
Akce Bosch- workshop pro školy

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

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

Náplň

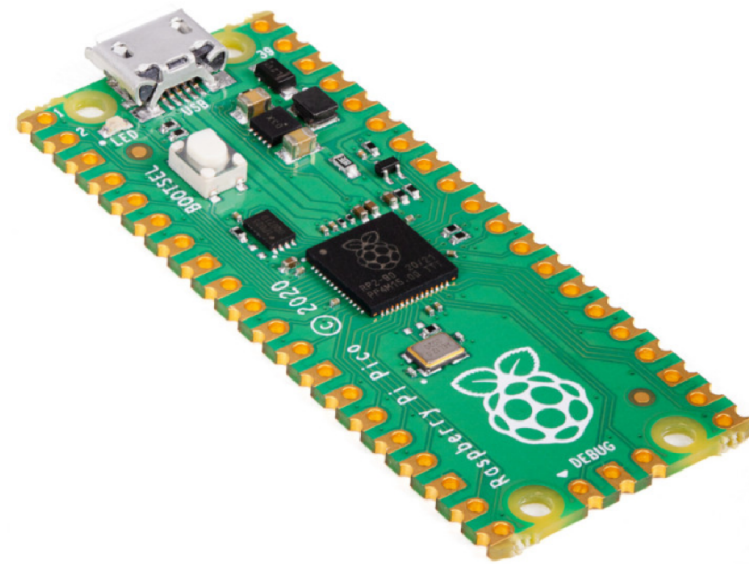
**Mikropočítač Raspberry PI PICO –
co to je a jak s ním jednoduše začít pracovat**

**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 zkusit**



Mikropočítač - Raspberry PI PICO

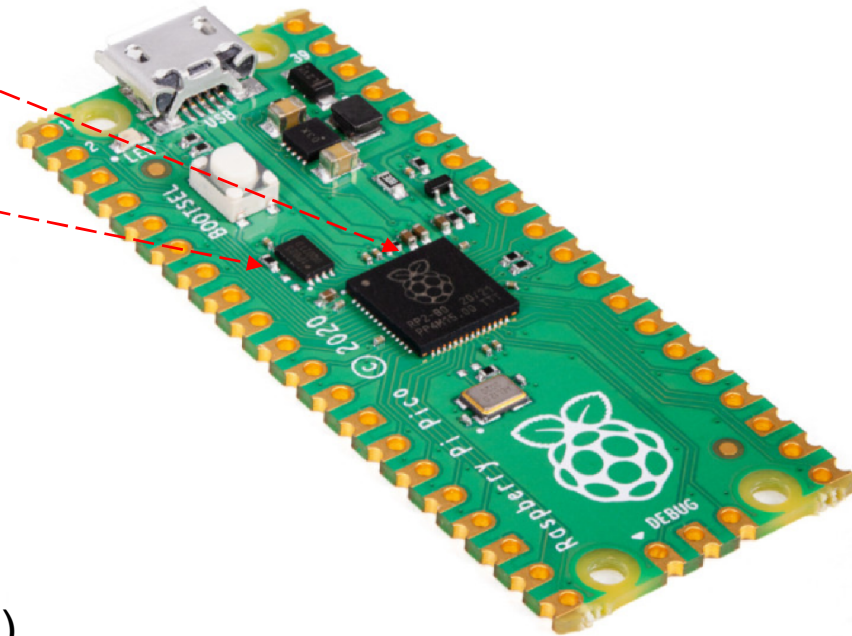
Mikropočítač Raspberry PI PICO

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, I²C BUS, SPI

Převodník ADC 12 bitový. Až 500 000 vzorků za sekundu

https://embedded.fel.cvut.cz/procesory/RP_PICO

Informace

https://embedded.fel.cvut.cz/procesory/RP_PICO

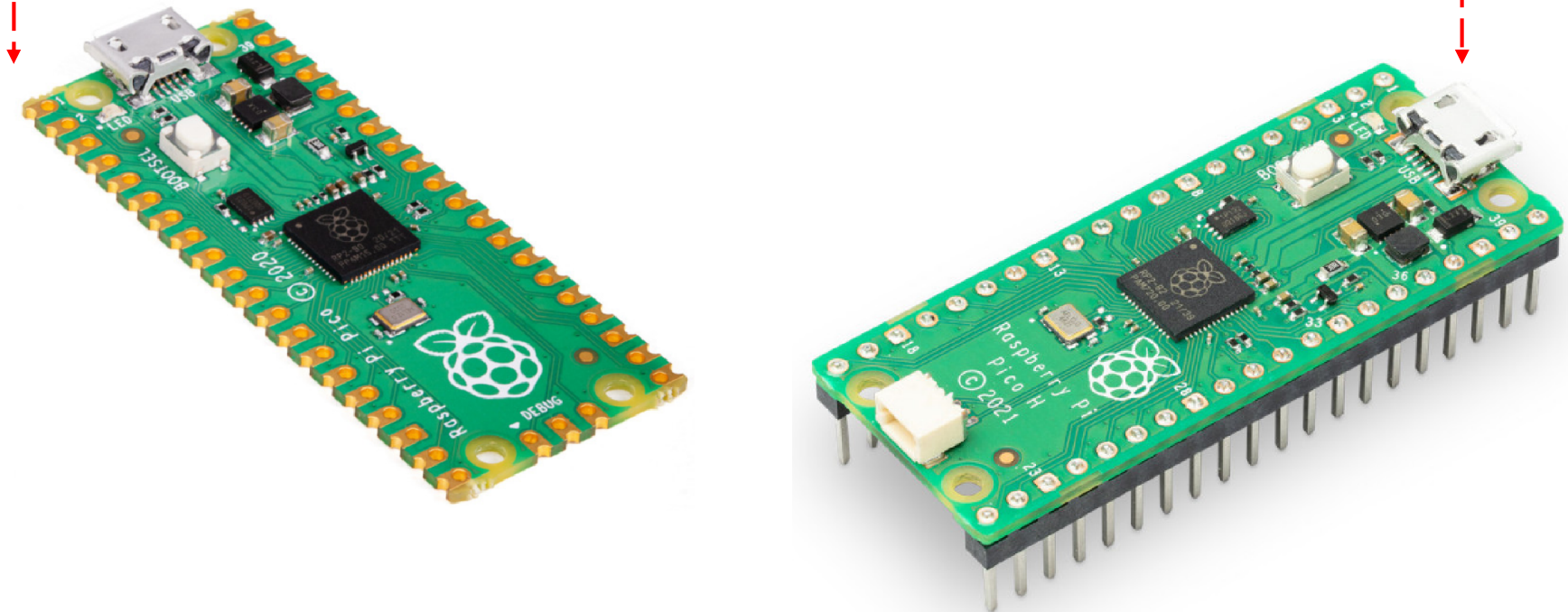
<https://www.raspberrypi.com/products/raspberry-pi-pico/>

—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>



Mikroprocesor, Mikropočítač

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

Náš procesor Raspberry PI PICO má napájecí napětí **+3,3 V**.

To je to označeno na desce jako **3V3**

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

Jak náš procesor „řekne“ svému okolí, že signalizuje **jedničku (true)** 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))

Jak signalizuje **nulu (false)** na daném kontaktu- připojí jej (vnitřně) ke svorce **GND** – zem.

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.

Takto zapojená (s rezist. mezi **PIN a GND**) LED bude svíí při 1- true.

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

Pokud bychom LED s rezistorem zapojili **mezi PIN** a napájení **3V3**, **fungovala** by v inverzní logice svítila by při **nule – false**.

V naší desce máme čtyři různé LED (červená, žlutá, zelená, modrá) zapojené mezi pin a GND, tedy budou svítit při **true**.

Připojená LED **nebude** svítit.

Dále na desce máme bzučák (**buzzer**) a tlačítko, které připíná pin k zemi - GND

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/download/rp2-pico/>

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 článku RC

Sledování blikání zářivek pomocí fototranzistoru

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

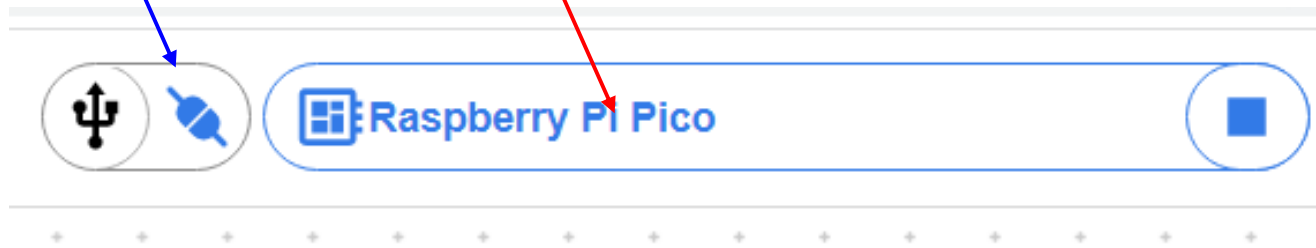
Více informací je v závěru této prezentace.

•

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

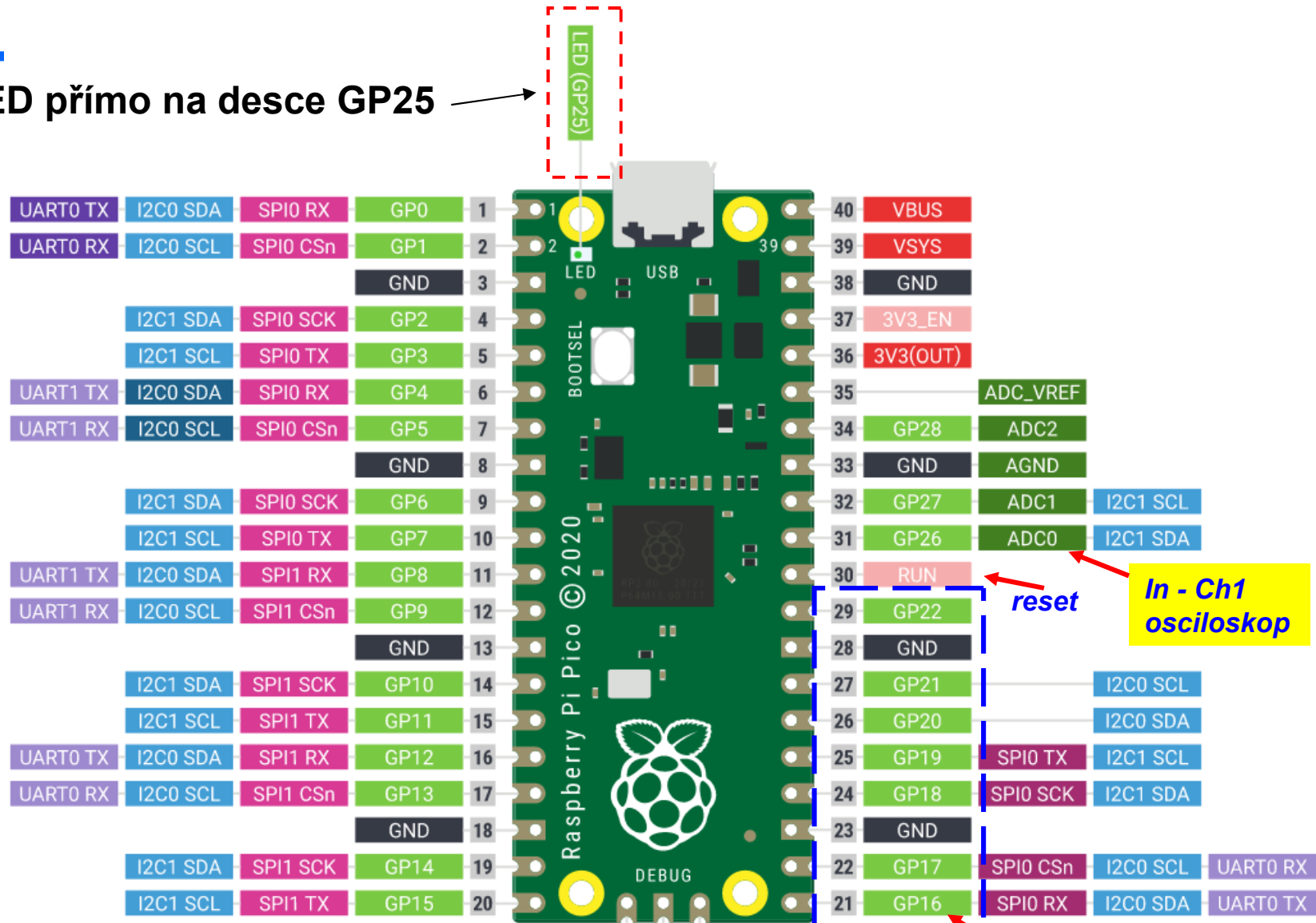
Nabídky panelů

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

The screenshot shows the BIPES interface with the 'Device' tab selected. The main content is a detailed pinout diagram for the Raspberry Pi Pico. The pins are numbered 1 through 40, with some pins having multiple functions. A legend at the bottom identifies the functions by color: Power (red), Ground (black), UART / UART (default) (purple), GPIO, PIO, and PWM (green), ADC (dark green), SPI / SPI (default) (pink), I2C / I2C (default) (blue), System Control (light red), and Debugging (orange).

Pin	Function(s)
1	UART0 TX, I2C0 SDA, SPI0 RX, GP0
2	UART0 RX, I2C0 SCL, SPI0 CSn, GP1
3	GND
4	I2C1 SDA, SPI0 SCK, GP2
5	I2C1 SCL, SPI0 TX, GP3
6	UART1 TX, I2C0 SDA, SPI0 RX, GP4
7	UART1 RX, I2C0 SCL, SPI0 CSn, GP5
8	GND
9	I2C1 SDA, SPI0 SCK, GP6
10	I2C1 SCL, SPI0 TX, GP7
11	UART1 TX, I2C0 SDA, SPI1 RX, GP8
12	UART1 RX, I2C0 SCL, SPI1 CSn, GP9
13	GND
14	I2C1 SDA, SPI1 SCK, GP10
15	I2C1 SCL, SPI1 TX, GP11
16	UART0 TX, I2C0 SDA, SPI1 RX, GP12
17	UART0 RX, I2C0 SCL, SPI1 CSn, GP13
18	GND
19	I2C1 SDA, SPI1 SCK, GP14
20	I2C1 SCL, SPI1 TX, GP15
21	GP16, SPI0 RX, I2C0 SDA, UART0 TX
22	GP17, SPI0 CSn, I2C0 SCL, UART0 RX
23	GND
24	GP18, SPI0 SCK, I2C1 SDA
25	GP19, SPI0 TX, I2C1 SCL
26	GP20, I2C0 SDA
27	GP21, I2C0 SCL
28	GND
29	GP22
30	RUN
31	GP26, ADC0, I2C1 SDA
32	GP27, ADC1, I2C1 SCL
33	GND, AGND
34	GP28, ADC2
35	ADC_VREF
36	3V3(OUT)
37	3V3_EN
38	GND
39	VSYS
40	VBUS

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*

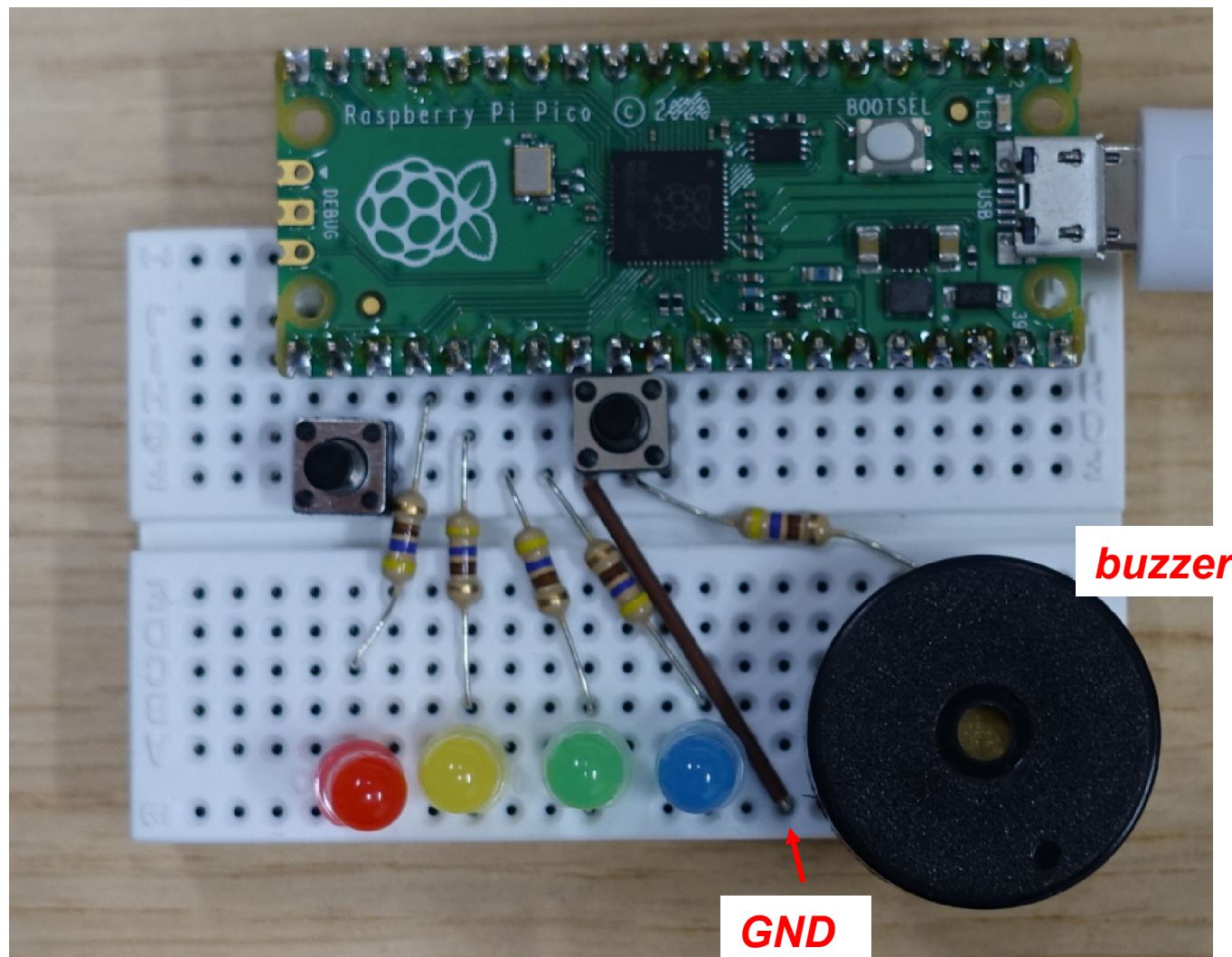
470 R značí zapojení rezistoru o odporu 470 Ohmů do série s pinem

Úroveň „true“ na pinu LED rozsvítí,

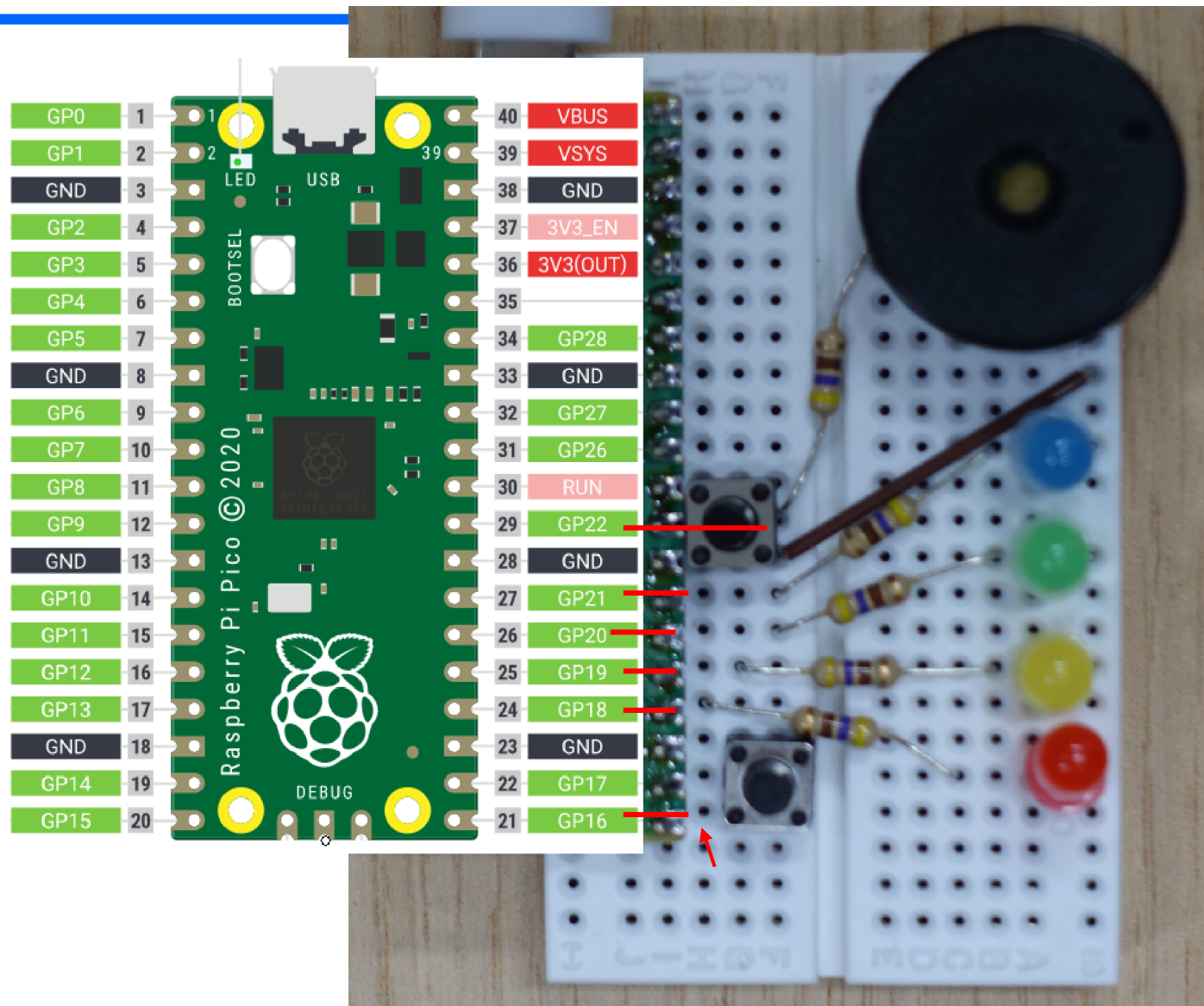
úroveň „false“ LED na pinu zhasne

Osazení experimentálního pole

-



Výstupní piny

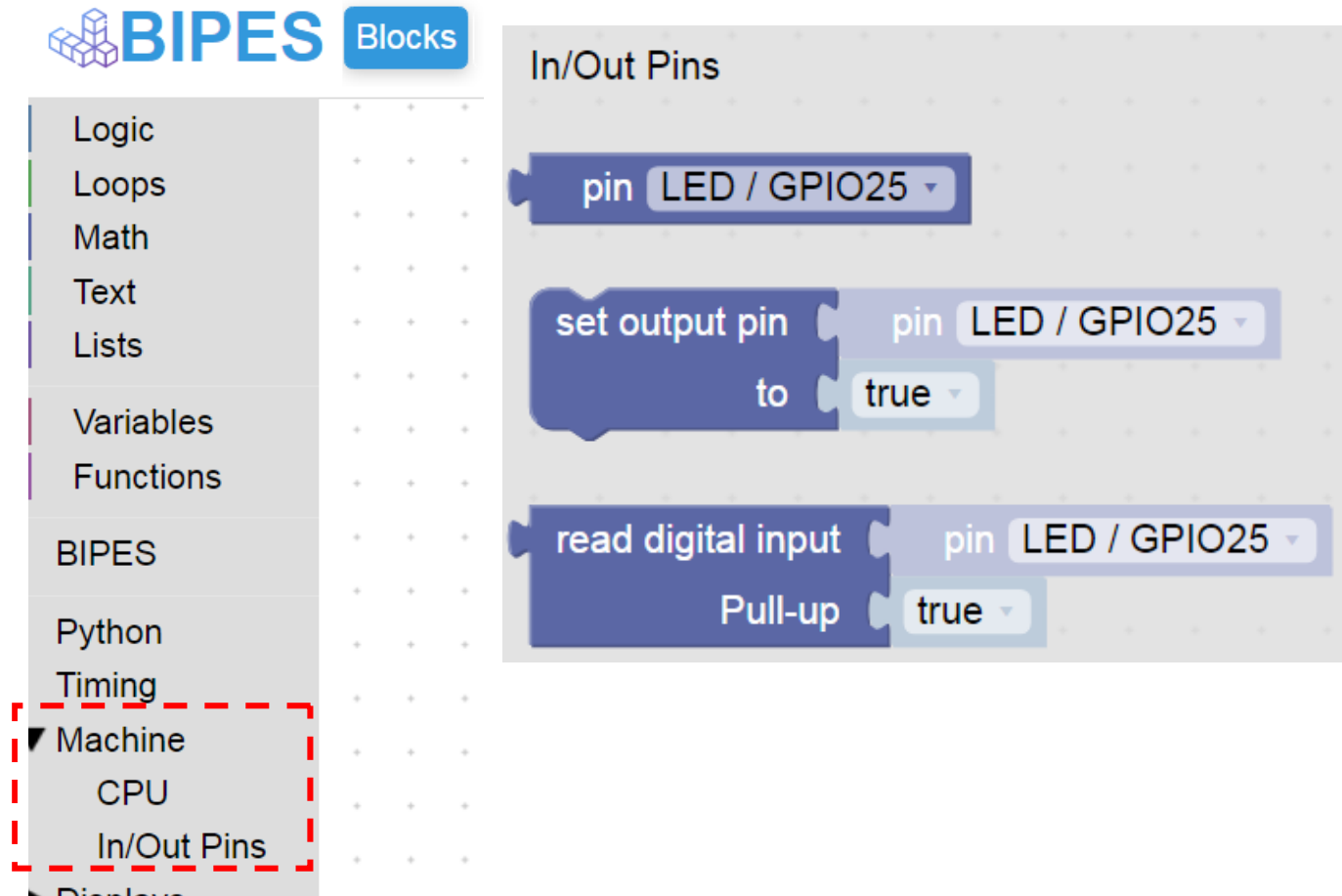


.

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

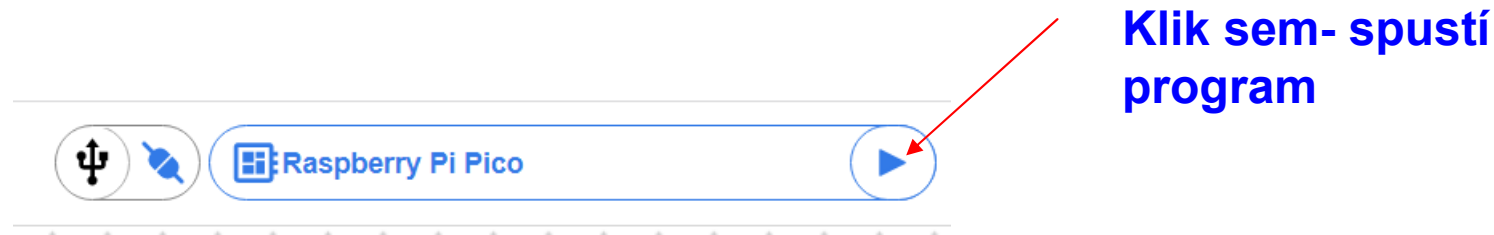
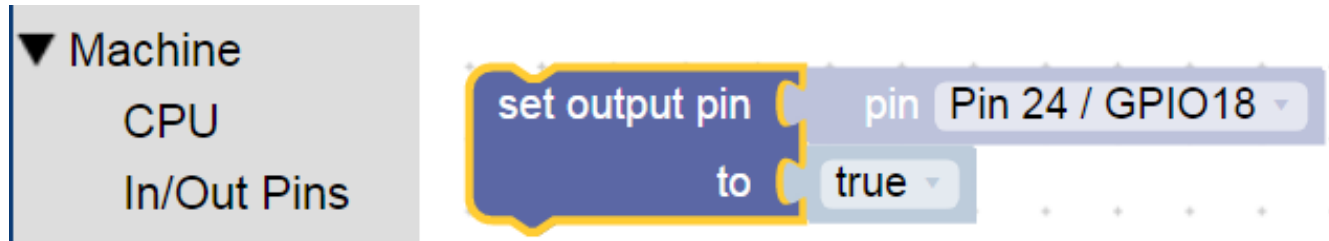
Rozsvítit LED- nástroje



The image shows the BIPES Blocks interface. On the left is a sidebar with a category list: Logic, Loops, Math, Text, Lists, Variables, Functions, BIPES, Python, Timing, Machine (expanded to show CPU and In/Out Pins), and Displays. The 'In/Out Pins' category is highlighted with a red dashed box. On the right, a workspace titled 'In/Out Pins' contains three blocks: a 'pin' block with 'LED / GPIO25' selected, a 'set output pin to' block with 'LED / GPIO25' and 'true' selected, and a 'read digital input Pull-up' block with 'LED / GPIO25' and 'true' selected.

Jednorázový běh programu (bez opakování)

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



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

The image shows a Scratch script for controlling three LEDs. The script consists of a sequence of blocks: 'set output pin to true' for Pin 24 / GPIO18, a 1-second delay, 'set output pin to false' for Pin 24 / GPIO18, another 1-second delay, 'set output pin to true' for Pin 25 / GPIO19, a 1-second delay, 'set output pin to false' for Pin 25 / GPIO19, a 1-second delay, 'set output pin to true' for Pin 26 / GPIO20, a 1-second delay, 'set output pin to false' for Pin 26 / GPIO20, and a final 1-second delay.

Stálý běh programu, použití nekonečné smyčky



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

Menu Loops

The screenshot shows the 'Menu Loops' in the MicroPython IDE. It contains several loop-related blocks: a 'repeat 10 times' block, a 'repeat while' block (highlighted with a red dashed box), a 'count with i from 1 to 10 by 1' block, a 'for each item i in list' block, and a 'break out of loop' block.

Menu Logic

The screenshot shows the 'Menu Logic' in the MicroPython IDE. It contains several logic blocks: an 'if do' block, an '=' (equality) block, an 'and' block, a 'not' block, a 'true' block (highlighted with a red dashed box), and a 'null' block.

This is a detailed view of a 'repeat while' loop block. The 'repeat' dropdown is set to 'while' and the condition is 'true'. The 'do' block contains four sub-blocks: 'set output pin pin Pin 24 / GPIO18 to true', 'delay 1 seconds', 'set output pin pin Pin 24 / GPIO18 to false', and 'delay 1 seconds'. A red dashed arrow points from the 'true' block in the 'Menu Logic' to the 'true' condition in this loop block.

Stálý běh programu, použití nekonečné smyčky

Menu Loops

repeat 10 times
do

repeat while
do

count with i from 1 to 10 by 1
do

for each item i in list
do

break out of loop

Menu Logic

if do

=

and

not


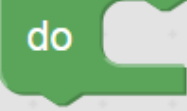
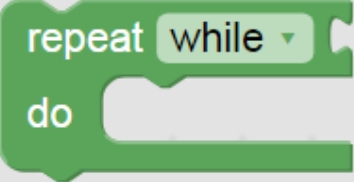


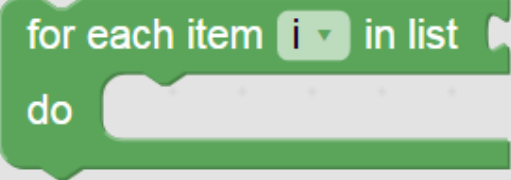
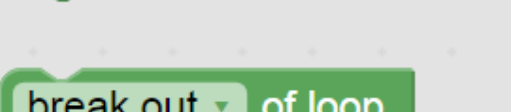
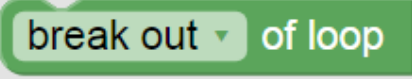
true

null

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

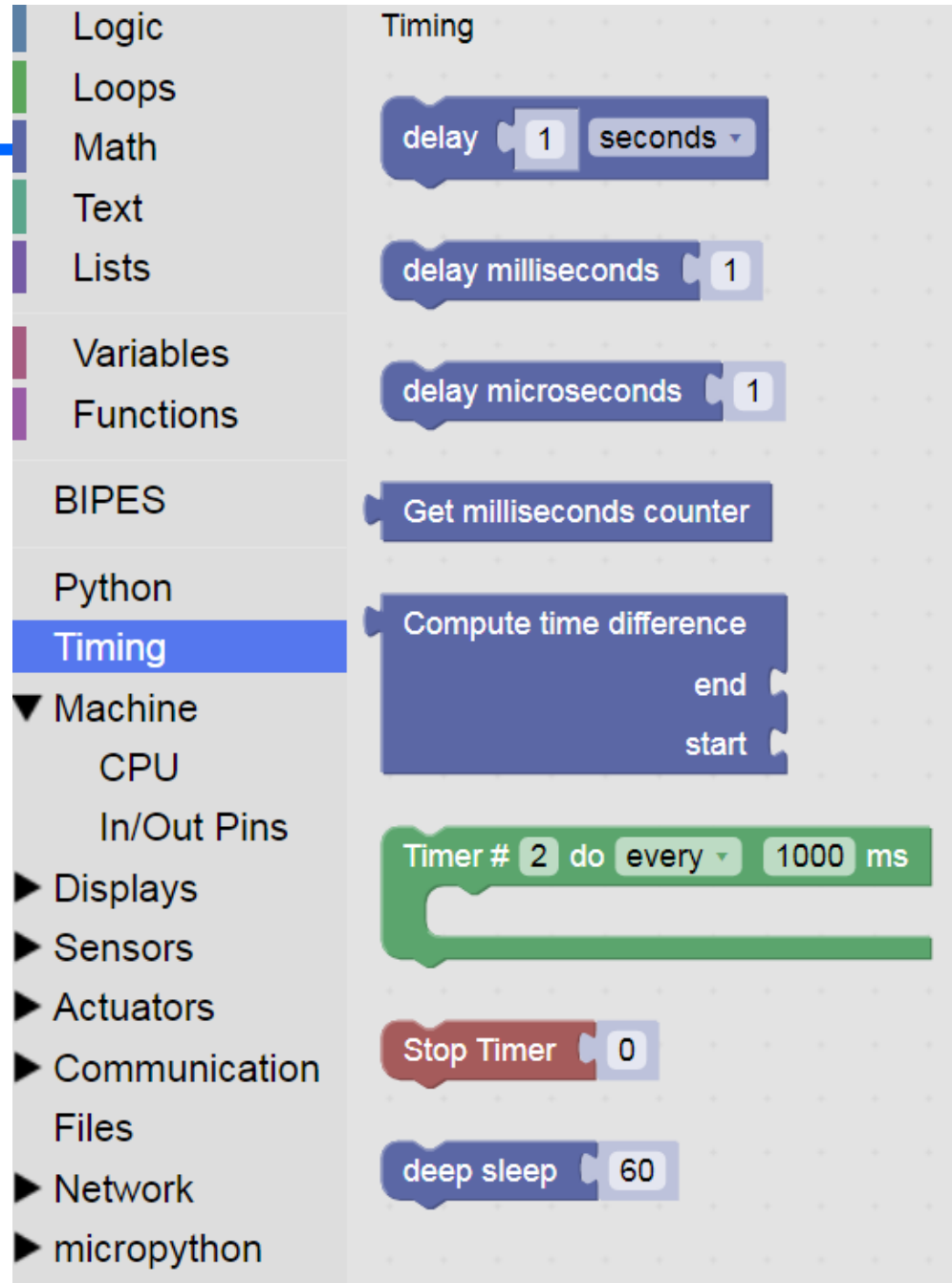
Páce s PINY

The screenshot displays a block-based programming interface with a sidebar on the left and a workspace on the right. The sidebar lists various categories: Logic, Loops, Math, Text, Lists, Variables, Functions, BIPES, Python, Timing, Machine (with sub-items CPU and In/Out Pins), Displays, Sensors, Actuators, Communication, Files, Network, and micropython. The 'In/Out Pins' category is selected and highlighted in blue. The workspace contains the following code blocks:

- pin** LED / GPIO25
- set output pin** to pin LED / GPIO25, to true
- read digital input** pin LED / GPIO25, Pull-up true
- Read RPI Pico ADC Input** pin LED / GPIO25
- RPI Pico PWM # 0** Pin 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



Logic

Loops

Math

Text

Lists

Variables

Functions

BIPES

Python

Timing

Machine

 CPU

 In/Out Pins

▶ Displays

▶ Sensors

▶ Actuators

▶ Communication

Files

▶ Network

▶ micropython

Timing

delay 1 seconds

delay milliseconds 1

delay microseconds 1

Get milliseconds counter

Compute time difference

end

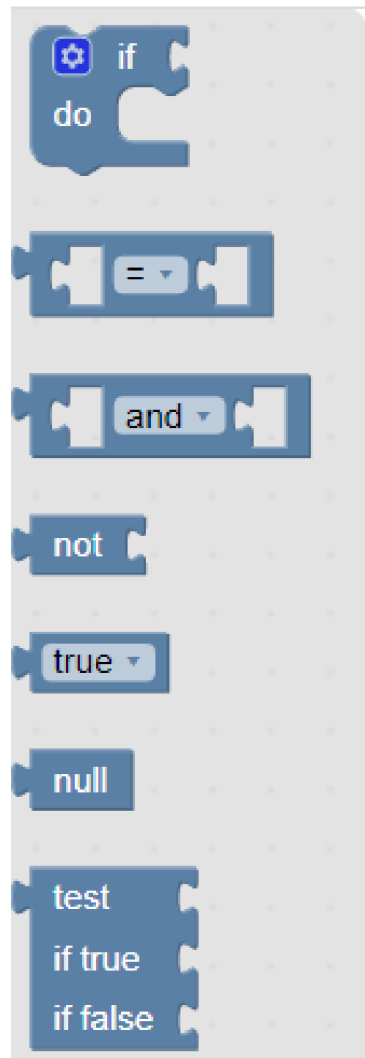
start

Timer # 2 do every 1000 ms

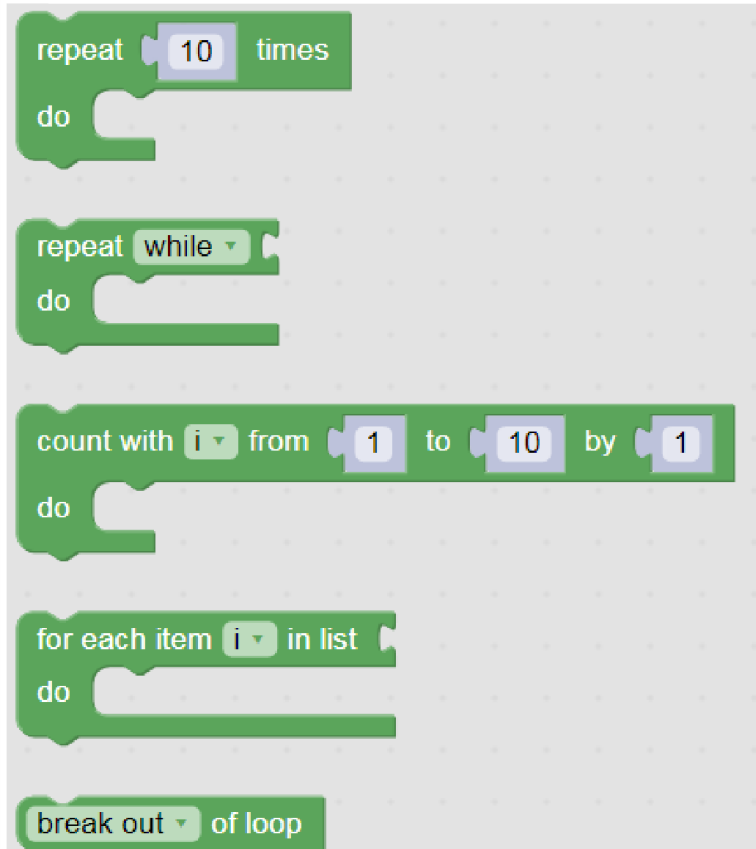
Stop Timer 0

deep sleep 60

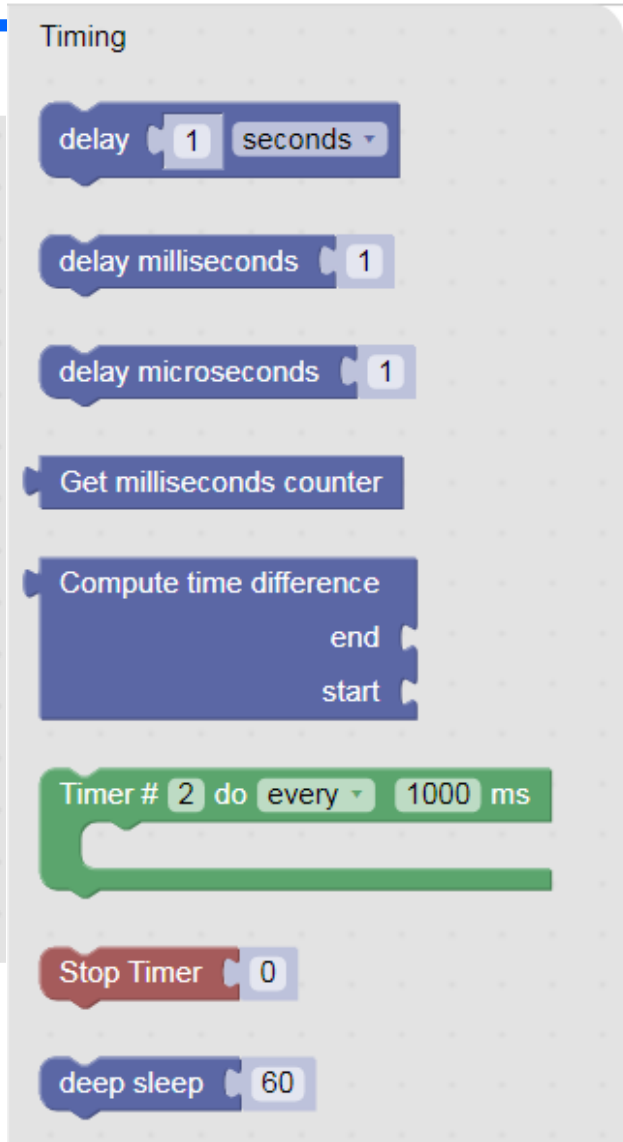
Bloky logika, smyčky, timing stroj-pin



A vertical palette of logic blocks for a programming environment. The blocks include: an 'if' block with a gear icon and a 'do' block; an equals sign block; an 'and' block; a 'not' block; a 'true' block with a dropdown arrow; a 'null' block; and a 'test' block with 'if true' and 'if false' options.



A vertical palette of loop and iteration blocks. From top to bottom: a 'repeat' block with '10' and 'times' fields; a 'do' block; a 'repeat while' block; a 'do' block; a 'count with' block with 'i' as the variable, '1' as the start, '10' as the end, and '1' as the step; a 'do' block; a 'for each item' block with 'i' as the variable and 'in list' as the target; a 'do' block; and a 'break out' block with 'of loop' as the target.



A vertical palette of timing and control blocks. From top to bottom: a 'delay' block with '1' and 'seconds' fields; a 'delay milliseconds' block with '1' field; a 'delay microseconds' block with '1' field; a 'Get milliseconds counter' block; a 'Compute time difference' block with 'end' and 'start' fields; a 'Timer # 2 do every' block with '1000 ms' field; a 'Stop Timer' block with '0' field; and a 'deep sleep' block with '60' field.

Zrychlit blikání LED

Timing- delay 50 ms.

Pak dát **10 ms/10 ms**, zrychlení

Nebude vidět blikání

Změnit **poměr**

Dále 2 ms /18 ms

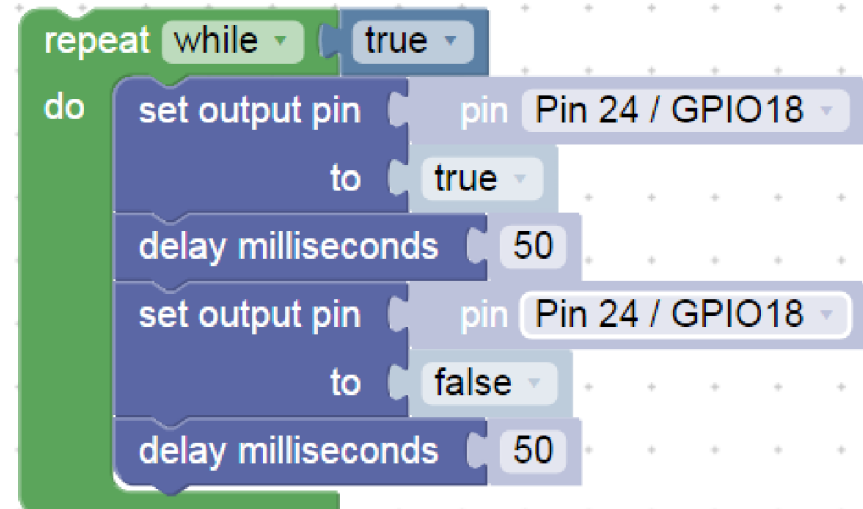
Zkusit jinak **měnit poměr časů**

Co se změnilo? Nestačíme sledovat blikání, ale jen vidíme

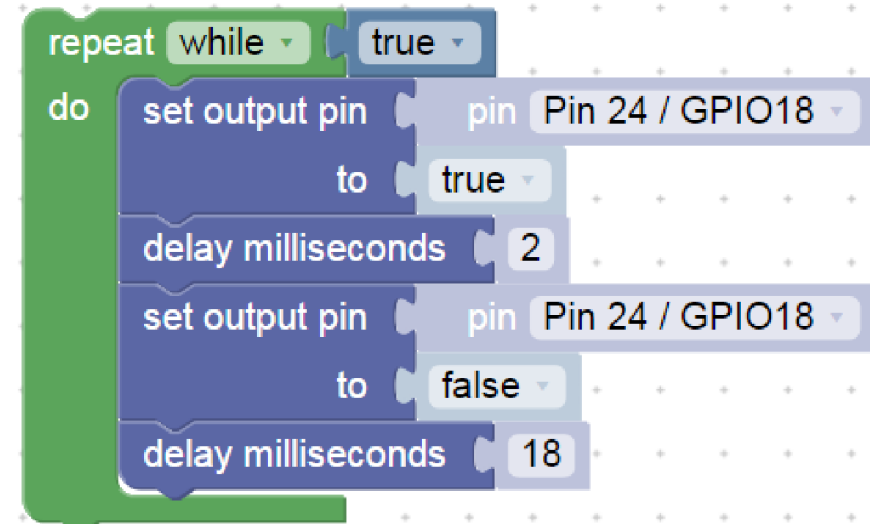
pokles jasů LED.

To jsme vytvořili programově řízenou **PWM**

Pulse Width Modulation



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



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

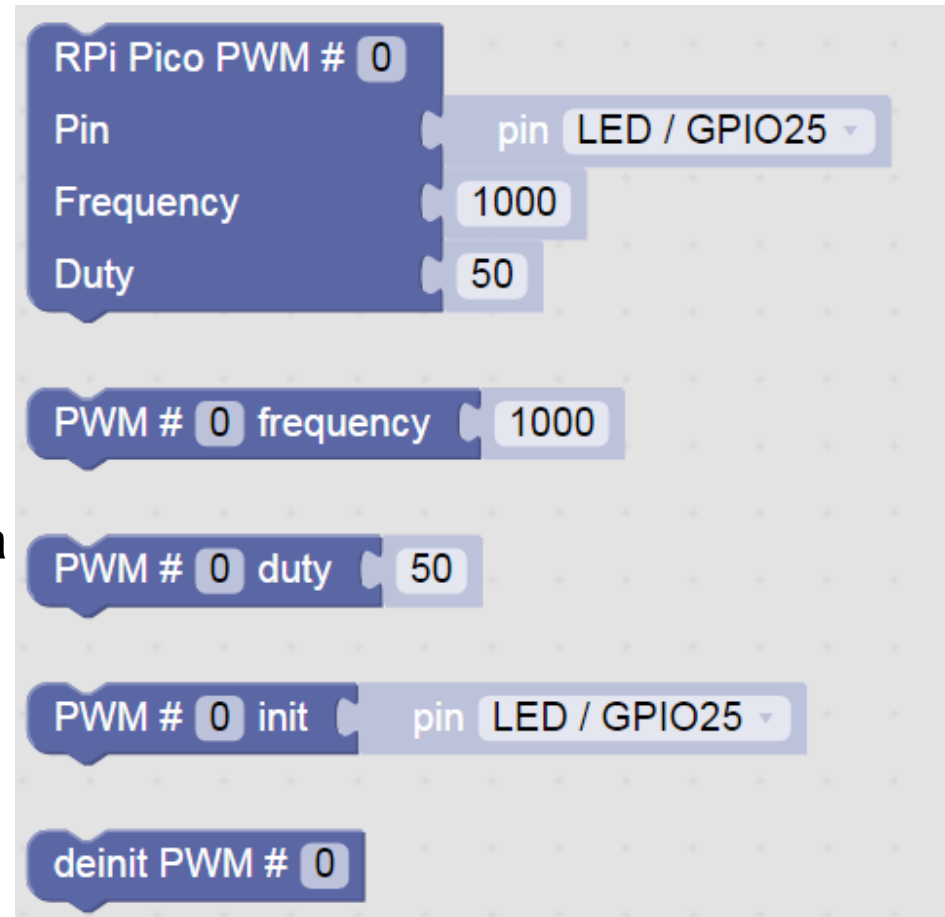
32768 znamená 50 procent !!

To souvisí s formou vyjádření 16- bitového binárního čísla

Pokud dáme 50, tak to skoro nebude svítit.

Nula- 0 nesvítí vůbec,

65535 svítí plně

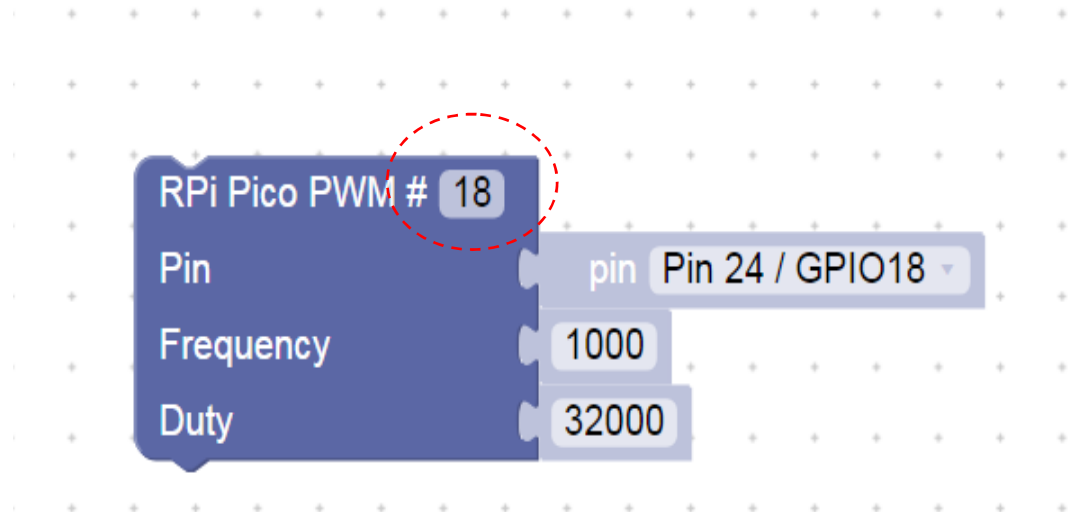


Jednorázová inicializace PWM

Červená LED bude svítit plovičným 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
```


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

Pulse **W**idth **M**odulation

```
repeat while true  
do  
  set output pin pin Pin 29 / GPIO22 to true  
  delay microseconds 1000  
  set output pin pin Pin 29 / GPIO22 to false  
  delay microseconds 1000
```

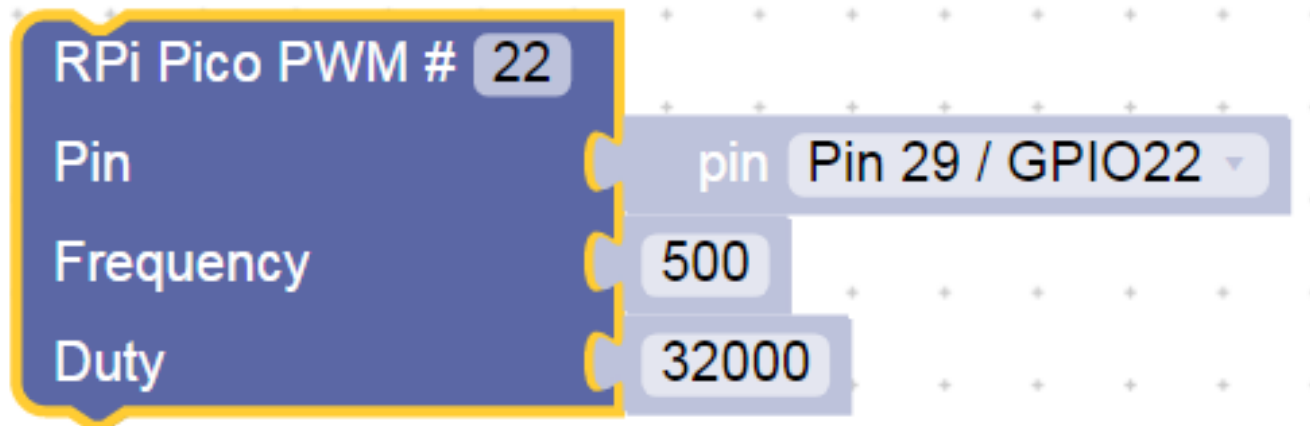
Jednorázová inicializace PWM pro Buzzer

Jednorázová inicializace- buzzer na **GPIO22**

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

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

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



Ovládání bzučáku (buzzer) pomocí PWM

Změna frekvence PWM

```
RPi Pico PWM # 22
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
```

Bzučák s PWM s deinitializací PWM

Na chvíli úplně vypneme PWM

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

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

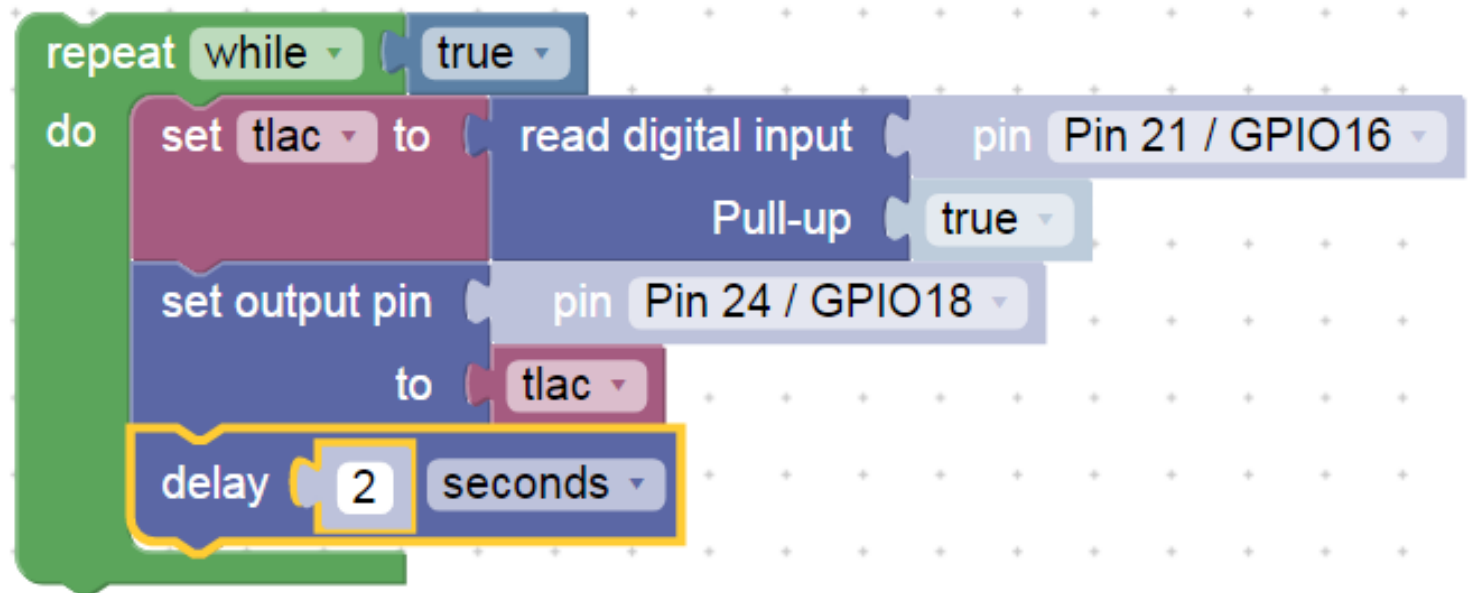
```
repeat while true do
  set output pin pin Pin 24 / GPIO18 to read digital input pin Pin 21 / GPIO16 Pull-up true
```

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**.

```
repeat while true
do
  set output pin pin Pin 24 / GPIO18 to true
  delay 0.5 seconds
  set output pin pin Pin 24 / GPIO18 to false
  delay 0.5 seconds
  if false = read digital input pin Pin 21 / GPIO16 Pull-up true
  do break out of loop
```

▪

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

```
RPi Pico PWM # 18
Pin Pin 24 / GPIO18
Frequency 10000
Duty 0
set LED_duty to 0
repeat while true
do
  change LED_duty by 1000
  PWM # 18 duty LED_duty
  delay milliseconds 100
  if LED_duty >= 30000
  do
    set LED_duty to 0
```

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

```
set poc to 0
repeat while true
do
  change poc by 1
  set output pin Pin 24 / GPIO18 to true
  delay 0.5 seconds
  set output pin Pin 24 / GPIO18 to false
  delay 0.5 seconds
  if poc >= 10
  do
    break out of loop
```

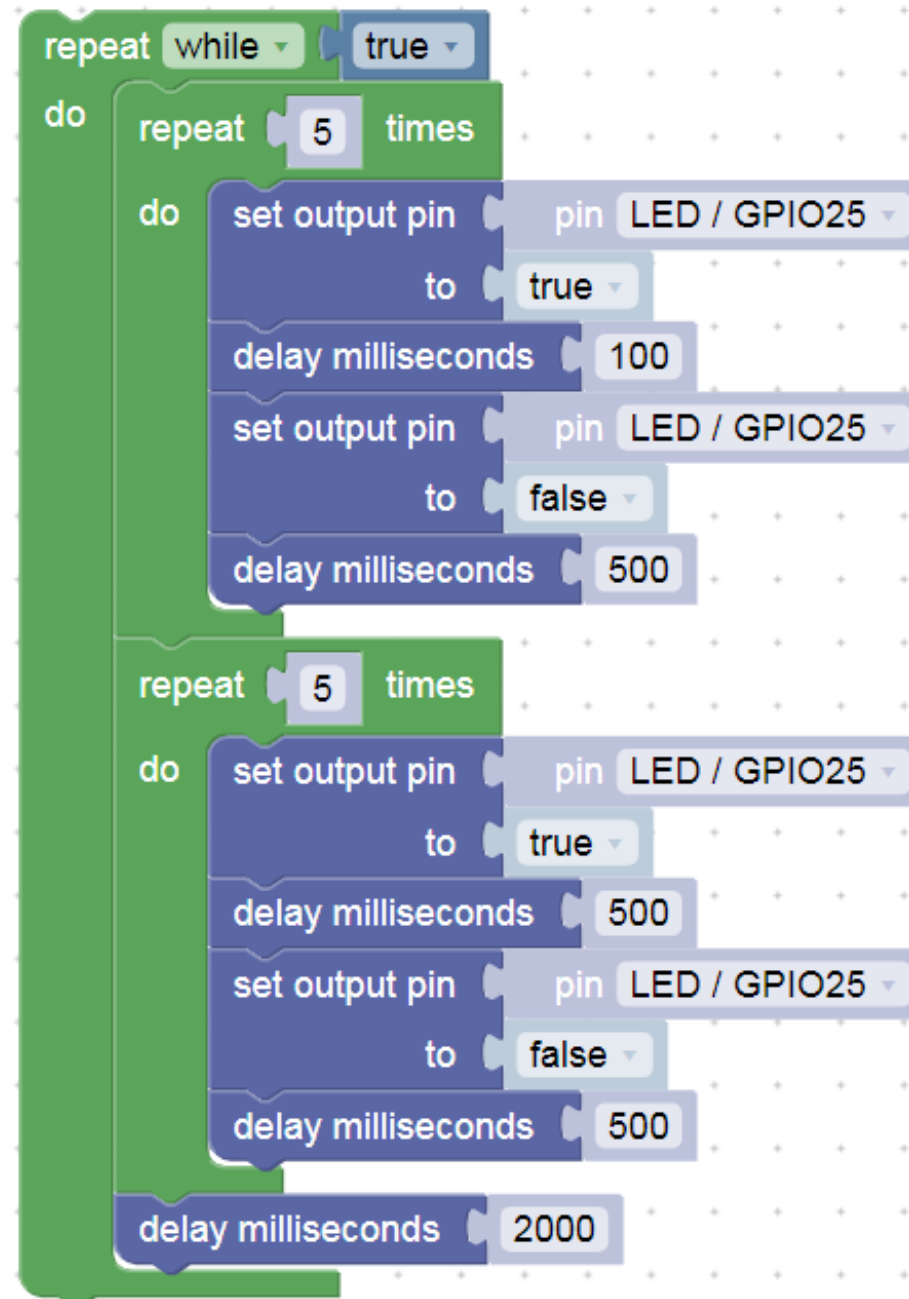
The image shows a Scratch script on a grid background. It starts with a 'set poc to 0' block. This is followed by a 'repeat while true' loop block. Inside the loop, there is a 'do' block containing several sub-blocks: 'change poc by 1', 'set output pin Pin 24 / GPIO18 to true', 'delay 0.5 seconds', 'set output pin Pin 24 / GPIO18 to false', 'delay 0.5 seconds', and an 'if poc >= 10' block. The 'if' block has a 'do' sub-block containing 'break out of loop'. A red dashed box highlights the 'if' and 'do' blocks.

Postupná změna frekvence PWM pro bzučák (buzzer)

Frekvence 100 Hz až 2000 Hz

The code block is a Scratch-style script for an RPi Pico. It starts with a 'RPi Pico PWM #' block set to 22. Below it are three input fields: 'Pin' set to 'Pin 29 / GPIO22', 'Frequency' set to 100, and 'Duty' set to 32000. The main logic begins with a 'set Frkvence to 100' block. This is followed by a 'repeat while true' loop. Inside the loop, there is a 'change Frkvence by 50' block, which is highlighted with a yellow callout box containing the text 'Zvětšovat frek. po 50 Hz' and a red arrow pointing to the '50' value. Below this is a 'PWM # 22 frequency' block with 'Frkvence' as the input. This is followed by a 'delay milliseconds 200' block. The loop ends with an 'if Frkvence >= 2000' block, which contains a 'do set Frkvence to 100' block to reset the frequency.

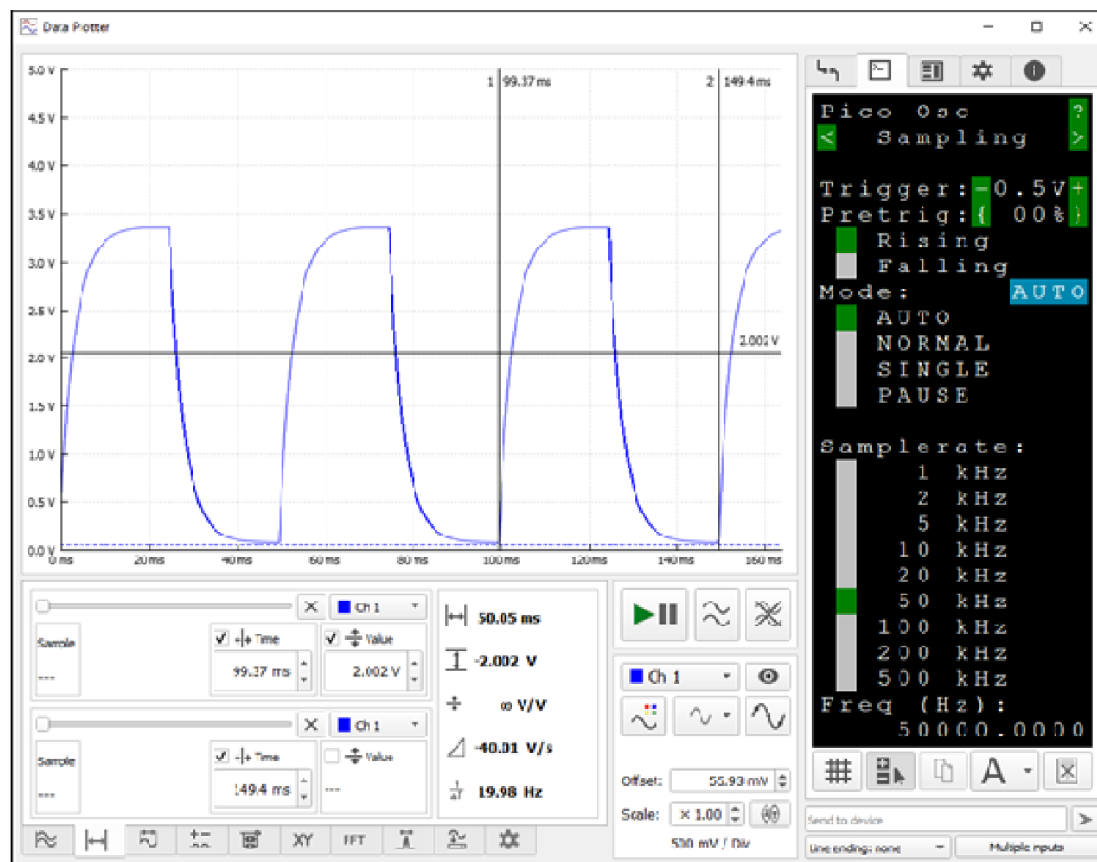
Blikání s využitím smyčky
Blikne 5x krátce 100 ms
S pauzou 500 ms, pak
blikne 5 x 500 ms s pauzou
500 ms
A tak dále „kolem dokola“



Experimenty pro pokročilé

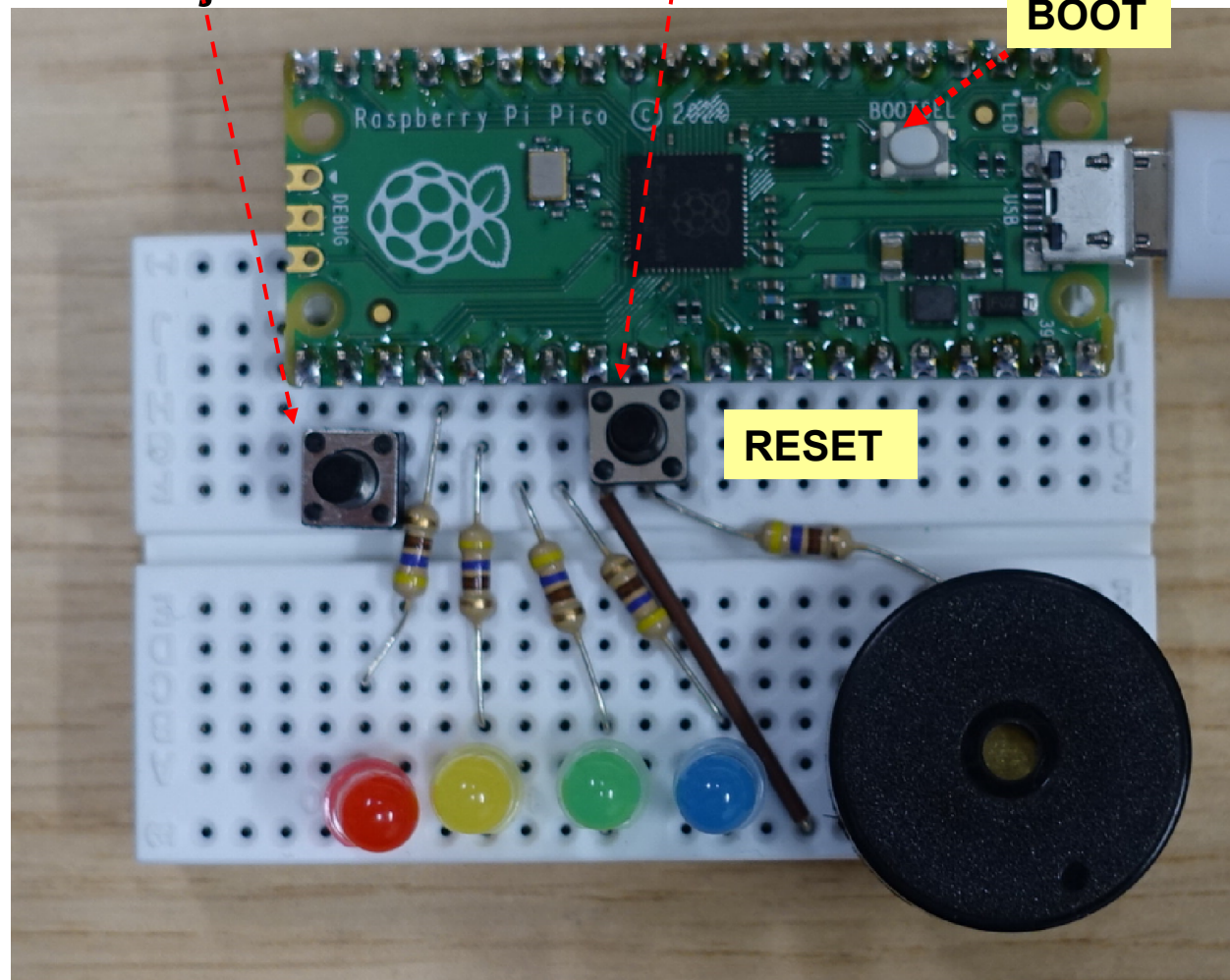
Raspberry Pi PICO nám může také (improvizovaně) nahradit měřicí přístroje, osciloskop, logický analyzátor.

Na https://embedded.fel.cvut.cz/SDI/RP_PICO/Osciloskop



Experimenty pro pokročilé

Příprava, vyjmout tlačítko přemístit do pozice mezi GND a RUN
Levý kontakt tlačítka je u hnědé vodiče - GND

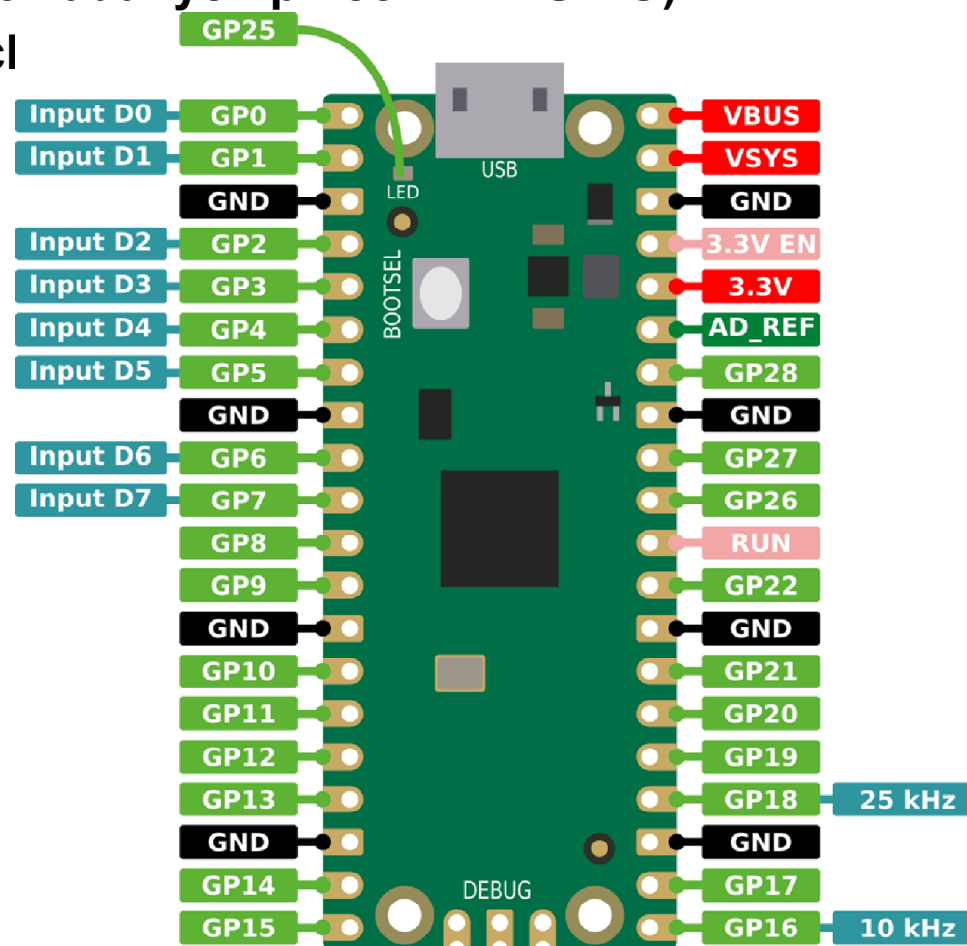
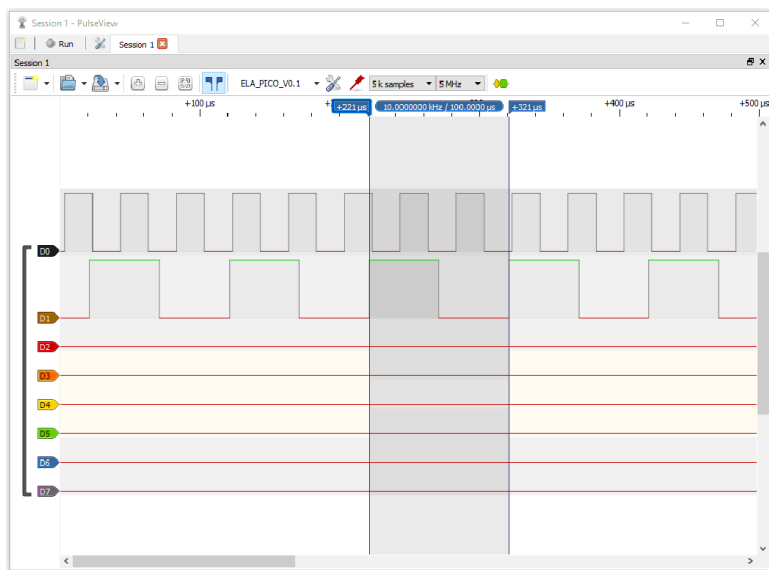


Logický analyzátor ELA s RP PICO

. https://embedded.fel.cvut.cz/SDI/RP_PICO/ELA

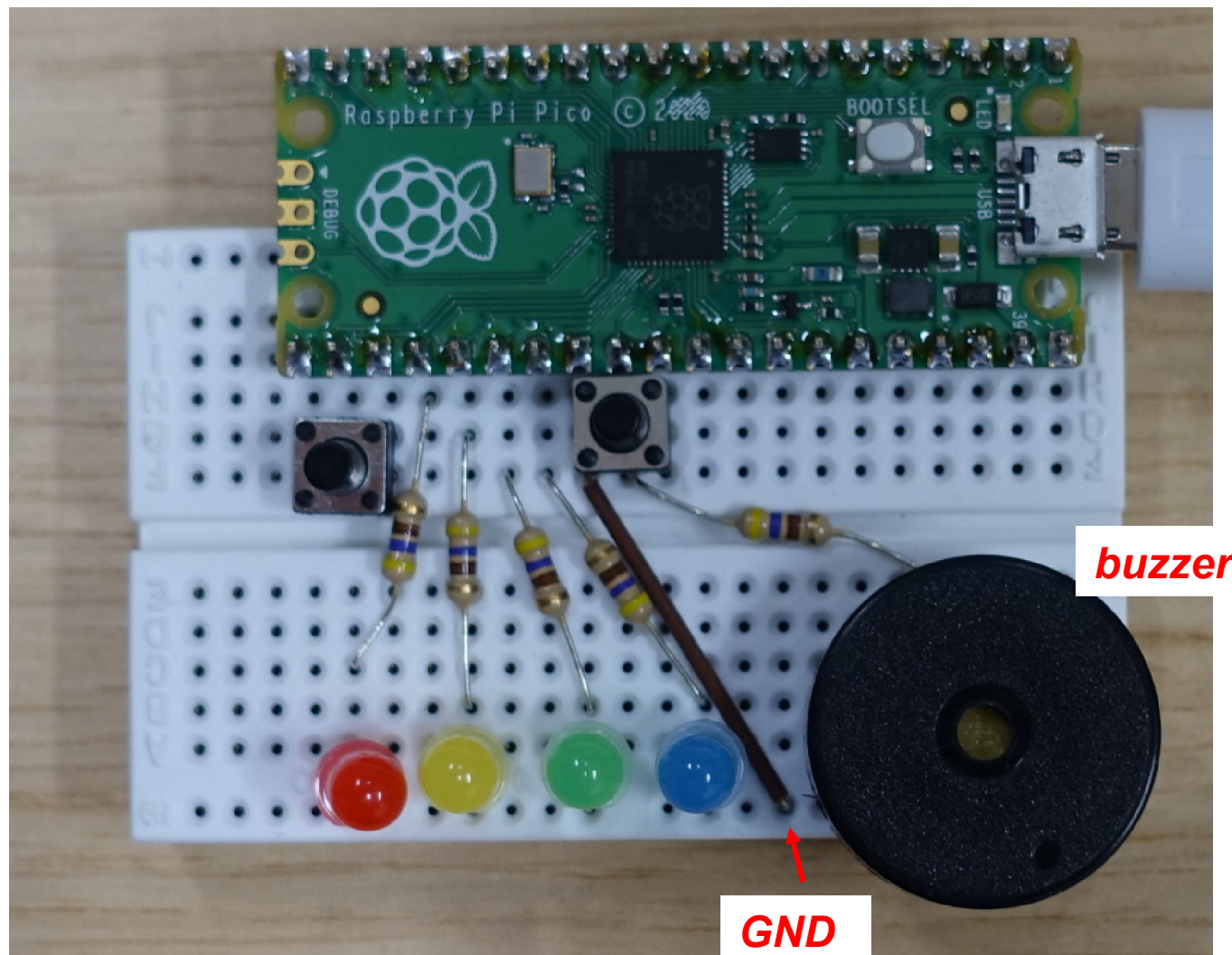
Logický analyzátor - přístroj pro sledování časového průběhu logických signálů (např. na programem ovládaných pinech ARDUINO)

Podporuje analýzu komunikačních
UART, SPI, I2C BUS,.....



Osazení experimentálního pole

-



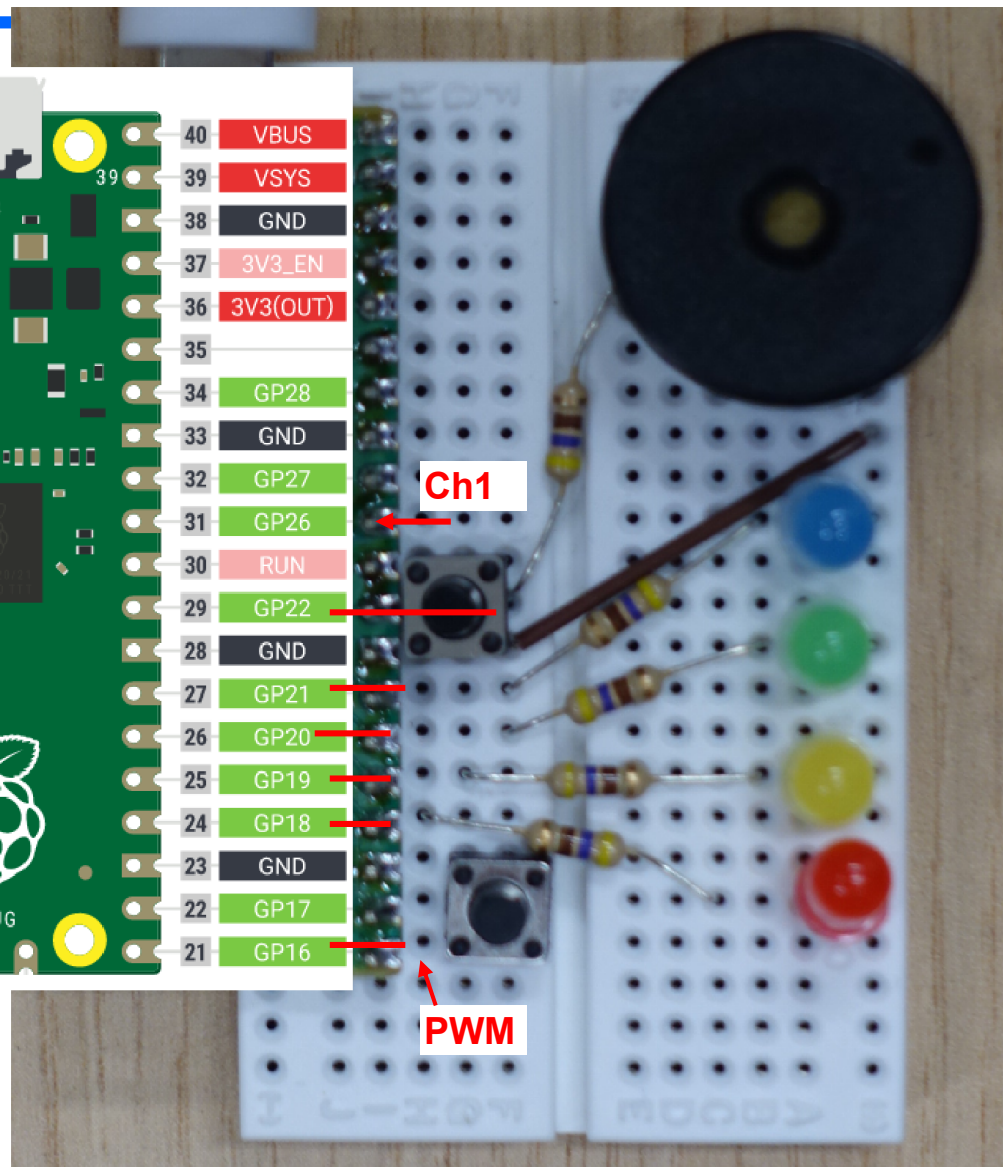
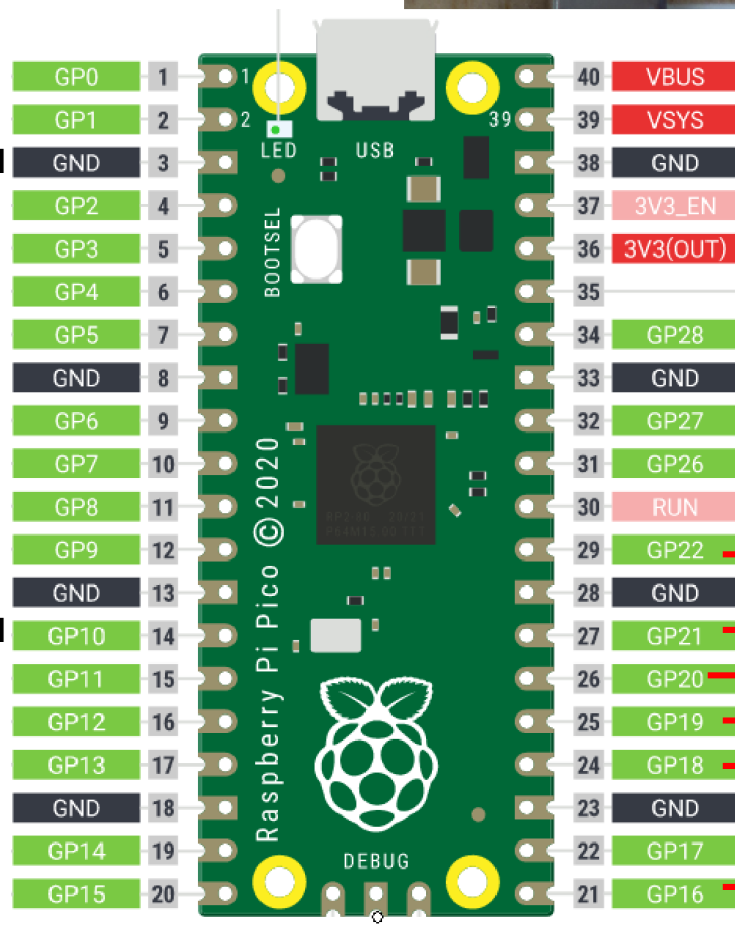
Ch1

vstup
osciloskopu

PWM

výstup
PWM
generátoru
při funkci
osciloskopu

GND – zem
RUN - reset



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**

Pro grafické programování , nahrát micropython - stránky [https://embedded.fel.cvut.cz/stredni skoly/akce BOSCH](https://embedded.fel.cvut.cz/stredni_skoly/akce_BOSCH)

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

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

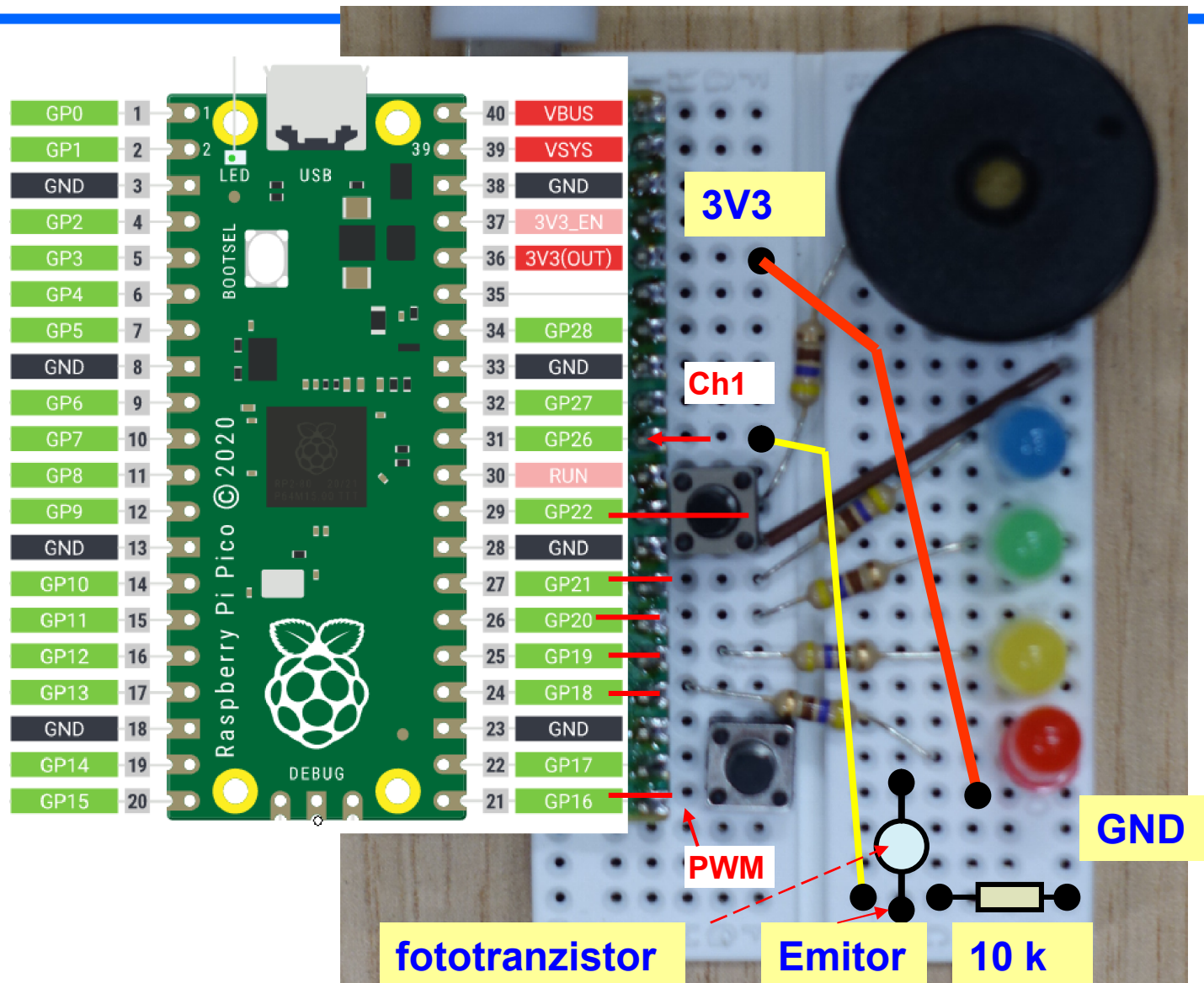
Experimety s oscilokopem

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).

Zapojení fototranzistoru s osciloskopem s PR PICO



Panel osciloskopu s PC aplikací Data plotter

- Připojení k USB volba menu

