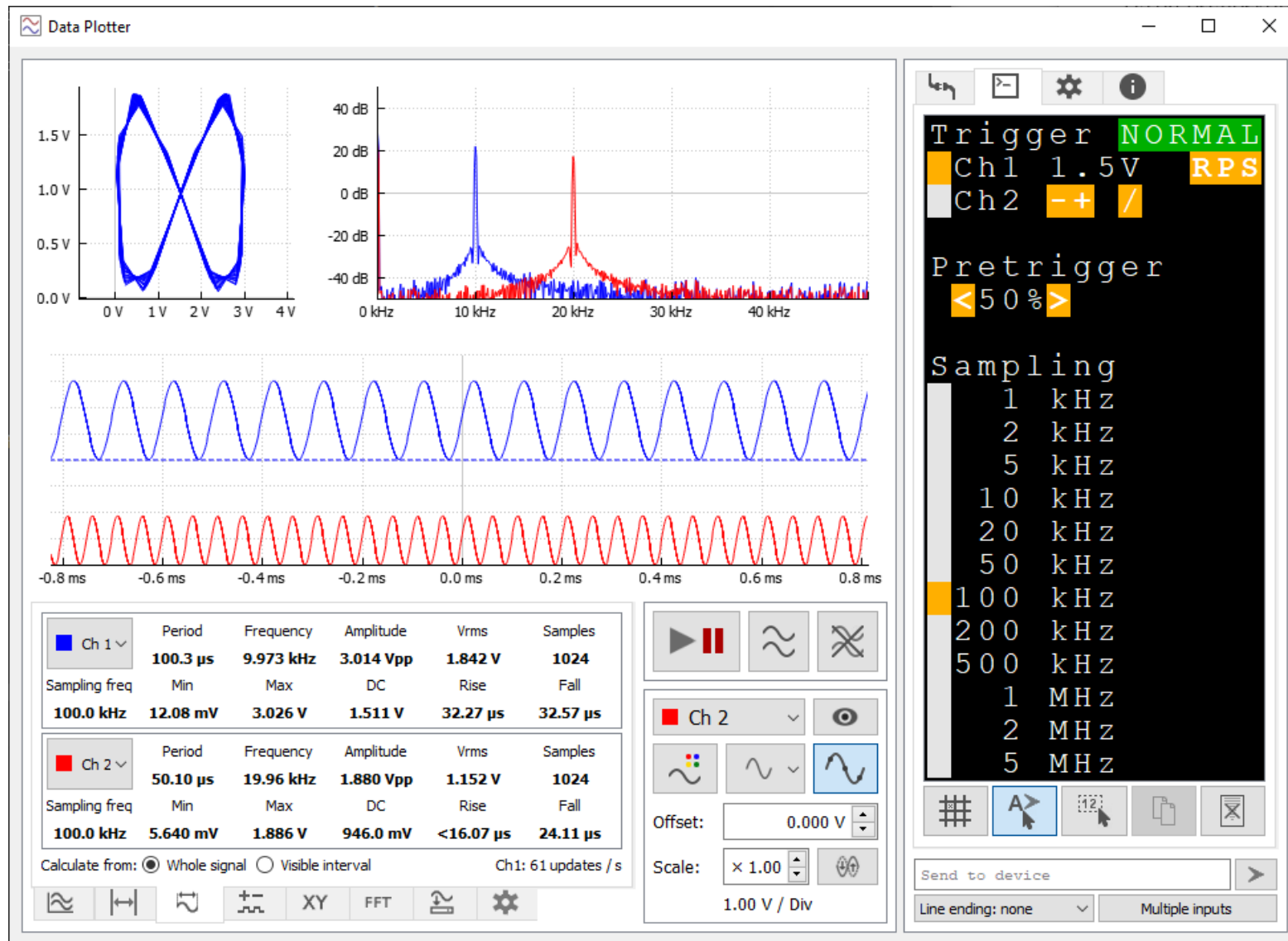
<https://embedded.fel.cvut.cz/platformy/embo>



# <https://embedded.fel.cvut.cz/platformy/dataplotter>



# <https://embedded.fel.cvut.cz/platformy/dataplotter>

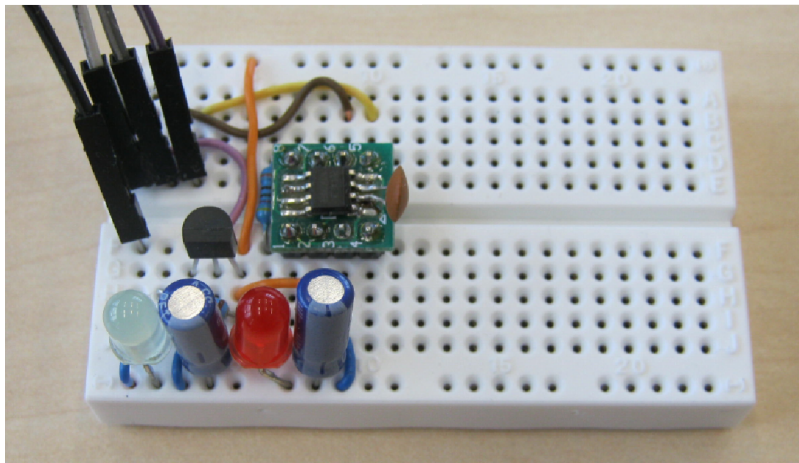
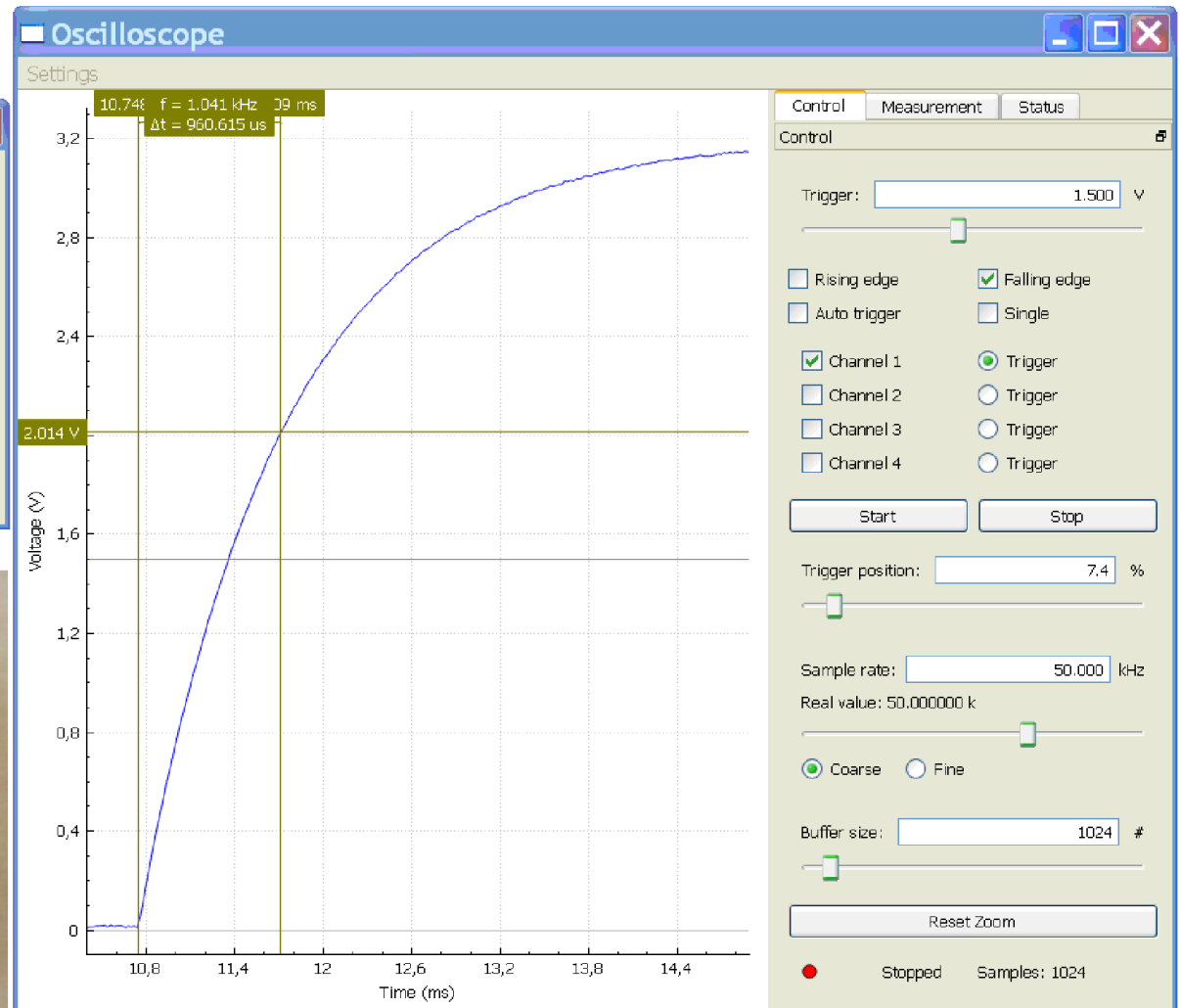


### PWM output

Frequency:  Hz  
Real value: 50.000000

Coarse  Fine

Duty cycle:  %



# **Možnosti využití SDI v laboratořích**

---

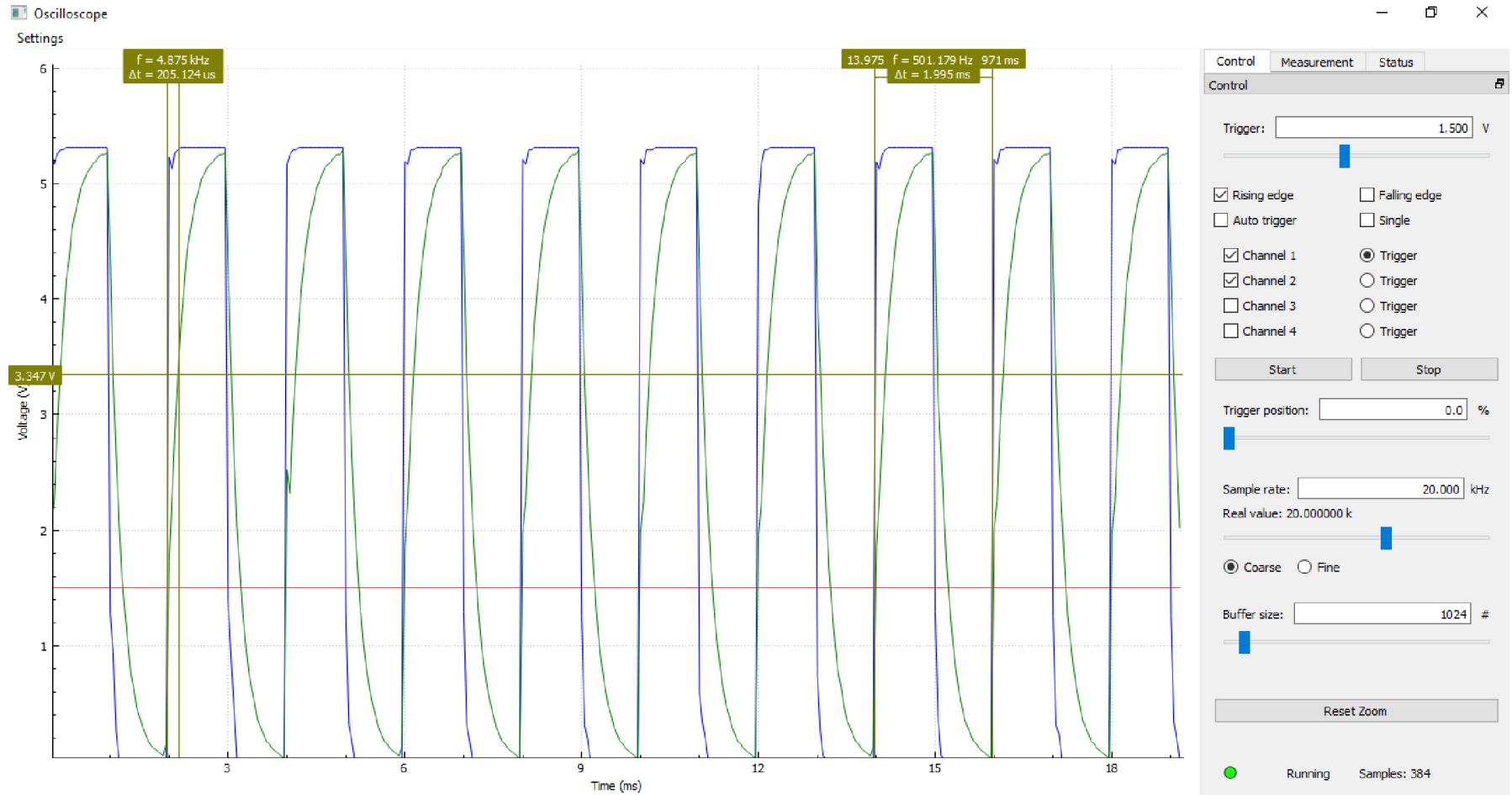
**Možnosti využití SDI v laboratořích fyziky, případně v odborných laboratořích – elektronika**

**Domácí práce studentů na projektech**

**Možnosti SDI z hlediska funkčních vlastností**

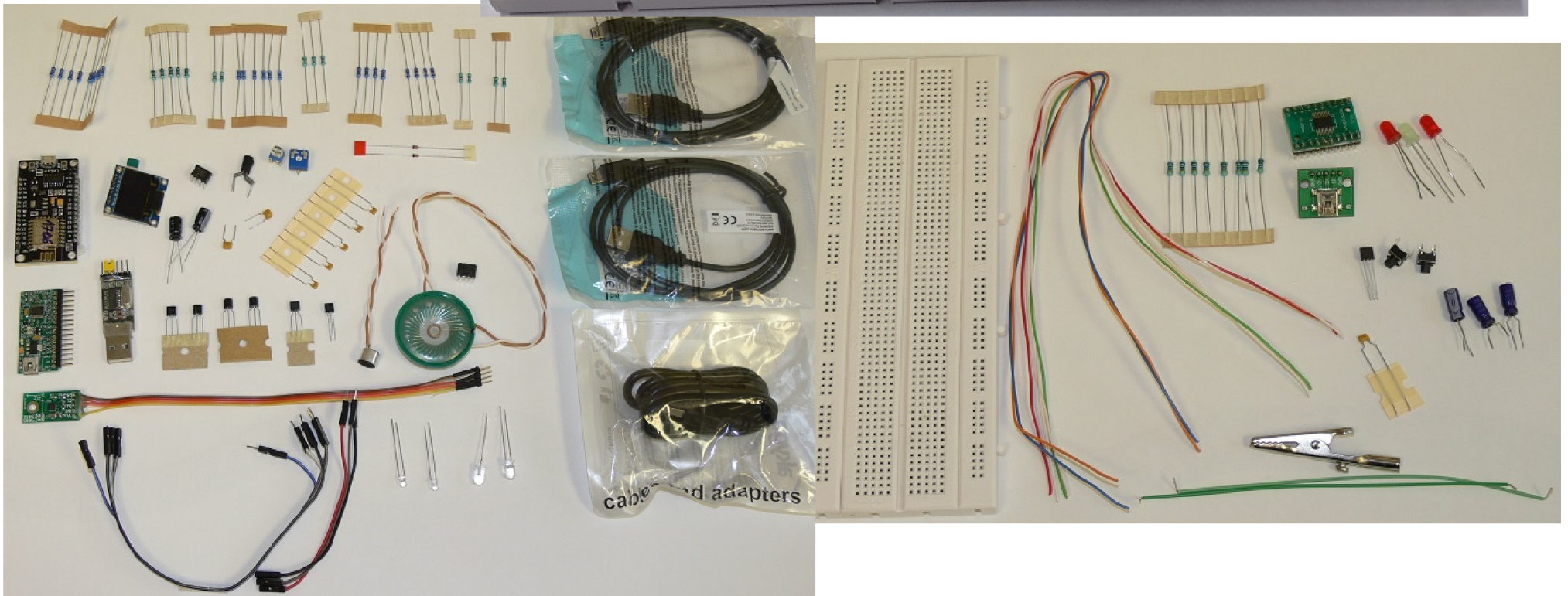
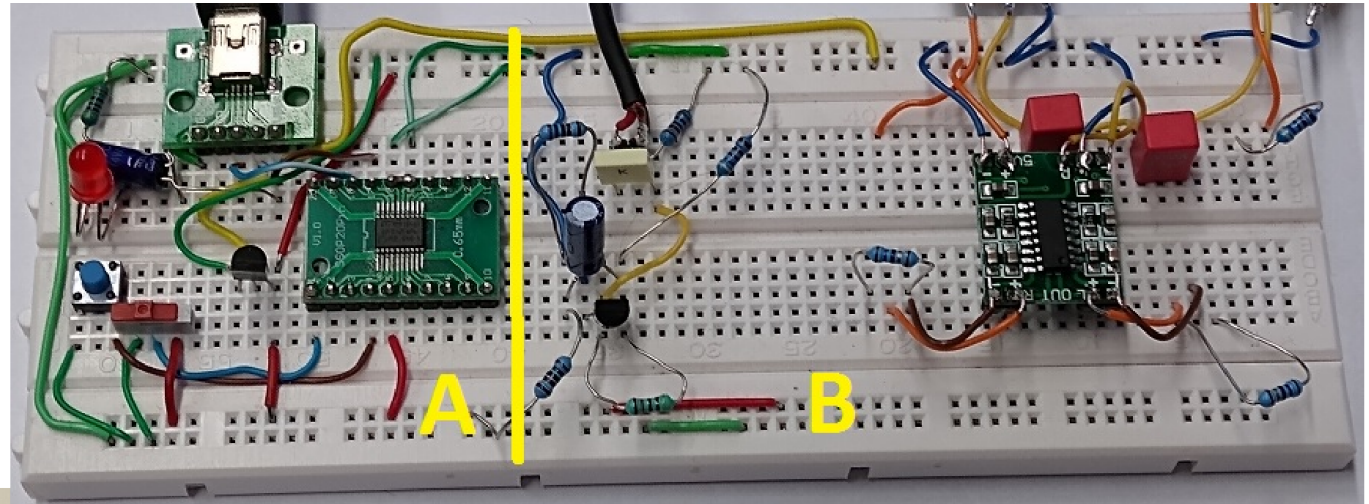


# <https://embedded.fel.cvut.cz/platformy/Arduino/oscilloskop>



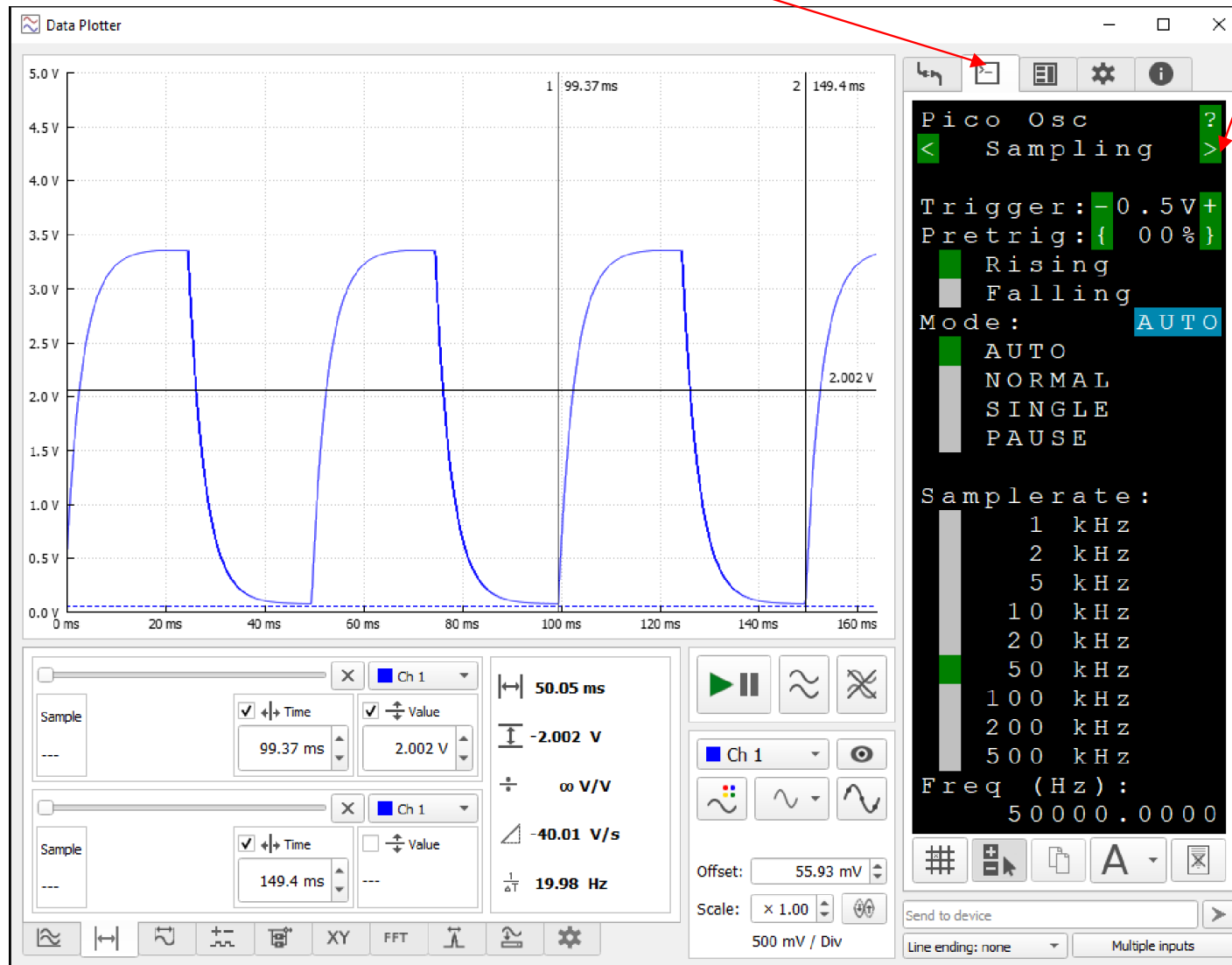
# SDI v distanční výuce

Info dle <https://embedded.fel.cvut.cz/kurzy/lpe/LPE2021>



# Panel osciloskopu s PC aplikací Data plotter

- Připojení k USB volba menu



# RP PICO jako měřicí přístroj

---

Pro pokročilejší zájemce – je možný experiment; nahráním našeho programu se RP PICO změní v **jednoduchý digitální oscioskop a impulsní generátor**.

Možné **experimenty** k provedení zde:

**Měření odezvy integračního članku RC**

**Sledování blikání zářivek pomocí fototranzistoru**

*Je však třeba si experiment sestavit – **zapojit** nakontaktním poli*

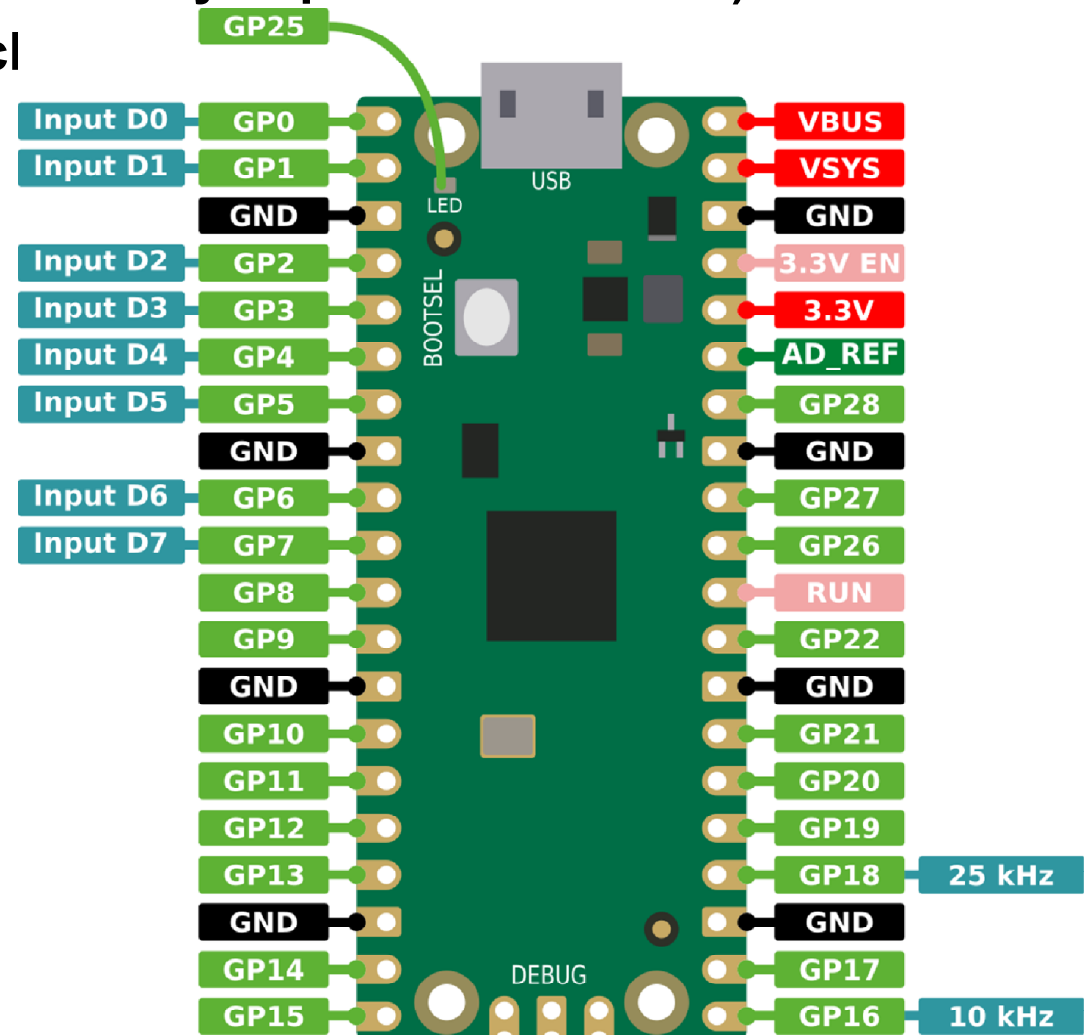
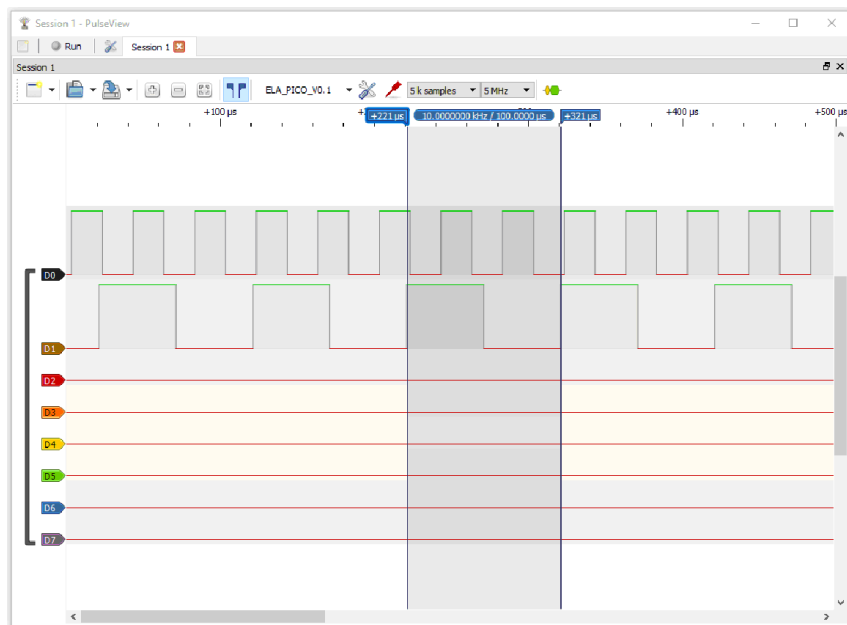
# Logický analyzátor ELA s RP PICO

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

Logický analyzátor - přístroj pro sledování časového průběhu logických signálů (např. na programem ovládaných pinech ARDUINO)

Podporuje analýzu komunikačních

UART, SPI, I2C BUS,.....





# Experiment s osciloskopem s PICO

---

program osciloskop [vanecvit-pico-osc-2022-05-10.uf2](#)

Je již náhrán v deskách

Na **PC** spustíme **aplikaci DATA Plotter**

Experimenty s osciloskopem

První - můžeme připojit PWM výstup na vstup osciloskopu Ch1 a pozorovat jeho signál.

Druhý- lze sledovat blikání zářivek pomocí fototranzistoru připojeného k osciloskopu.

Fototranzistor- **delší** vývod je **emitor**. Rezistor 10 000 = 10 kiloOhmů mezi emitor a GND, Kolektor na napájení **3V3** (+3,3 V).

# Nahrávání programu

---

**Stisk bílého tlačítka BOOT na kitu, stisk RESET, uvolnění RESET, uvolnění BOOT.**

**Kit se nyní počítači jeví jako externí paměť FLASH, dokteré nakopírujeme soubor s příponou **.UF2****

**Máme dva kity**

**První s firmware osciloskop**

**Druhý s micropython, tedy nemusíme nahrávat.**

**Nicméně později bude třeba náhrát firmware -Micropython**

# 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*, které jsou schopny komunikovat s deskou přes USB.

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

# 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.*

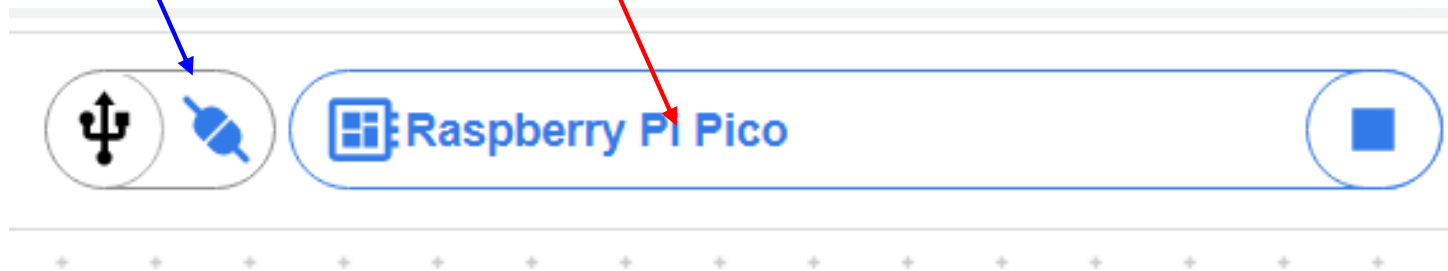
*Ke stažení je na [micropython.org](https://micropython.org)*

<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

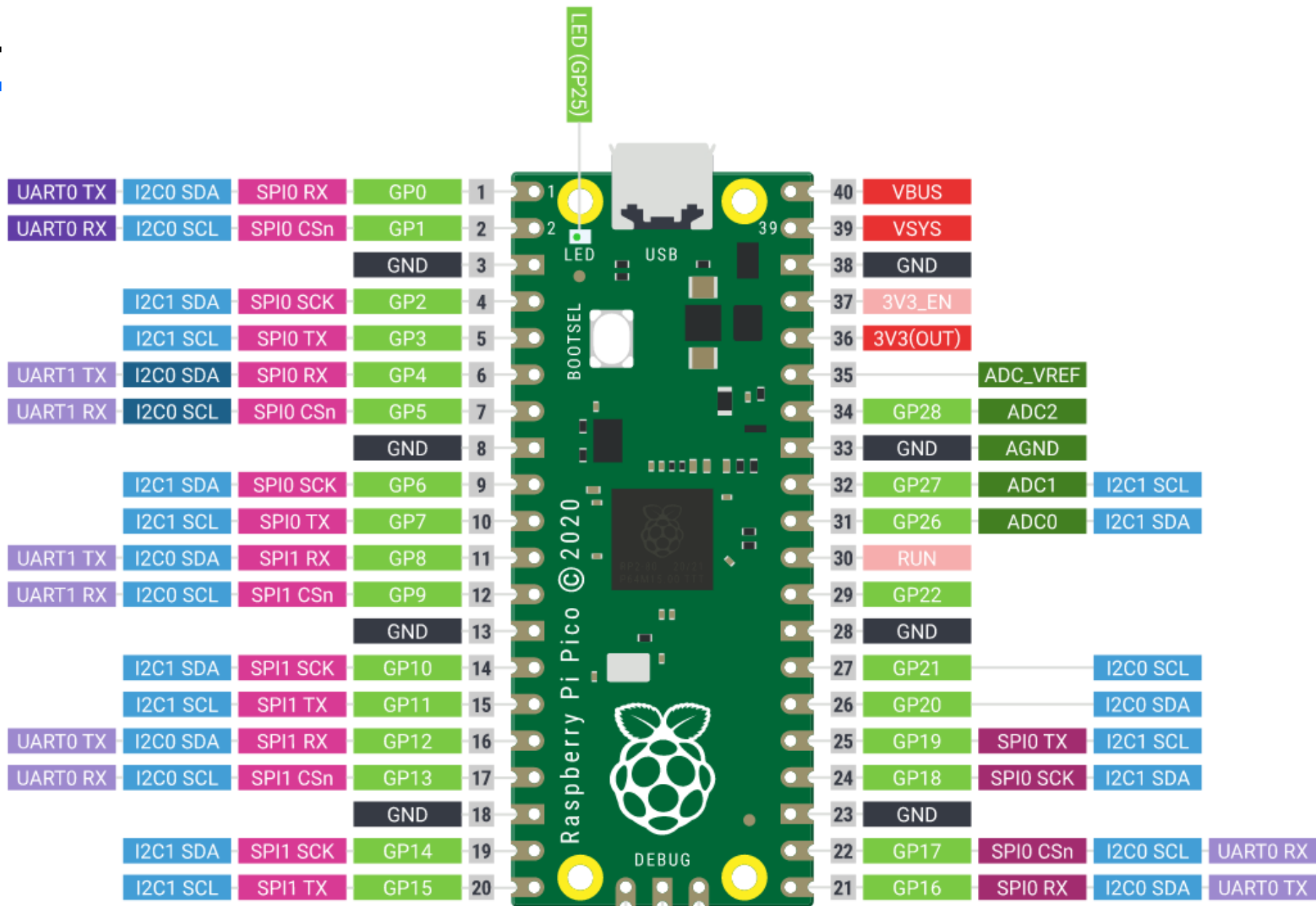
▶ Network

▶ micropython

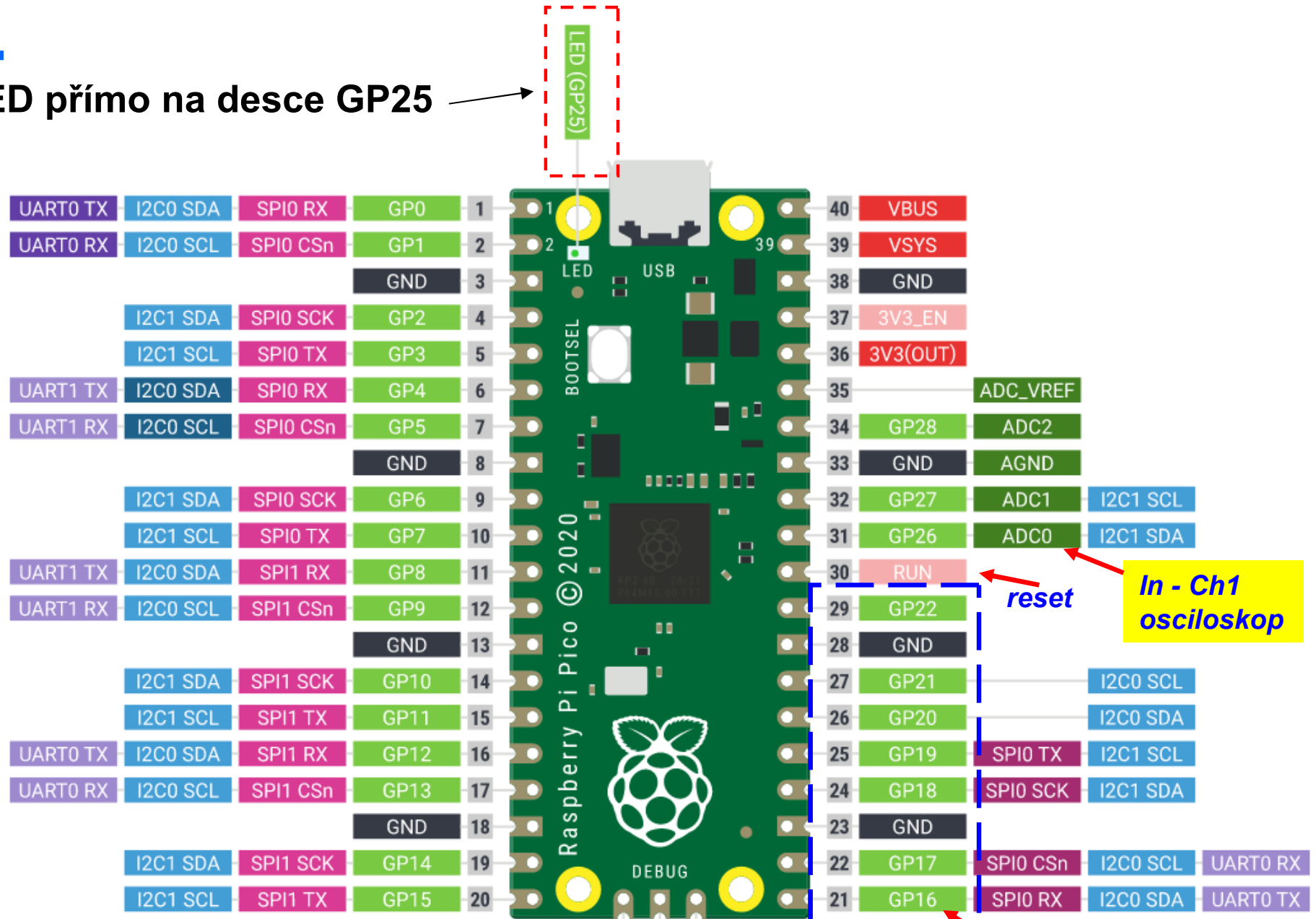
**Připojení** desky, volba rozhraní **USB**  
někdy potřebuje **opakovat dvakrát**  
***připojit, odpojit, připojit***







# LED přímo na desce GP25



Toto budeme využívat my

PWM OUT osciloskop

# Zapojení experimentální desky- vývody

---

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 *osciloskop PWM OUT*

**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

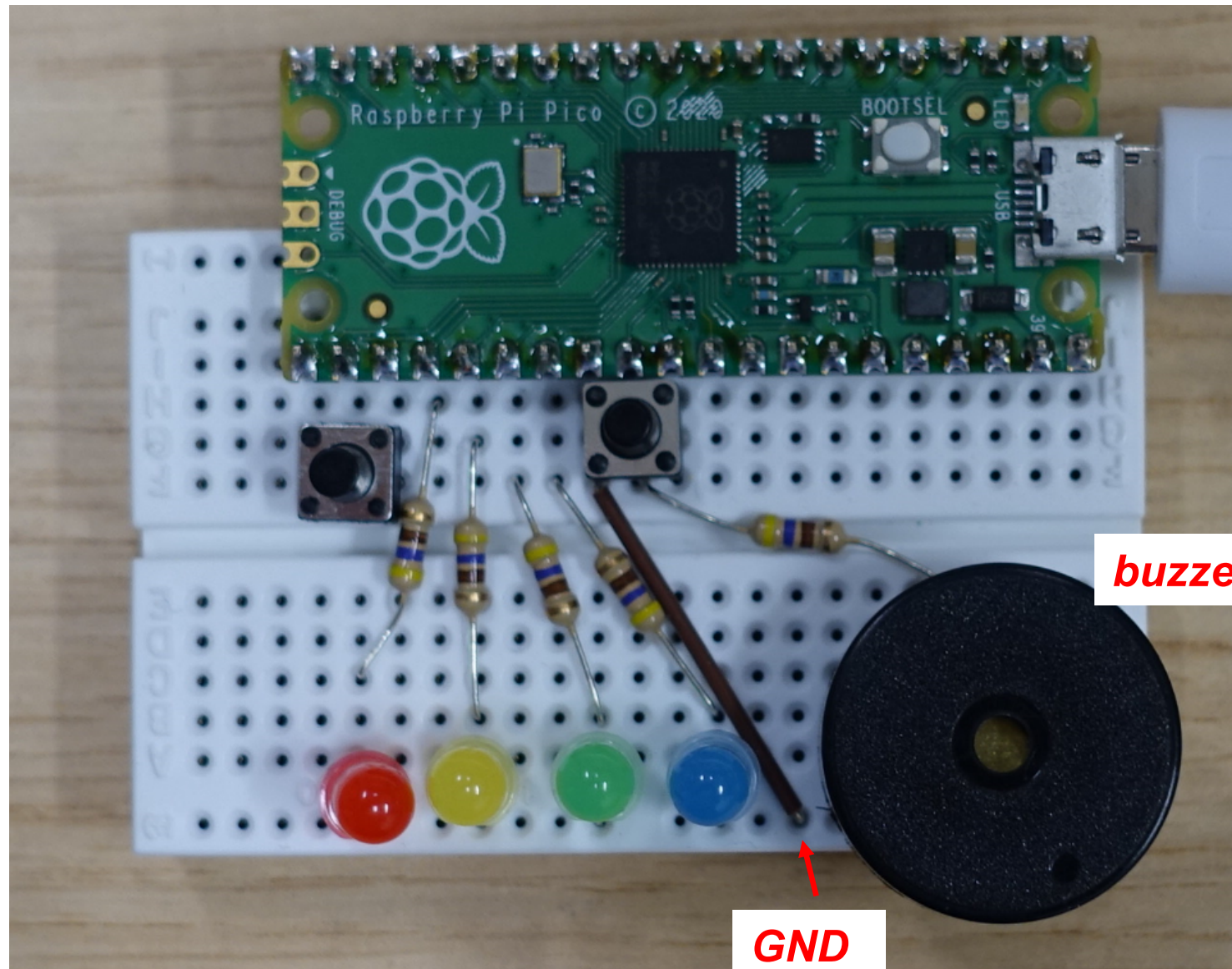
**GP26** ..... *Ch1 In Osciloskop*

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

Úroveň „*true*“ na pinu LED rozsvítí,

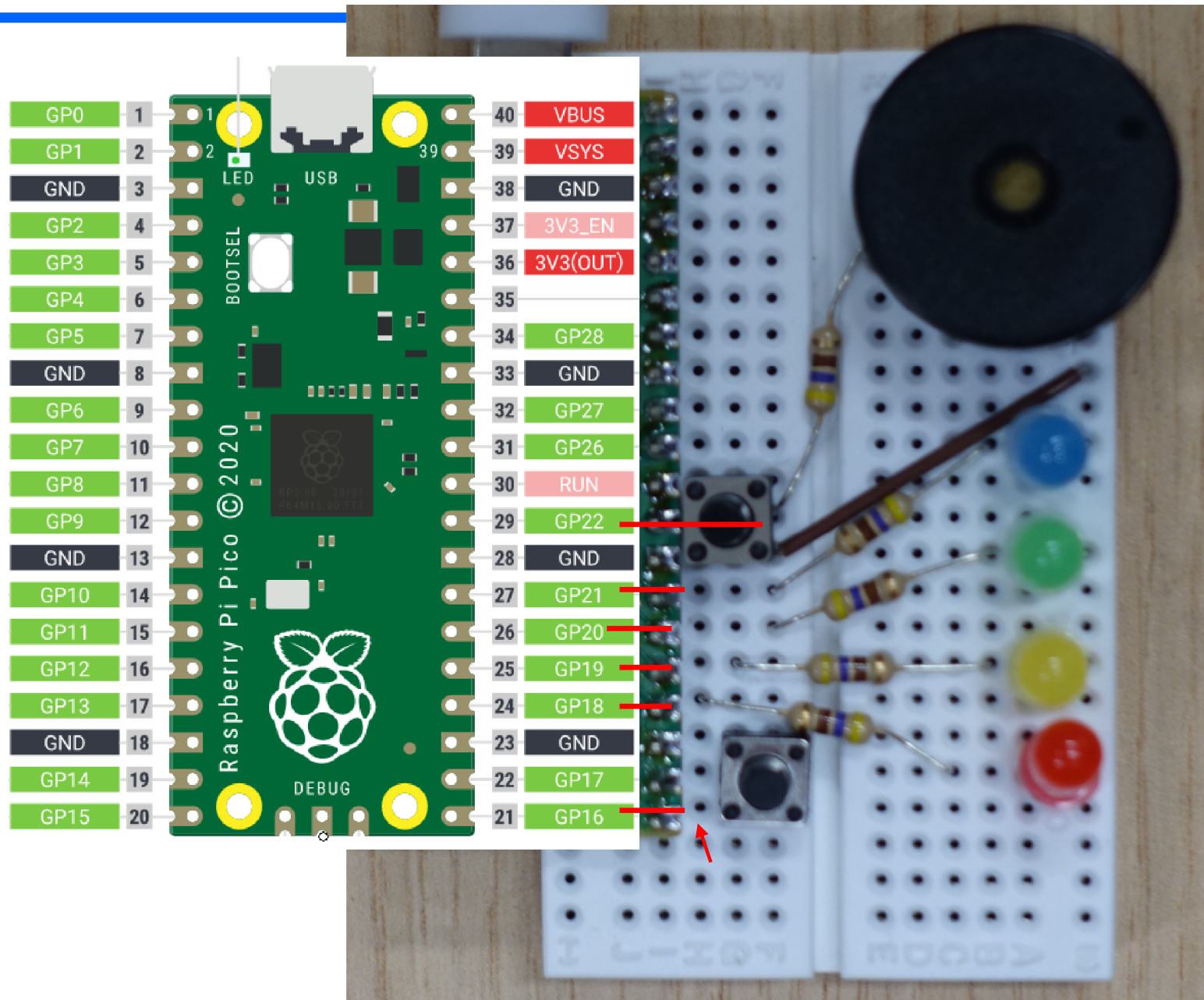
úroveň „*false*“ LED na pinu zhasne

# Osazení experimentálního pole





# Výstupní piny





.

---

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**

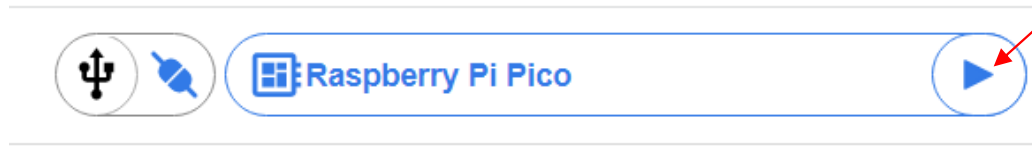
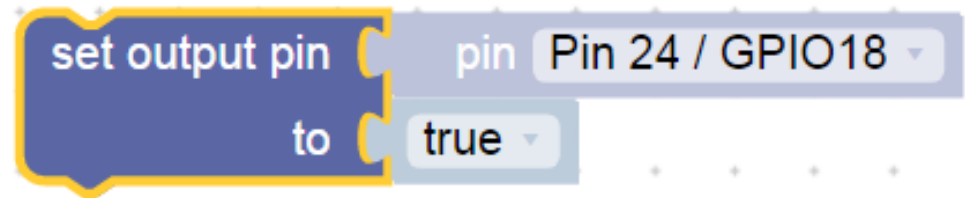
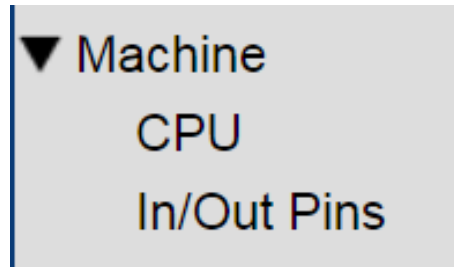
# Rozsvítit LED- nástroje

The image shows the BIPES Blocks interface. On the left is a category menu with the following items: Logic, Loops, Math, Text, Lists, Variables, Functions, BIPES, Python, Timing, Machine (expanded to show CPU and In/Out Pins), and Display. The 'Machine' category is highlighted with a red dashed box. On the right is a workspace titled 'In/Out Pins' containing three blocks:

- A 'pin' block with a dropdown menu set to 'LED / GPIO25'.
- A 'set output pin to' block with a dropdown menu set to 'true' and a 'pin' dropdown menu set to 'LED / GPIO25'.
- A 'read digital input' block with a dropdown menu set to 'true' and a 'pin' dropdown menu set to 'LED / GPIO25'.

# 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 Je 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,  
vše provede několikrát a ukončí se LOOP

Program s podmínkou

Nekonečný program s podmínkou true

# Rozsvícení a zhasnutí LED, 1x

---

```
set output pin pin Pin 24 / GPIO18 to true
delay 1 seconds
set output pin pin Pin 24 / GPIO18 to false
delay 1 seconds
set output pin pin Pin 25 / GPIO19 to true
delay 1 seconds
set output pin pin Pin 25 / GPIO19 to false
delay 1 seconds
set output pin pin Pin 26 / GPIO20 to true
delay 1 seconds
set output pin pin Pin 26 / GPIO20 to false
delay 1 seconds
```

The image shows a Scratch script for controlling three LEDs. The script consists of a sequence of blocks: 'set output pin' followed by 'to true', a 'delay 1 seconds' block, 'set output pin' followed by 'to false', and another 'delay 1 seconds' block. This sequence is repeated for three different pins: Pin 24 / GPIO18, Pin 25 / GPIO19, and Pin 26 / GPIO20. The blocks are arranged in a vertical stack, with each 'set output pin' block connected to a 'to true' or 'to false' block, which is then connected to a 'delay 1 seconds' block. The script starts with Pin 24, then moves to Pin 25, and finally to Pin 26.



# Stálý běh programu, použití nekonečné smyčky



Aktuální **běh** našeho programu – symbol **čtvereček**

Menu Loops

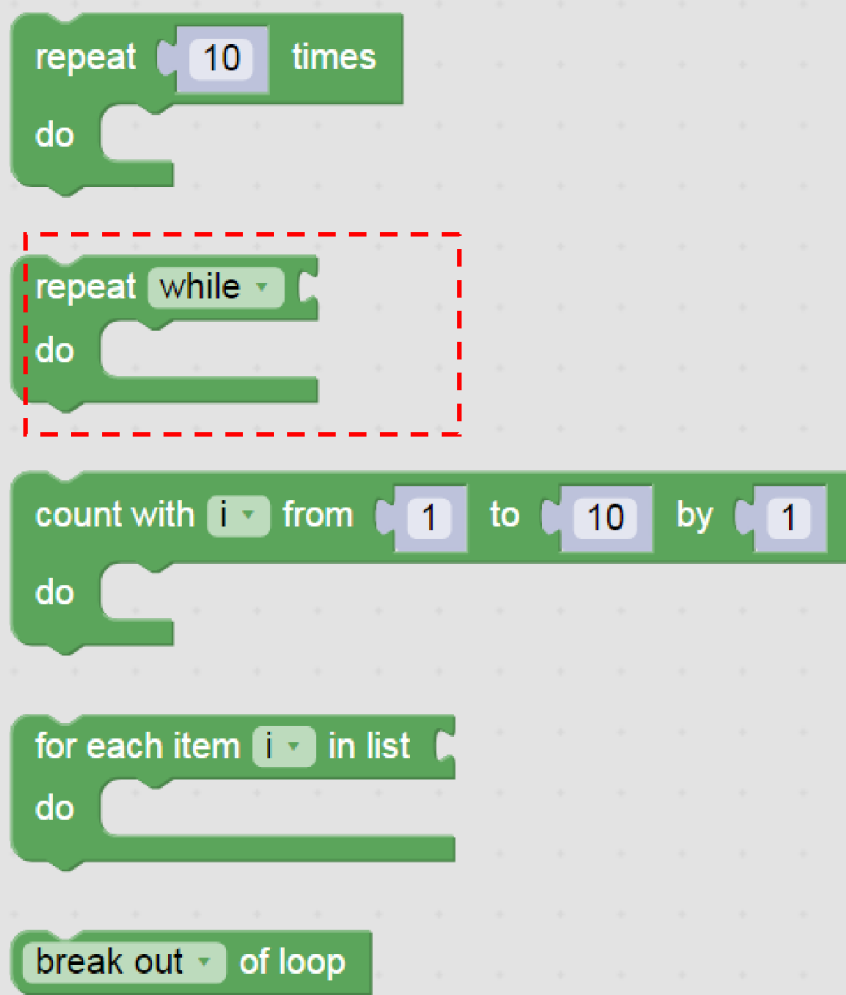
Menu Logic



The image shows a screenshot of the MicroPython IDE code editor. The code is written in a block-based language. A 'repeat while' loop is highlighted with a green border. The loop's condition is 'true'. The loop body contains four blocks: 'set output pin to true', 'delay 1 seconds', 'set output pin to false', and 'delay 1 seconds'. The 'true' block in the Logic menu is highlighted with a red dashed box, and a red dashed arrow points from it to the 'true' block in the loop's condition. The 'repeat while' loop is also highlighted with a yellow border.

# Stálý běh programu, použití nekonečné smyčky

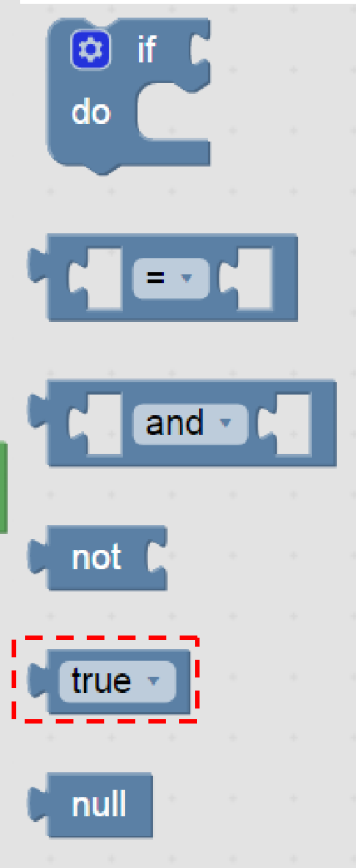
## Menu Loops



Scratch menu for loops containing the following blocks:

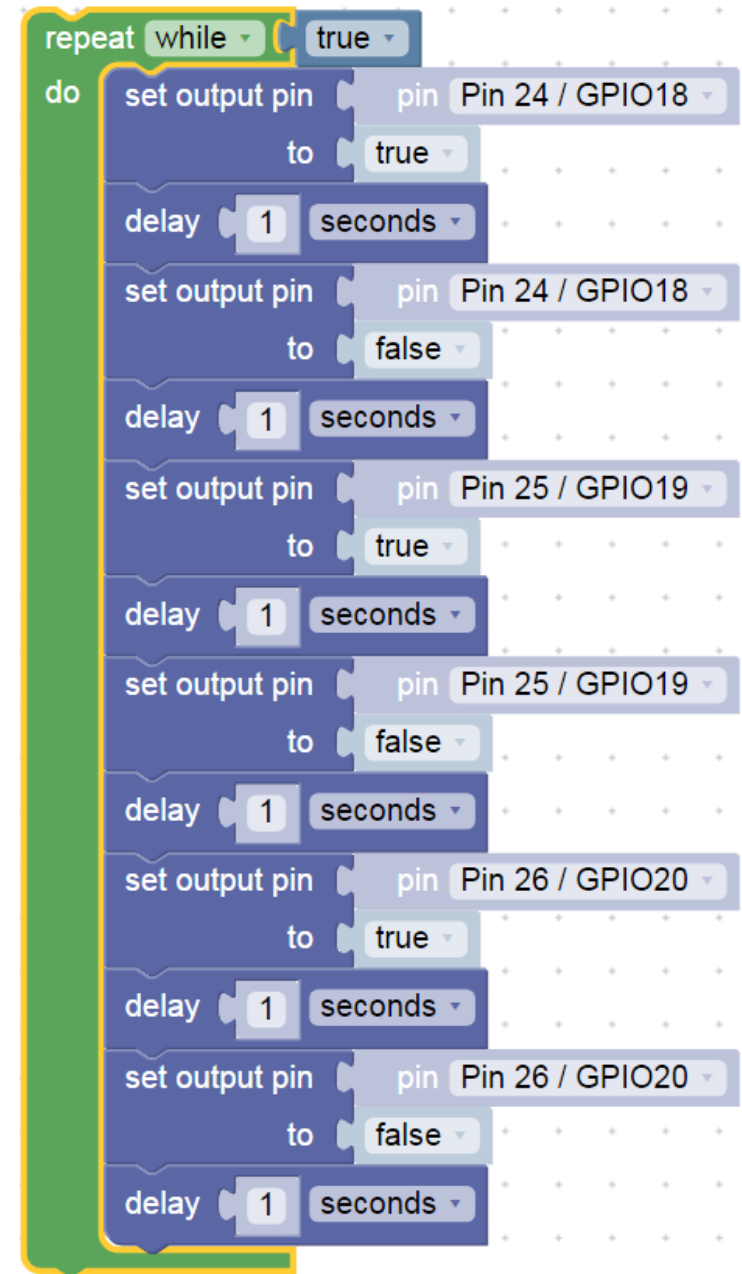
- repeat 10 times do
- repeat while do (highlighted with a red dashed box)
- count with i from 1 to 10 by 1 do
- for each item i in list do
- break out of loop

## Menu Logic



Scratch menu for logic containing the following blocks:


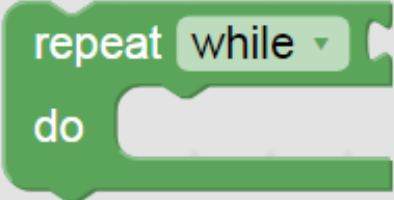
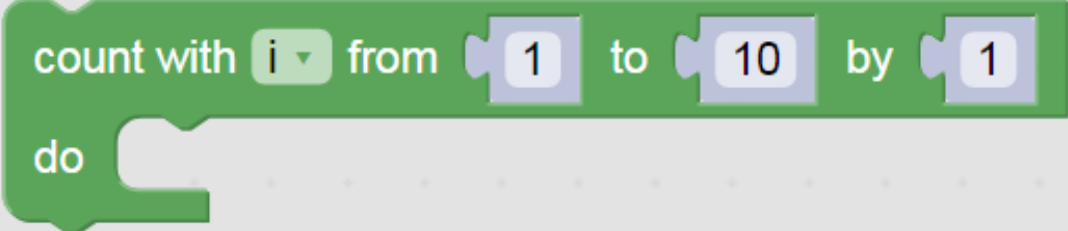
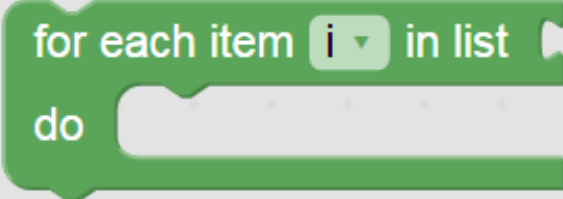
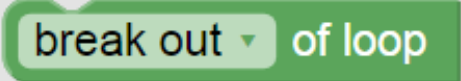
- if do
- =
- and
- not
- true (highlighted with a red dashed box)
- null



Scratch script for a continuous loop:

```
repeat while true do
  set output pin pin Pin 24 / GPIO18 to true
  delay 1 seconds
  set output pin pin Pin 24 / GPIO18 to false
  delay 1 seconds
  set output pin pin Pin 25 / GPIO19 to true
  delay 1 seconds
  set output pin pin Pin 25 / GPIO19 to false
  delay 1 seconds
  set output pin pin Pin 26 / GPIO20 to true
  delay 1 seconds
  set output pin pin Pin 26 / GPIO20 to false
  delay 1 seconds
```

# Smyčky

Logic	
<b>Loops</b>	
Math	
Text	
Lists	
Variables	
Functions	
BIPES	
Python	
Timing	
Machine	
CPU	
In/Out Pins	
Displays	
Sensors	
Actuators	
Communication	
Files	

# Páce s PINY

The screenshot shows a block-based programming environment with a sidebar on the left and a workspace on the right. The sidebar contains a list of categories: Logic, Loops, Math, Text, Lists, Variables, Functions, BIPES, Python, Timing, Machine (with a dropdown arrow), CPU, In/Out Pins (highlighted in blue), Displays, Sensors, Actuators, Communication, Files, Network, and micropython. The workspace contains a script titled "In/Out Pins" with the following blocks:

- pin LED / GPIO25
- set output pin to true (pin LED / GPIO25)
- read digital input Pull-up true (pin LED / GPIO25)
- Read RPi Pico ADC Input (pin LED / GPIO25)
- RPi Pico PWM # 0 (Pin LED / GPIO25, Frequency 1000, Duty 50)
- PWM # 0 frequency 1000
- PWM # 0 duty 50
- PWM # 0 init (pin LED / GPIO25)
- deinit PWM # 0

# Timing

## Bloky pro časování

The image shows the MicroPython IDE's block palette with the 'Timing' category selected. The palette is organized into several sections:

- Logic**
- Loops**
- Math**
- Text**
- Lists**
- Variables**
- Functions**
- BIPES**
- Python**
- Timing** (highlighted)
- Machine**
  - CPU
  - In/Out Pins
- Displays**
- Sensors**
- Actuators**
- Communication**
- Files**
- Network**
- micropython**

The 'Timing' section contains the following blocks:

- delay**: A block with a value of 1 and a dropdown menu set to 'seconds'.
- delay milliseconds**: A block with a value of 1.
- delay microseconds**: A block with a value of 1.
- Get milliseconds counter**: A block that returns the current milliseconds counter.
- Compute time difference**: A block with 'end' and 'start' ports for timing a code block.
- Timer #**: A block with a value of 2, a dropdown menu set to 'every', and a value of 1000 ms.
- Stop Timer**: A block with a value of 0.
- deep sleep**: A block with a value of 60.



# Bloky logika, smyčky, timing stroj- pin

The image displays three panels of Scratch-style code blocks. The left panel shows logic blocks: 'if do', '=', 'and', 'not', 'true', 'null', 'test if true if false'. The middle panel shows loop blocks: 'repeat 10 times do', 'repeat while do', 'count with i from 1 to 10 by 1 do', 'for each item i in list do', and 'break out of loop'. The right panel, titled 'Timing', shows blocks for delays: 'delay 1 seconds', 'delay milliseconds 1', 'delay microseconds 1', 'Get milliseconds counter', 'Compute time difference end start', 'Timer # 2 do every 1000 ms', 'Stop Timer 0', and 'deep sleep 60'.

# Zrychlit blikání LED

Timing- delay 50 ms.

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 jasů LED.**

To jsme vytvořili programově řízenou **PWM**

**Pulse Width Modulation**

```
repeat while true
do
  set output pin Pin 24 / GPIO18 to true
  delay milliseconds 50
  set output pin Pin 24 / GPIO18 to false
  delay milliseconds 50
```

```
repeat while true
do
  set output pin Pin 24 / GPIO18 to true
  delay milliseconds 2
  set output pin Pin 24 / GPIO18 to false
  delay milliseconds 18
```

# PWM Pulse- Width Modulation, modulace šířky impulsu

Pokud chceme **řídit 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.

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

Pozor, zde číslo

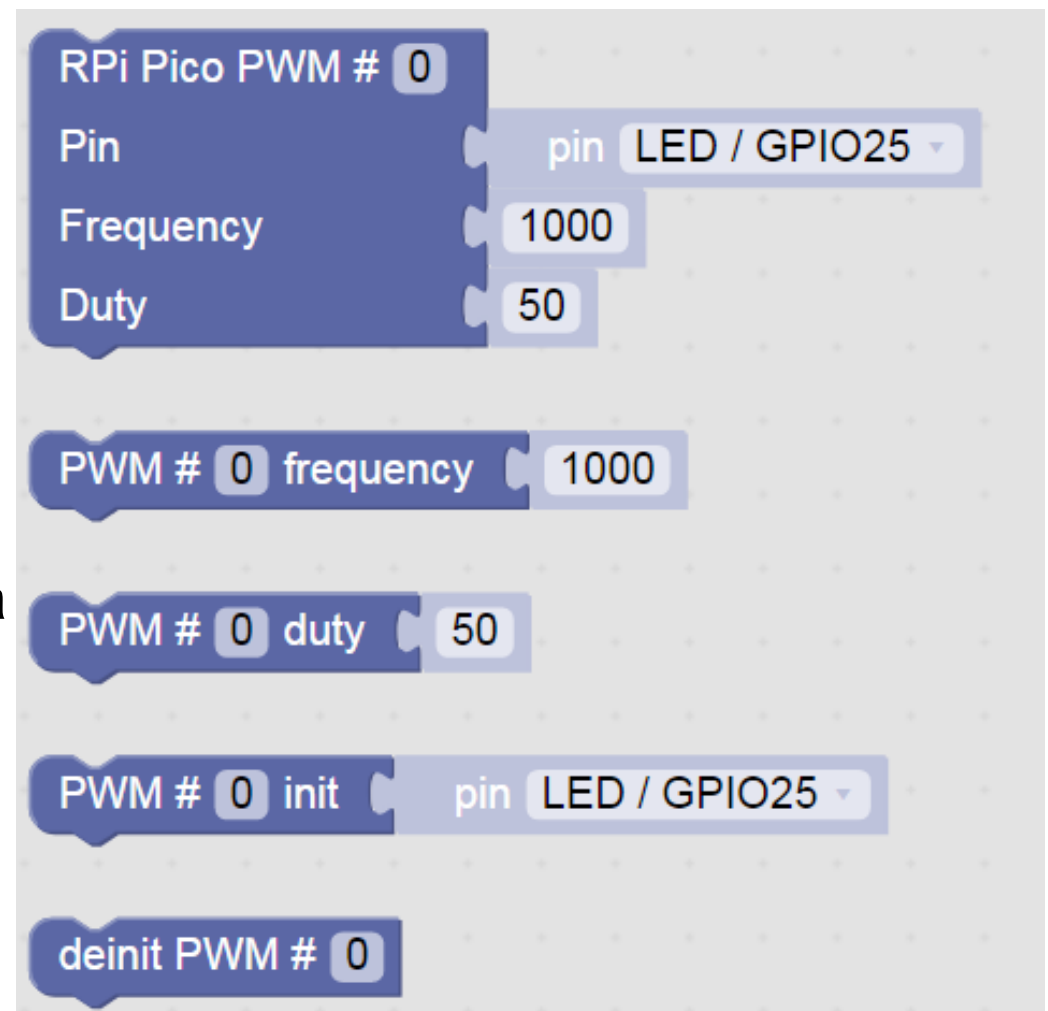
**32768 znamená 50 procent !!**

To souvisí s formou vyjádření 16- bitového binárního čísla

Pokud dáme 50, tak to skoro nebude svítit.

Nula- 0 nesvítí vůbec,

**65535** svítí plně



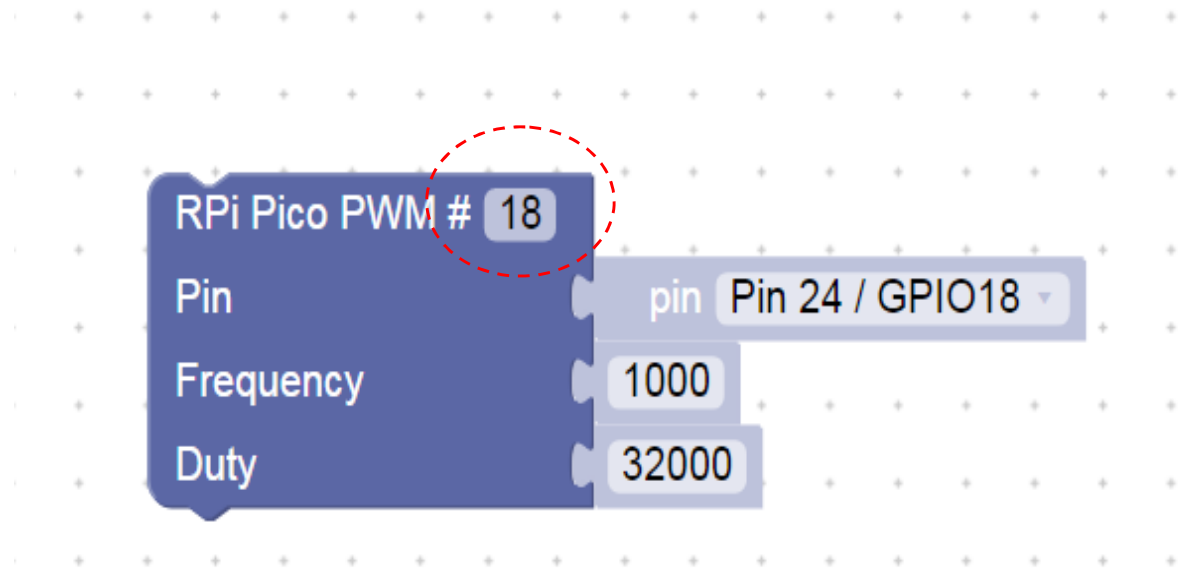
# Jednorázová inicializace PWM

---

Červená LED bude svítit plovicním jasem

Můžeme zkusit i frekvenci 10 Hz – budeme vidět blikání

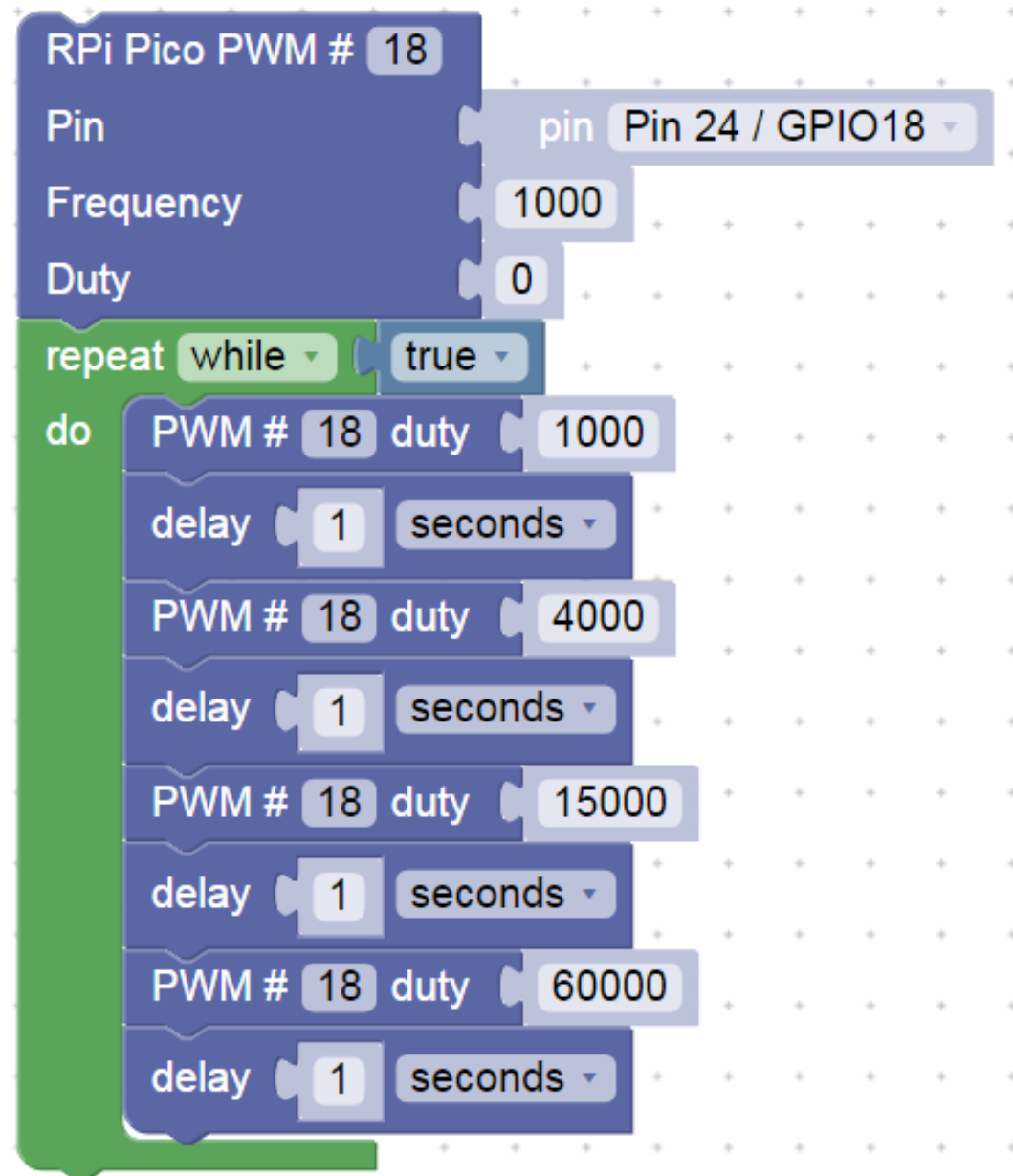
Frekvenci **1 Hz nelze** použít (procesor to nepodporuje)



PWM se **používá** např. pro řízení jasu LED , posvícení mobilu, řízení světel, řízení elektromotorů,.....

# Nekonečná postupná změna jasu LED ve stupních

Postupné rozsvícení  
a naráz zhasnutí LED



```
RPi Pico PWM # 18
Pin pin Pin 24 / GPIO18
Frequency 1000
Duty 0
repeat while true
do
  PWM # 18 duty 1000
  delay 1 seconds
  PWM # 18 duty 4000
  delay 1 seconds
  PWM # 18 duty 15000
  delay 1 seconds
  PWM # 18 duty 60000
  delay 1 seconds
```

The image shows a Scratch-style code editor for an RPi Pico. The code is written in a block-based language. It starts with a 'RPi Pico PWM # 18' block, followed by 'Pin pin Pin 24 / GPIO18', 'Frequency 1000', and 'Duty 0'. A 'repeat while true' loop contains a 'do' block with the following sequence: 'PWM # 18 duty 1000', 'delay 1 seconds', 'PWM # 18 duty 4000', 'delay 1 seconds', 'PWM # 18 duty 15000', 'delay 1 seconds', 'PWM # 18 duty 60000', and 'delay 1 seconds'.

# Generování akustického signálu - buzzer

Program generuje napinu GP22 střídavě jedničku (3 V) a nulu (0 V)

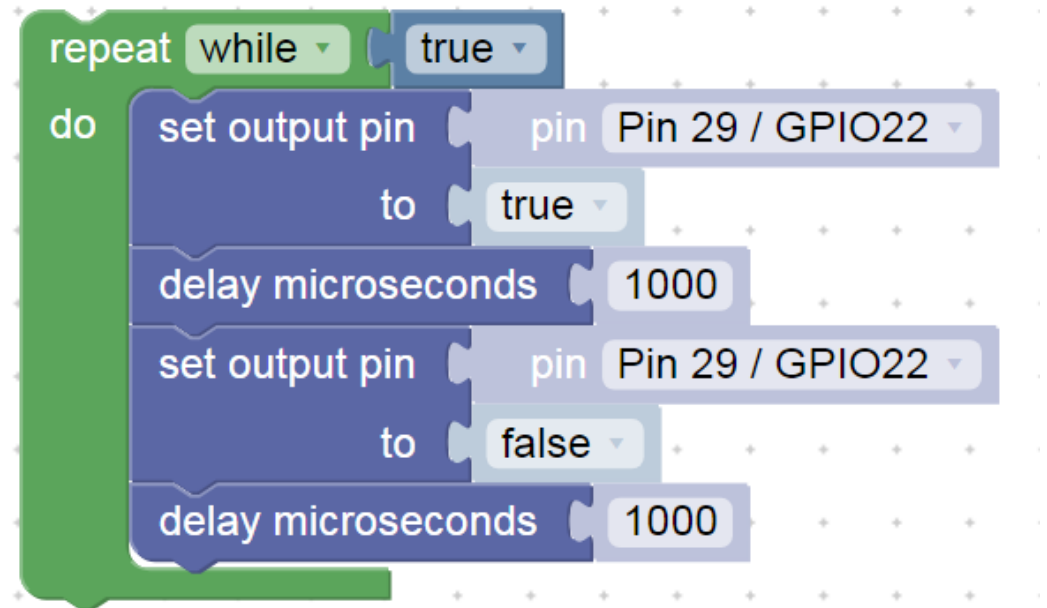
Tím se budí „reproduktorek- „bzučák“ – buzzer

Zkusit měnit časy čekání

Toto však není dobrý způsob,  
jak generovat zvuk

Na to využijme vlastní funkci  
procesoru PWM

**P**ulse **W**idth **M**odulation





# Jednorázová inicializace PWM pro Buzzer

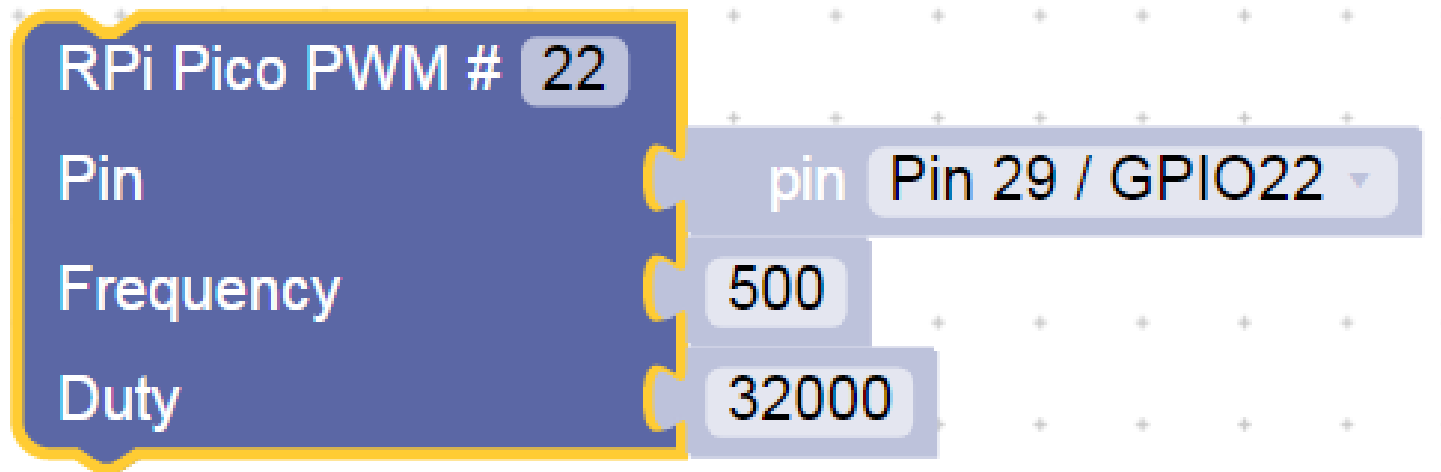
---

Jednorázová inicializace- buzzer na **GPIO22**

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

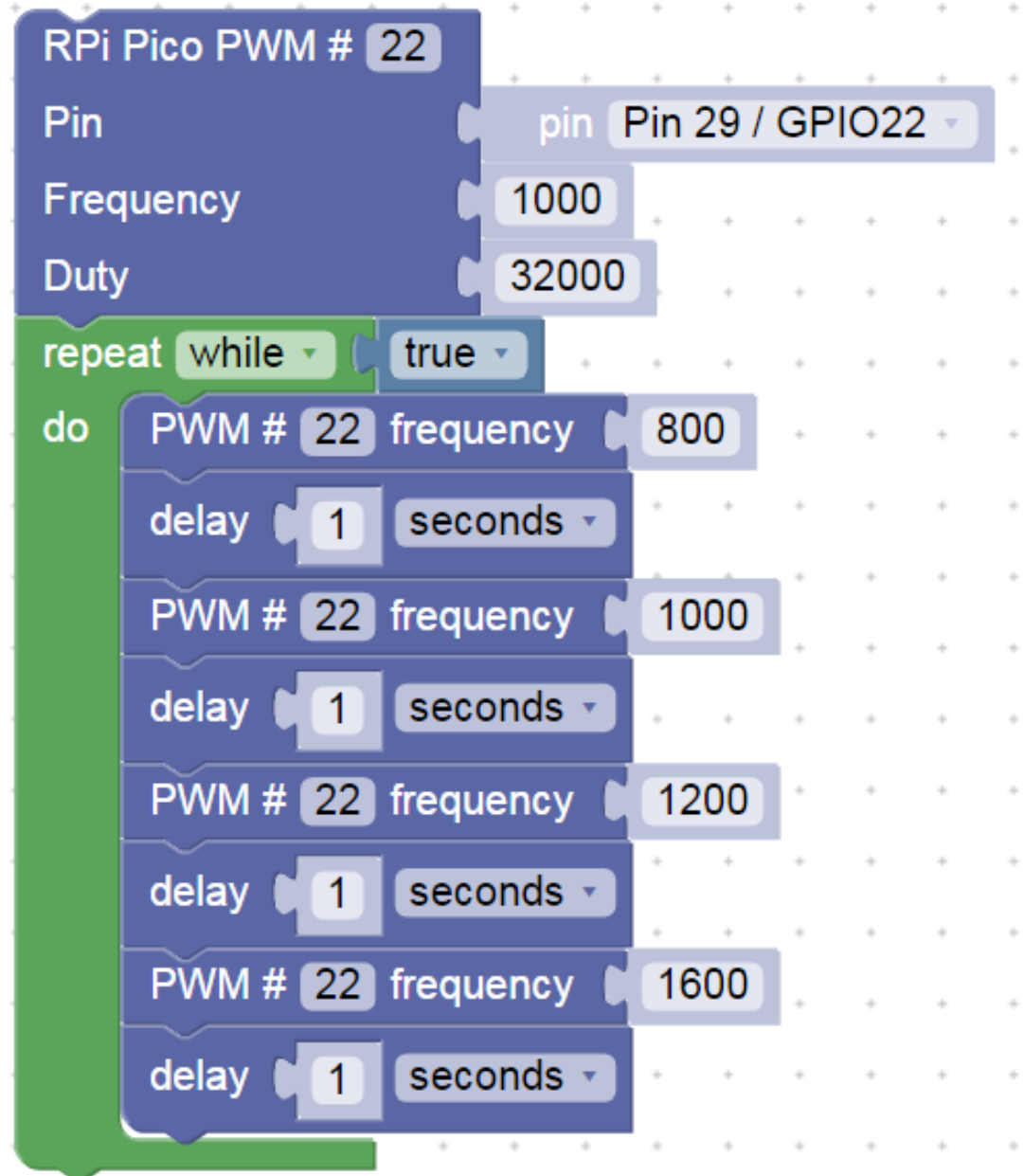
Frekvence 500 Hz, střída cca 49 % =  $32000 / 65536$ , ( $65536 = 2^{12}$ )

Pokud chceme, aby nás to (*hlasitě*) nerušilo, dát malou střihu např. duty = 10, nebo 0 a pak se nic negeneruje



# Ovládání bzučáku (buzzer) pomocí PWM

## Změna frekvence PWM

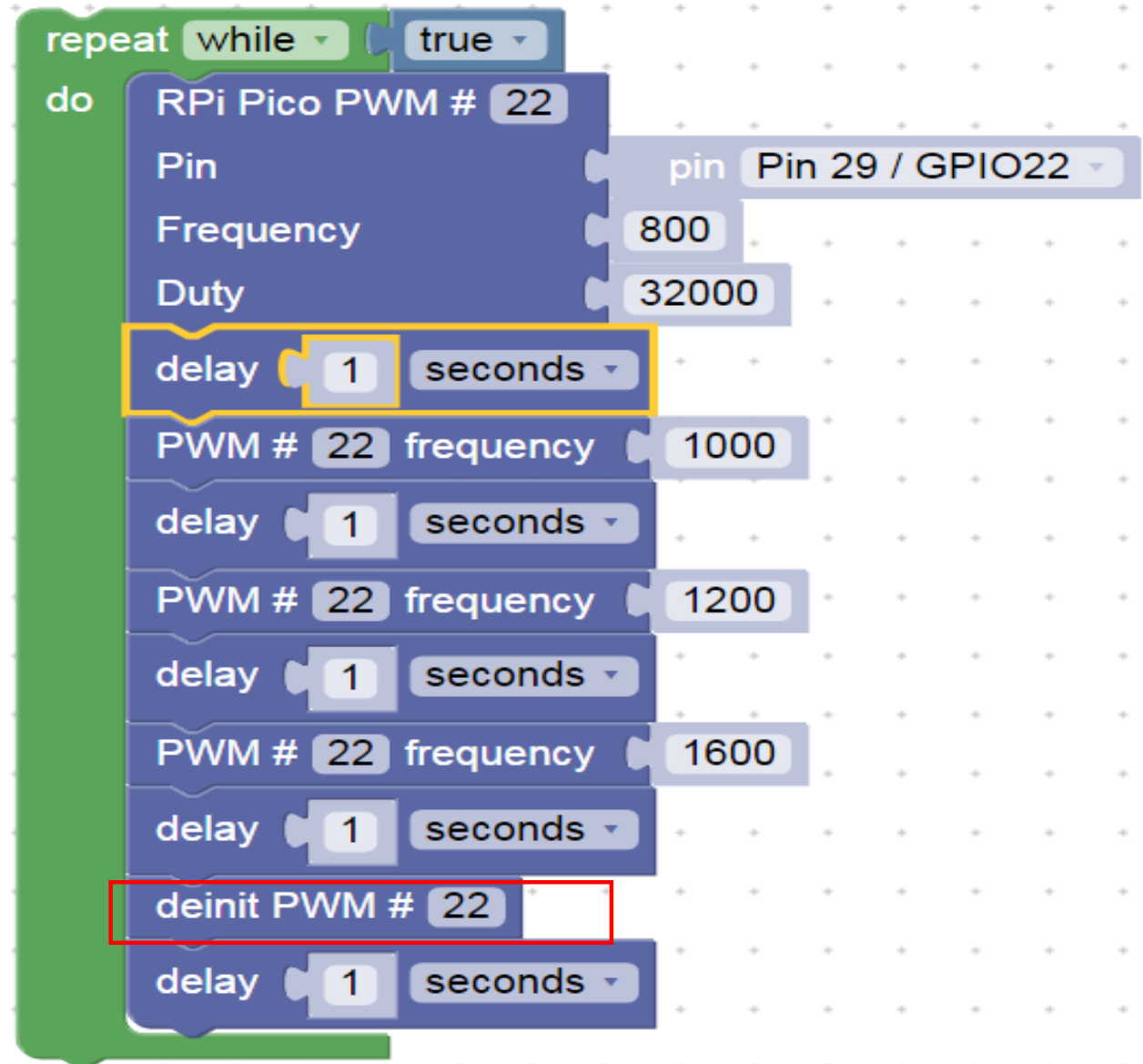


```
RPi Pico PWM # 22
Pin pin Pin 29 / GPIO22
Frequency 1000
Duty 32000
repeat while true
do
  PWM # 22 frequency 800
  delay 1 seconds
  PWM # 22 frequency 1000
  delay 1 seconds
  PWM # 22 frequency 1200
  delay 1 seconds
  PWM # 22 frequency 1600
  delay 1 seconds
```

The image shows a Scratch script for an RPi Pico. It starts with an 'RPi Pico PWM # 22' block, followed by 'Pin' (Pin 29 / GPIO22), 'Frequency' (1000), and 'Duty' (32000). A 'repeat while true' loop contains a 'do' block with four iterations: each iteration sets the PWM frequency to a specific value (800, 1000, 1200, 1600) and is followed by a 1-second delay.

# Bzučák s PWM s deinicializací PWM

Na chvíli úplně vypneme PWM



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

Nekonečná smyčka, stále opakovaně čte stav tlačítka, ukládá do proměnné **“tlac”** dle toho nastavuje červenou LED na GPIO18.

```
repeat while true do
  set tlac to read digital input pin Pin 21 / GPIO16 Pull-up true
  set output pin pin Pin 24 / GPIO18 to tlac
```

Logic  
Loops  
Math  
Text  
Lists  
**Variables**

Create variable...  
set tlac to  
change tlac by 1  
tlac

Jednodušší řešení bez proměnné  
**Kopírujeme** stav tlačítka **přímo na** výstupní **pin**

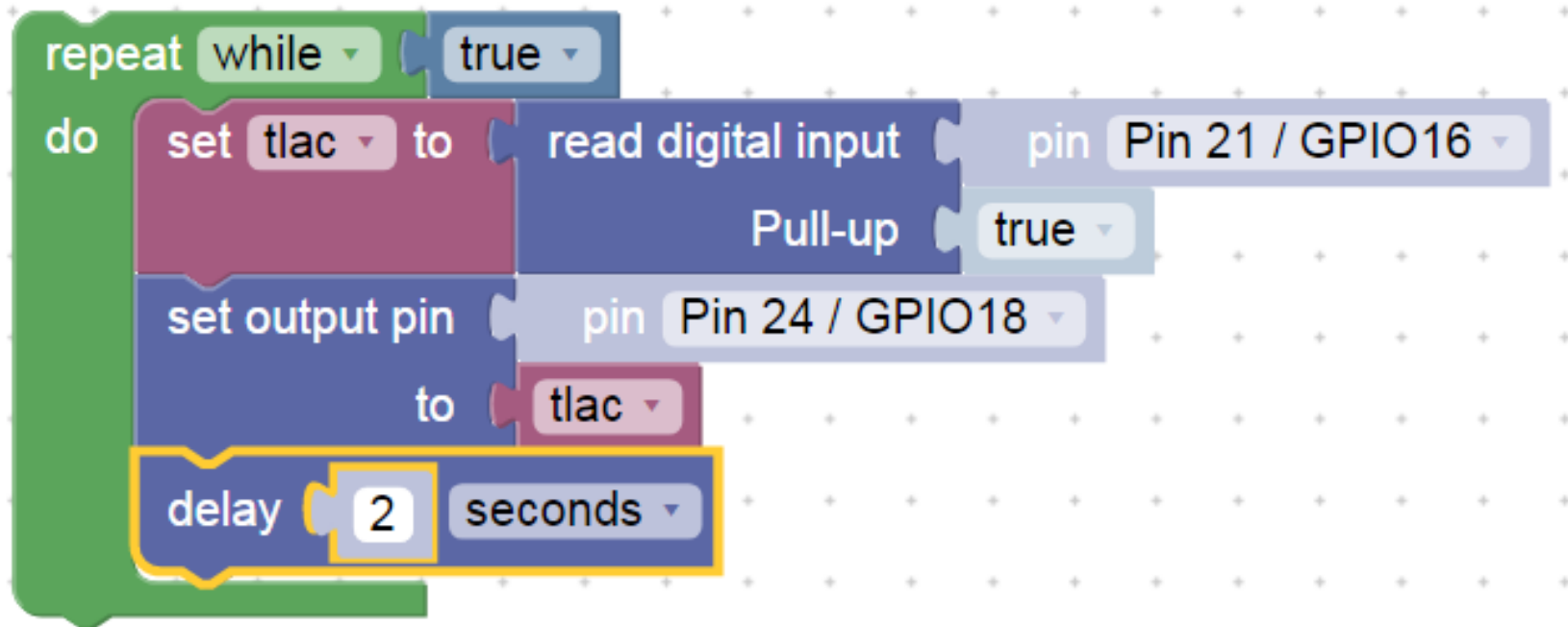
```
repeat while true do
  set output pin pin Pin 24 / GPIO18 to read digital input pin Pin 21 / GPIO16 Pull-up true
```

# Čtení stavu tlačítka a uložení do proměné

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

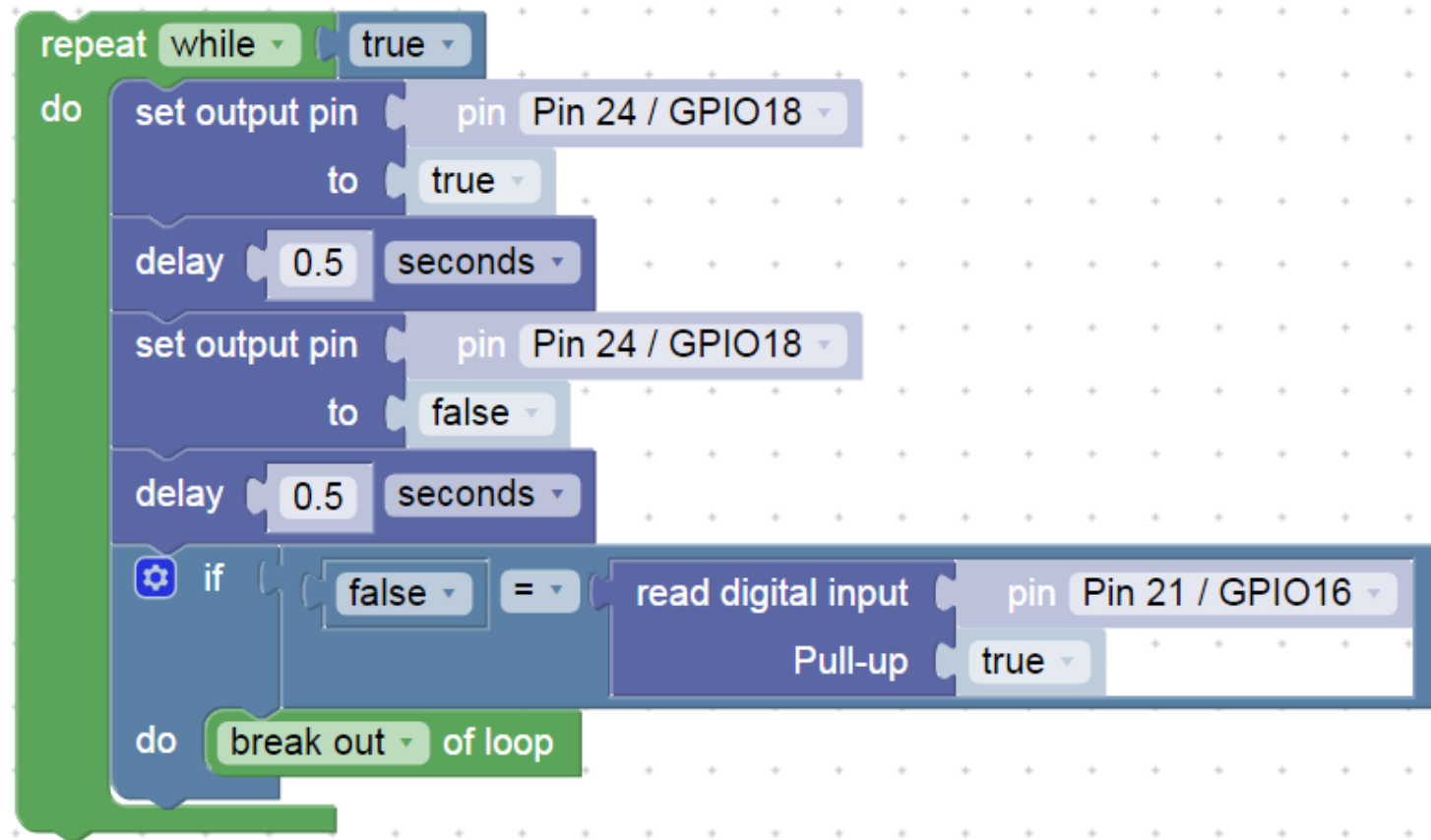
Důsledek - zpomalení reakce „**pomalé tlačítko**“

*( analogie – vybírání poštovní shránky 1x za den)*



# Čtení tlačítka, ukončení smyčky programu

Program stále v nekonečné smyčce bliká LED na GPIO18 po dobu, kdy není stisknuto tlačítko. Po stisku tlačítka (připojení GPIO16 ke GND) se **smyčka ukončí**.





▪

---

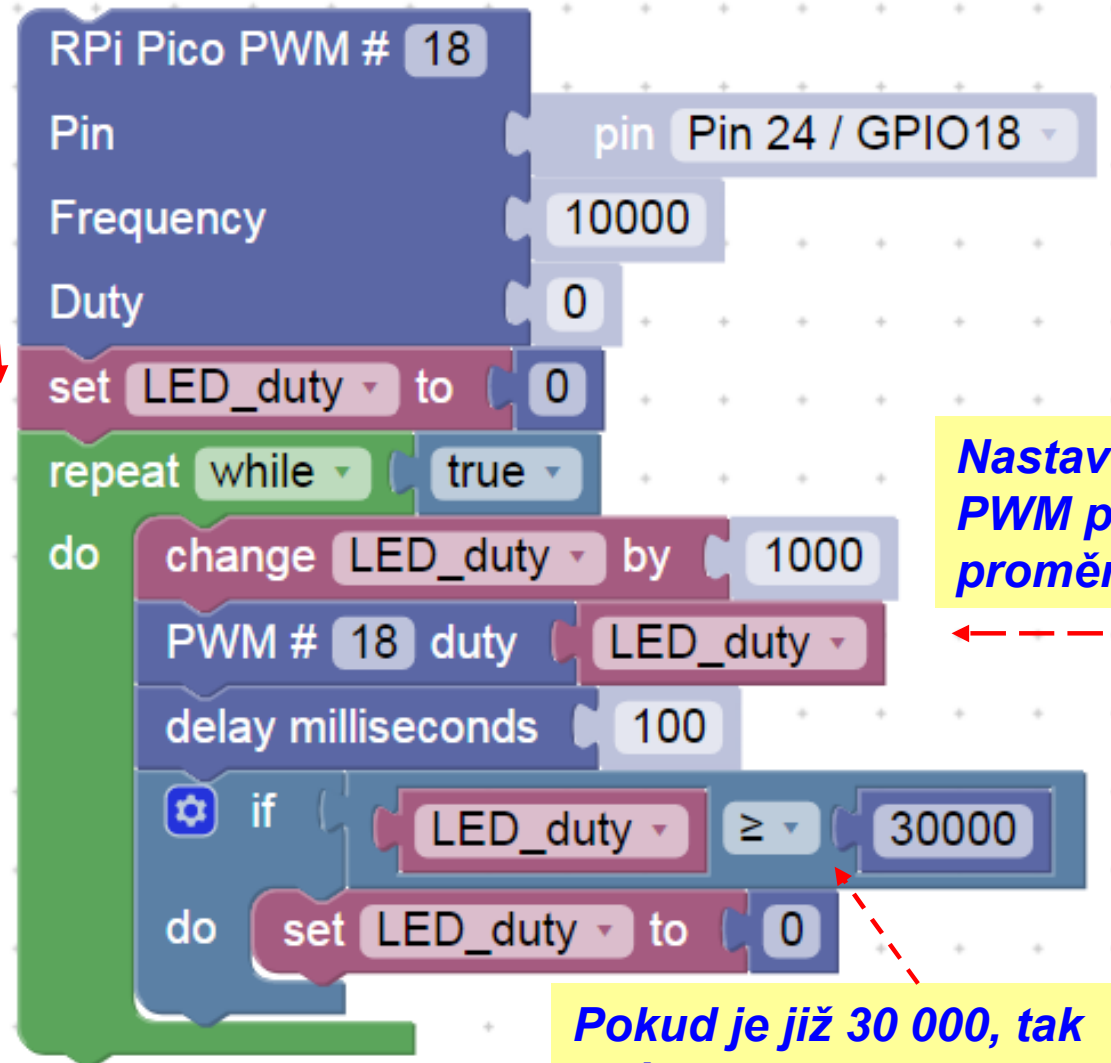
## Dále – další příklady pro inspiraci

# Postupná změna jasu LED na pinu GPIO18

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

Postupné rozsvěování

LED a naráz pohasnutí

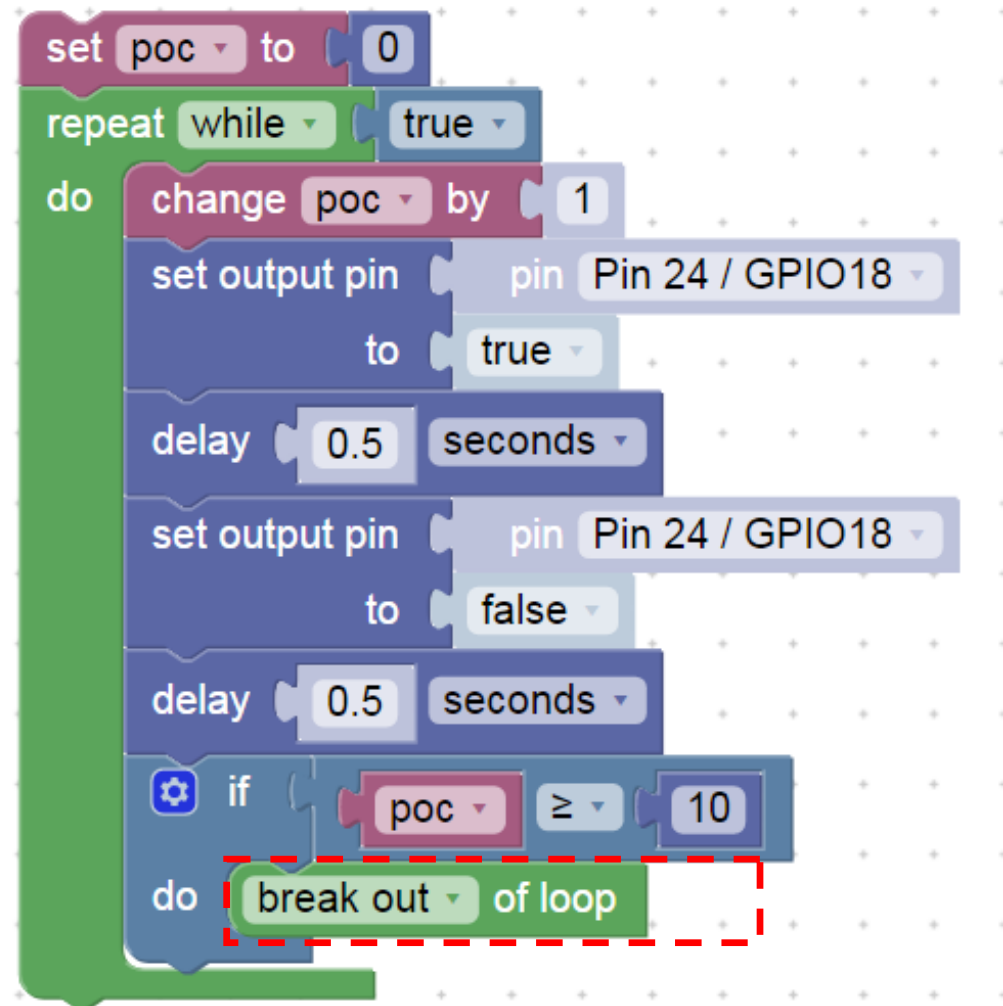


*Nastavení PWM podle proměnné*

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

# Ukončení smyčky podle podmínky

Při splnění podmínky se ukončí  
smyčka i celý program



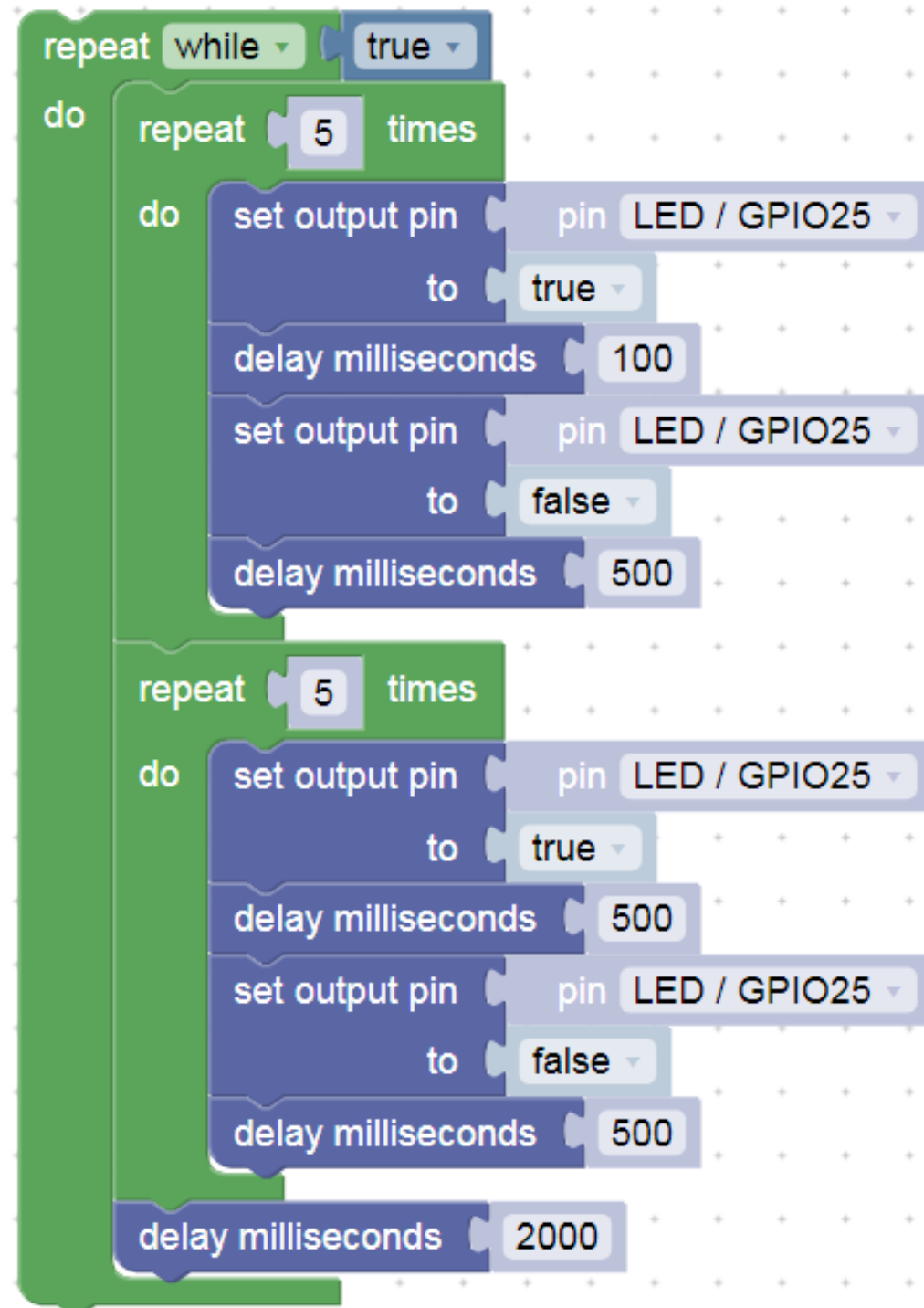
# Postupná změna frekvence PWM pro bzučák (buzzer)

Frekvence 100 Hz až 2000 Hz

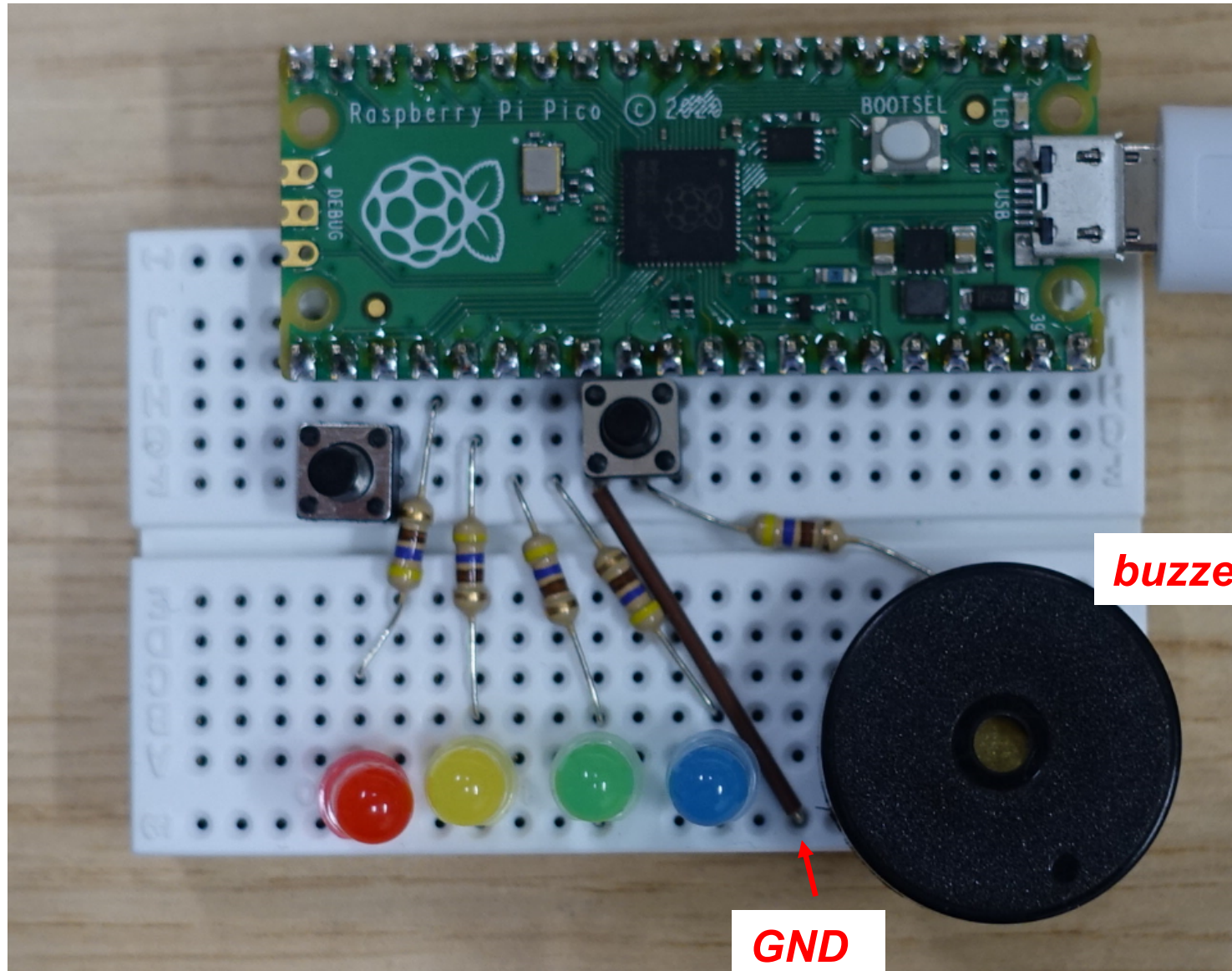
```
RPi Pico PWM # 22
Pin pin Pin 29 / GPIO22
Frequency 100
Duty 32000
set Frkvence to 100
repeat while true
do
  change Frkvence by 50
  PWM # 22 frequency Frkvence
  delay milliseconds 200
  if Frkvence >= 2000
  do
    set Frkvence to 100
```

Zvětšovat frek.  
po 50 Hz

**Blikání s využitím smyčky**  
**Blikne 5x krátce 100 ms**  
**S pauzou 500 ms, pak**  
**blikne 5 x 500 ms s pauzou**  
**500 ms**  
**A tak dále „kolem dokola“**



# Osazení experimentálního pole





▪

---

**.Konec**