
Workshop pro učitele fyziky a informatiky z gymnázií

**Mikropočítač Raspberry PI PICO a možnosti jeho využití při praktické
výuce fyziky, informatiky a programování na gymnáziích**

Prezentující: doc. Fischer, katedra měření ČVUT – FEL, 2.2.2023

Náplň

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

Softwarově definované přístroje SDI - Software Defined Instruments

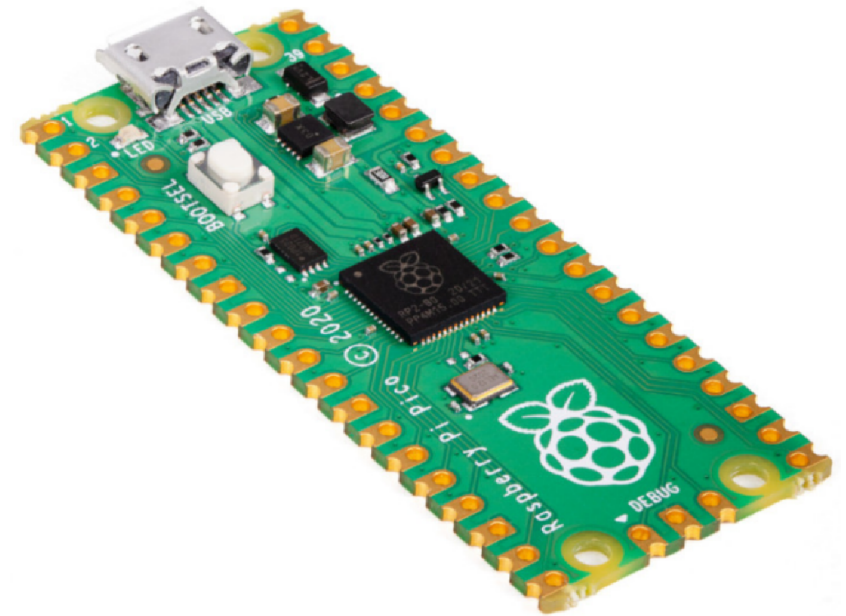
Možnosti realizace SDI

SDI, které jsou k dispozici

RP PICO jako SDI (osciloskop, ...)

Programování RP PICO na základě

Micropython s využitím nadstavby pro grafické programování BIPES



Workshop s RP PICO, experimenty

- Určení **rychlosti zvuku**
- Měření **rychlosti pohybu**
- **RP PICO jako měřicí přístroj** pro laboratorní experimenty
- **Grafické programování**

Náhrada přístrojů pro laboratorní experimenty, SDI

Laboratorní experimenty – potřeba laboratorní (měřicí) přístroje.

Měření neelektrických a elektrických veličin

Pro výuku **fyziky** na gymnáziích – asi hlavně měření neelektrických veličin; možné s převodem na elektrické veličiny

Fyzikální neelektrická měření - často použity elektronické měřicí přístroje- **měření času, zobrazení průběhu měřené veličiny, vyhodnocení světla, teploty...**

Náhrada přístrojů mikropočítačem

Růst výkonu současných mikrořadičů (**embedded microcontrollers**) nebo také **MCU – Microcontroller Unit**,

Jejich vybavení periferiemi (převodníky analogo - číslicové, čítače, převodníky číslico- analogové,..ADC) umožňuje **realizovat měřicí přístroje jen programovými prostředky s využitím vnitřních bloků MCU.**

To označujeme jako **SDI Software Defined Instruments**

Motivace začátku řešení **SDI** na kat. měření

Úlohy, samostatné projekty, např, v mikroprocesorové technice, student potřebuje při řešení úlohy ruční multimetr a osciloskop
Pokud nestíhal, potřeboval se dostat i mimo čas standardní výuky do laboratoře,.. Komplikace, čas navíc..

Proto byl vyvinut před cca 10 lety náš první přístroj
LEO - Osciloskop s generátorem

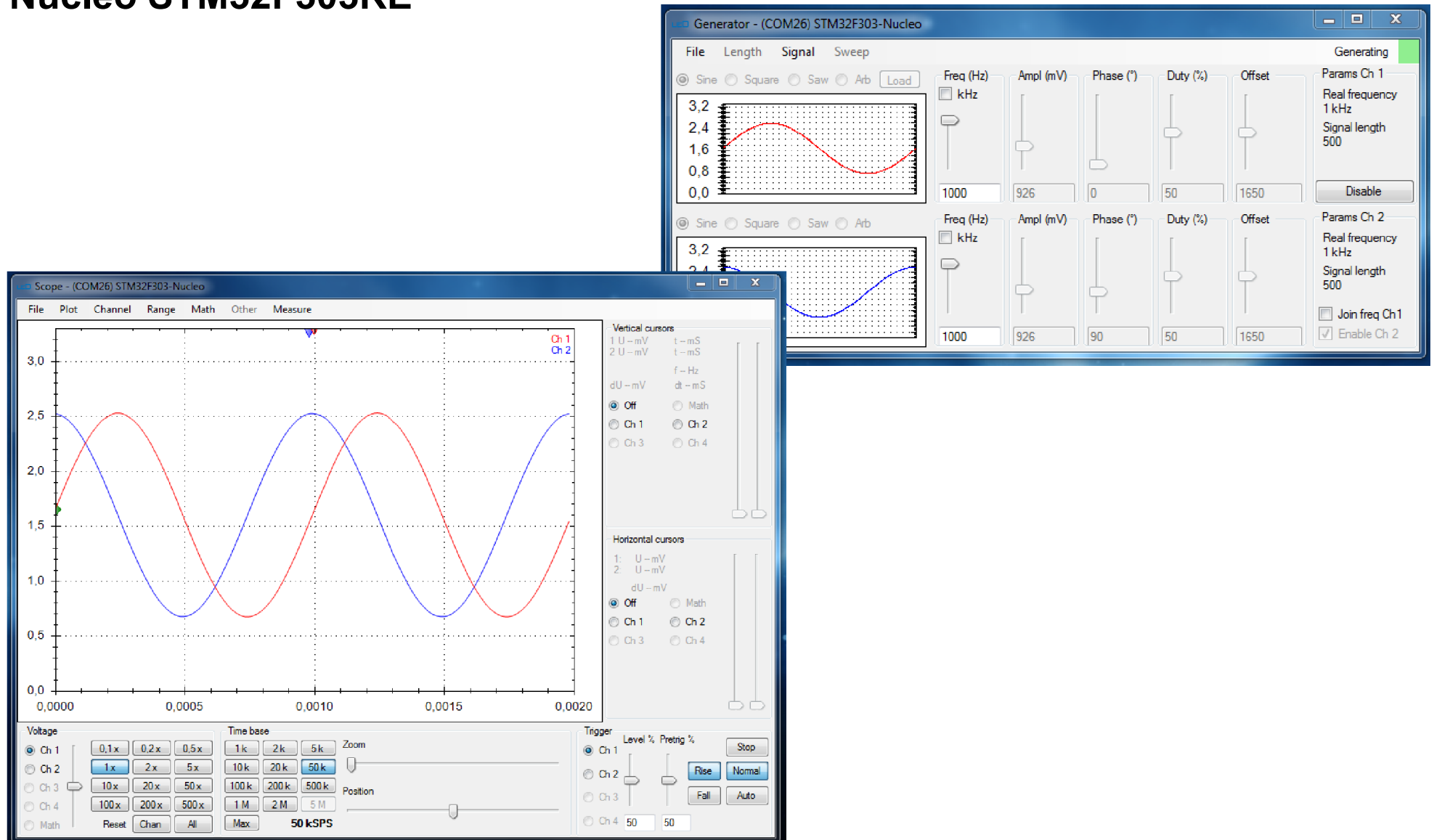
Zatím nejlépe vyvinutý a hromadně používaný,

Osvědčil se při praktické distanční výuce,

Veškerý SW a návody k použití jsou k dispozici na
embedded.fel.cvut.cz

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

LEO - Little Embedded Oscilloscope Nucleo STM32F303RE



Možnost použití Arduino jako SDI

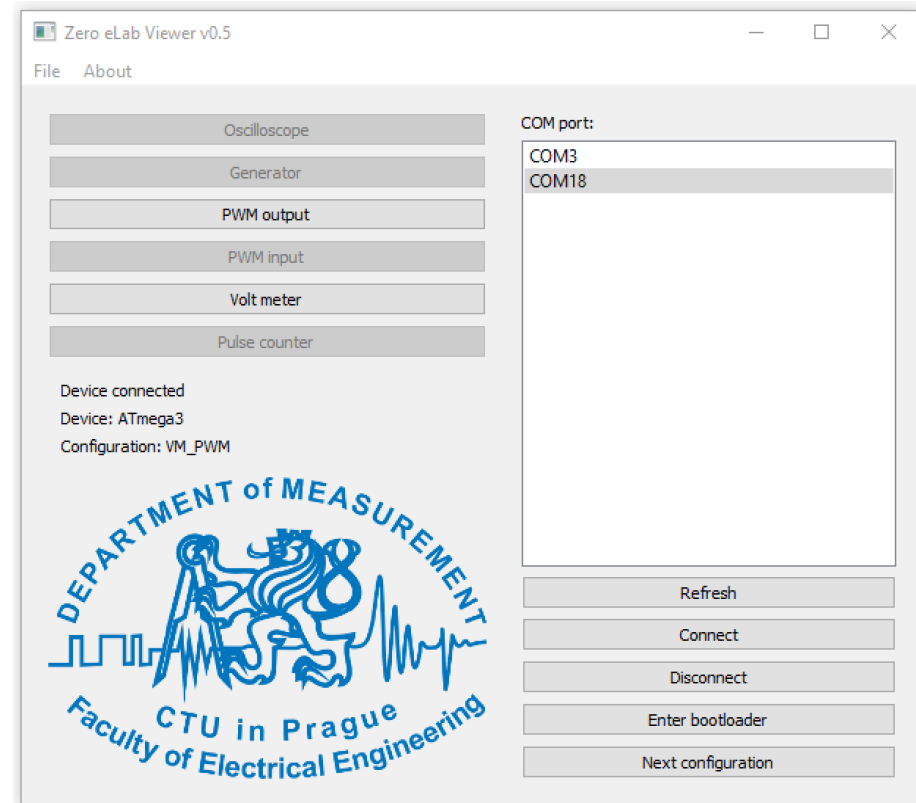
Mnoho uživatelů Arduina, po řadě výkonných SDI jsme v r. 2021 vyvinuli nejméně výkonný **SDI s Arduino**.

Pro řadu **fyzikálních experimentů** může postačovat, „**ukázat princip**“

Přístrojové funkce: voltmetr, záznamník (waweform recorder), impulzní generátor, osciloskop

k dispozici na

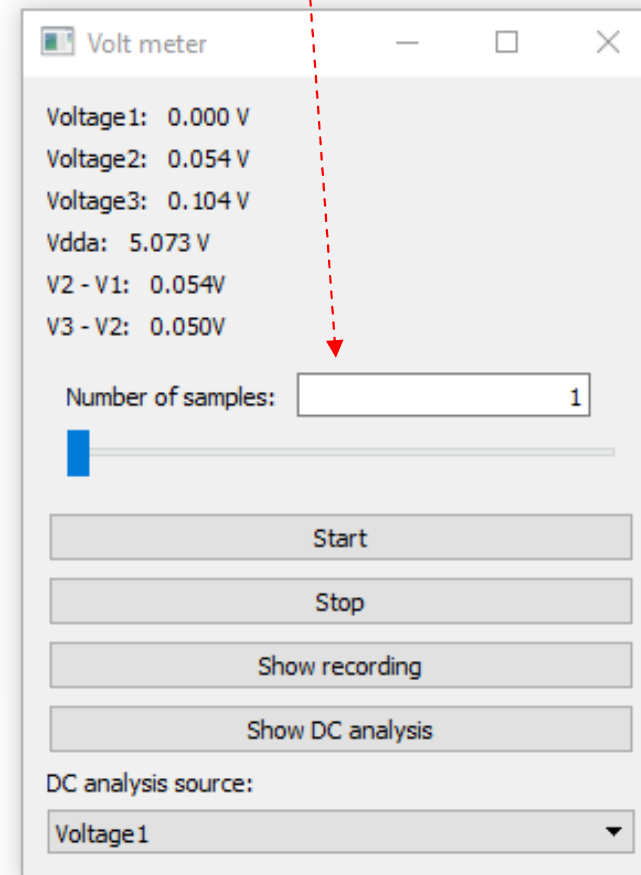
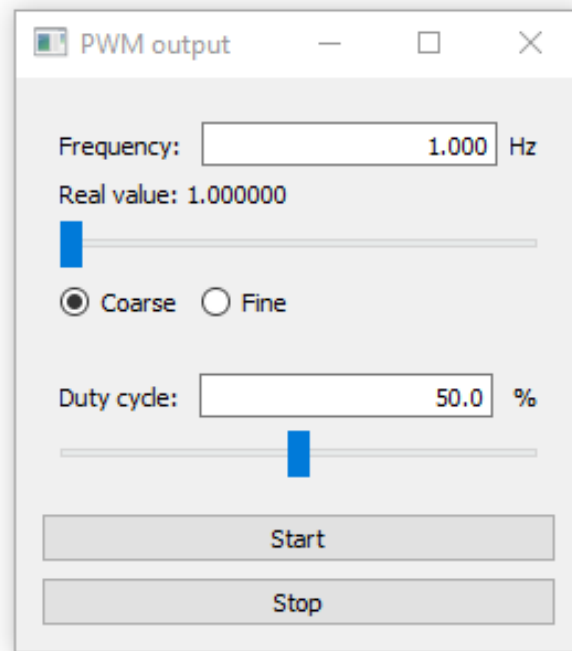
<https://embedded.fel.cvut.cz/SDI>



Arduino – voltmetr, záznamník + impulzní generátor

Impulzní generátor PWM + tříkanálový voltmetr

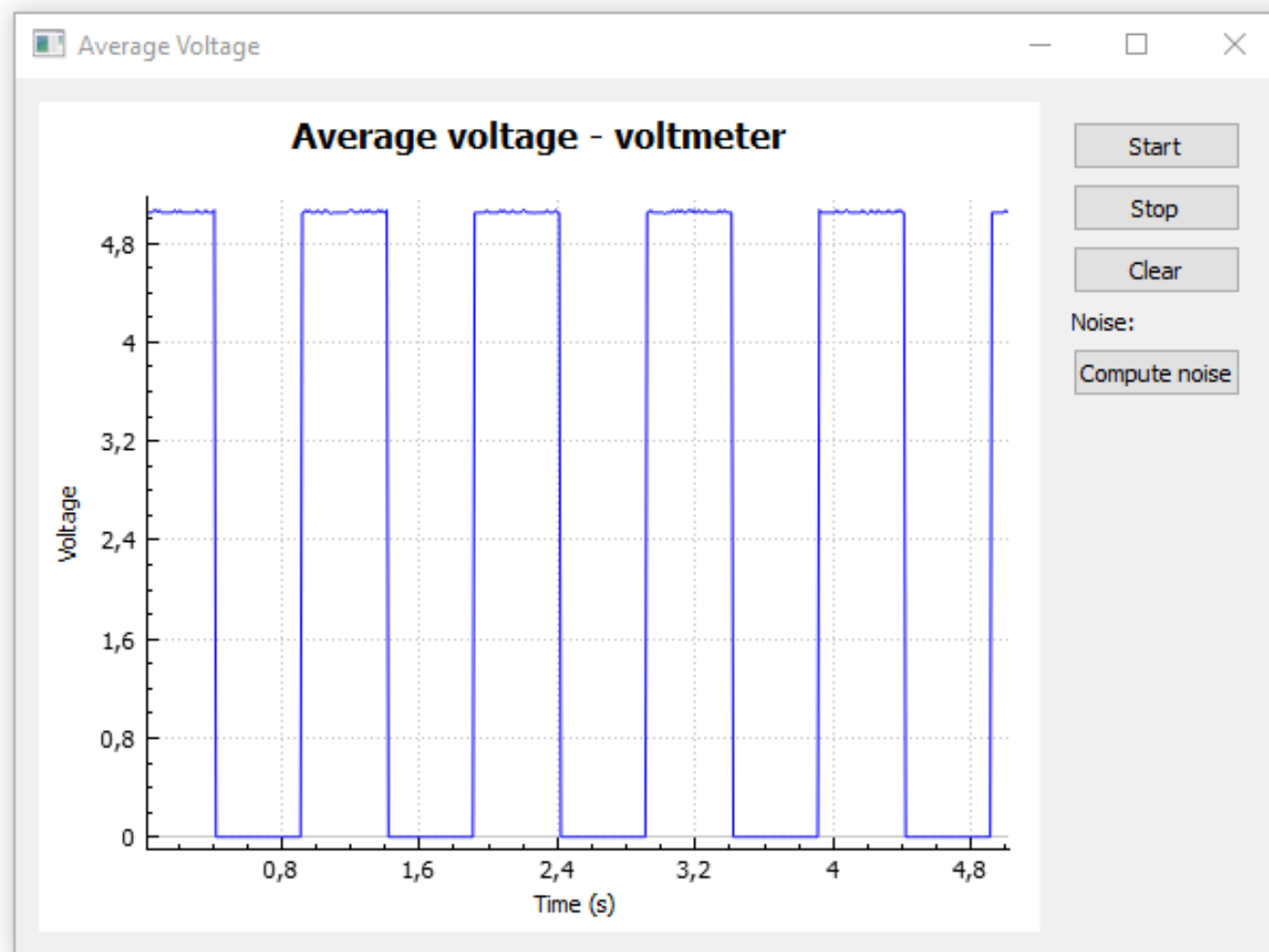
Vzorkování **frekvencí 100 Hz**, možnost **průměrování** pro snížení šumů



Arduino - záznamník

Záznam signálu z fyzikálního experimentu (signál optické závory u kyvadla, záznam změny teploty, změna osvětlení,...)

Grafické znázornění dat z voltmetru...

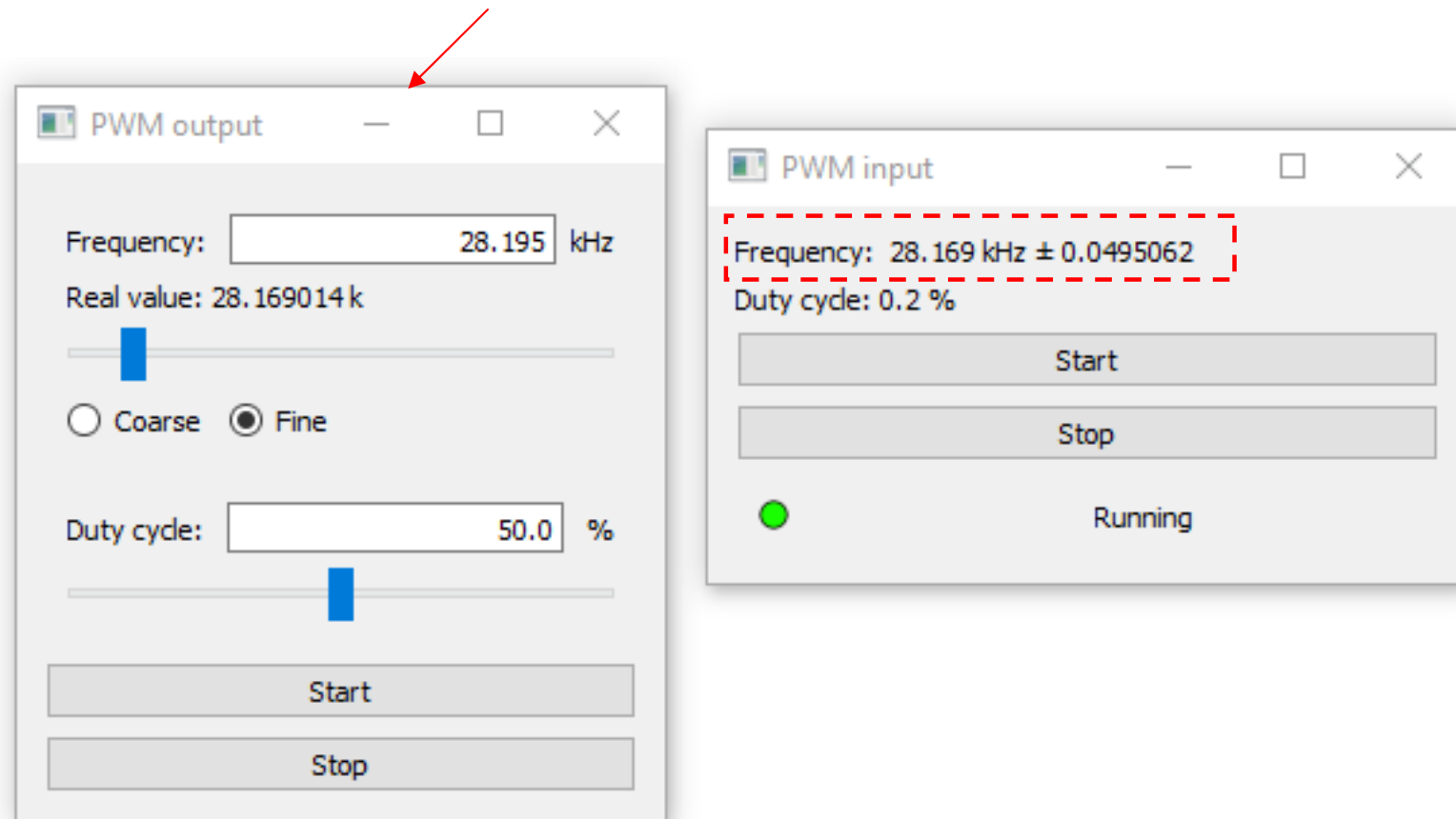


Arduino - čítač

Měření **periody signálu**, z toho výpočet **frekvence** –
reciproční měření frekvence, časové rozlišení **0,125 mikrosekundy**

Pokusy – měření frekvence signálu, určení intervalu impulsů,..

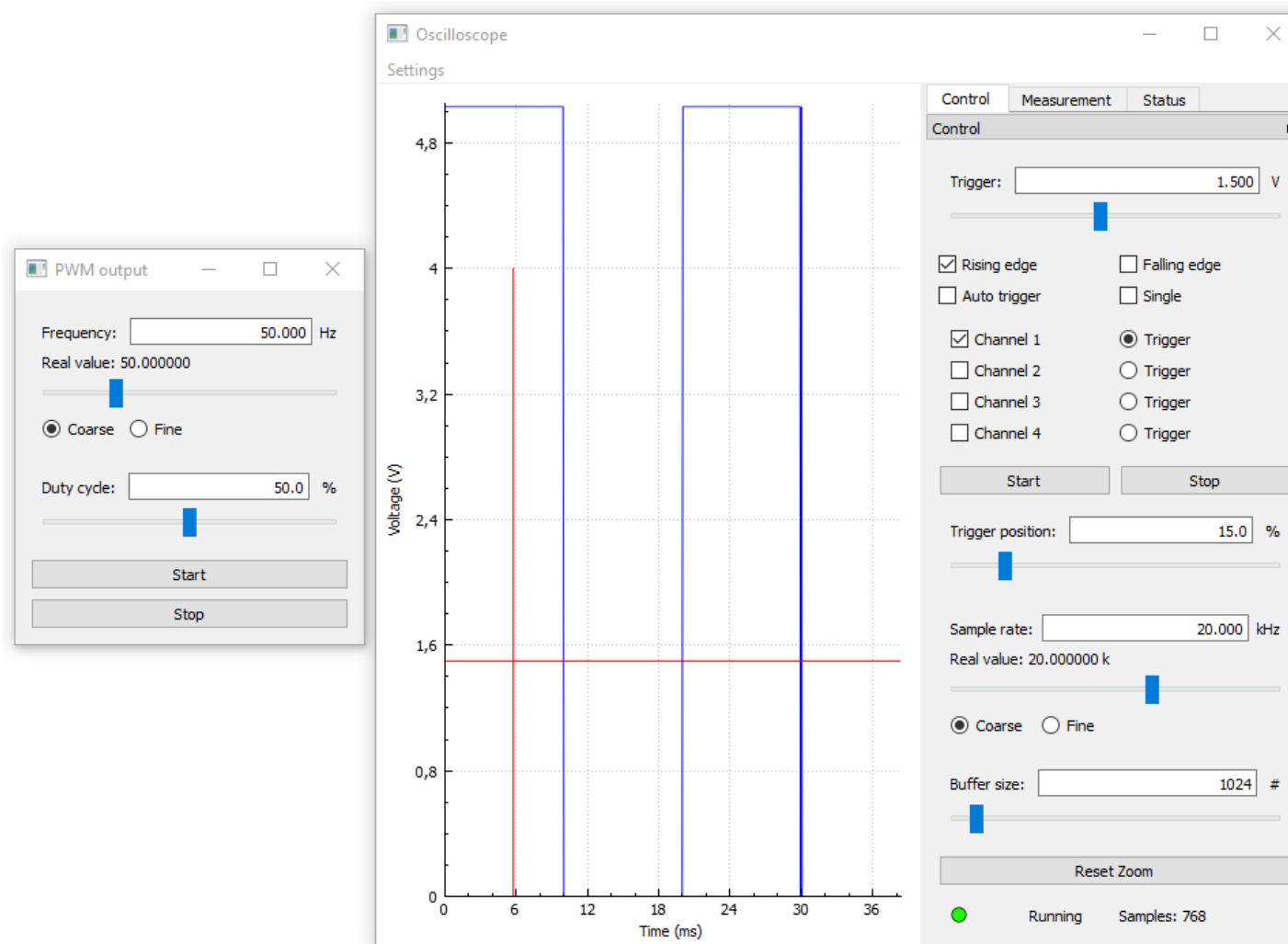
K dispozici též **impulzní generátor PWM** pro testování funkce



Arduino osciloskop

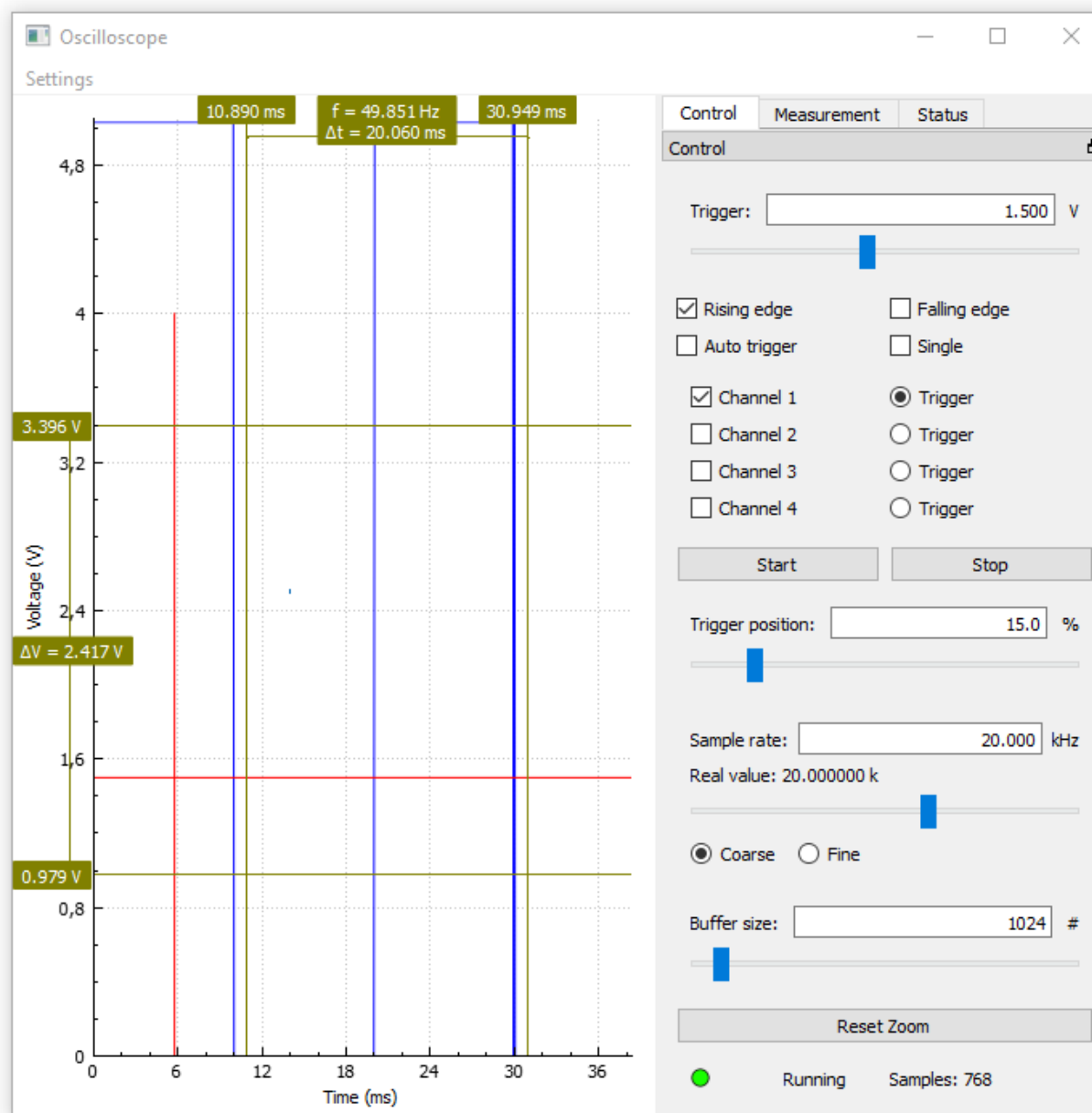
Impulzní generátor PWM + osciloskop jedno/ dvoukanálový

Vzorkování až **1x 80 KS/s**, až 768 S, nebo **2x 50 Ks/s** (až 2 x 384 S)



Arduino – osciloskop - měření

Měření parametrů
signálu
pomocí kurzorů
Export záznamu
do xxx.CSV
Pro **MS Excel**



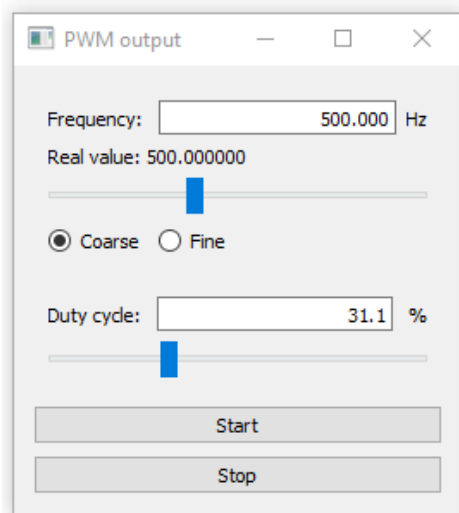
Arduino- logický analyzátor

L.A. 6 kanálů až 100 kS/s

Délka záz

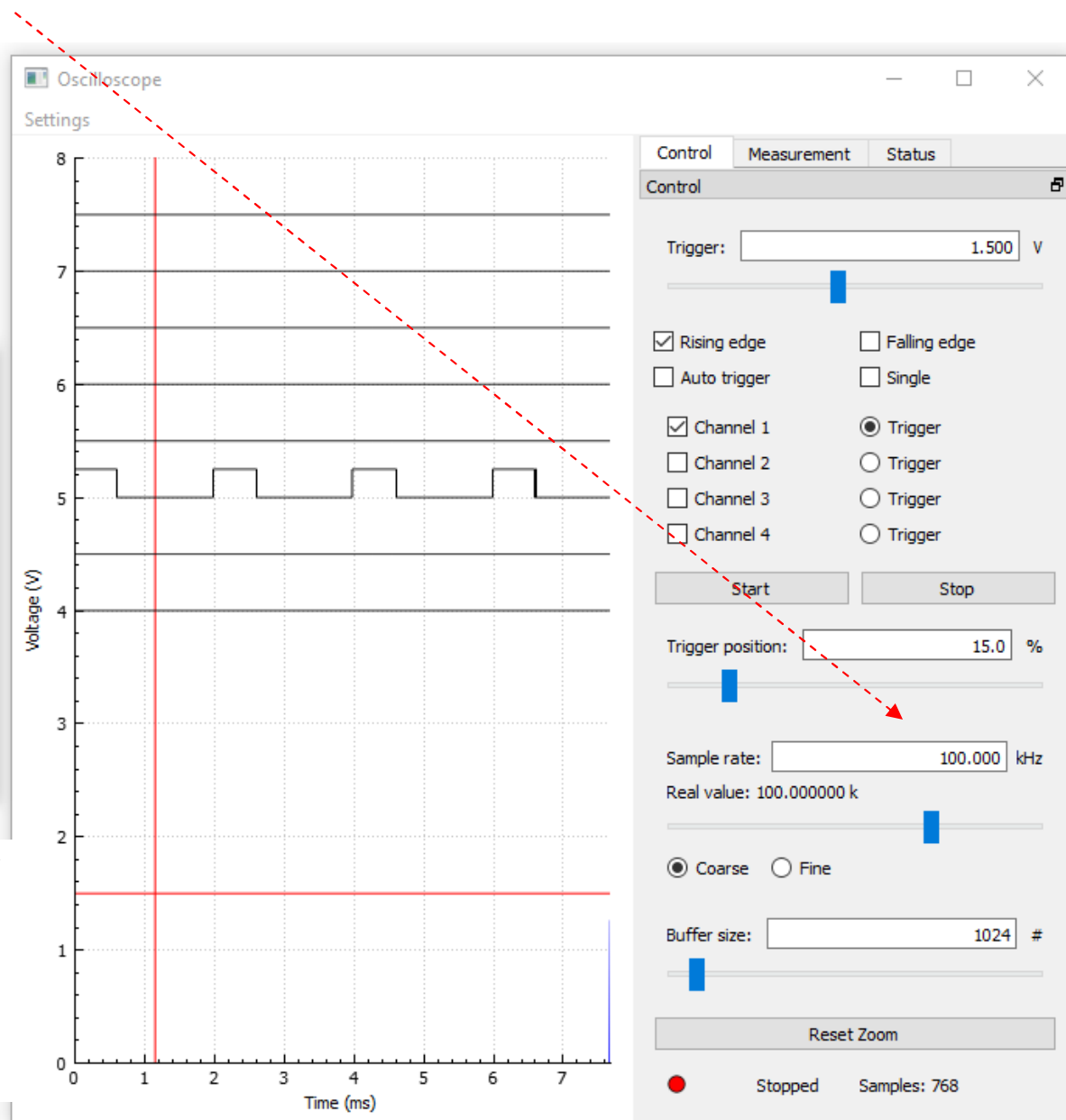
namu- až 768 S

pro sledování výstupních
signálů jiného mikrooč.



Vhodné do výuky programování
mikroočítačů,

Možnost sledovat
stav jejich výstupů



SDI pro experimentální výuku

Na katedře měření ČVUT – FEL – vyvinuta řada SDI, především s **32-bitovými** mikrořadiči řady **STM32** s jádrem **ARM Cortex**

Využívány ve výuce, osvědčily se také při **distanční výuce**.

Vhodné pro **hromadné akce - kurzy**

Kurz praktické elektroniky pro nastupující studenty FEL

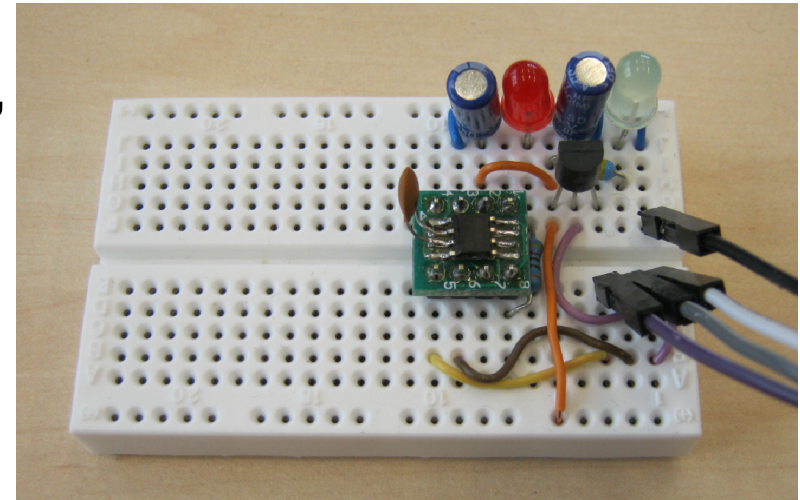
ETC22 - klub pro studenty gymnázií,..



SDI nabízené katedrou měření

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

Hlavní kritérium, jednoduchost konstrukce,
pokud možno nic složitého,
využít pro napájení, ovládání SDI
a zobrazení výsledků PC s USB.



**Nejmenší a
nejjednodušší
osciloskop s
STM32G030J6M6**

Přístroje založené na STM32G030 zatím nejmenší a nejlevnější řešení přístroje

Přístroje založené na STM32F042

Přístroje založené na STM32L072 - Data Plotter L072KZ zde

Přístroje založené na STM32F303RE

Přístroje založené na STM32F401 (ve vývoji)

Přístroje založené na STM32F411 (Osciloskop - připravuje se k zveřejnění)

Přístroje založené na STM32F103 (STM32F103C8 - Blue Pill a STM32F103RE)

Přístroje založené na STM32L412KB - Data Plotter L412 zde

Přístroje založené na STM32G431KBT6 - Data Plotter G431KB zde , VSVI SW zde , Stroboskopický osciloskop

Přístroje - osciloskopy založené na Arduino Data Plotter Arduino zde, Osciloskop pro Arduino Ard-eLabViewer

Přístroje založené na Raspberry PI PICO

Raspberry PI PICO- jako nástroj pro výuku

Po rozšířeném Arduinu se nedávno objevil nový mikropočítač Raspberry PI PICO, který se rozšiřuje mezi mládeží a kutily

Výhody:

- dostupnost (zatím)
- velká komunita uživatelů
- výhodná cena
- populární
- **výkonný** mikropočítač (nesrovnatelně výkonnější oproti Arduino)

Mikropočítač - Raspberry PI PICO

Mikropočítač Raspberry PI PICO

S procesorovým obvodem RP2040

2 MByte externí FLASH

Cena cca 120 Kč s DPH

Vlastní procesorový obvod RP2040

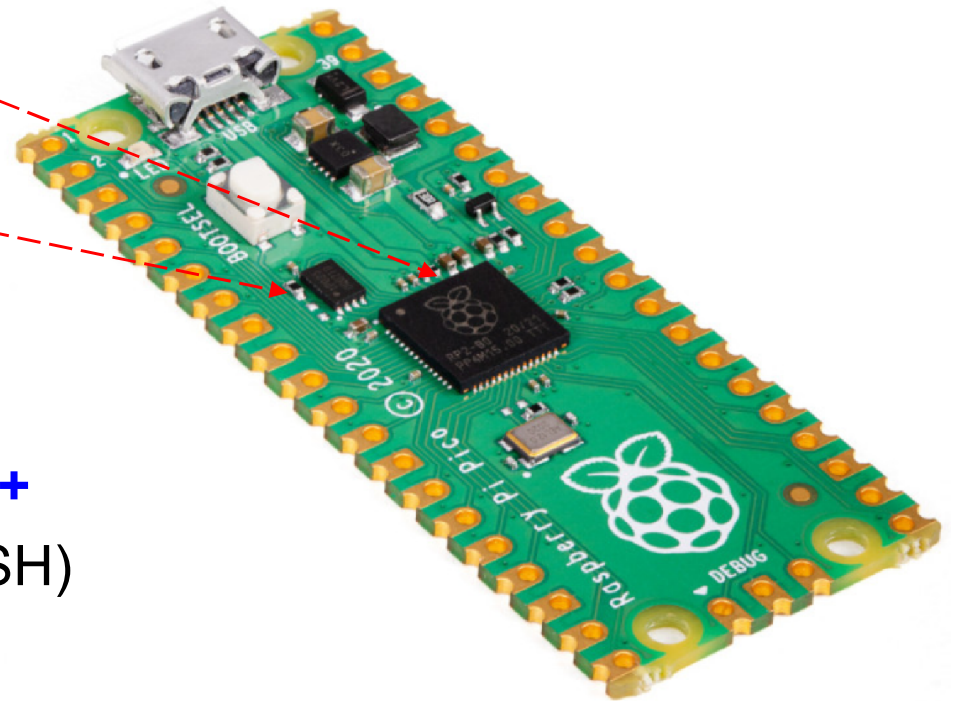
Dvě 32-bitová jádra ARM Cortex M0+

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 –
to je zásadní pro realizaci osciloskopu



Informace a odkazy

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

Informace

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

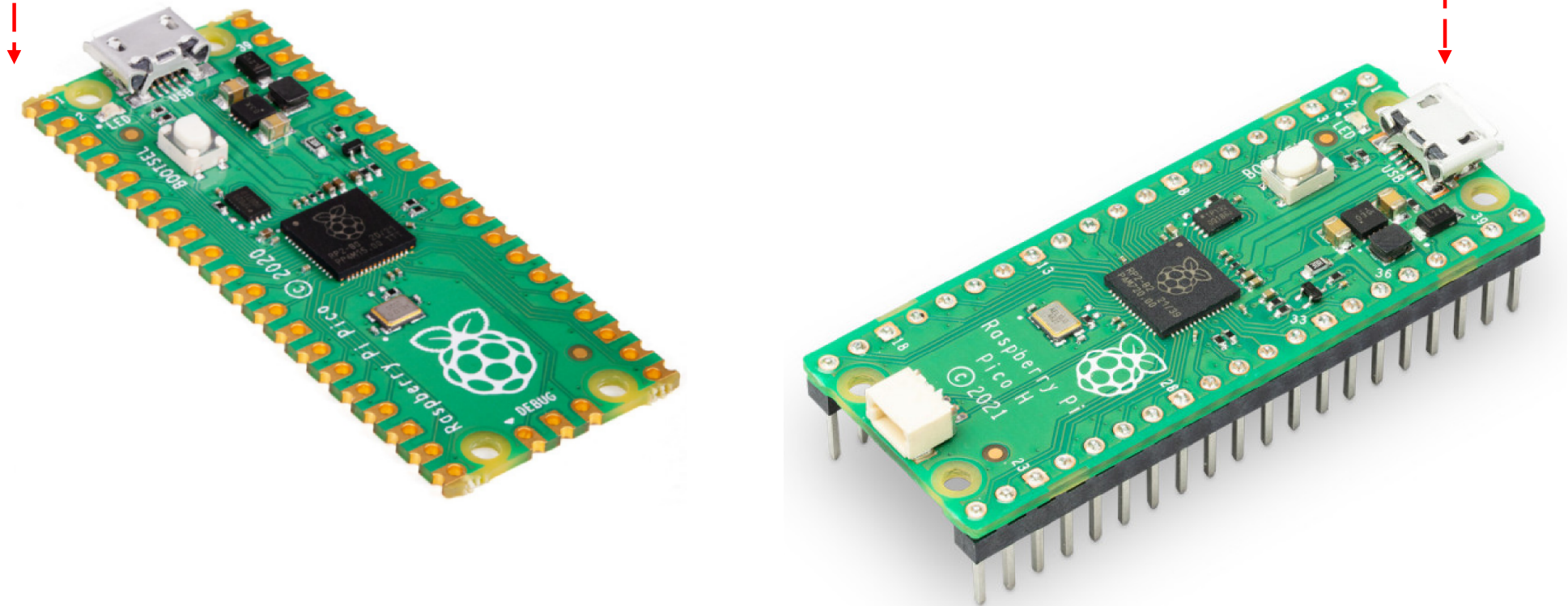
<https://www.raspberrypi.com/products/raspberry-pi-pico/>

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

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

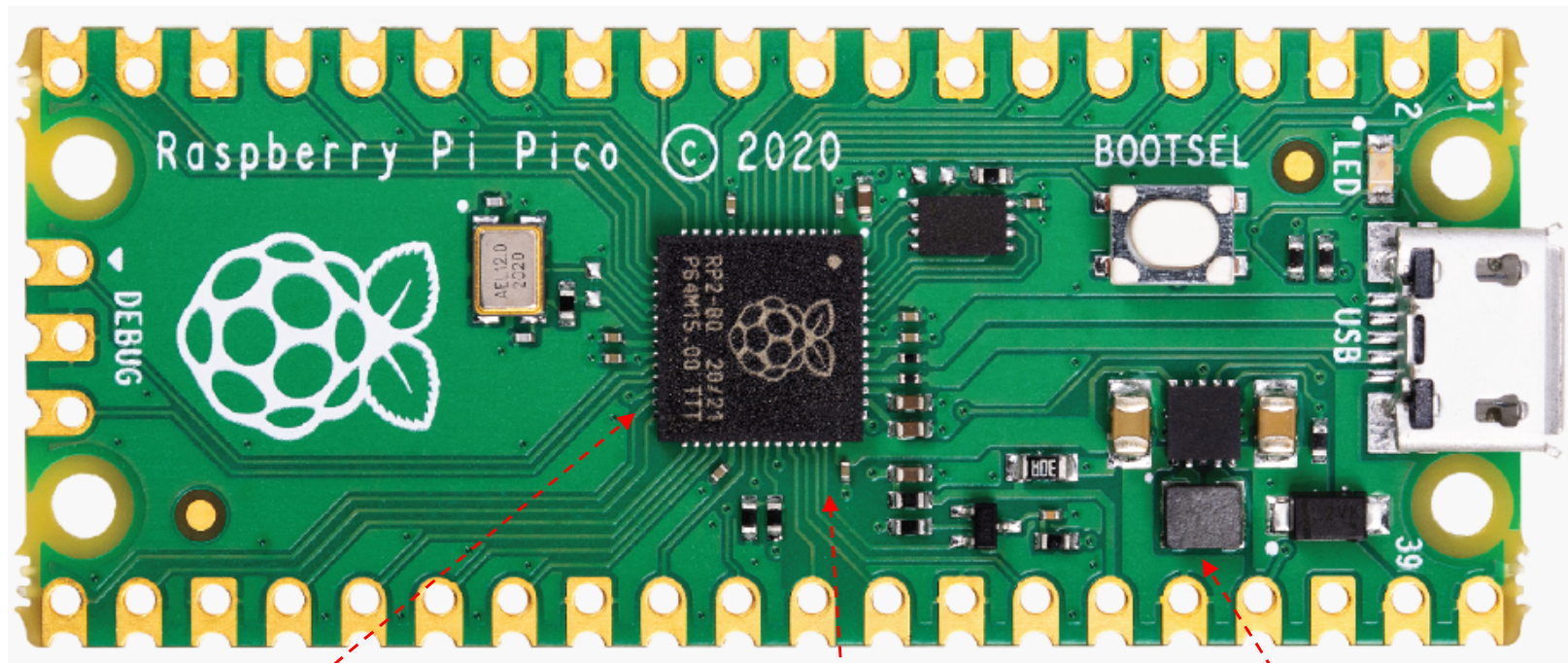
Pokud **nemáme možnost pájet headery**, koupit již verzi **PICO H**

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



Mikropočítač - PI PICO

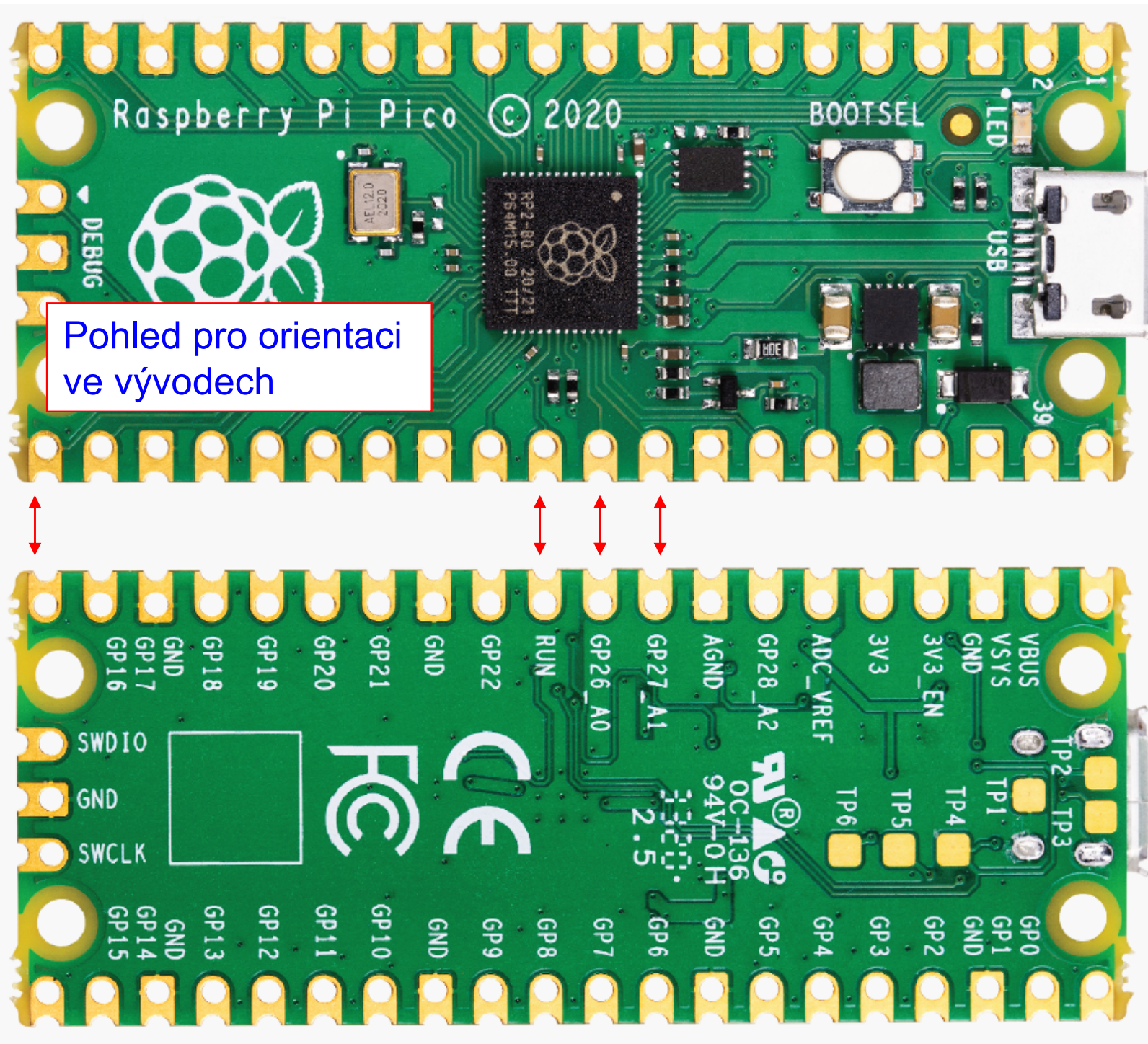
Pohled shora



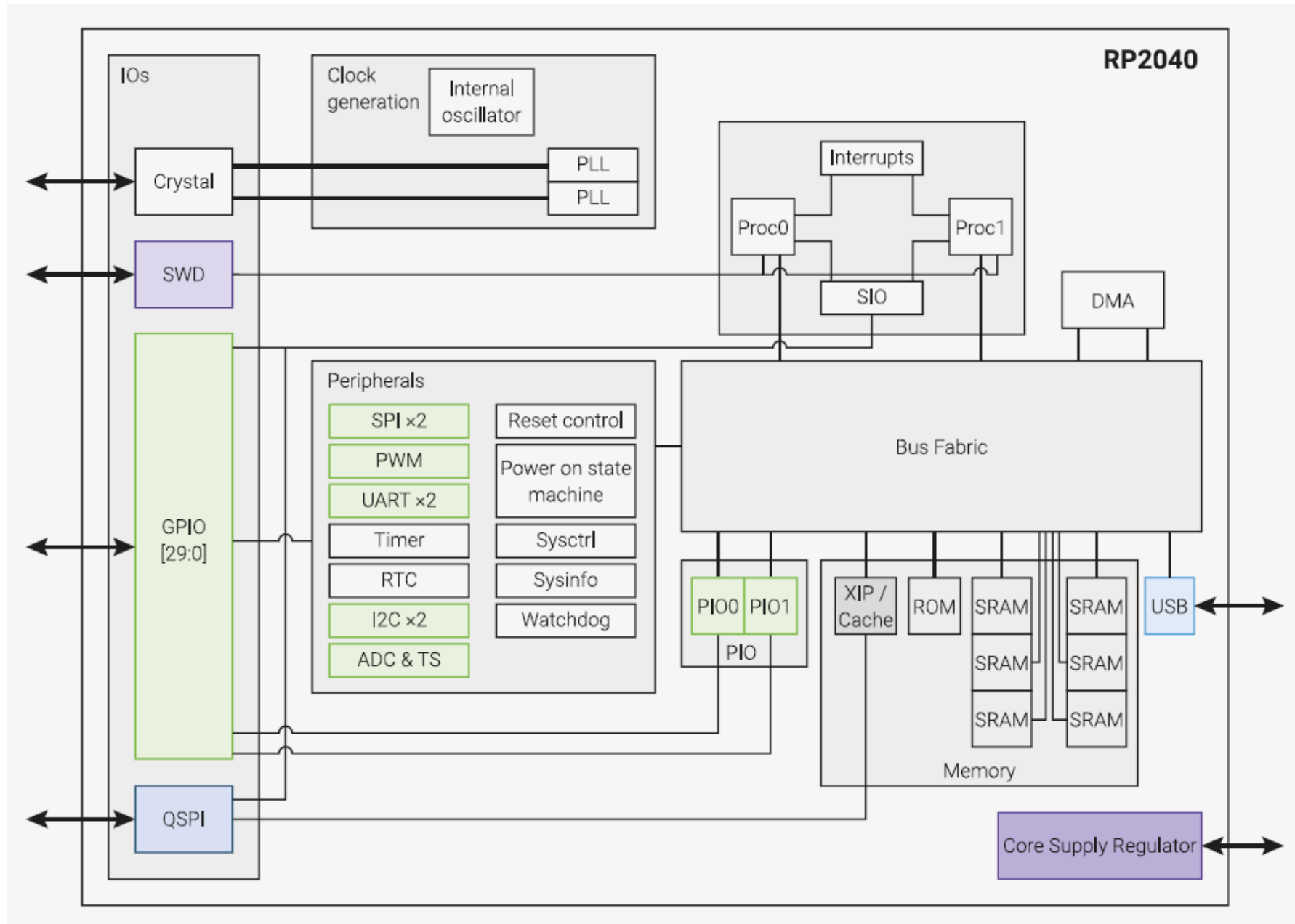
procesor

Externí
FLASH

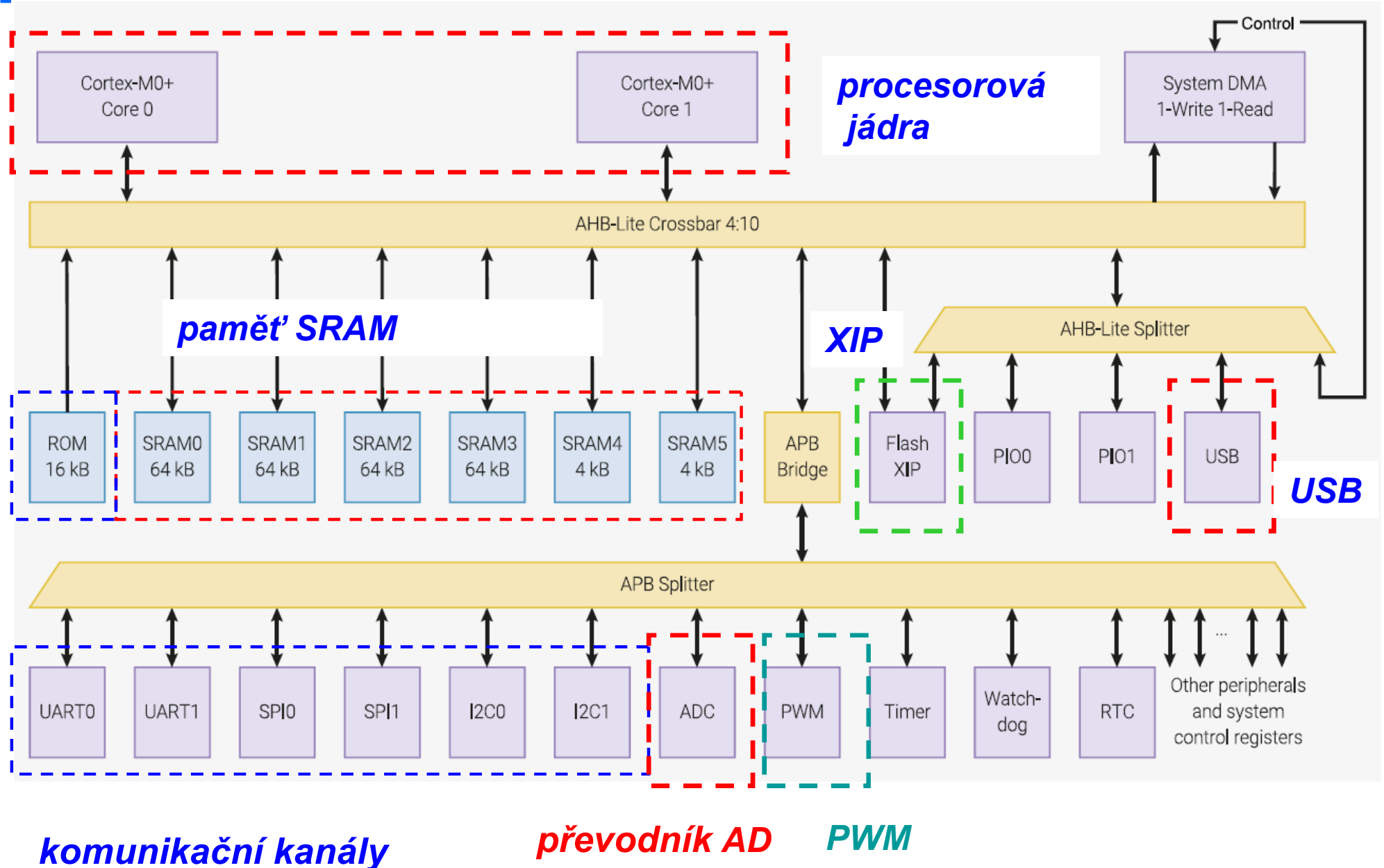
Pulzní
zdroj



PR PI PICO struktura



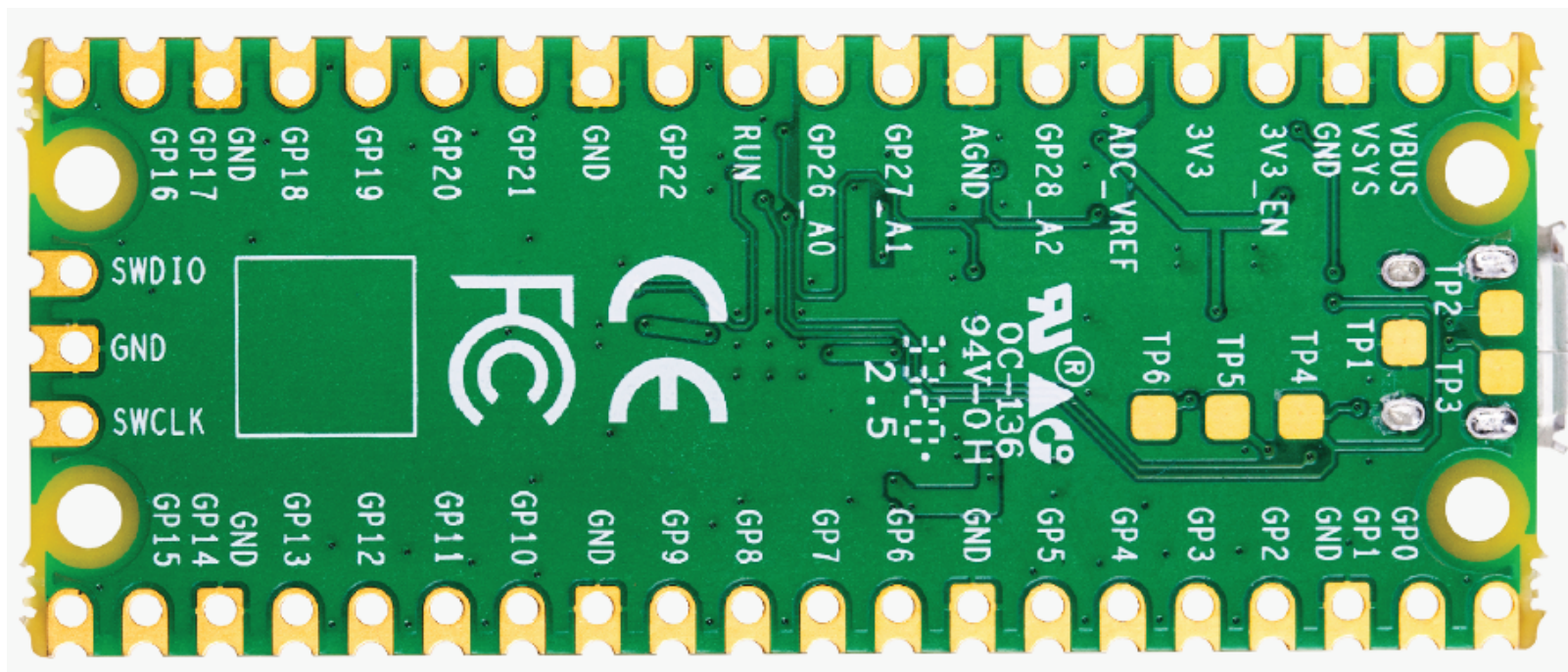
PI PICO Paměťové bloky a periferie



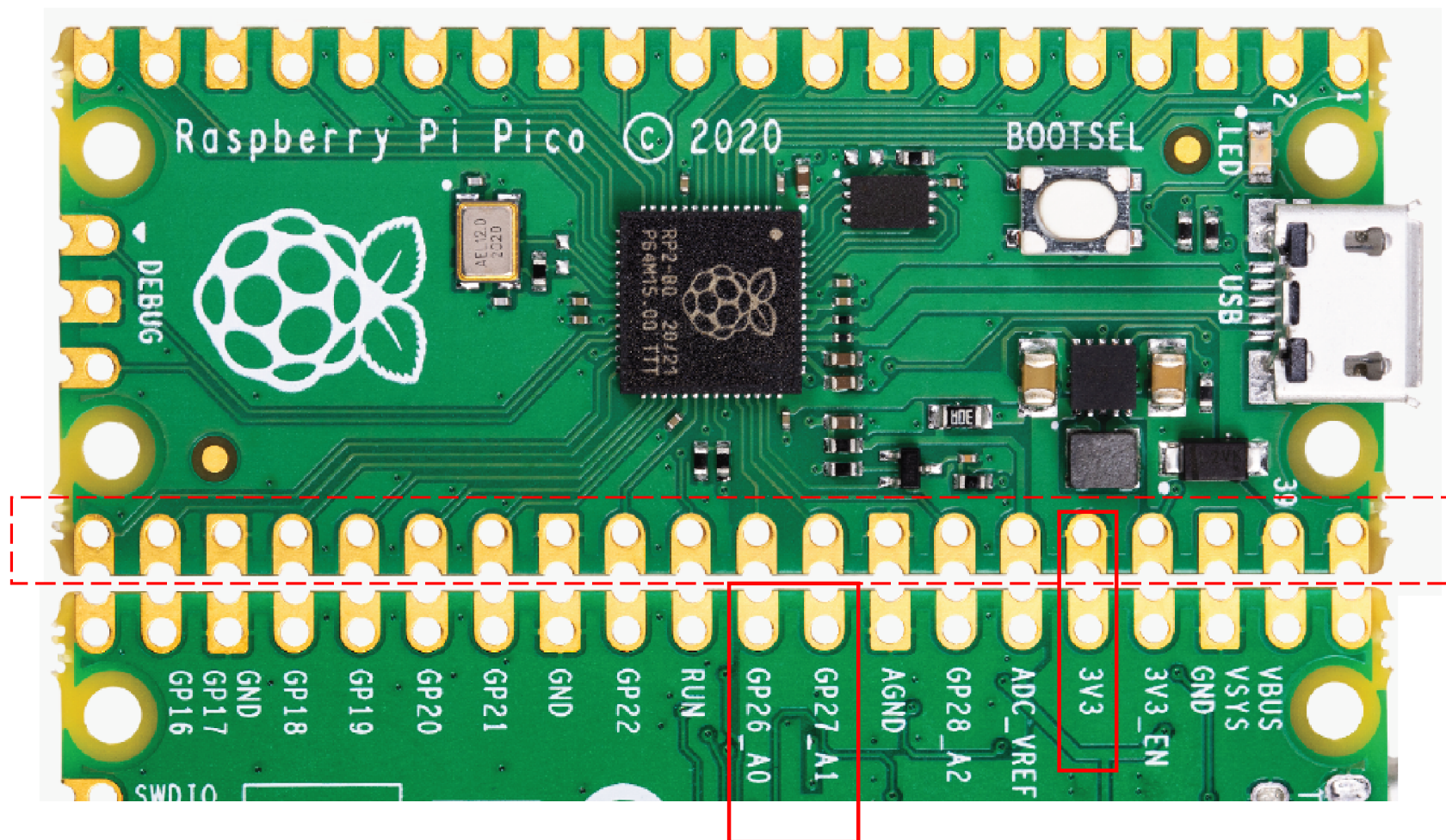
Mikropočítač - Raspberry PI PICO

Vývody RP PICO - popis na spodní straně

Tuto stranu desky využíváme



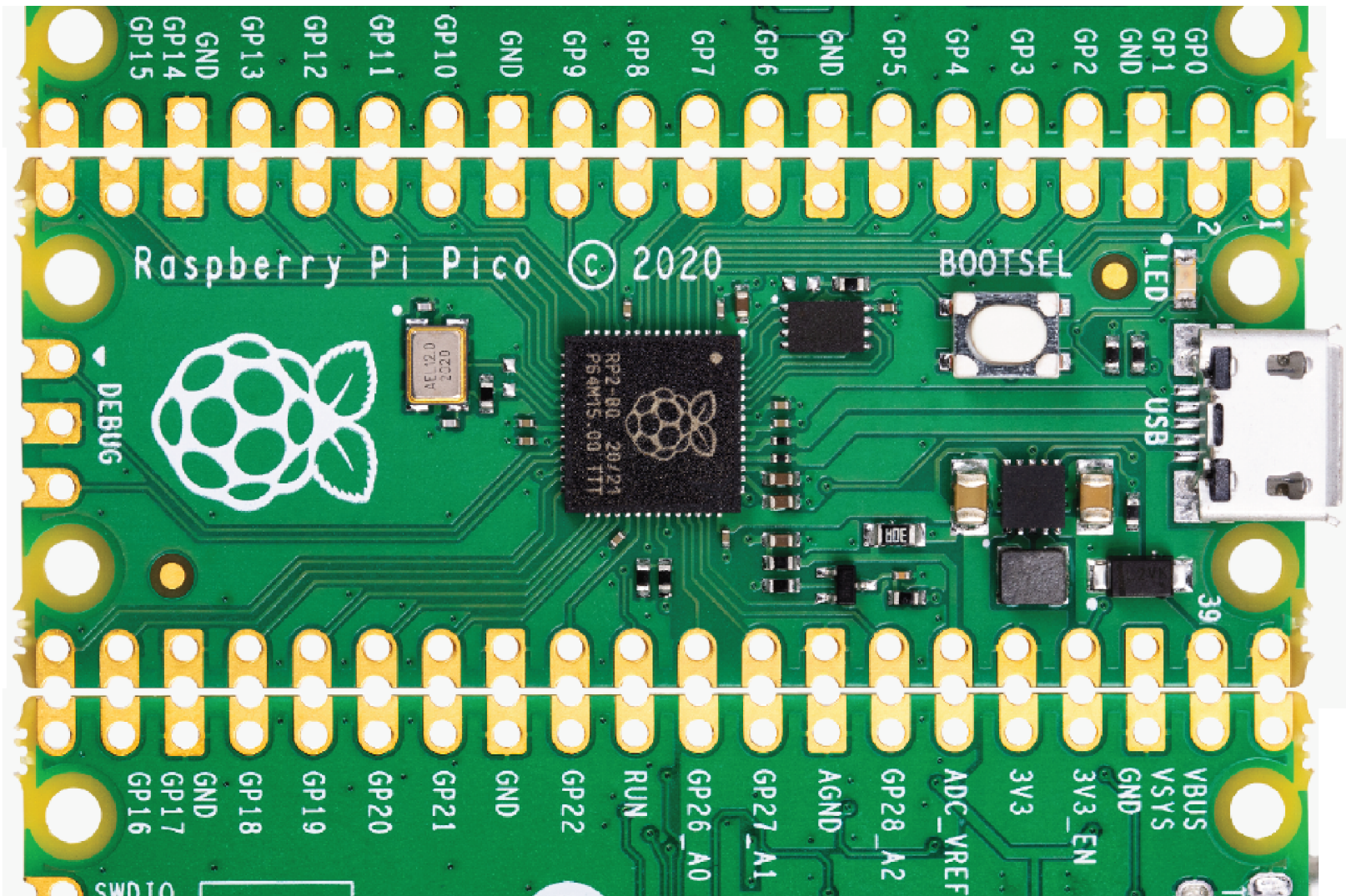
RP PI PICO - strana s vývody pro naše experimenty



Vstupy našeho osciloskopu

Napětí +3,3 V

PI PICO vývody- obě strany



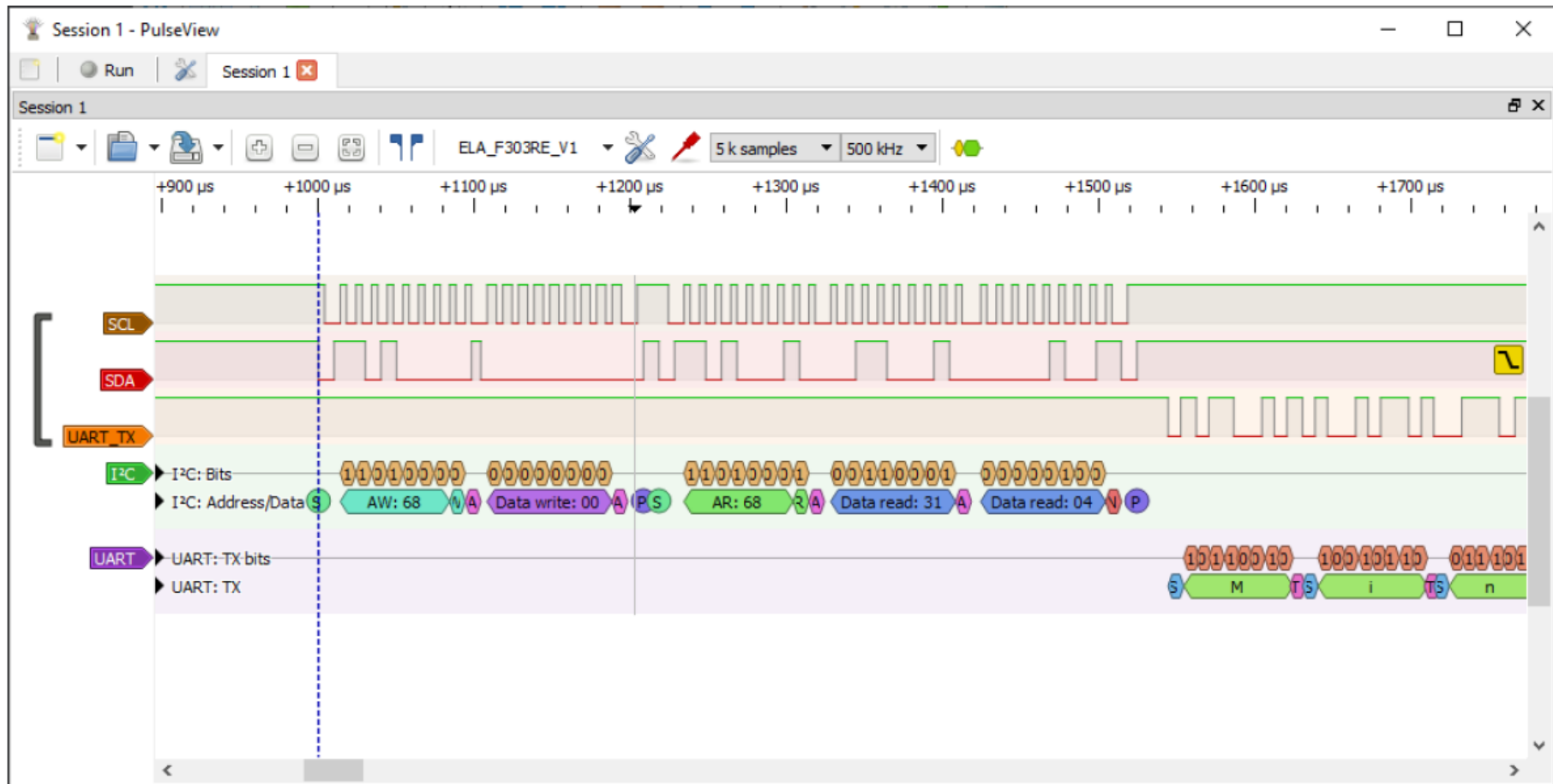
Přístroje s RP PICO pro experimenty

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

Přístroje – osciloskop, **logický analyzátor**, stačí jen nahrát náš SW (volné)

Logický analyzátor - sledování stavu výstupních signálů mikročítače
(BBC Micro:bit, Arduino, RP PICO,..) záznam log. sign. **až 40 MS/s !!**

Délka záz. **50 000 S !!**, analýza dat- na rozhraní SPI, IIC Bus, UART,..



Logický analyzátor ELA s RP PICO

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

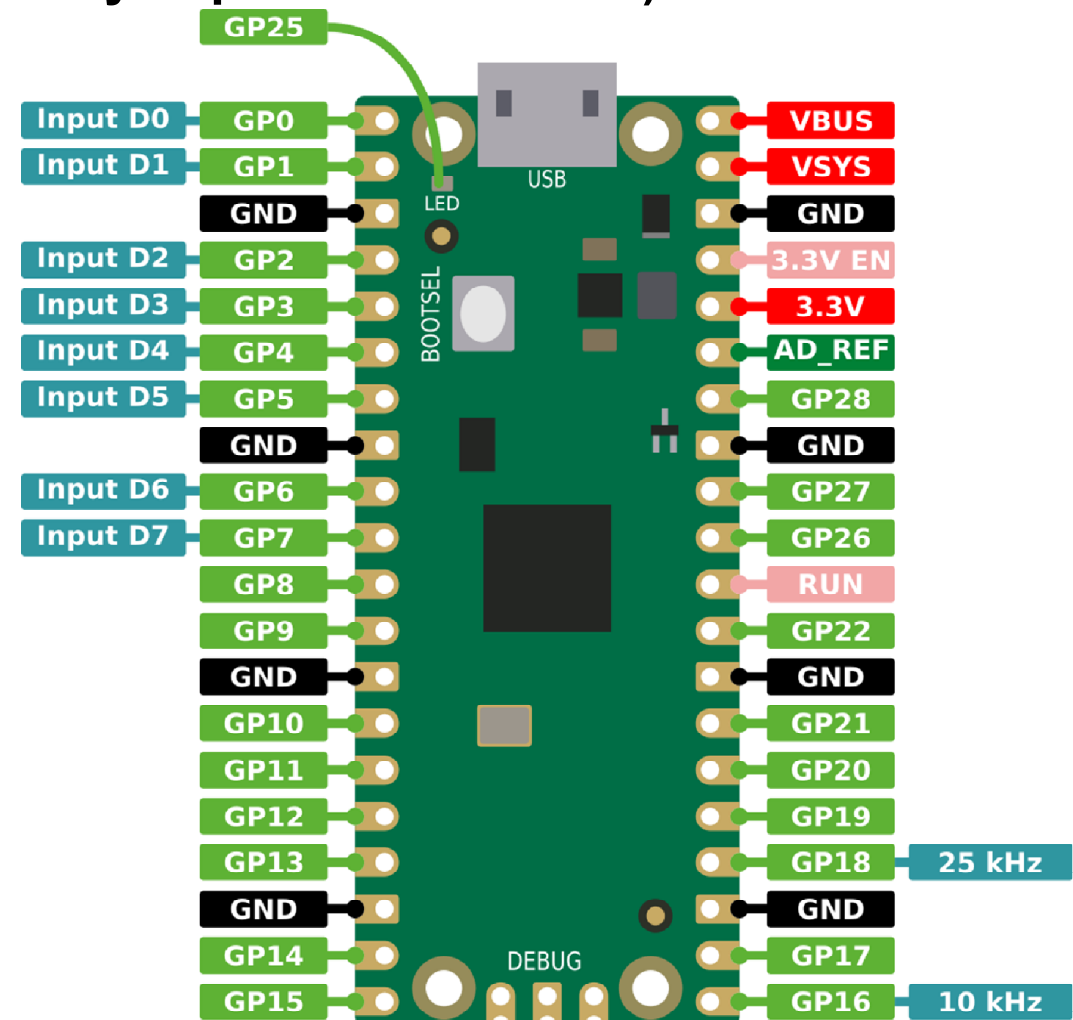
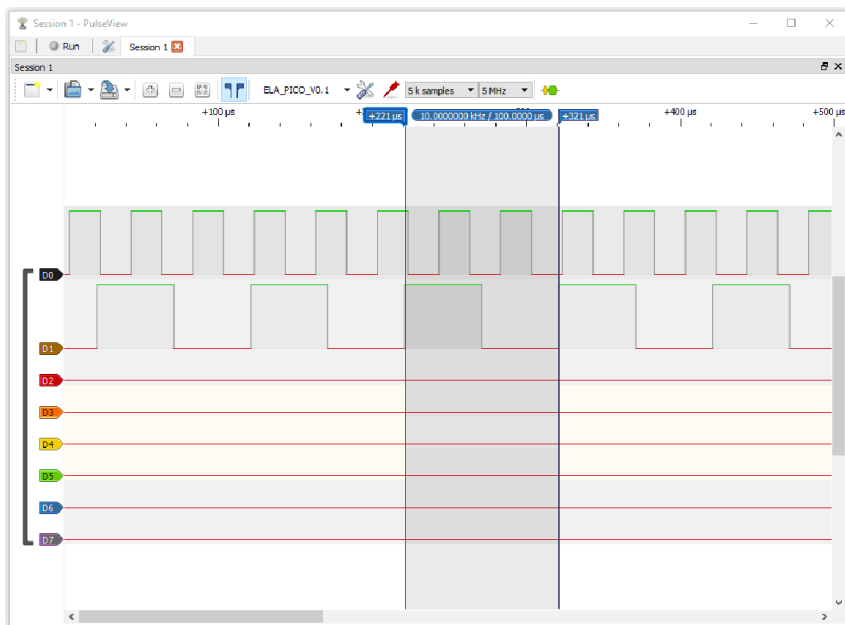
Logický analyzátor - přístroj pro sledování časového průběhu logických signálů (např. na programem ovládaných pinech ARDUINO)

Analýza komunikačních protokolů
s využitím Sigrok Pulseview

UART, SPI, I2C BUS,....

Autor firmware pro PI PICO

Ing. Vít Vaněčk



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

Dataplotter - univerzální zobrazovací PC aplikace, možno využít ve výuce informatiky, **přenos dat** z mikropročítače **do PC**, **zobrazení časového průběhu**

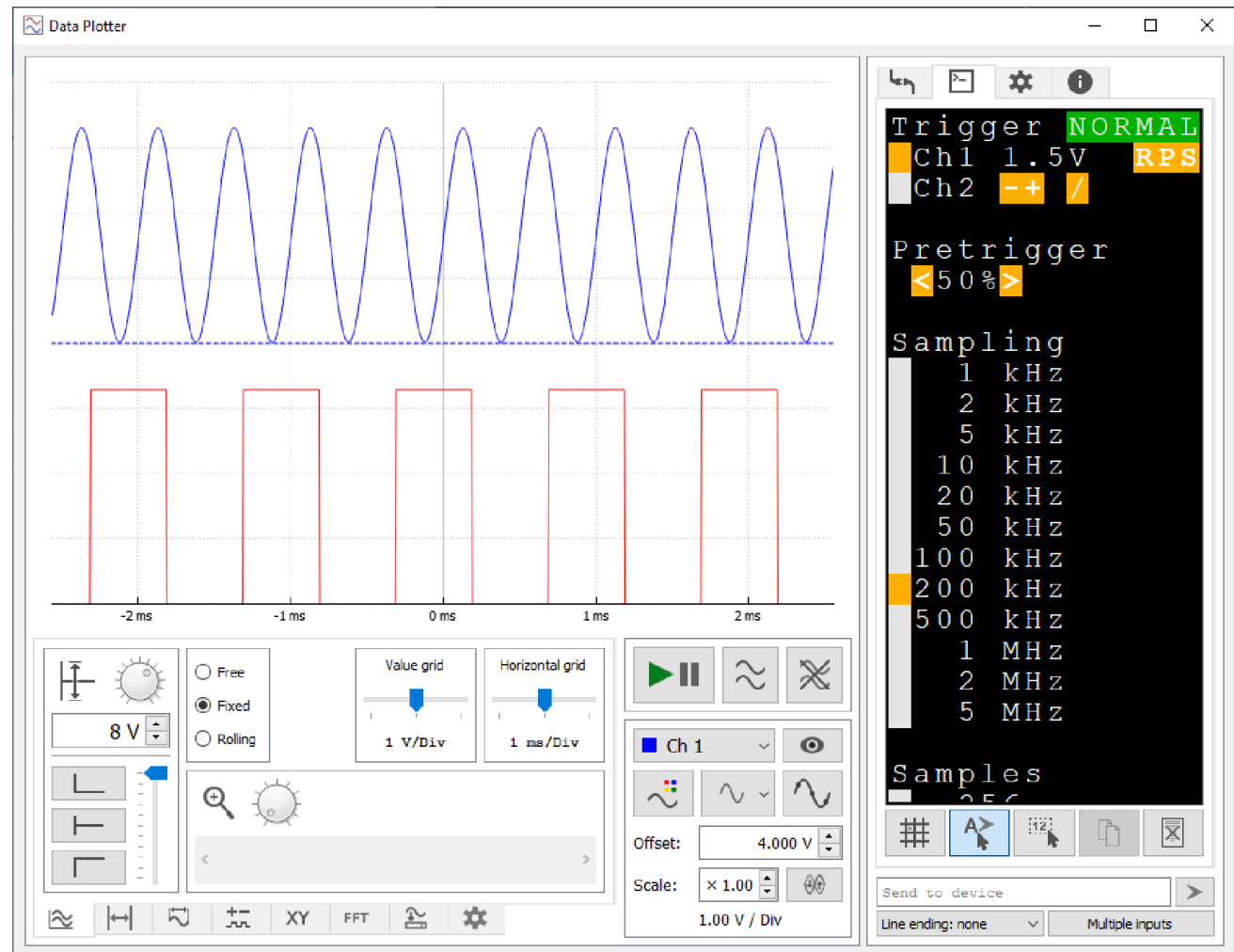
Volně k použití

Původ-

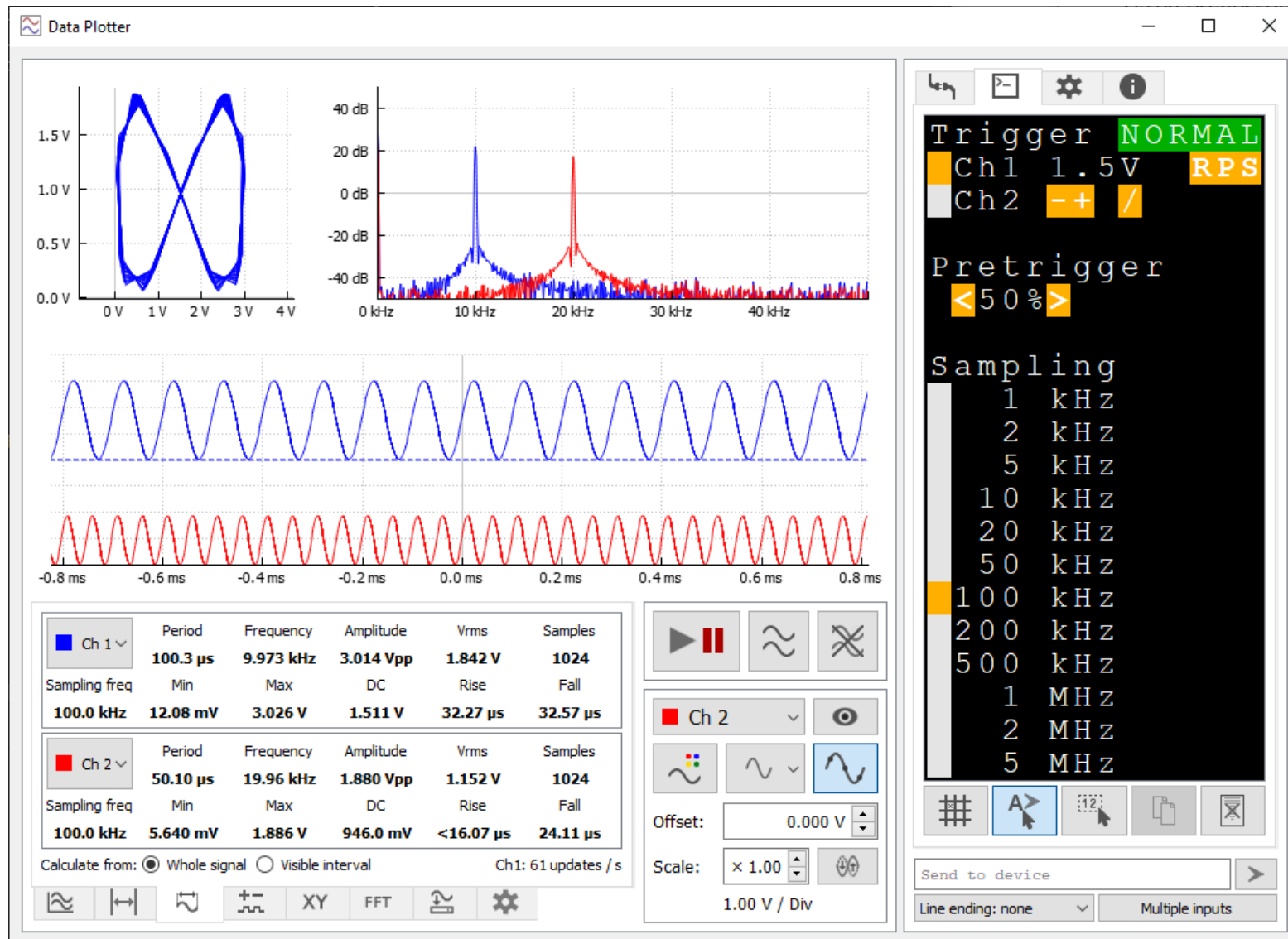
katedra měření

ČVUT – FEL

Autor. Bc. Jiří Maier



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



Osciloskop ELA Scope s **PI PICO** a **Dataplotter**

Funkce - dvoukanálový osciloskop

Až 1x 500 kS/s , délka záznamu 8192 S, 2x 250 kS/s, 2x 4096 S

Minimální vzorkovací rychlost 1 kS/s

Komfortní funkce zobrazení dat zajišťuje Dataplotter

Pozn.: Omezení pro vstupní napětí

0 až + 3,3 V !

Napětí **nesmí být záporné,**

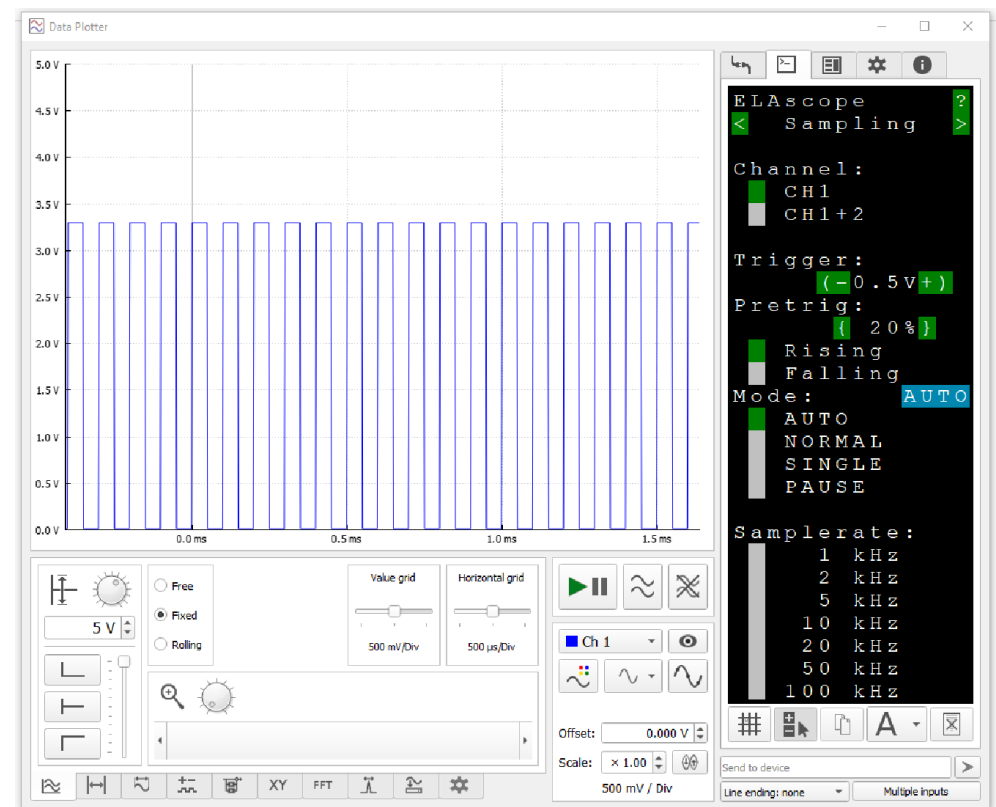
ani větší než +3,3 V

Řešení případných chybných
zapojení -

Ochrana – do série se vstupy
zapojit rezistory **2200 Ohmů** (2k2)

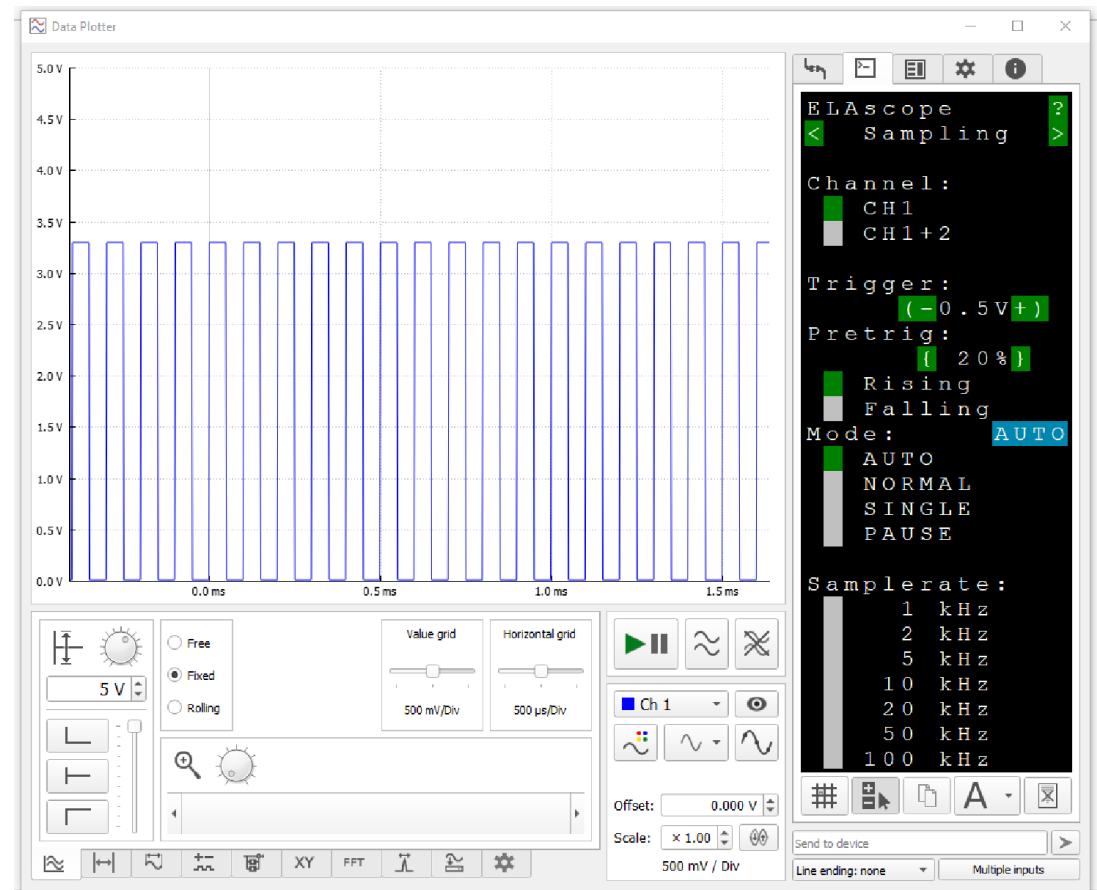
Autor firmware ELASOPE

Ing. Vít Vaněček



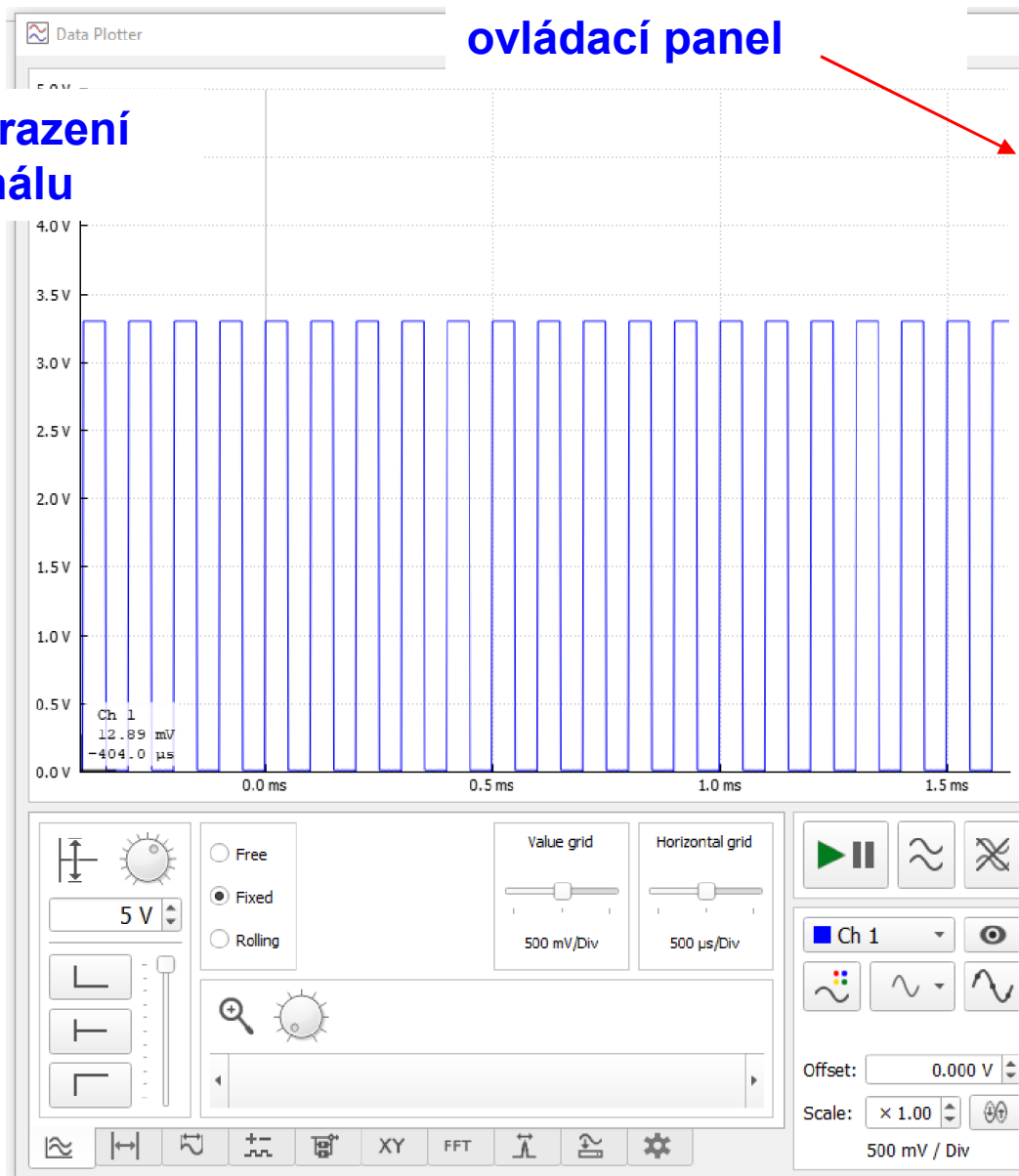
Oscilloskop ELA Scope s Raspberry PI PICO

Nahrát firmware do RP PICO, připojit jej k PC přes USB,
spustit aplikaci Data plotter,



Zobrazení signálu

Komunikační a ovládací panel



```
ELAscope
PWM Gen

Function:
OFF
PWM {050%}
PWM Sine
PWM Tria

Apply Max min
+
0010000.00
-

Freq (Hz):
10000.0000
PWM Div:
001+00/16
PWM Max:
0010000 Hz

Max freq:
2 kHz
10 kHz
50 kHz
100 kHz
500 kHz
```

Několik přepínaných panelů

pomocí šipek, klik myši

barevná pole- pro „klikání“ volba funkce nebo parametru

Volba velikosti fontu

„aby se to vešlo“

Přístroje s Data plotter – způsob ovládání

Proč tak nezvyklé ovládání přístroje s programem Dataplotter

Aplikace Data plotter je **univerzální** pro všechny možné přístroje

Funguje jako „**slave**“ – zobrazuje data, která se aplikaci posílají (v daných formátech). Jsou formáty pro zobrazení analogových dat, digitálních dat,...

? Jak řešit individuální ovládání přístroje

Komunikační panel je **terminálové okno** - textové pole – generuje sám mikropočítač (zde RP PICO). Pod barevnými čtverečky jsou **skryta písmena se stejnou barvou** (tedy nejsou vidět). Pokud se klikne na daný čtvereček, nebo **označený znak**, ten se **vyšle z PC** do RP PICO.

Program mikropočítače „ví“, co má po přijetí daného znaku dělat

Program **Dataplotter můžete využít ve své výuce**. Třeba i jen pro zobrazení naměřených dat bez využití ovládacího panelu

Naši studenti s Dataplotter řeší např. snímač tepu, improvizovaný elektrokardiograf, monitor svalové aktivity,...

Autorem programu Data plotter je náš **student Bc. Jiří Maier**

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

Panel impulsního generátoru PWM

nastavení střídavy PWM signálu

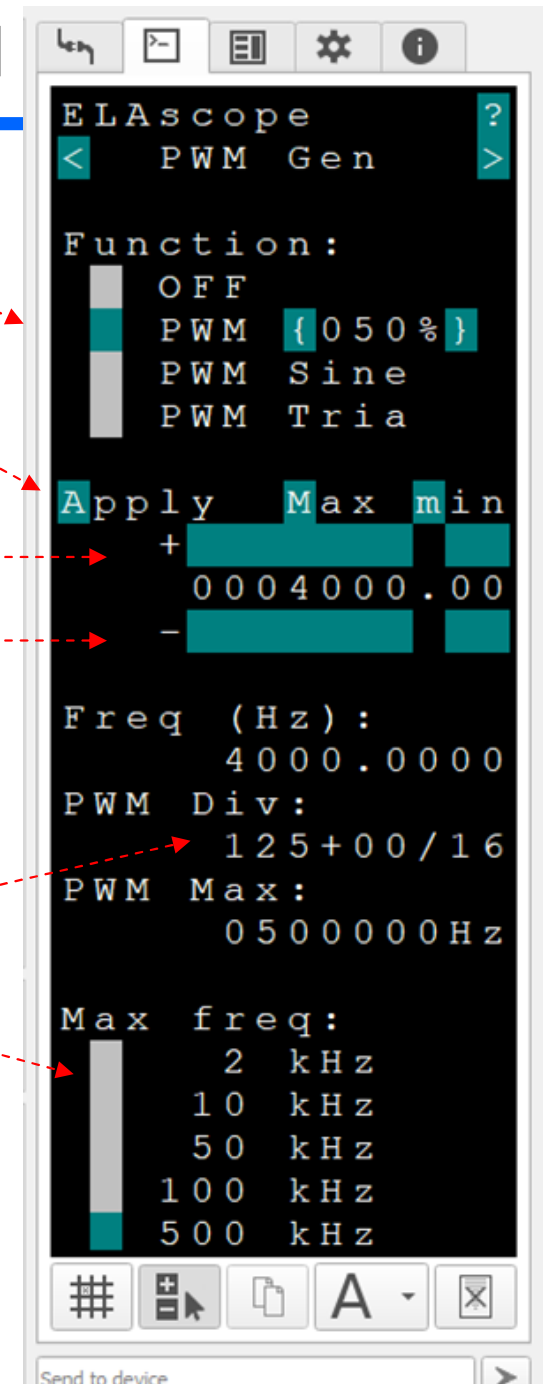
A - vyšli nastavení

+ klikem zvýš digit

- klikem sniž digit

Volba maximální frekvence PWM

dělením číslem 1 až 255 (PWM Div)
se získá požadovaná frekvence



Panel nastavení osciloskopu

- přepínání dalších panelů
- volba 1 nebo 2 kanály
- volba trigger level
 - + zvyš hrubě o 0,5 V;) zvyš jemně o 0,1 V
 - sniž hrubě o 0,5 V; (sniž jemně od 0,1 V

volba druhu spouštěcí hran

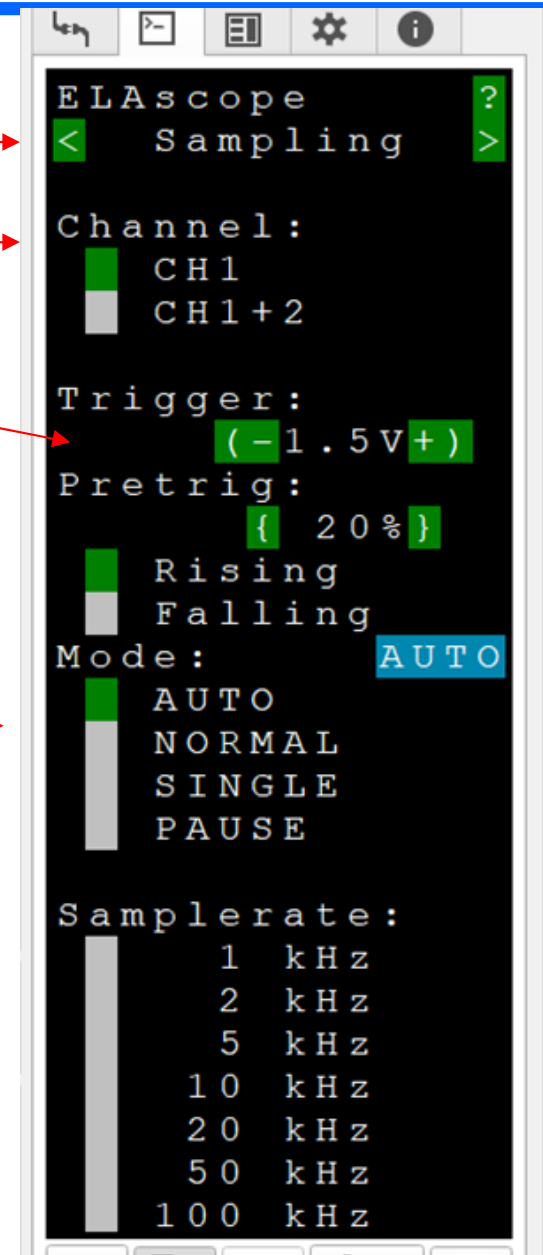
Režim Auto, Normal, Single, Pause

Auto - pokud se nenajde podmínka, spustí se sám

Normal - čeká , dokud nepřijde podmínka

Single - jednorázový odměr a stop

Pause – stop



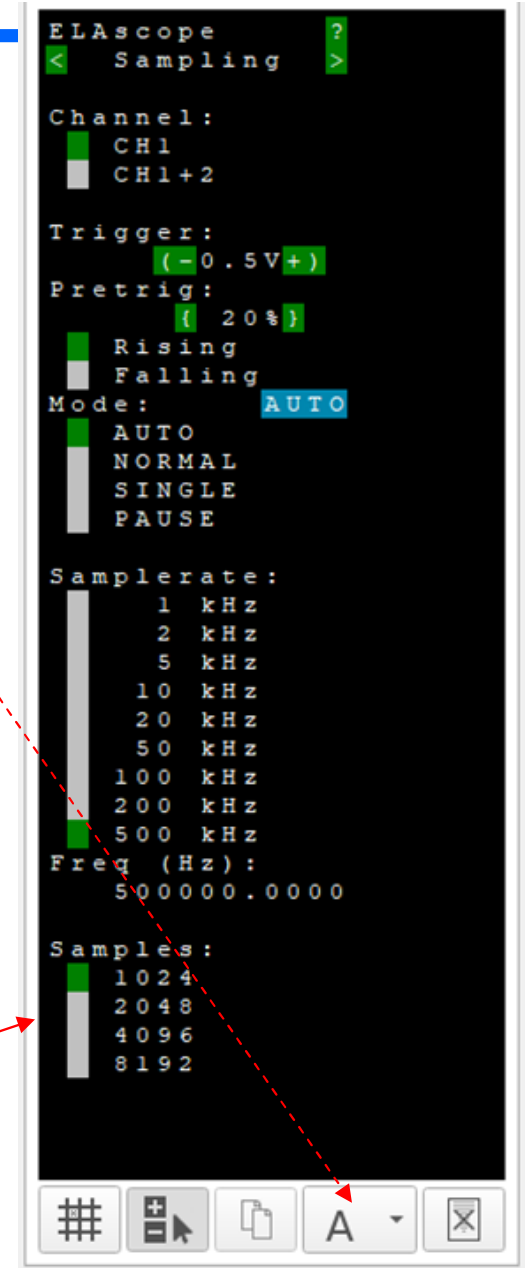
▪

Ovládací panel s volbou menšího fontu

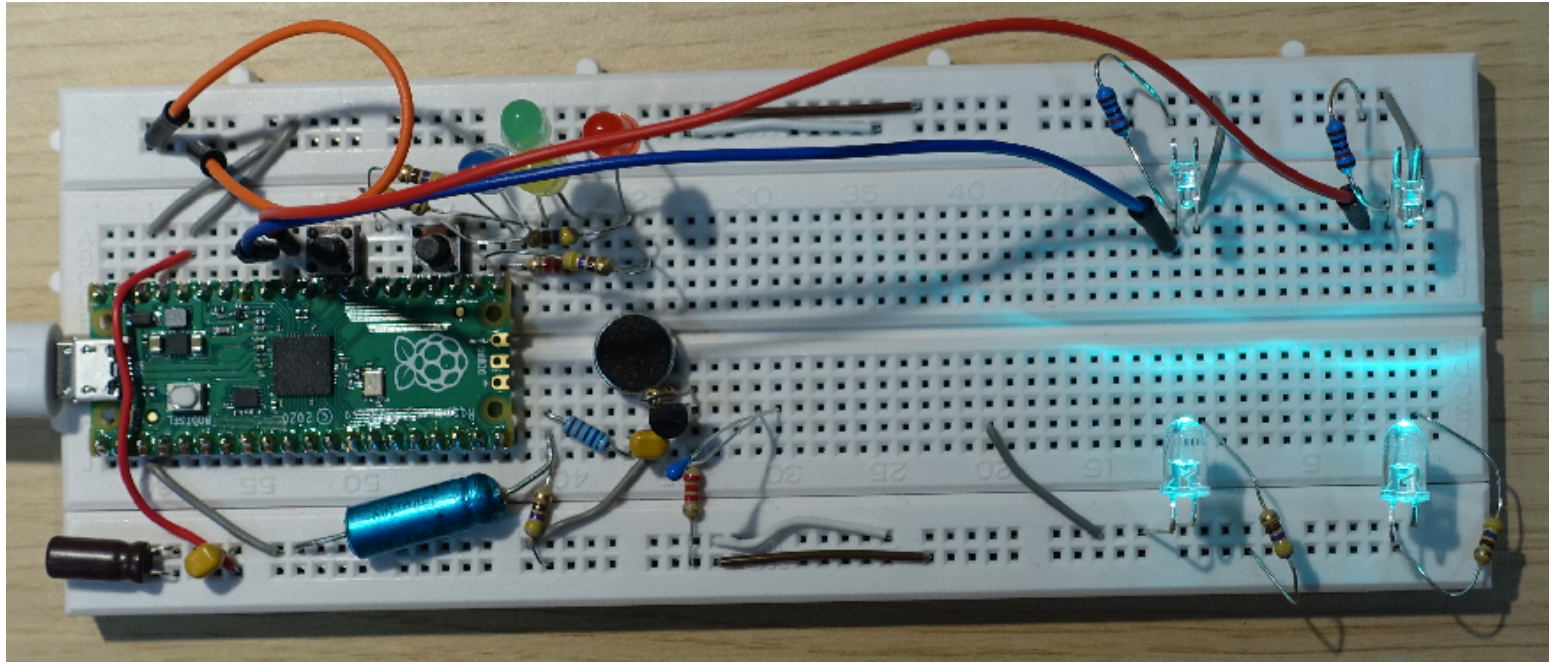
„aby se to všechno vešlo“

Pokud se text „nevejde“ najednou do panelu, lze jej myší vertikálně posouvat

volba délky záznamu



Naše experimentální sestava s RP PICO



Budeme používat a zapojovat LED, rezistory, fototranzistory, mikrofon, proto krátce teorie po případné další studium

Světloemitující diody - LED

Indexová značka – ploška z boku na spodní straně pouzdra označuje katodu U nové LED katoda má **kratší vývod** (kratší vodič)



Zapojení LED do obvodu

LED při workshopu používáme v optické závoře a jako indikaci na výstupech procesoru

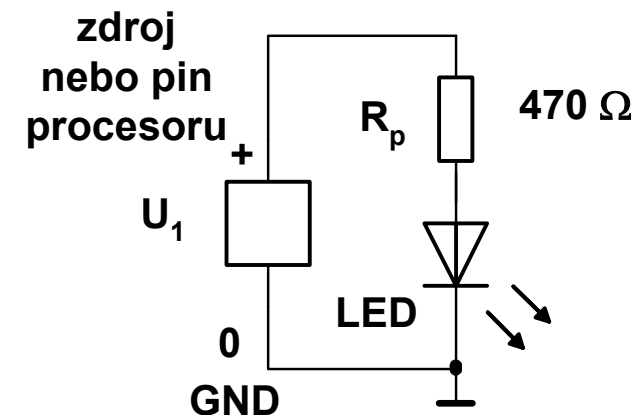
Lze prohodit i umístění –pořadí- LED a rezistoru

Katoda LED – směrem ke GND – zem

LED **nikdy** nepřipojovat na pin procesoru nebo napájení **bez rezistoru**

Otočení LED – prohození vývodů katoda, anoda, při našem malém napájecím napětí +3,3 V není problém, jen nebude svítit

Pozn. Pozor **přímé připojení** LED, např. na napětí + 5 V nebo +3,3 V nejen poškodí LED, ale může vést rozštípnutí pouzdra a vymrštění částičky plastu do okolí a poškození zdroje



Barevný kód značení odporu rezistorů

Barevné značení **velikosti** odporu standardních rezistorů s drátovými vývody

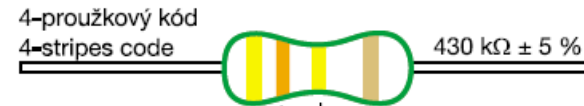
430 kOhmů = 43×10^4

označení 434

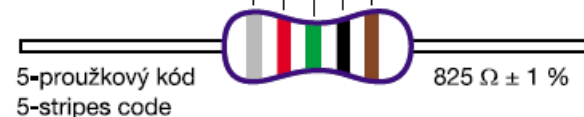
Tolerance výroby,
značena na konci
samostatným proužkem

Barevný kód

Color code



Barva Color	1. proužek 1. stripe	2. proužek 2. stripe	3. proužek 3. stripe	násobitel ratio	tolerance tolerance
černá - black	0	0	0	1	
hnědá - brown	1	1	1	10	± 1,00 % (F)
červená - red	2	2	2	10 ²	± 2,00 % (G)
oranžová - orange	3	3	3	10 ³	
žlutá - yellow	4	4	4	10 ⁴	
zelená - green	5	5	5	10 ⁵	± 0,5 % (D)
modrá - blue	6	6	6	10 ⁶	± 0,25 % (C)
fialová - violett	7	7	7	10 ⁷	± 0,10% (B)
šedá - grey	8	8	8	10 ⁸	± 0,05 % (A)
bílá - white	9	9	9	10 ⁹	
zlatá - gold	-	-	-	10 ⁻¹	± 5,00 % (J)
stříbrná - silver	-	-	-	10 ⁻²	± 10,00 % (K)



Pokud jsou smíchané rezistory o různých hodnotách odporu – nespolehat na čtení, ale raději zkontrolovat Ohmetrem,

Dle: <http://www.soucastky.chytrak.cz/Odpory/R%20-%20Uhlíkove.html>

Rezistory použité v úlohách

Použité rezistory „**modré**“, (resp. modrý podklad) jsou **metalizované**.

Odporová vrstva je na bázi kovu, tolerance je 1 %,

Hodnota (řada E48) je označena pomocí **5 proužků**, např. 10 k

Hnědá, černá, černá, rudá, hnědá 1, 0, 0, 2, 1 (procento)

$100 \times 10^2 = 10\,000 = 10\text{ k}$ → k značí kilo – 1000 Ohmů, Kiloohm

Rezistory „žluté“ – uhlíkové (přesněji uhlové), tolerance do 5%,

Značení hodnoty -pouze **4 proužky**

2k2, tolerance 5 %, používáme v experimentu, ochrana vstupu ADC

2, 2, 2, 5 (procent)

rudá, rudá, rudá, zlatá - **tři červené proužky** a jeden zlatý

470 Ohmů 4 7 1 žlutá, fialová, černá, zlatá

Použité rezistory: **metalizované 5k6, 10 k**,

uhlíkové 470, 2k2, 1M

Proč máme 10 k metalický (modrý) rezistor - pro snadnější rozlišení od ostatních hodnot podle barvy

Experimenty

Vyhodnocení blikání zářivek (LED žárovek fototranzistorem a osciloskopem)

Měření rychlosti pohybu pomocí dvou optických závor

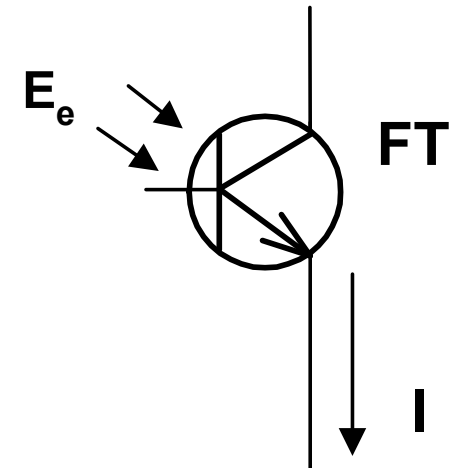
Určení rychlosti zvuku

Vyhodnocení intenzity osvětlení jasů objektů

Fototranzistor

Zde používáme **fototranzistor LL-304PTC4B-1AD**
výrobce FORYARD a další mají pouze dva vývody
- *Kolektor a Emitter* (báze není vyvedena)

Můžeme chápat jako **prvek, který propouští proud o velikosti odpovídající intenzitě dopadajícího záření** (jako „kohoutek ovládaný světlem“)



Čiré (průhledné) pouzdro, náš fototranzistor má **průměr 3 mm**
(pozor - vypadá jako LED, ale LED zde máme o průměru **5 mm**)

Kolektor označen – ploška na pouzdře *u vývodu kolektoru*,
u nové součástky je kratší vývod kolektoru

Poznámky do laboratoří se studenty:

Prohození vývodů C a E fototranzistoru, nic se neděje -nepokazí se, ale nefunguje ,

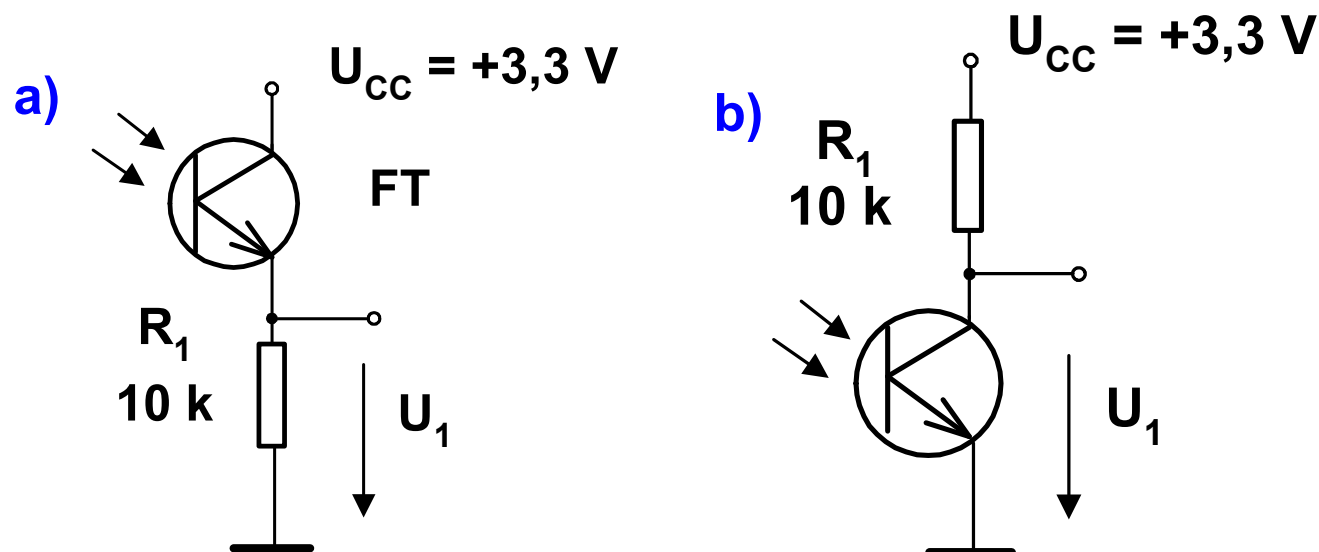
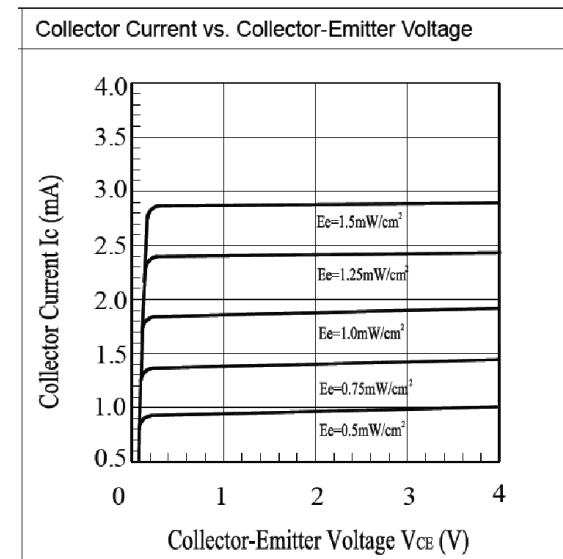
(fotocitlivost velmi malá bude několik set krát menší než normálně)

Zapojení fototranzistoru do obvodu

Dvě možná zapojení

zap. a) napětí U_1 se světlem **roste**
roste proud a tedy i úbytek napětí na R_1

zap. b) napětí U_1 se světlem **klesá**



Zapojení fototranzistoru pro „hraní“ studentů

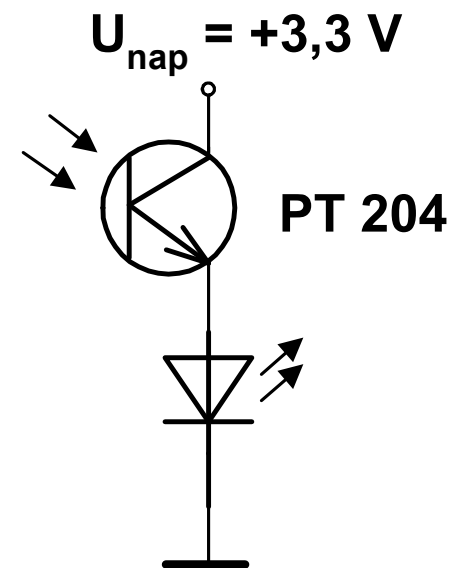
Demonstrace funkce fototranzistoru pro studenty

Použít LED, který dobře svítí i při malém proudu (naše zelené)

Není použít **rezistor**, fototranzistor sám **omezuje velikost proudu**

Případné použití rezistoru 470 Ohmů (pro ochranu proti případnému přímému připojení LED na napájení při experimentech studentů) nevadí - pouze se sníží velikost proudu LED.

Při **osvícení** fototranzistoru se mírně **rozsvítí i LED**



První experiment s osciloskopem s PICO

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

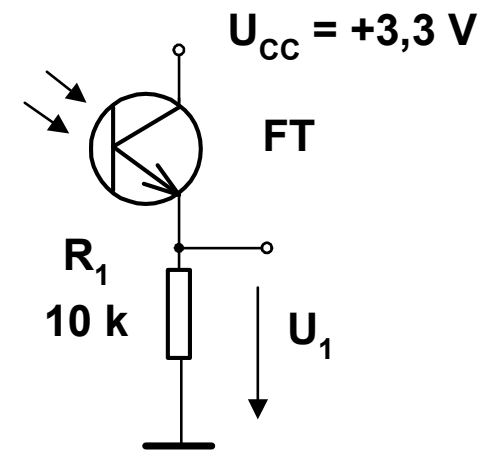
je již náhrán v deskách, na PC spustíme aplikaci **DATA Plotter**

První experiment - připojit výstup PWM (GP16) na vstup osciloskopu(GP26) **Ch₁** a pozorovat jeho signál
(to je propojit GP16 na GP26 vodičem)

Druhý experiment - 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).



Snímač pro měření rychlosti

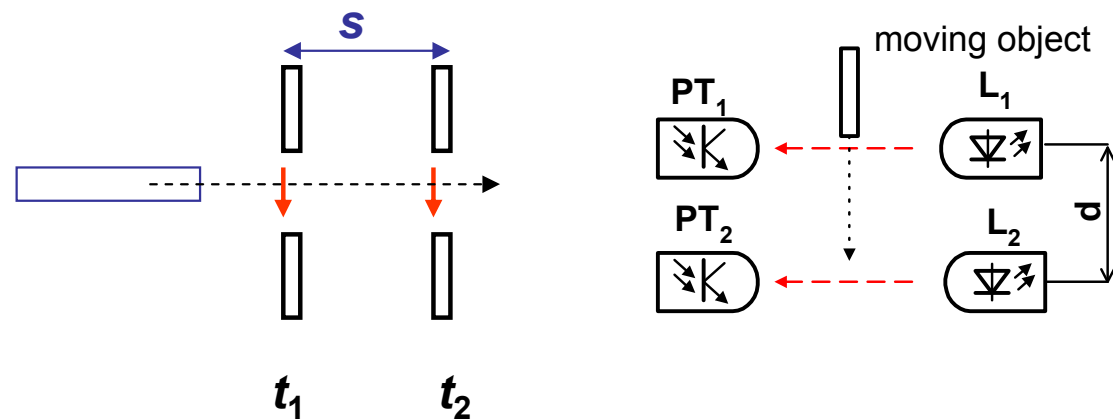
Rychlost v , dráha s , čas t

$v = s / t$ Realizace optické závory $t = t_2 - t_1$

Dvě optické závory posunuté o vzdálenost s

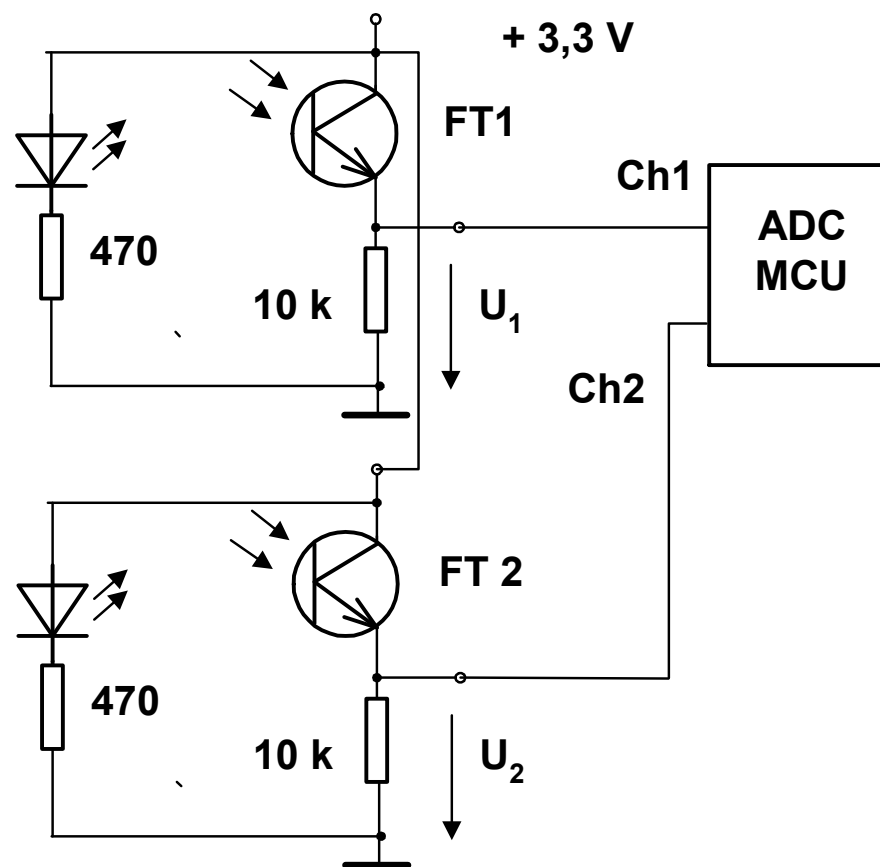
Přerušování chodu paprsků dvou optických závor hranou pohybujícího objektu

Osciloskop v režimu **dvou kanálů**, vyhodnotit **zpoždění** signálu závor pozorováním signálu v obou kanálech.



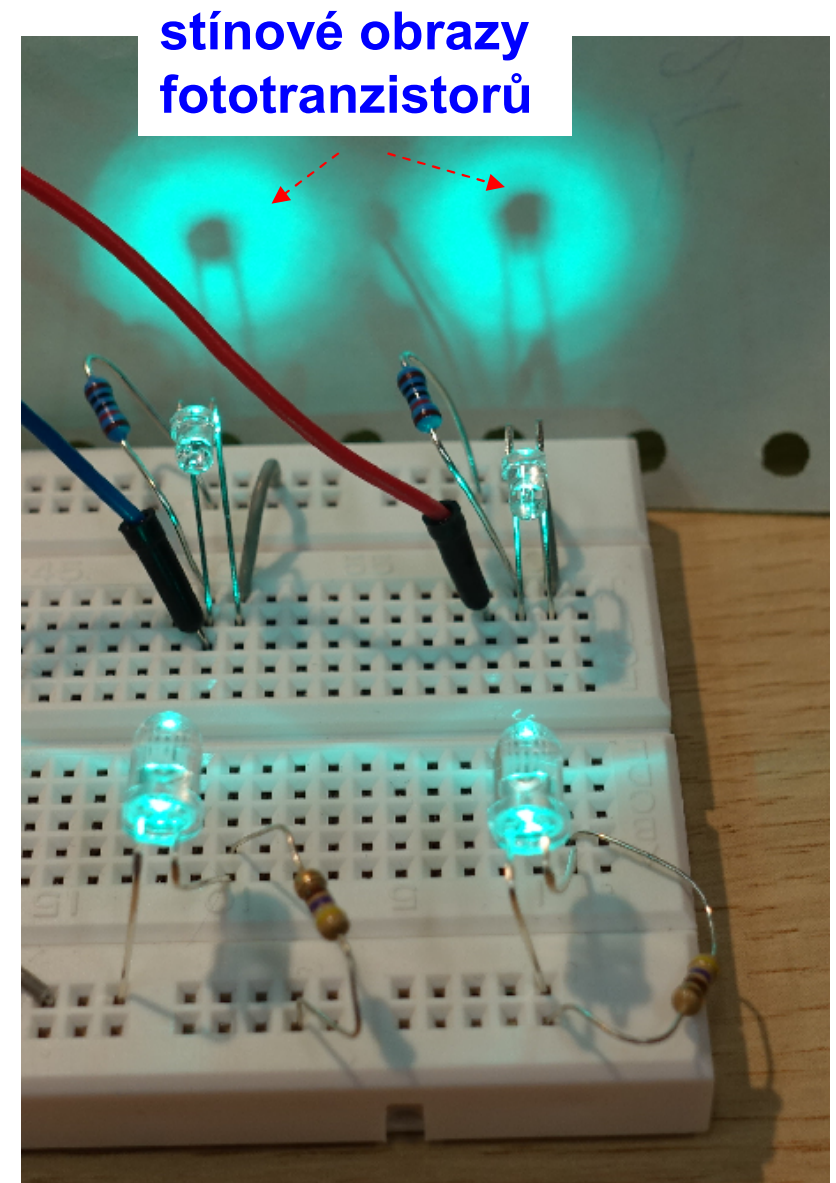
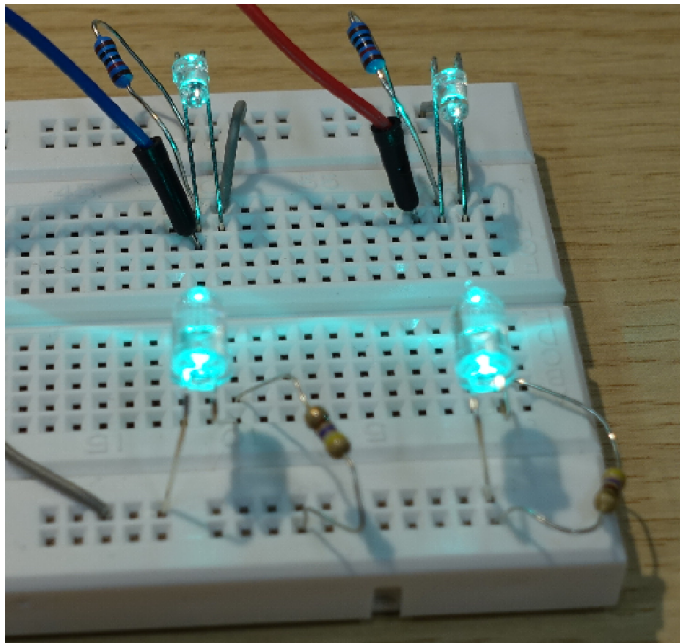
Zapojení optických závor pro měření rychlosti pohybu

• Výstupy připojeny na vstupy osciloskopu GP26, GP27



Uspořádání optických závor

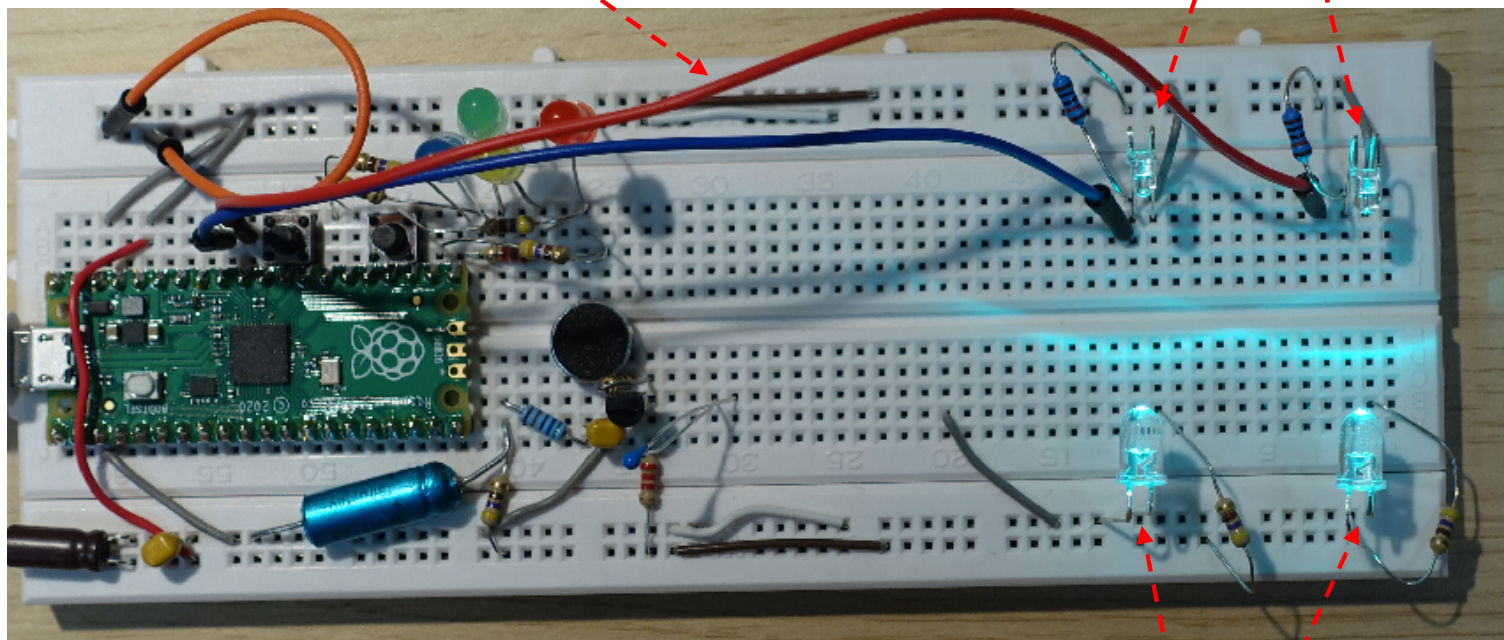
LED nasměrovat na fototranzistory tak, aby byly ve středu světelného kužele



Pole a dvojitá optická závora

•
připojení výst. sig. z emitorů
fototranzistorů na osciloskop

2x fototranzistor



2x LED

Experiment – určení rychlosti zvuku ve vzduchu

Vlnová délka akustického vlnění λ_z , kmitočet f_z

$$\lambda = \frac{v_z}{f_z}$$

Určení rychlosti zvuku v_z z úseku – dráhy

odpovídající vlnové délce λ_z a kmitočtu f_z

Potřebujeme určit vlnovou délku λ_z , vlnění

$$v_z = \lambda \cdot f_z$$

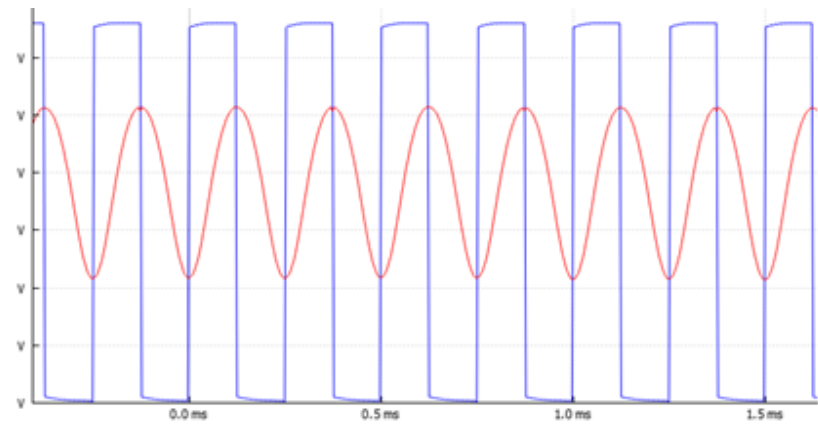
Metoda- generátor zvuku – buzzer budeme

vzdalovat od mikrofonu vyhodnotíme

zpoždění akustického signálu, které odpovídá **jedné periodě**.

Měření převedeme na měření fázového posuvu mezi generovaným a přijatým signálem

Wikipedia: Při teplotě 20 °C je rychlost zvuku v suchém vzduchu 343 m/s



Obvody pro experiment

Příjem akustického signálu- elektretový mikrofon s vnitřní zesilovačem

Vnější zesilovač s tranzistorem T_1 NPN typu BC546 (BC547)

$R_M = 5k6$, $R_B = 1M$, $R_C = 2k2$

Blok filtrace napájecího napětí s

$R_F = 100 M$ ($22 M, ..47 M$) a $R_F = 470$

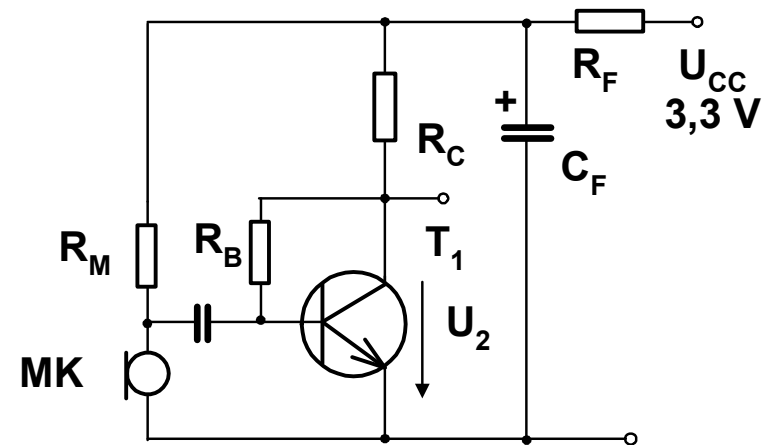
Jako akustický měnič je použit **piezo buzzer (bzučák)**

vykazuje velkou kapacitu,

řádově 30 nF

nutný ochranný rezistor 2k2 do série z výstupem buzení z impulzního generátoru PWM

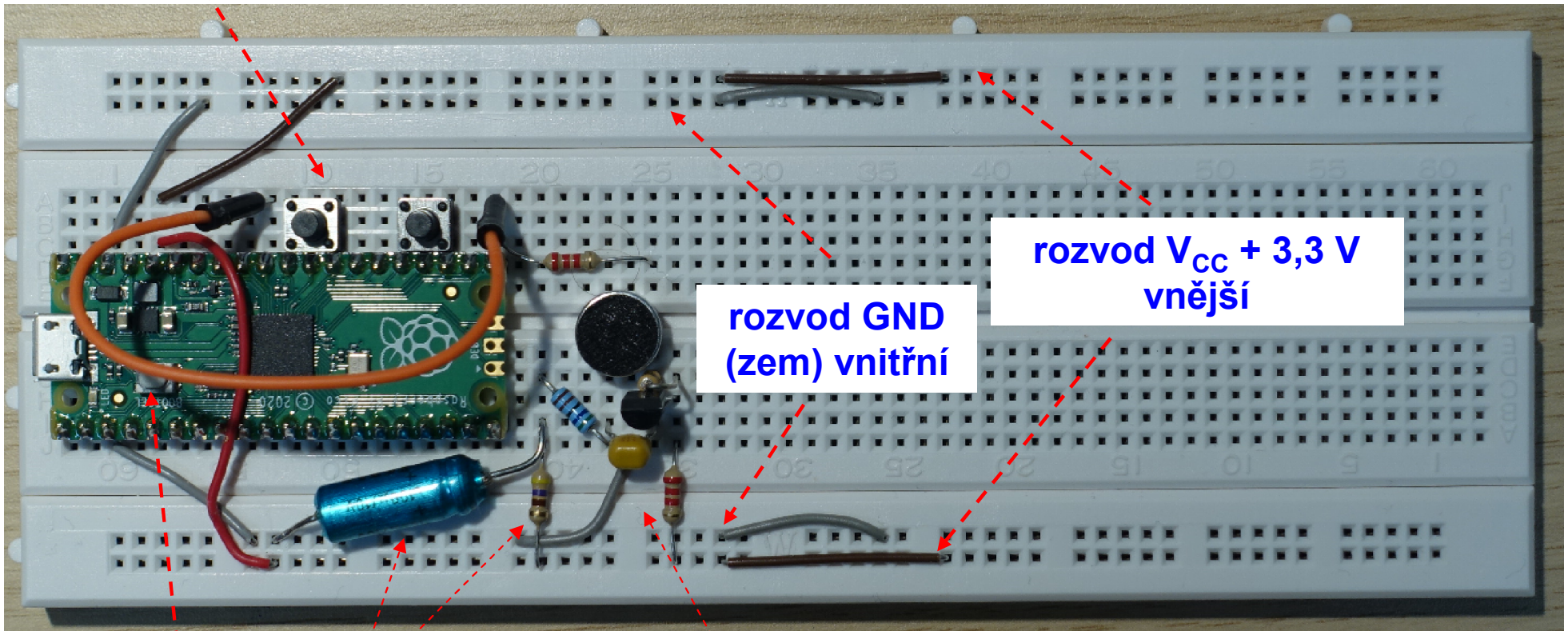
Pozn. : Z přechodového děje (exponenciála) a záznamu napětí osciloskopem přímo na buzzeru lze určit jeho kapacitu (bonus úloha)



Pole s RP PICO, mikrofonem a zesilovačem

▪

RESET tlačítko



rozvod $V_{CC} + 3,3 V$
vnější

rozvod GND
(zem) vnitřní

BOOT
tlačítko

RC filtrace napájení
mikrofonu

mikrofon + ext. zesilovač

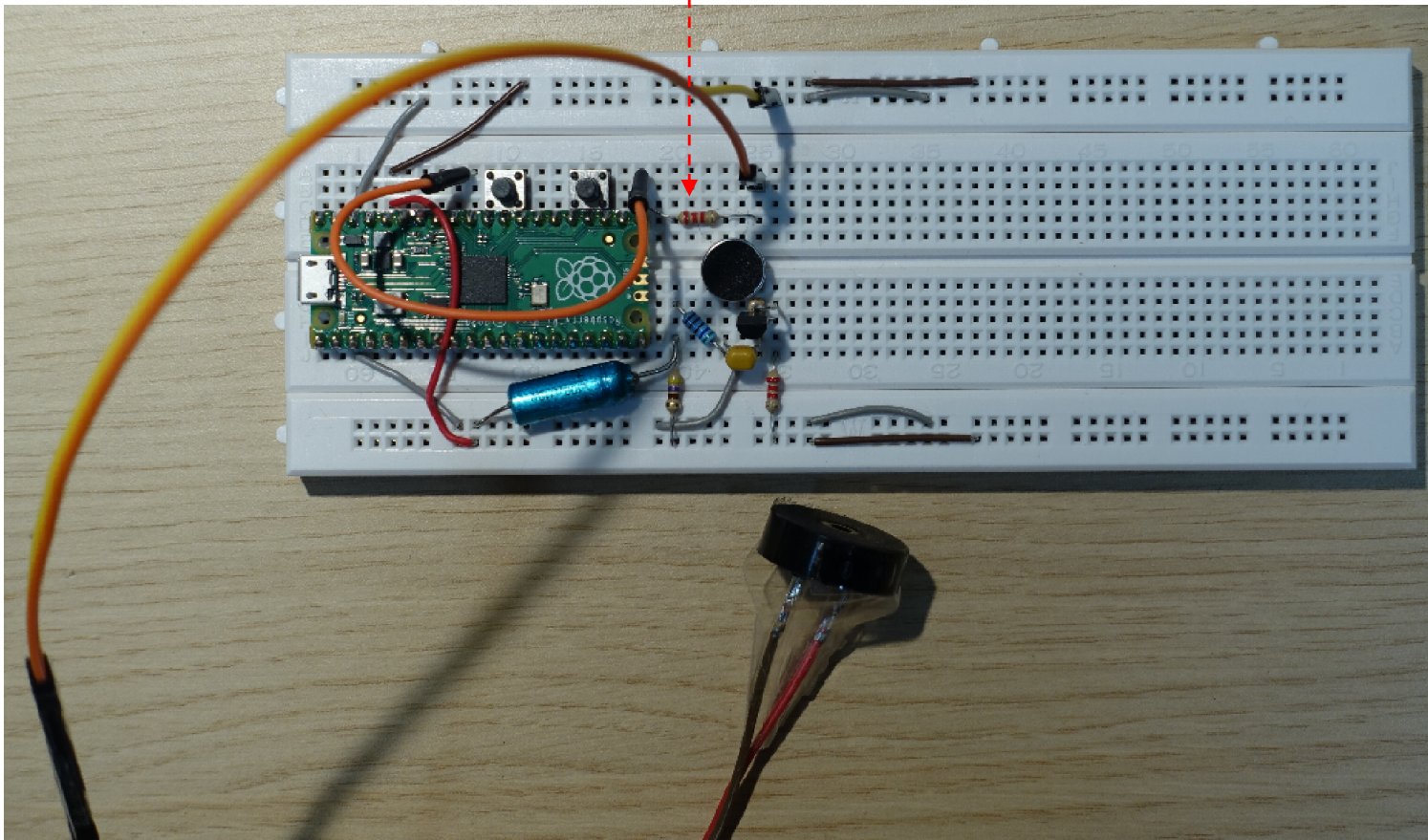
Připojení bzučáku

Ochranný rezistor 2k2 do série s bzučákem (BZ), druhý vývod BZ na GND

Rezonanční frekvence bzučáku cca 4 kHz (budeme používat 4 až 4,5 kHz)

- pokud to naše uši vydrží, jedná se o nepříjemný zvuk

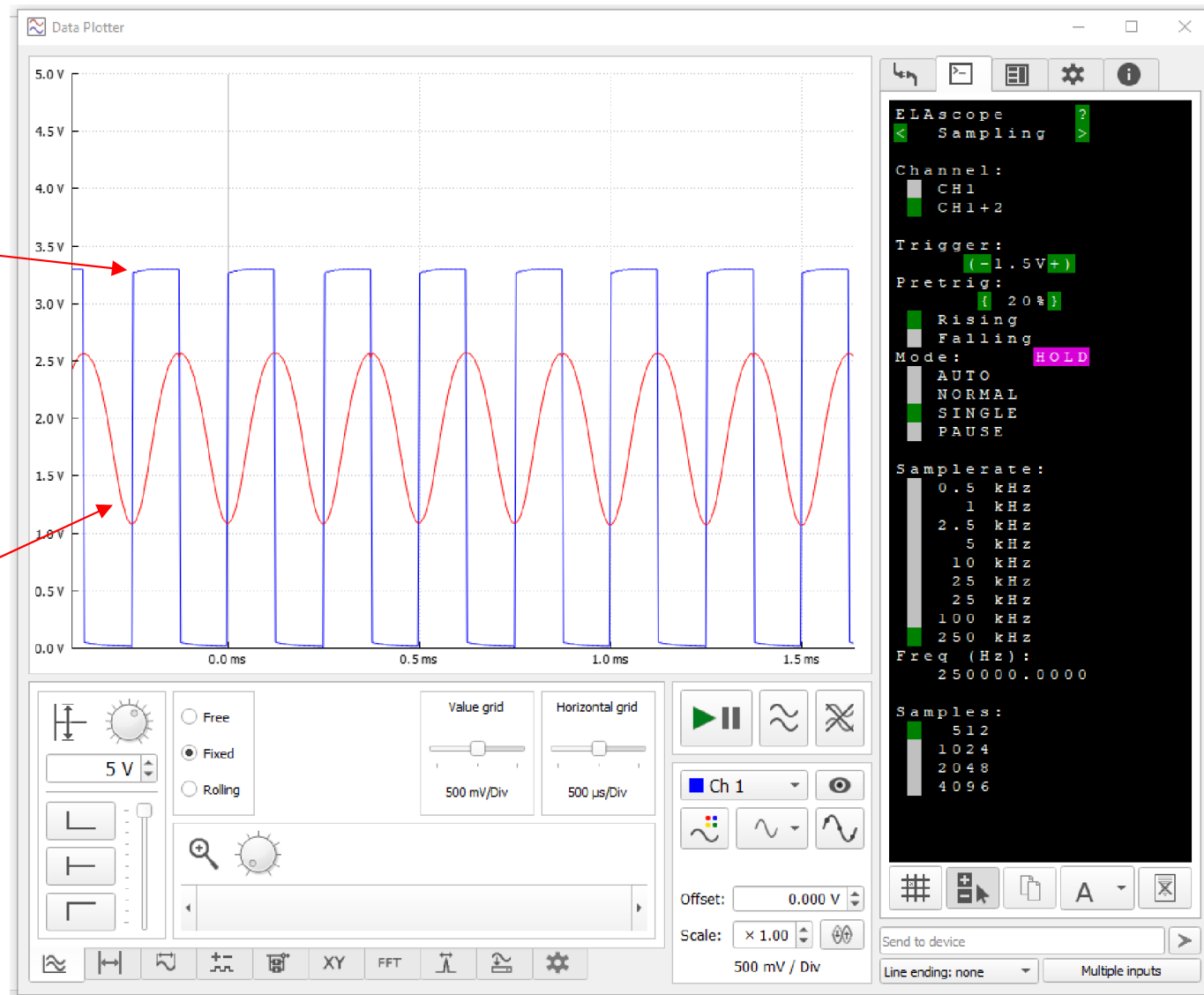
ochranný rezistor 2k2



Signály při experimentu – měření rychlosti zvuku

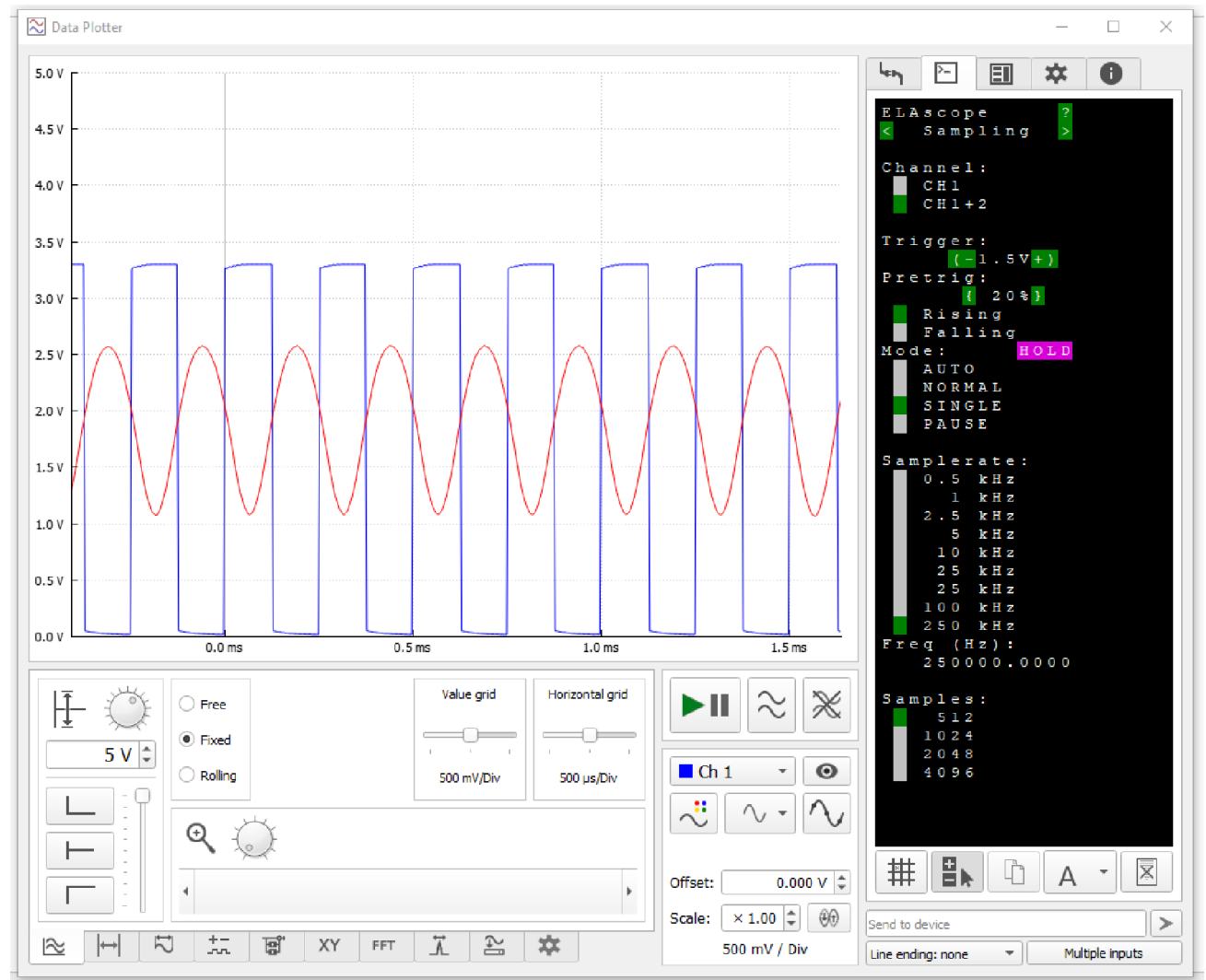
signál
PWM

zesílený
signál
mikrofonu
(sinus)



Měření rychlosti zvuku – posun bzučáku

Změna fáze signálu
(fázový posuv)
z mikrofonu
se zvětšením
vzdálenosti
bzučáku od
mikrofonu



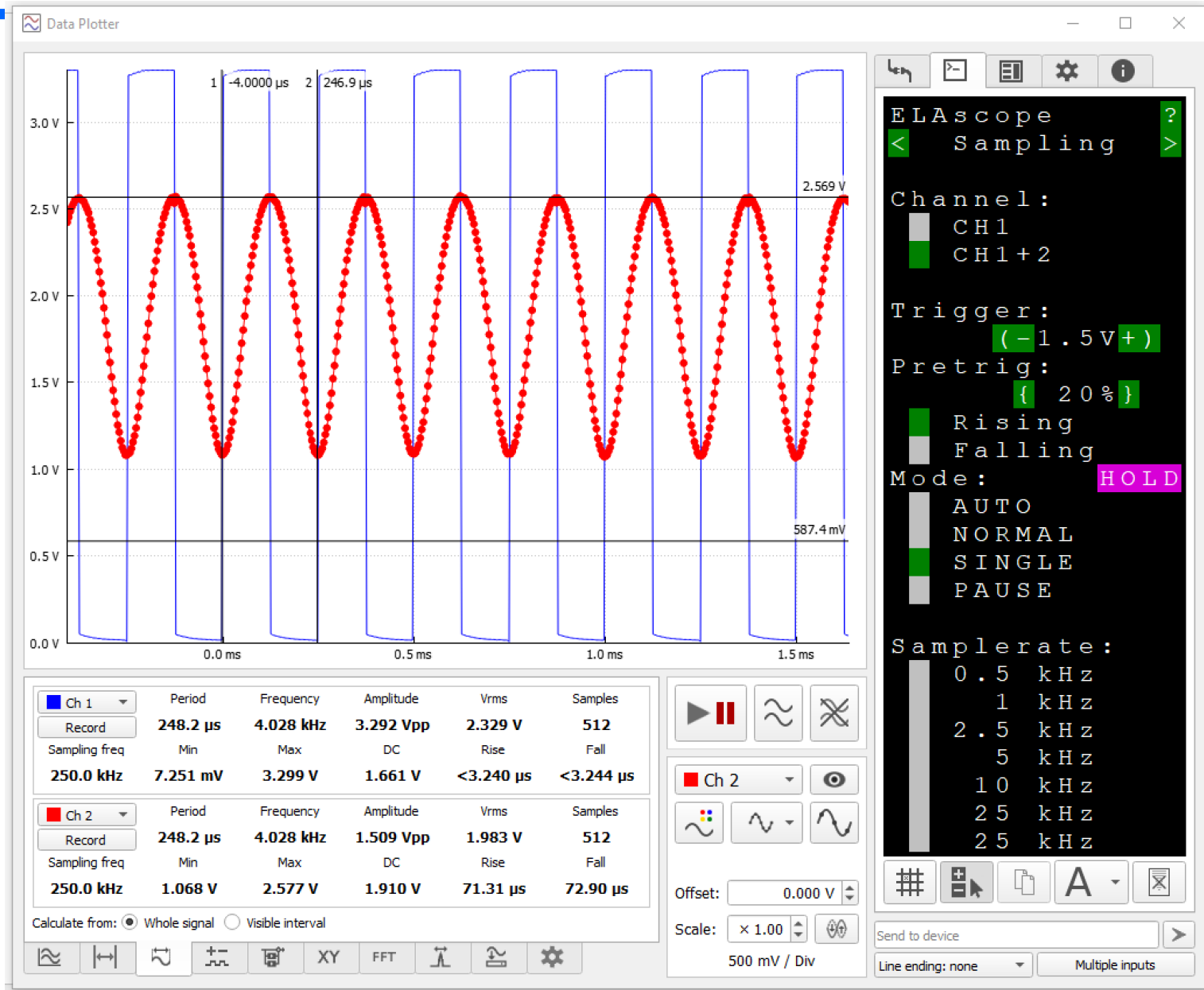
Určit velikost mechanického posuvu pro změnu fáze o 360 stupňů

Nastavení barvy zobrazení signálu

Jak to, že druhý signál je červený?? Lze nastavit barvu stopy

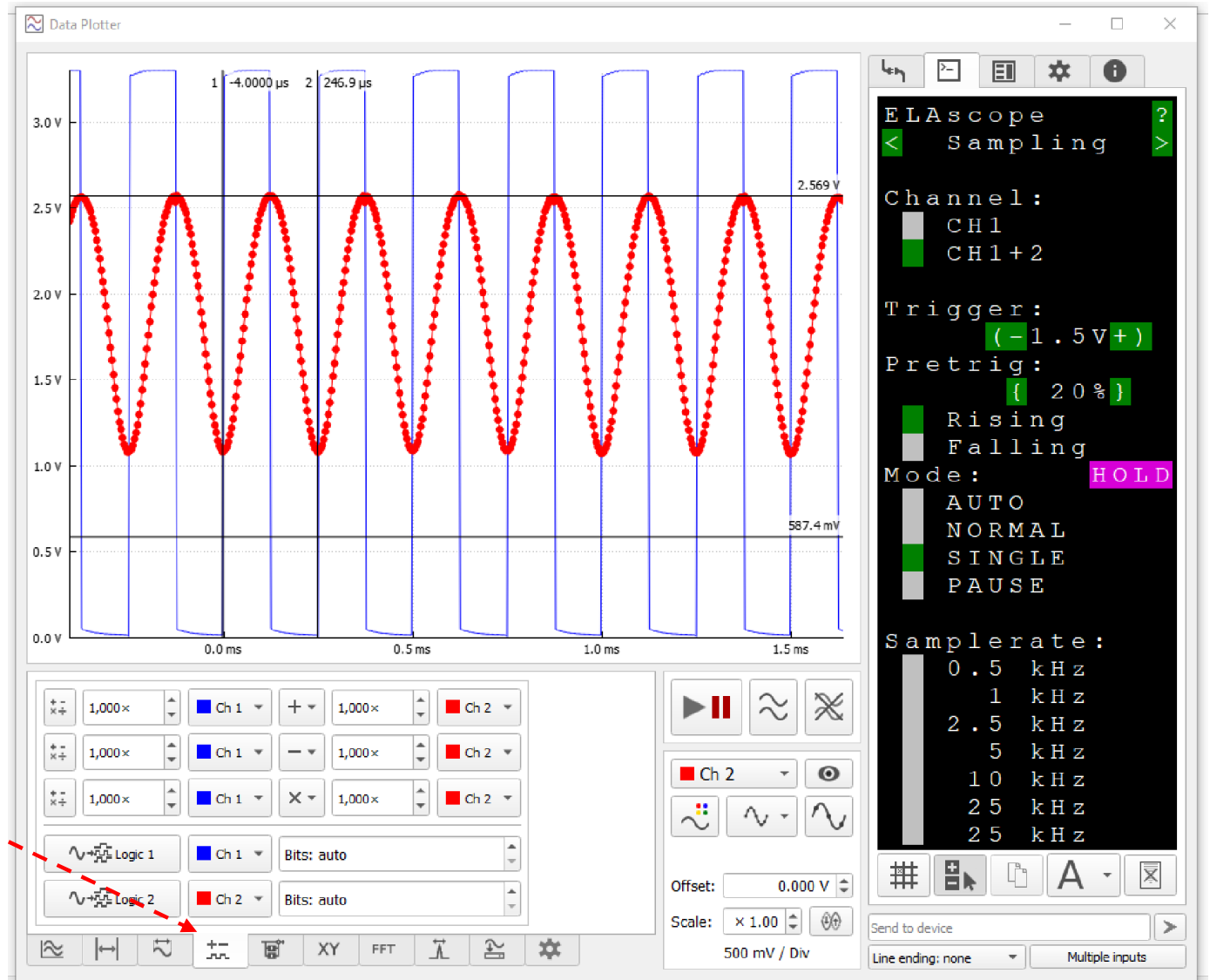
The screenshot displays the ELAScope software interface. The main window shows a signal plot with two traces: a blue square wave and a red parabolic curve. A 'Select Color' dialog box is open, showing a color selection tool with a rainbow gradient and a vertical color bar. A red arrow points from the text above to the color bar. The dialog box includes 'Basic colors', 'Custom colors', and 'HTML: #ff0000' fields. The main window also features a control panel with various settings like 'Value grid', 'Horizontal grid', 'Scale', and 'Offset'. On the right, a terminal window displays the ELAScope configuration, including 'Channel: CH1, CH1+2', 'Trigger: (-1.5V+)', 'Pretrig: { 20% }', 'Mode: HOLD', and 'Samplerate: 0.5 kHz, 1 kHz, 2.5 kHz, 5 kHz, 10 kHz, 25 kHz, 25 kHz'.

Panel měření parametrů signálu

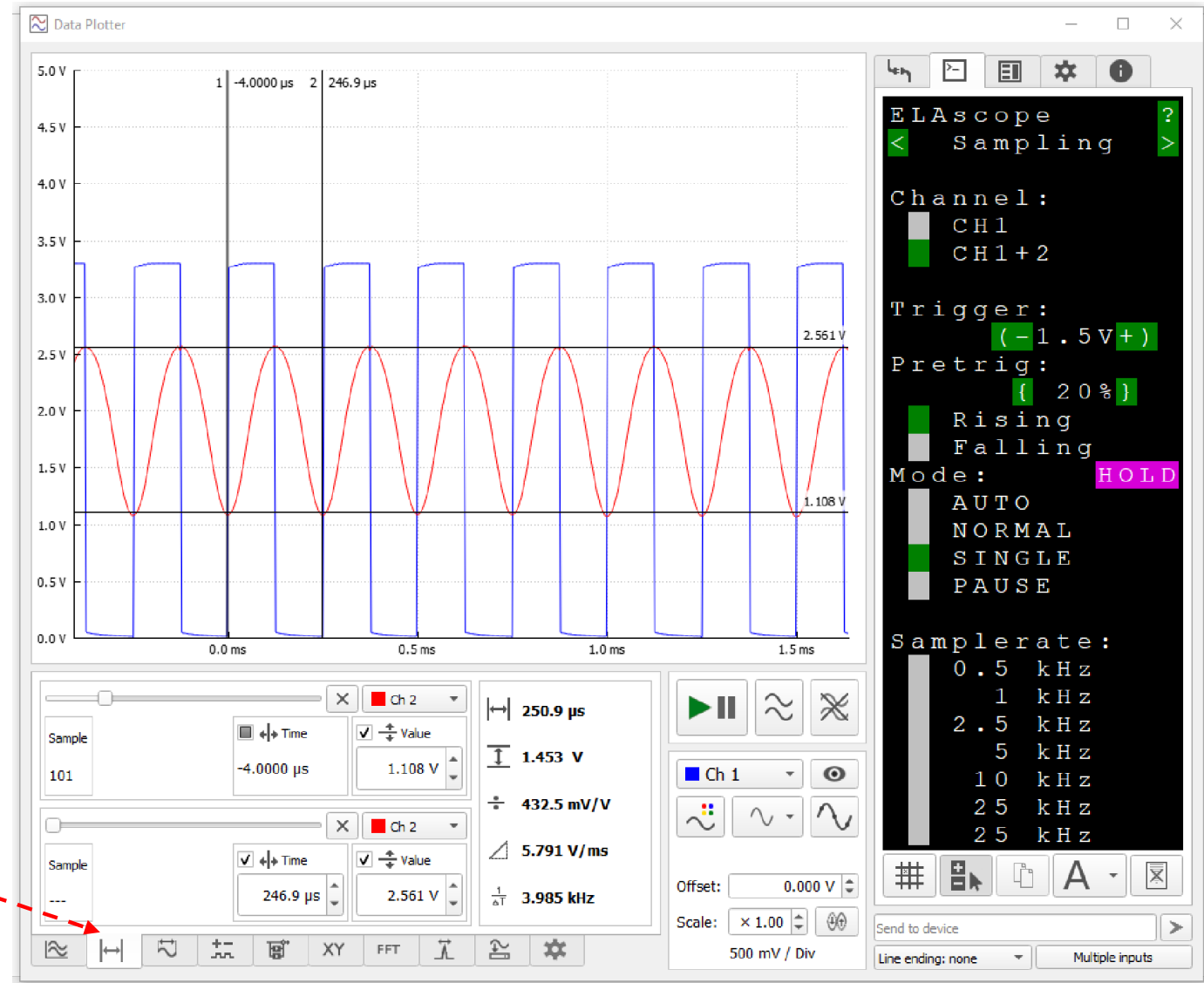


Panel matematických operací se signály

Funkce součet,
rozdíl, součin



Práce s kurzory- „ruční měření“ parametrů signálu

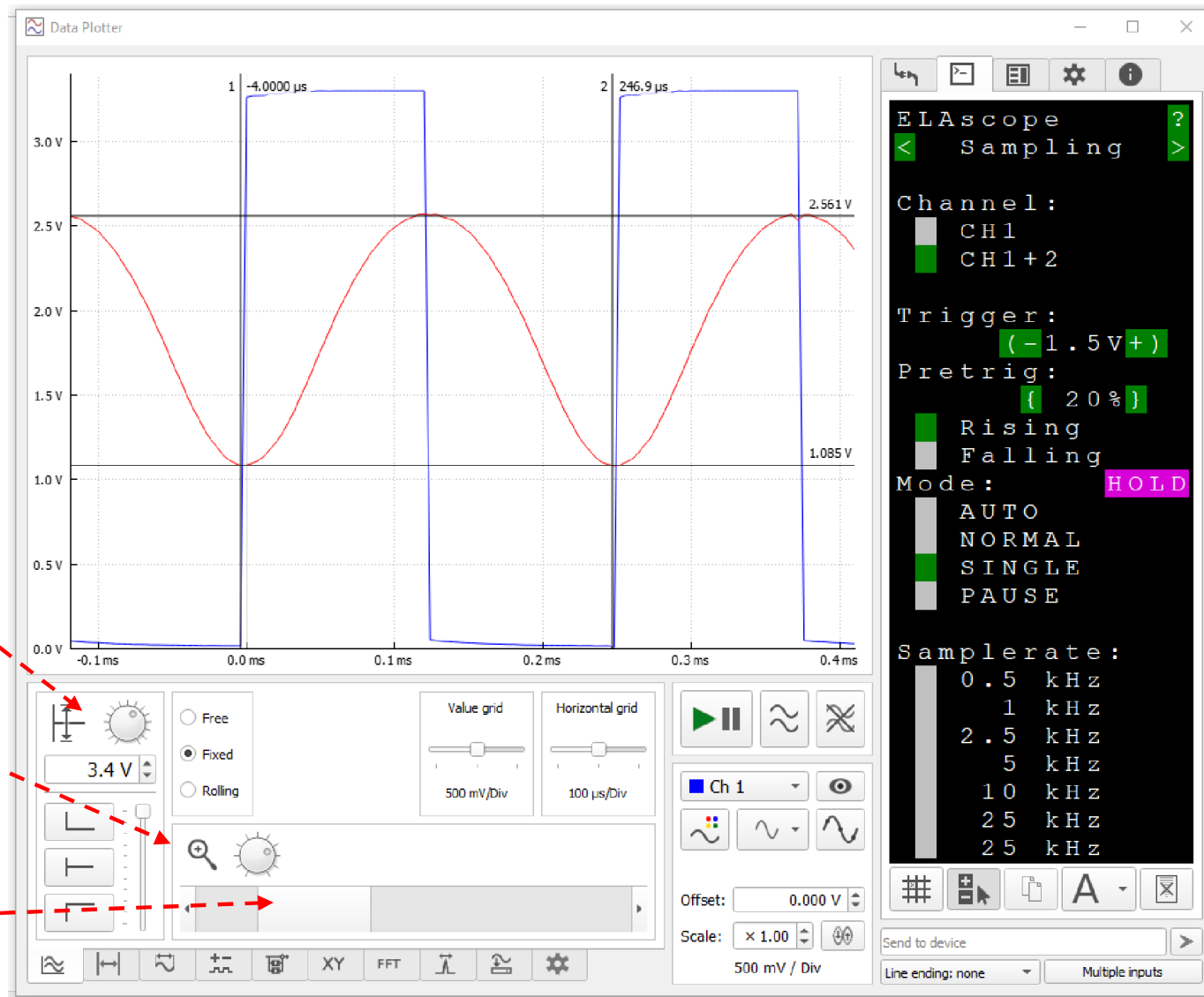


Zoom záznamu, horizontální a vertikální

vertikální zoom

horizontální zoom

horizontální posun



Použití RP PICO jako voltmetr pro statické měření

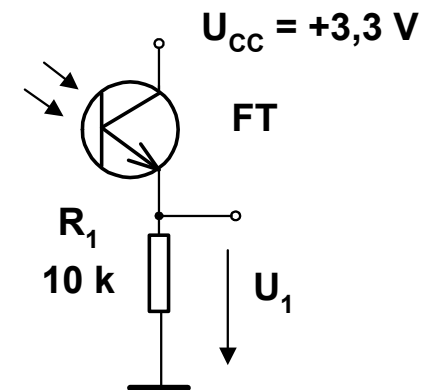
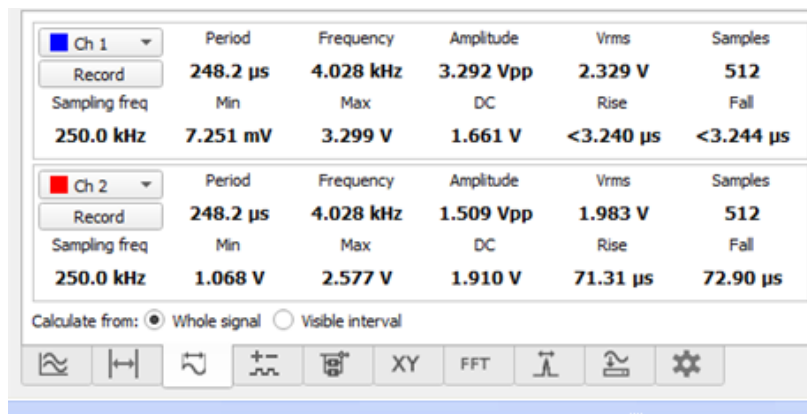
Nastavit auto trigger, záznam 512 vzorků ,10 KS/s nebo víc
zvolit panel měření zobrazení V_{rms}

Lineární měření světla- napětí U_1 by mělo být cca do 2 V, jinak se mohou projevovat nelinearity.

Větší proud- větší napětí, použij 2x paralelně 10 k, nebo 2k2

Fyzika, směrová charakteristika žárovek, srovnání svítivosti žárovek v ose, měřit ve vzdálenosti 1 m , před fototranzistor vložit **optický difuzér**

Vyhodnocení jasu povrchů- fototranzistor **bez difuzéru**, se stínicí bužírkou pro zamezení dopadu světla z boku.



Nahrávání programu

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

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

- buď firmware osciloskopu nebo firmware Micropython (používáme jej při grafickém programování)

Micropython, soubor je na stránce Workshopu

Pro práci s osciloskopem:

Firmware osciloskopu je také na stránce workshopu

PC Aplikace Dataplotter – odkaz na stránce workshopu

https://embedded.fel.cvut.cz/stredni_skoly/workshop_2_2_2023

Spolupráce s dalšími senzory

Termistor 10 k, s rostoucí teplotou klesá silně odpor
možnost staticky určovat teplotu
měření odporu - RP PICO „voltmetr“ na odporovém děliči
termistor + 10 k

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.*

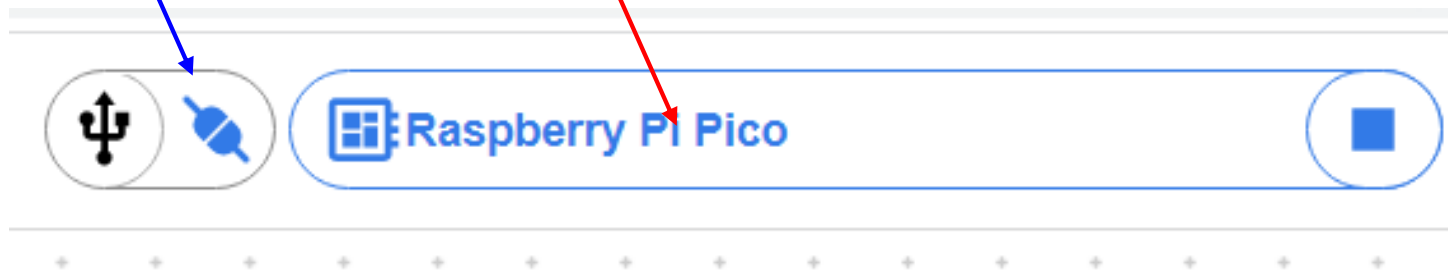
*Ke stažení je též na **micropython.org***

<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

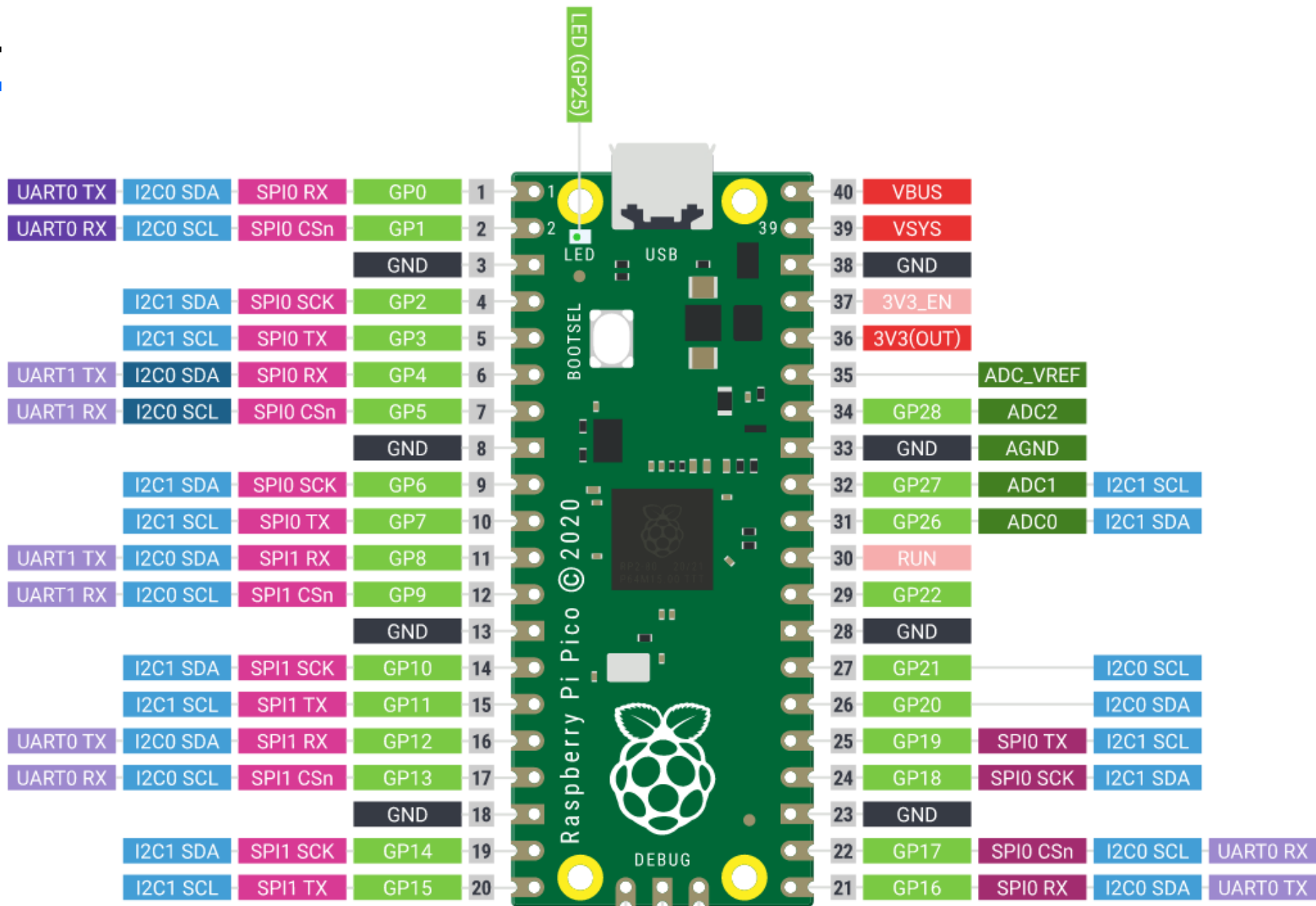
Připojení desky, volba rozhraní **USB**
někdy potřebuje **opakovat dvakrát**
připojit, odpojit, připojit

Nabídky panelů

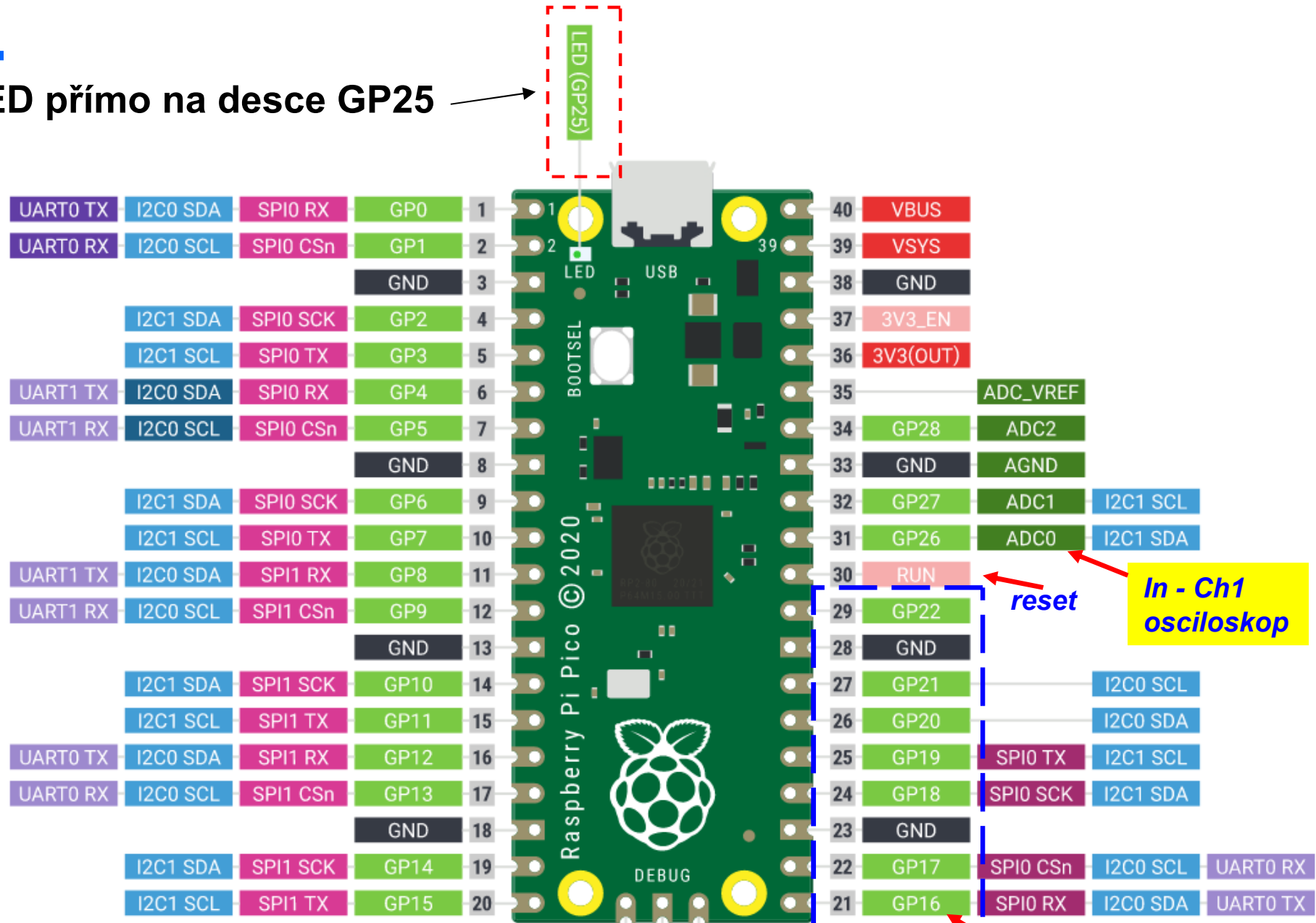
Device- zobrazí rozložení vývodů desky, to budeme potřebovat při programování

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 on the left and 41 through 45 on the right. Each pin is color-coded according to its function. A legend at the bottom of the diagram provides the key for these colors:

- 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)
- Debugging (Orange)



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*

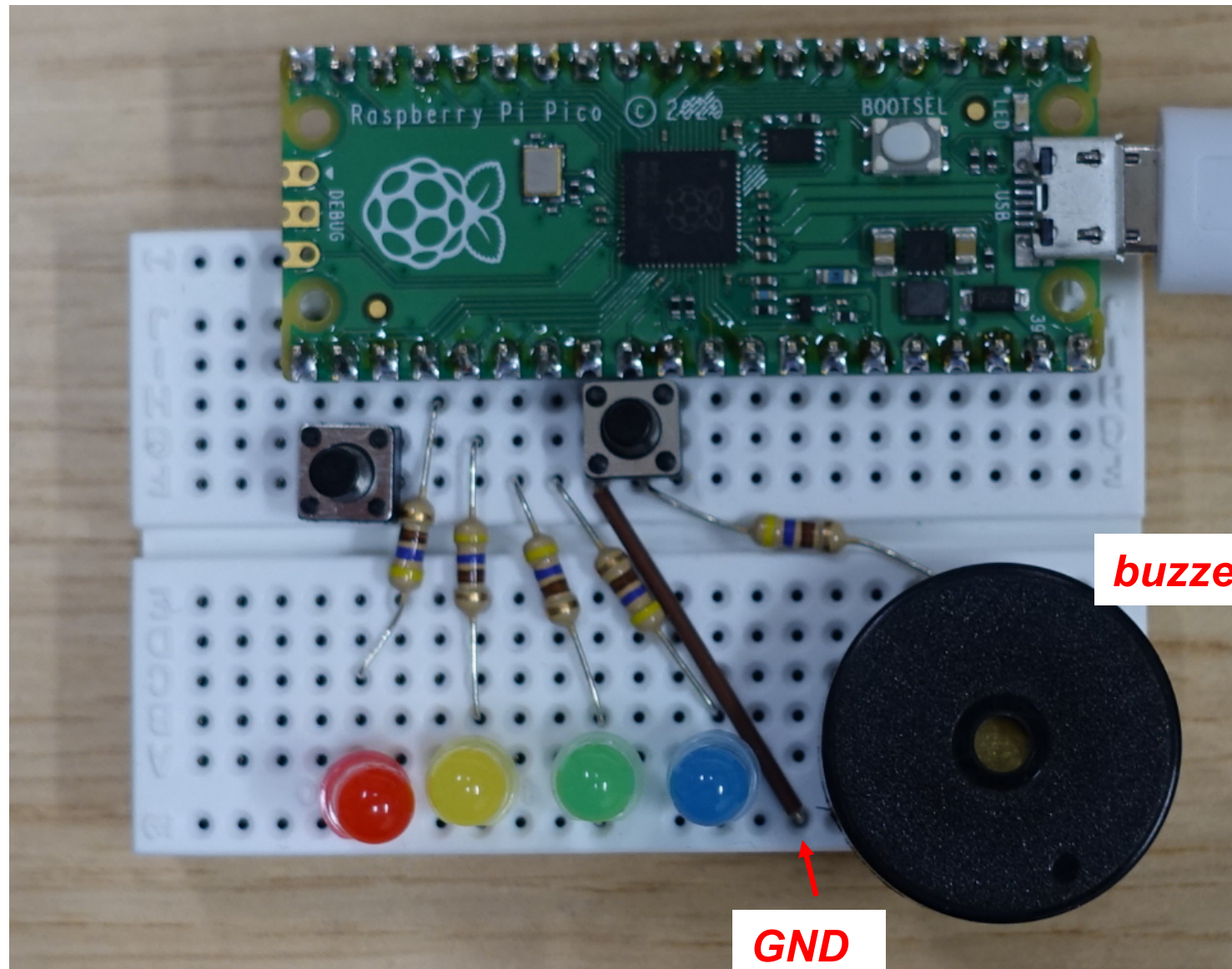
GP27 *Ch2 In*

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

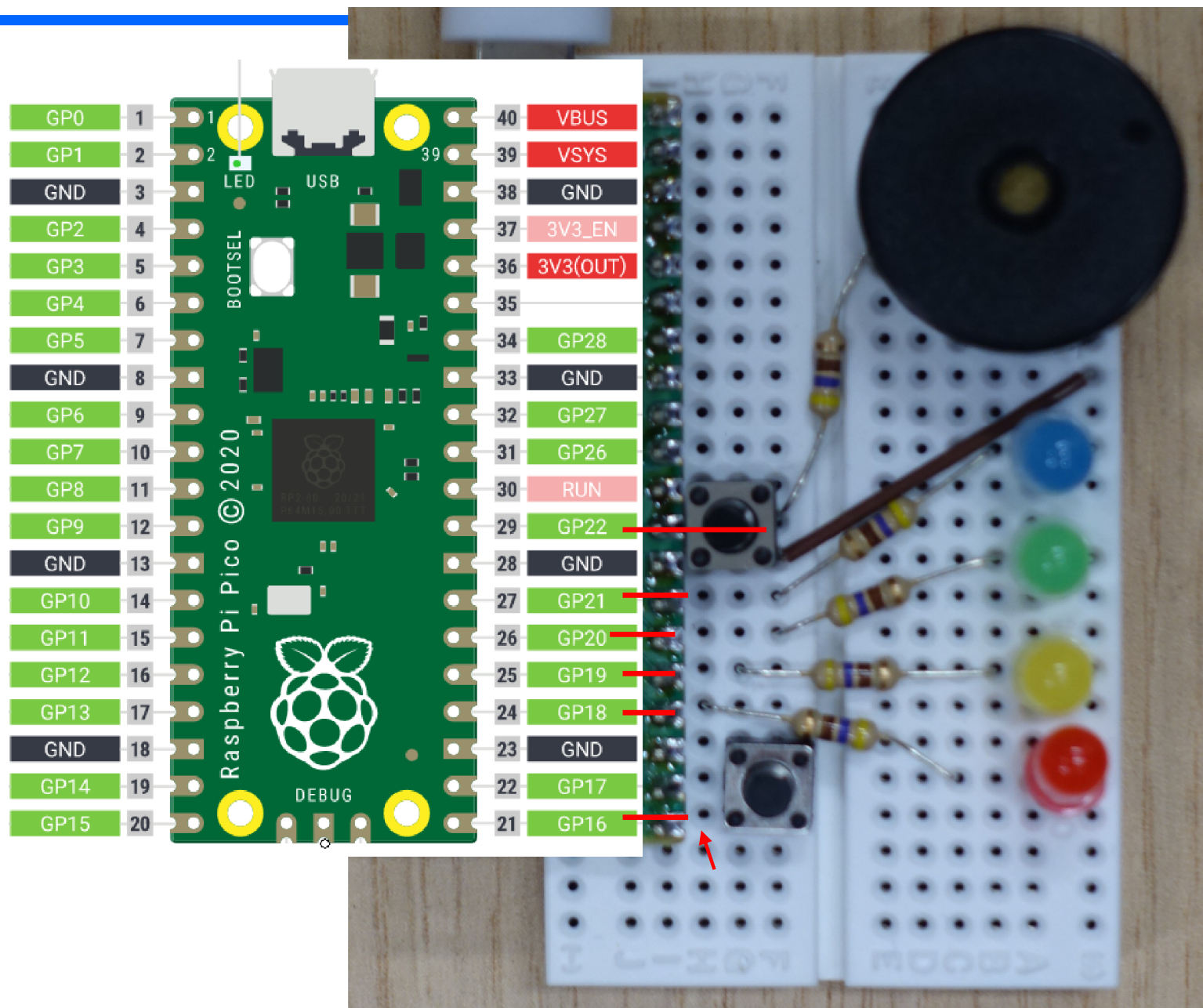
Úroveň „true“ na pinu LED rozsvítí,

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

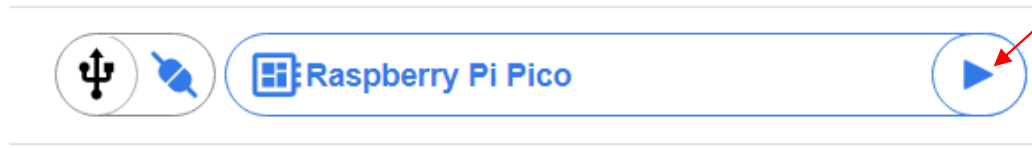
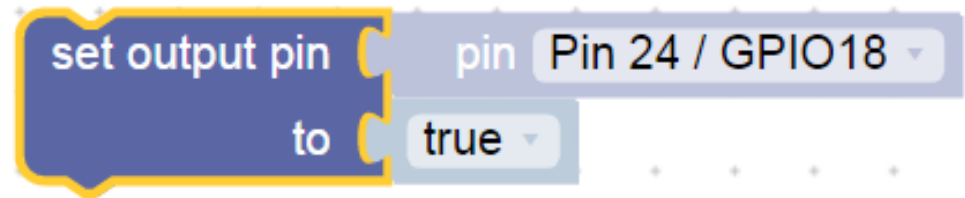
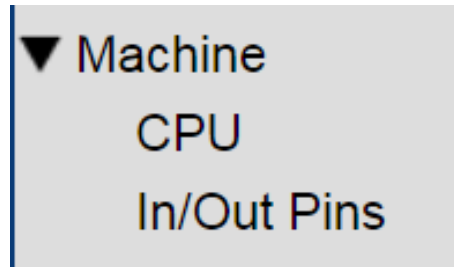
Rozsvítit LED- nástroje

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 'pin LED / GPIO25' and a value dropdown set to 'true'.
- A 'read digital input Pull-up' block with a dropdown menu set to 'pin LED / GPIO25' and a value dropdown set to '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 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 sequence of 12 code blocks in a Scratch-like environment. The blocks are arranged in a vertical stack. Each block is a 'set output pin' block followed by a 'delay' block. The 'set output pin' blocks are for pins Pin 24 / GPIO18, Pin 25 / GPIO19, and Pin 26 / GPIO20. The 'delay' blocks are all set to 1 seconds. The sequence of actions is: set Pin 24 to true, delay 1 second, set Pin 24 to false, delay 1 second, set Pin 25 to true, delay 1 second, set Pin 25 to false, delay 1 second, set Pin 26 to true, delay 1 second, set Pin 26 to false, delay 1 second.

Stálý běh programu, použití nekonečné smyčky



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

Menu Loops

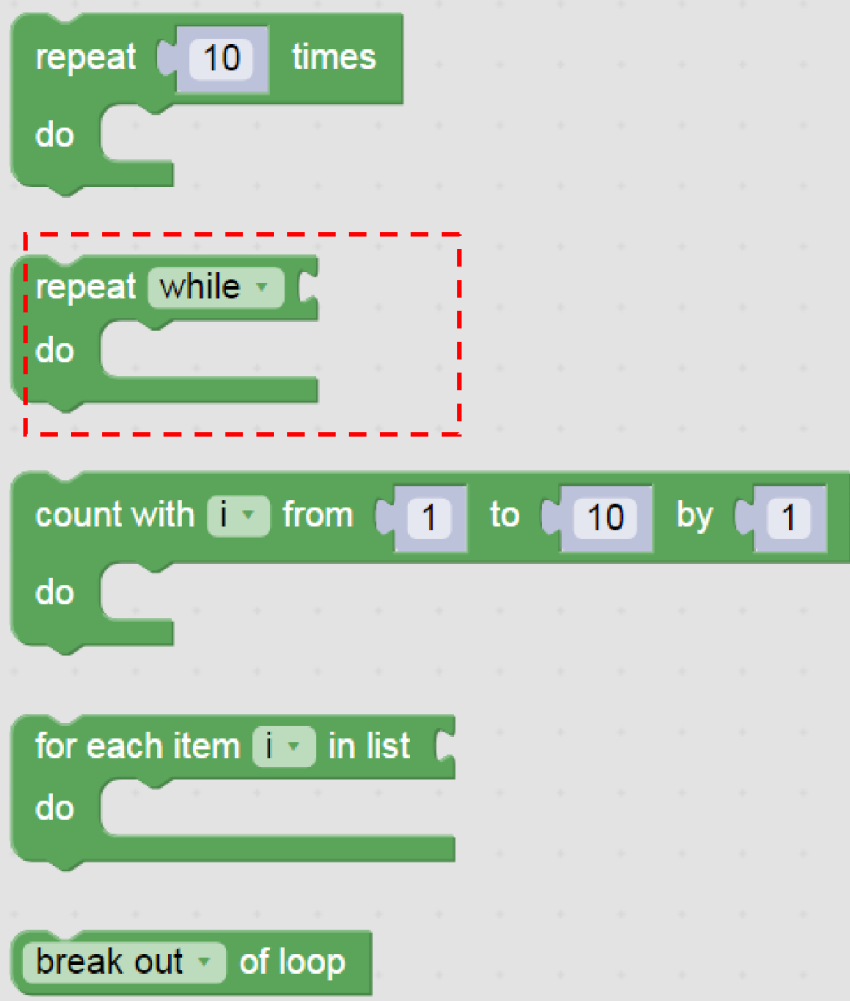
Menu Logic

The image shows a screenshot of the MicroPython IDE's block editor. On the left, the 'Menu Logic' is open, showing various logic blocks. A red dashed box highlights the 'true' block. A red dashed arrow points from this 'true' block to the 'while' condition of a 'repeat while' loop block in the main workspace. The loop block contains the following code:

```
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
```

Stálý běh programu, použití nekonečné smyčky

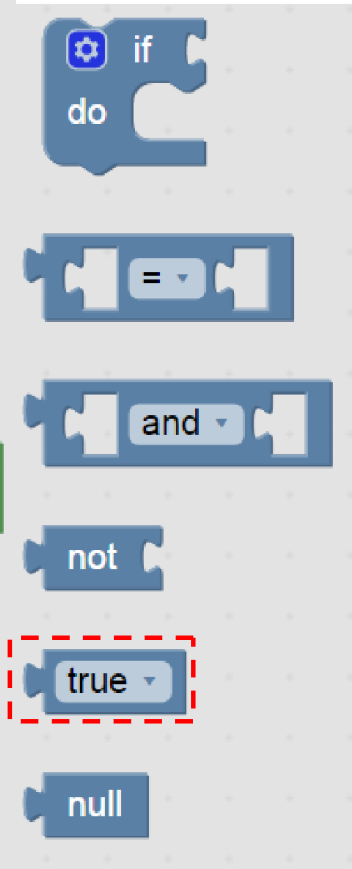
Menu Loops



Scratch menu for loops containing the following blocks:

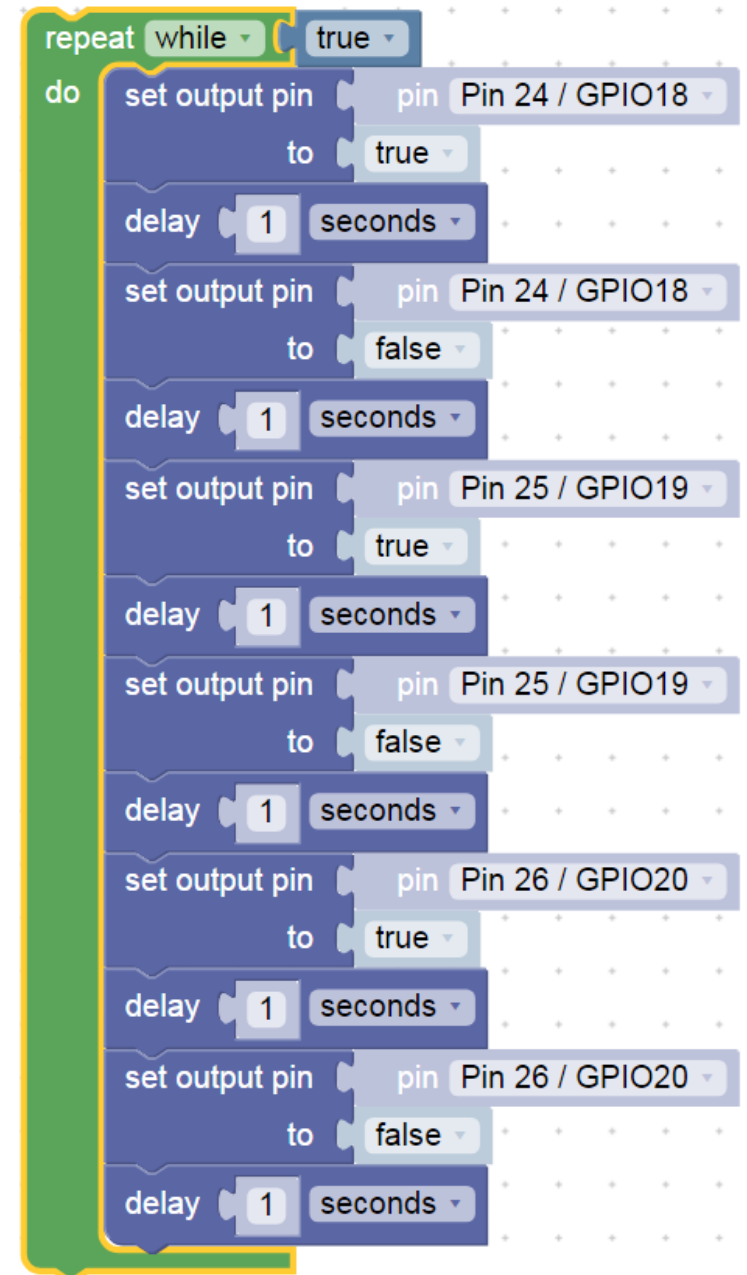
- repeat 10 times do
- repeat while do (highlighted with a red dashed box)
- count with i from 1 to 10 by 1 do
- for each item i in list do
- break out of loop

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

A Scratch-style block palette for loops and iteration. The palette is divided into several categories, with 'Loops' highlighted in green. The categories and their contents are:

- Logic**
- Loops** (highlighted):
 - repeat 10 times
 - do
- Math**
- Text**
- Lists**
- Variables**
- Functions**
- BIPES**
- Python**
- Timing**
- Machine**
 - CPU
 - In/Out Pins
- Displays**
- Sensors**
- Actuators**
- Communication**
- Files**

The 'Loops' category contains the following blocks:

- repeat 10 times
- do

The 'Python' category contains the following blocks:

- repeat while
- do

The 'Machine' category contains the following blocks:

- count with i from 1 to 10 by 1
- do

The 'Actuators' category contains the following block:

- for each item i in list
- do

The 'Communication' category contains the following block:

- break out of loop

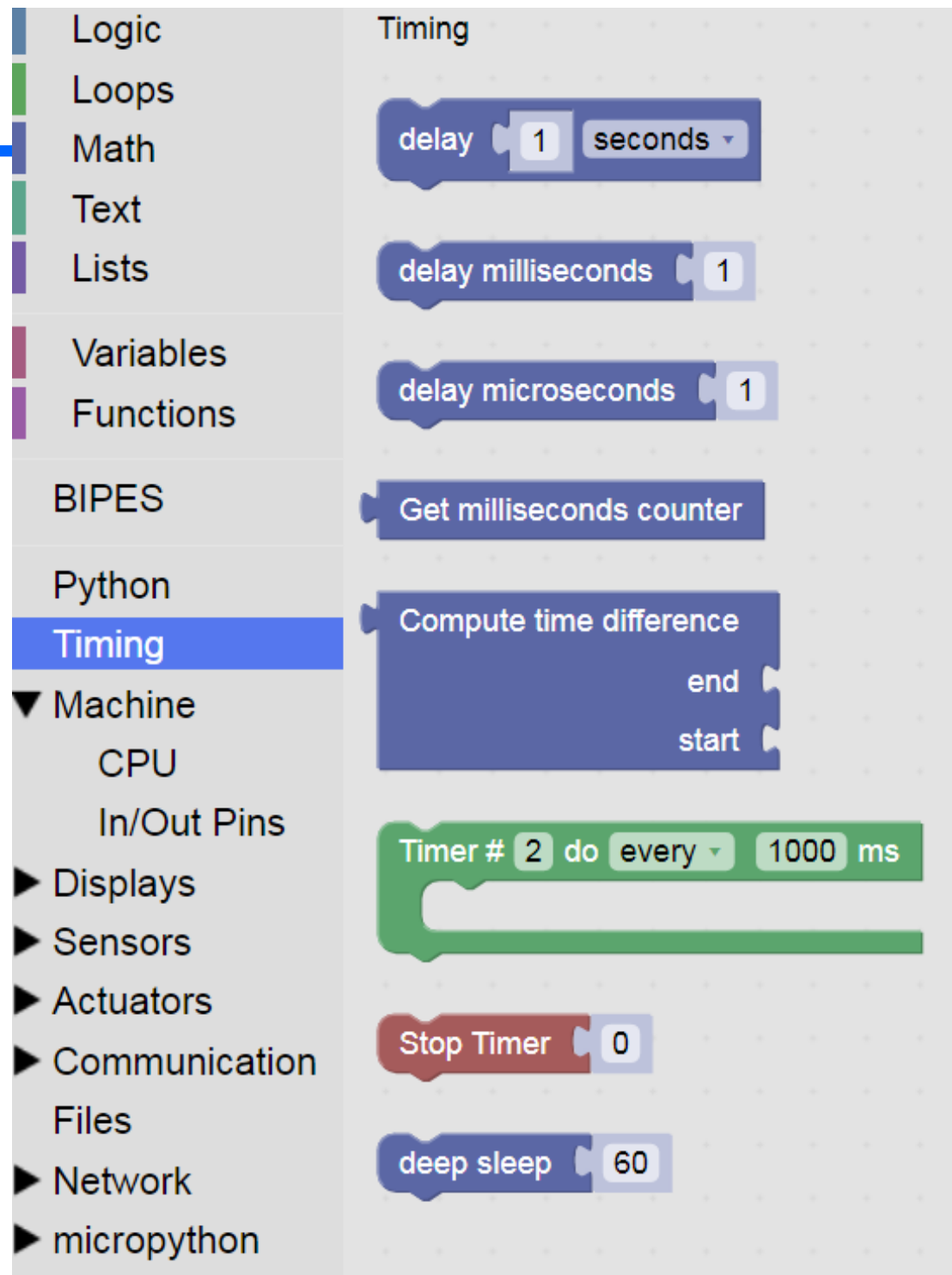
Páce s PINY

The screenshot shows the MicroPython IDE interface. On the left is a sidebar with a category list. The main workspace on the right contains a sequence of blocks for an RPi Pico program:

- Logic**: A 'pin' block with 'LED / GPIO25' selected.
- Logic**: A 'set output pin to' block with 'pin LED / GPIO25' and 'true' selected.
- Logic**: A 'read digital input' block with 'pin LED / GPIO25' and 'Pull-up true' selected.
- Logic**: A 'Read RPi Pico ADC Input' block with 'pin LED / GPIO25' selected.
- Logic**: An 'RPI Pico PWM #' block with '0' selected, connected to a 'Pin' block with 'pin LED / GPIO25' selected.
- Logic**: A 'Frequency' block with '1000' selected.
- Logic**: A 'Duty' block with '50' selected.
- Logic**: A 'PWM # 0 frequency' block with '1000' selected.
- Logic**: A 'PWM # 0 duty' block with '50' selected.
- Logic**: A 'PWM # 0 init' block with 'pin LED / GPIO25' selected.
- Logic**: A 'deinit PWM # 0' block.

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

The image displays three panels of Scratch-style code blocks. The left panel shows logic blocks: 'if do', comparison operators ('='), logical operators ('and', 'not'), constants ('true'), 'null', and conditional execution ('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'), timer management ('Get milliseconds counter', 'Compute time difference end start'), and a repeating timer ('Timer # 2 do every 1000 ms'), along with '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 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

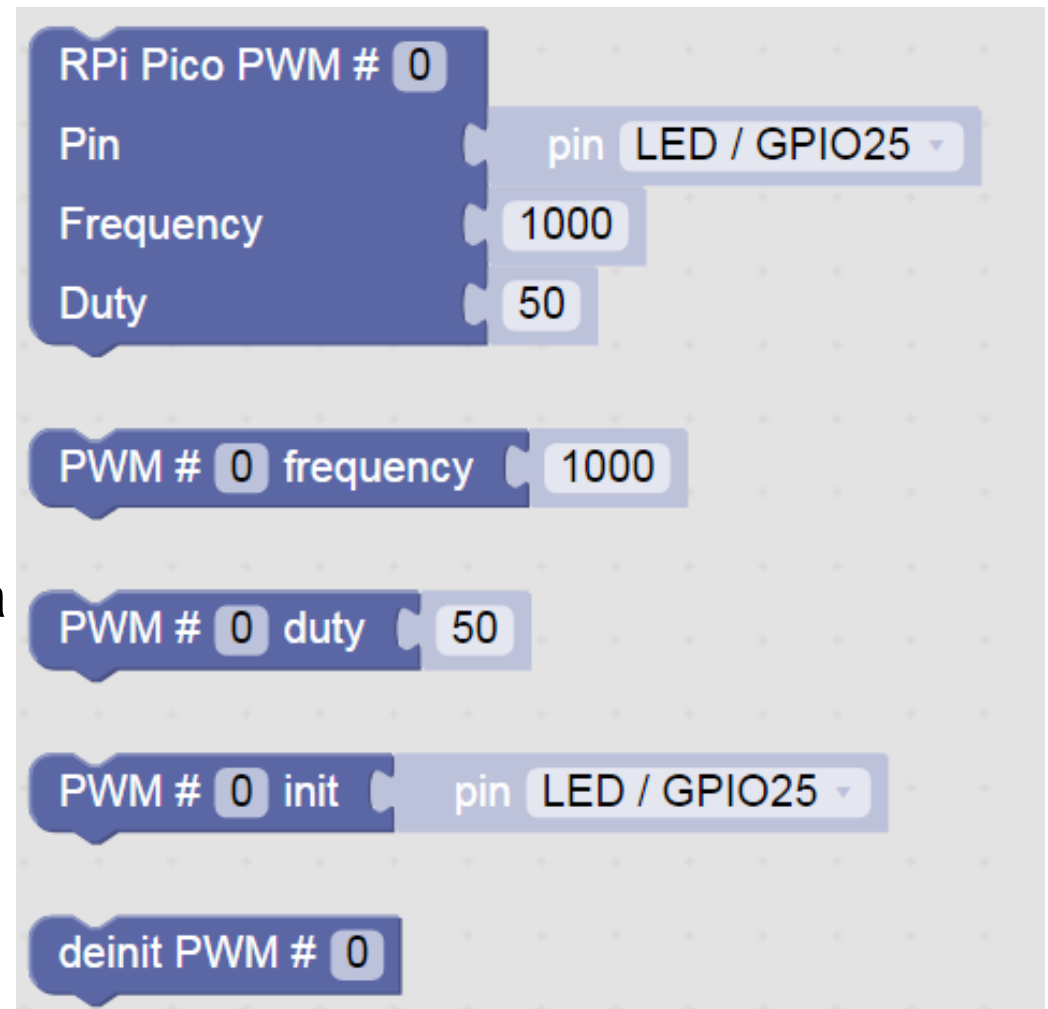
32768 znamená 50 procent !!

To souvisí s formou vyjádření 16- bitového binárního čísla

Pokud dáme 50, tak to skoro nebude svítit.

Nula- 0 nesvítí vůbec,

65535 svítí plně

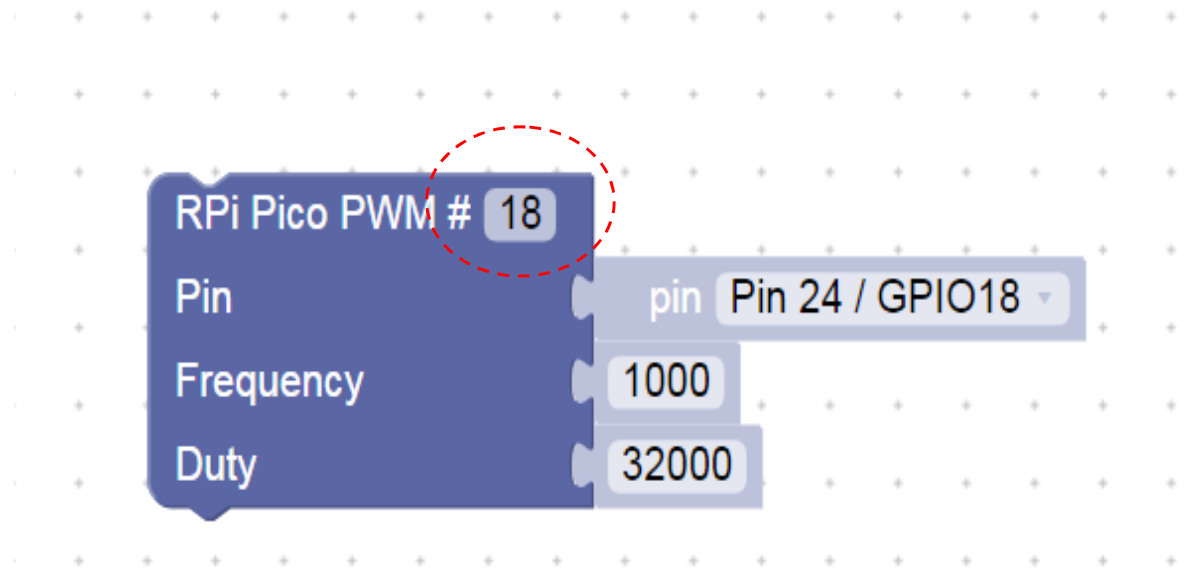


Jednorázová inicializace PWM

Červená LED bude svítit plovicním jasem

Můžeme zkusit i frekvenci 10 Hz – budeme vidět blikání

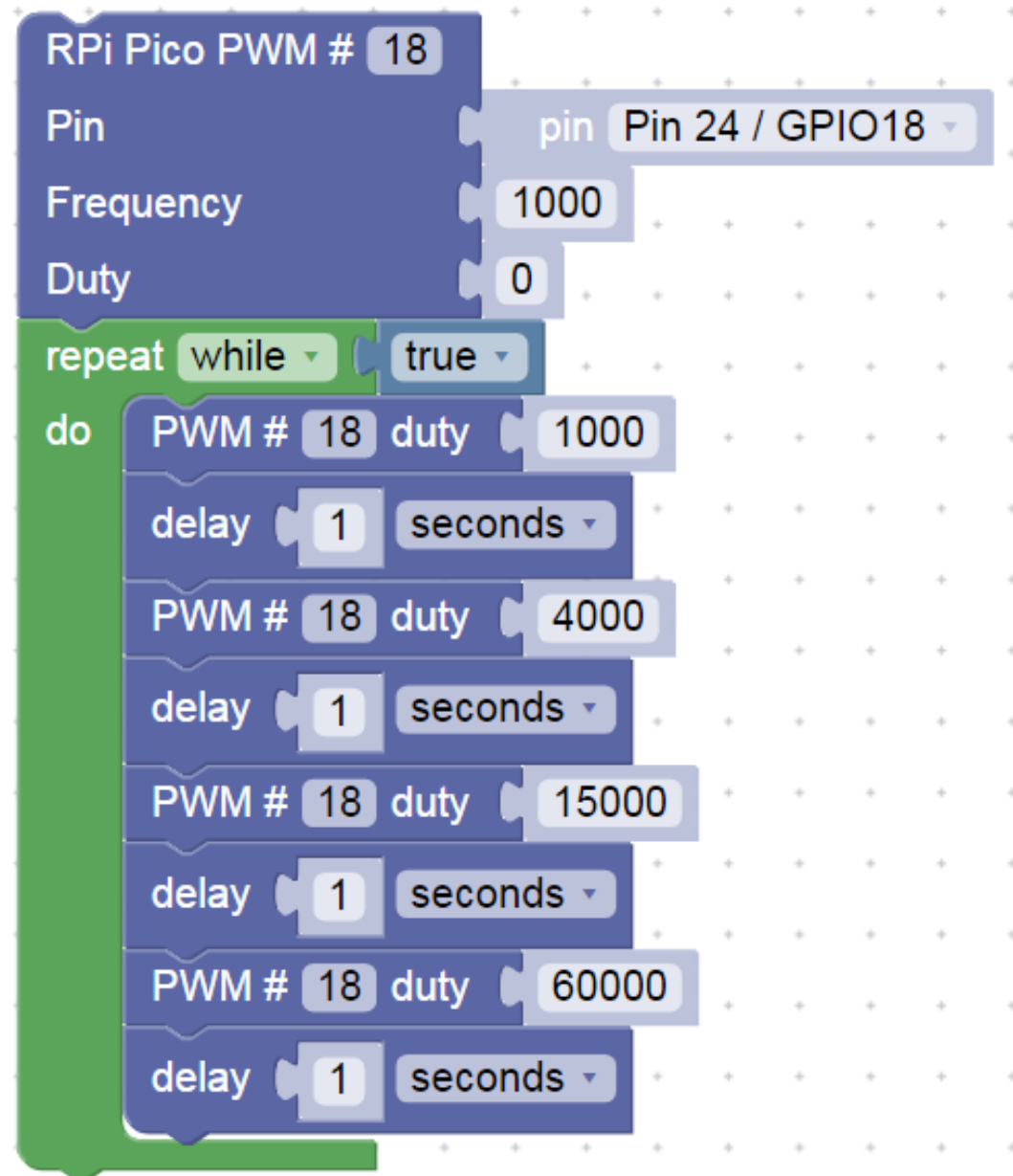
Frekvenci **1 Hz nelze** použít (procesor to nepodporuje)



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

Nekonečná postupná změna jasů LED ve stupních

Postupné rozsvícení a naráz zhasnutí LED



Generování akustického signálu - buzzer

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

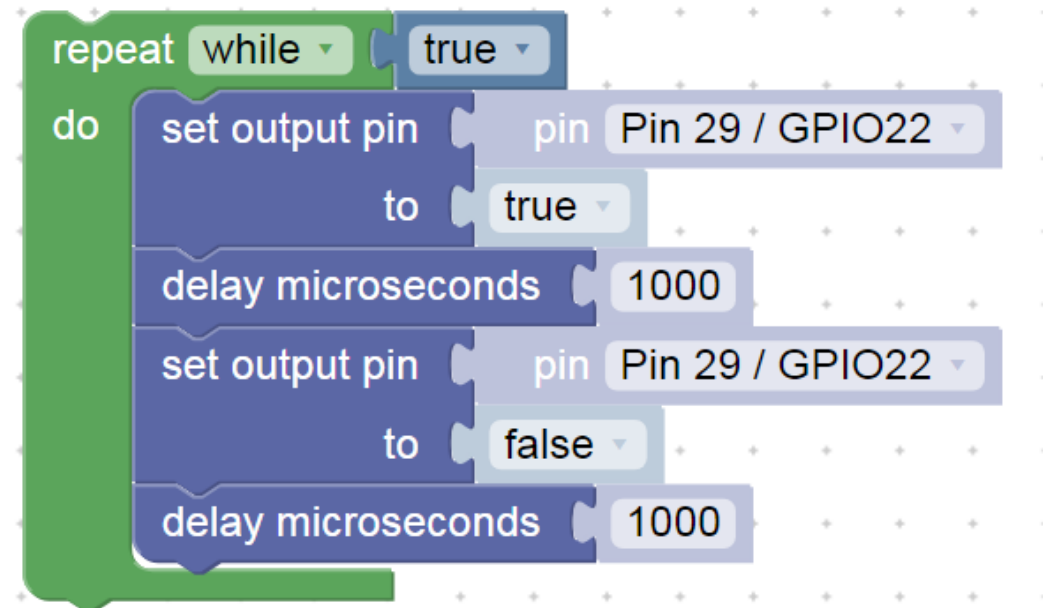
Tím se budí „reproduktorek- „bzučák“ – buzzer

Zkusit měnit časy čekání

Toto však není dobrý způsob,
jak generovat zvuk

Na to využijeme vlastní funkci
procesoru PWM

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



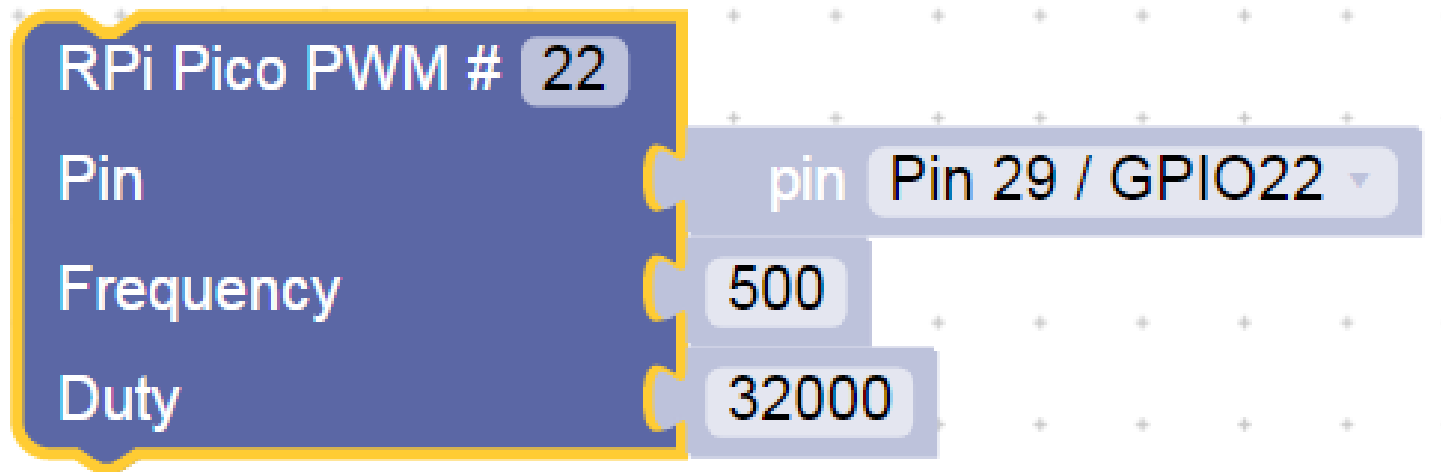
Jednorázová inicializace PWM pro Buzzer

Jednorázová inicializace- buzzer na **GPIO22**

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

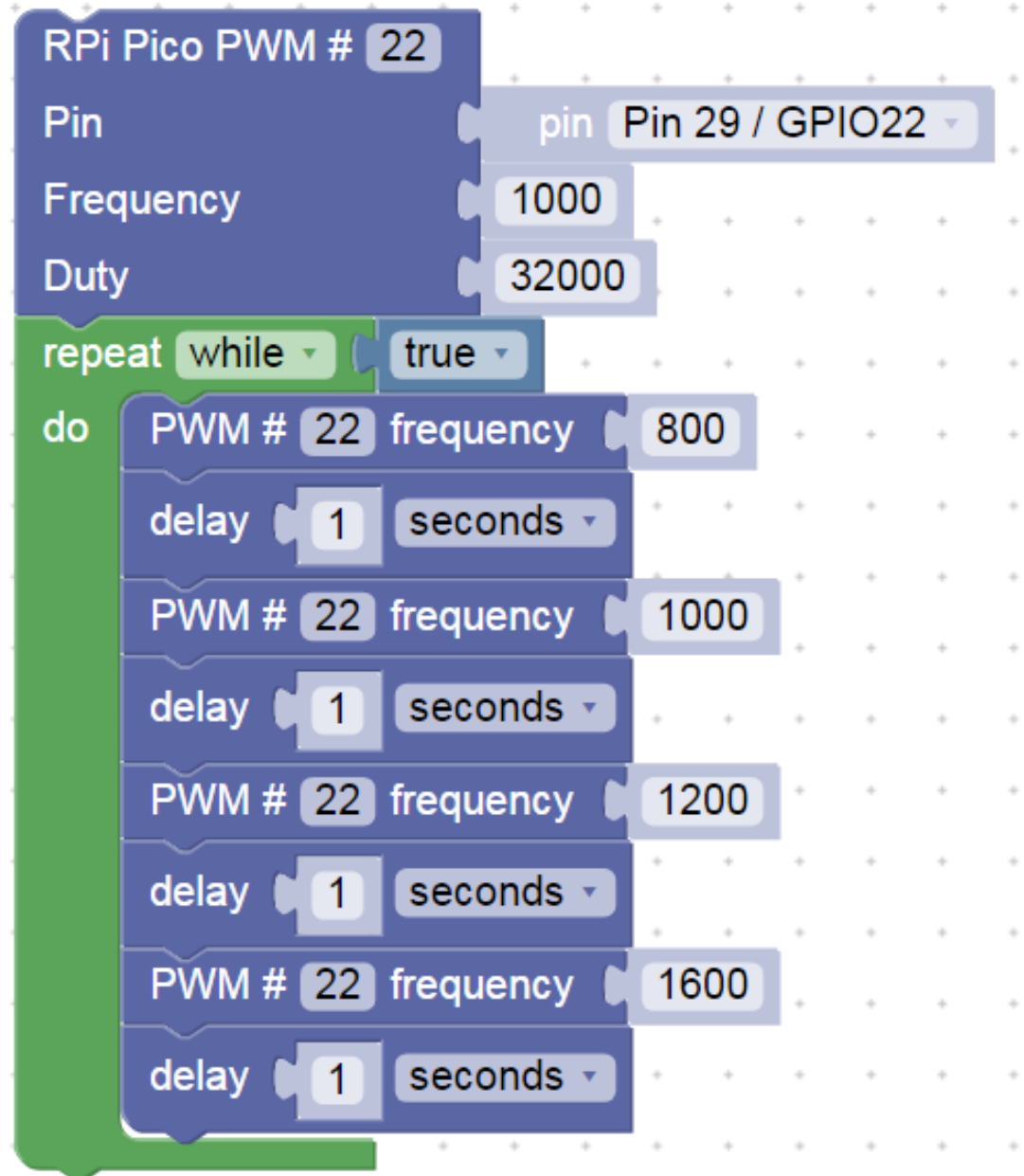
Frekvence 500 Hz, střída cca 49 % = $32000 / 65536$, ($65536 = 2^{12}$)

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



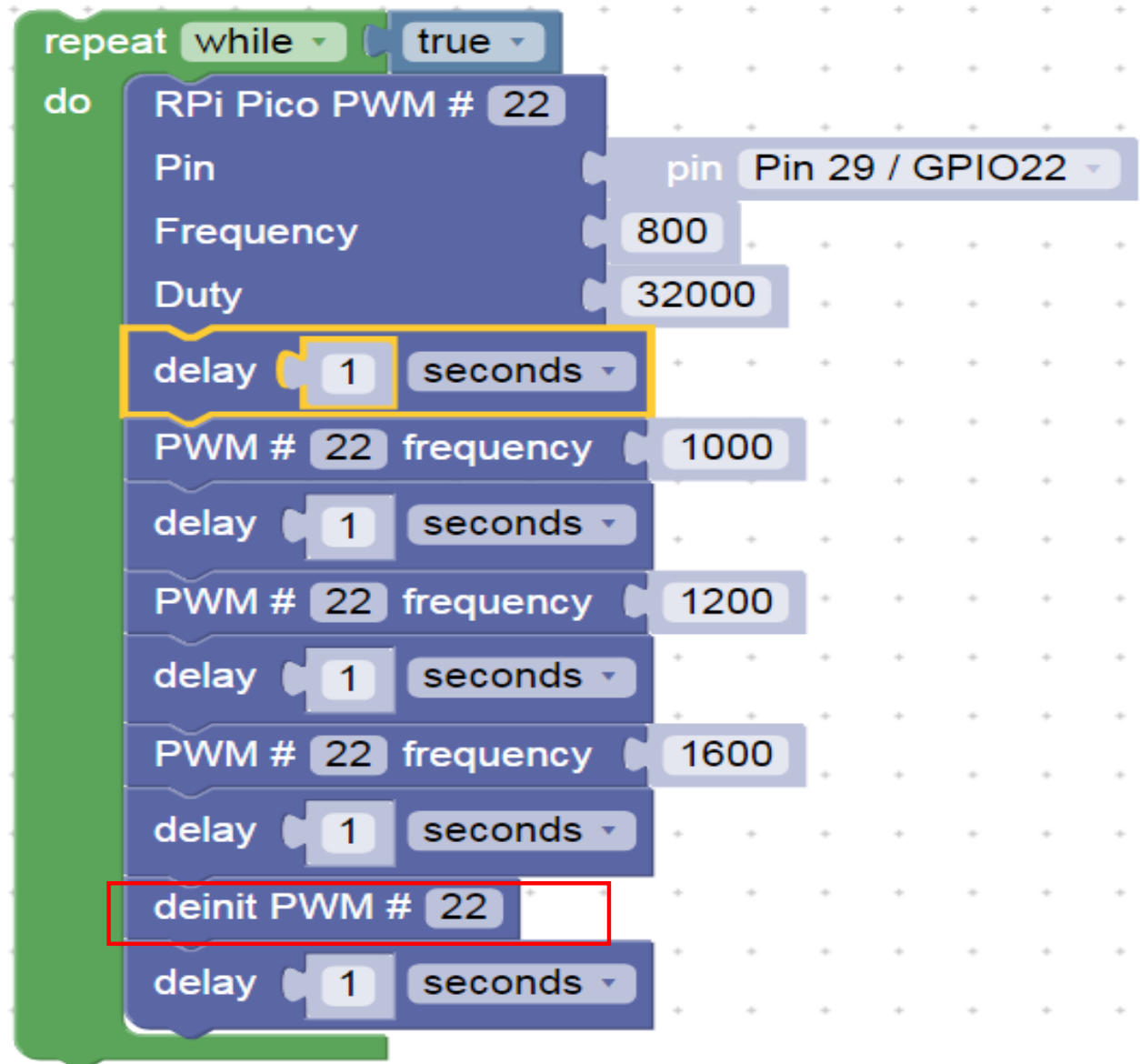
Ovládání bzučáku (buzzer) pomocí PWM

Změna frekvence PWM



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**

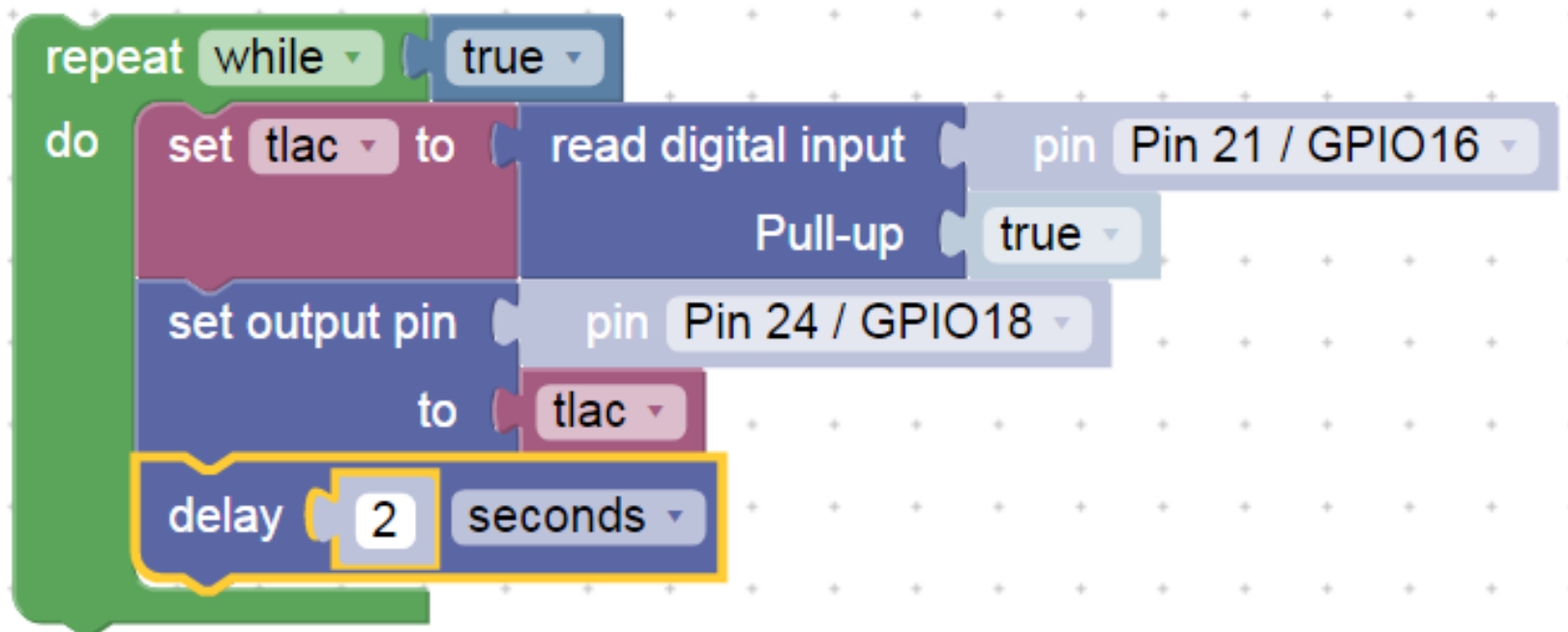
```
repeat while true do
  set output pin pin Pin 24 / GPIO18 to read digital input pin Pin 21 / GPIO16 Pull-up true
```


Čtení stavu tlačítka a uložení do proměné

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

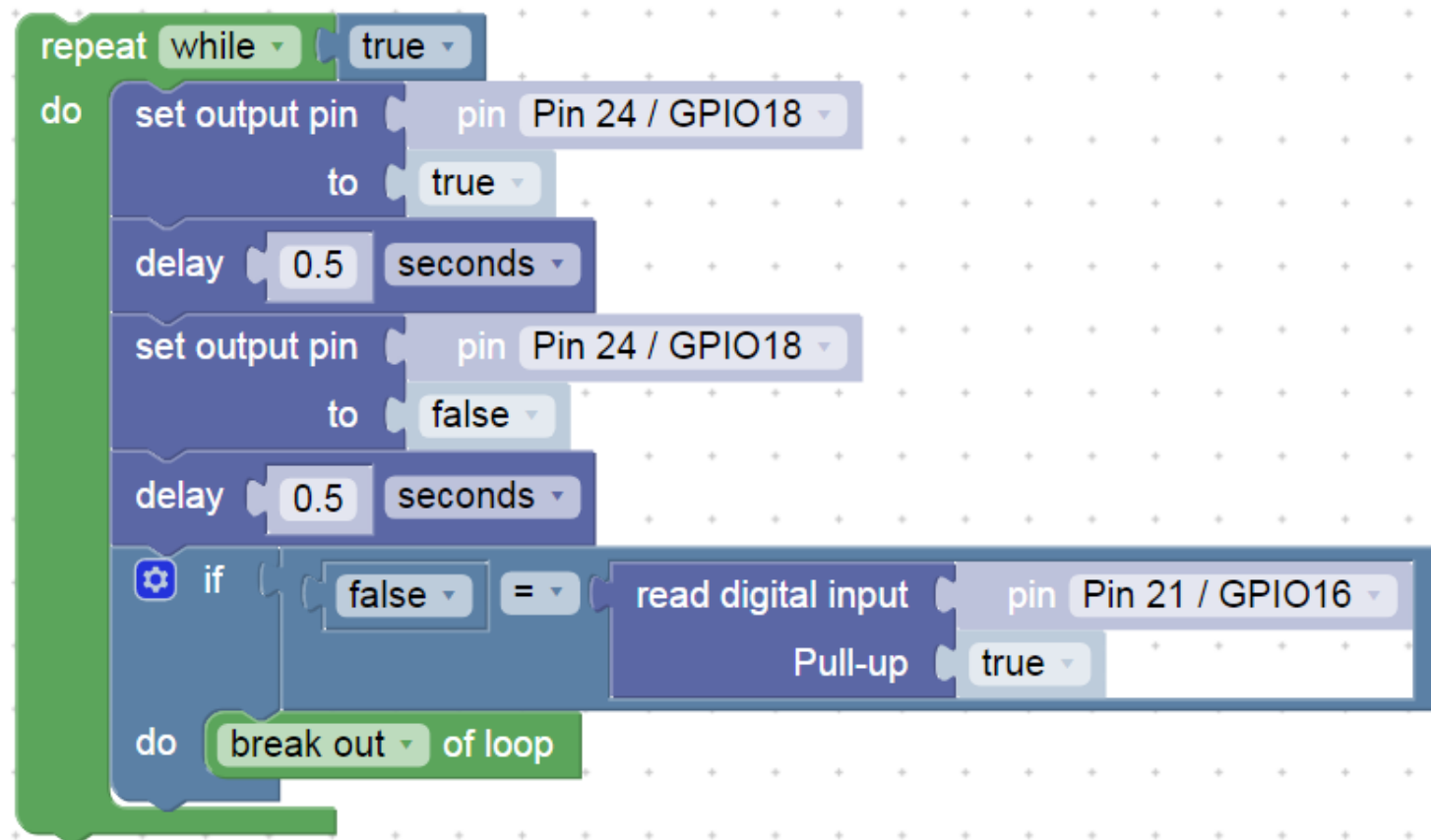
Důsledek - zpomalení reakce „**pomalé tlačítko**“

(*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**.



▪

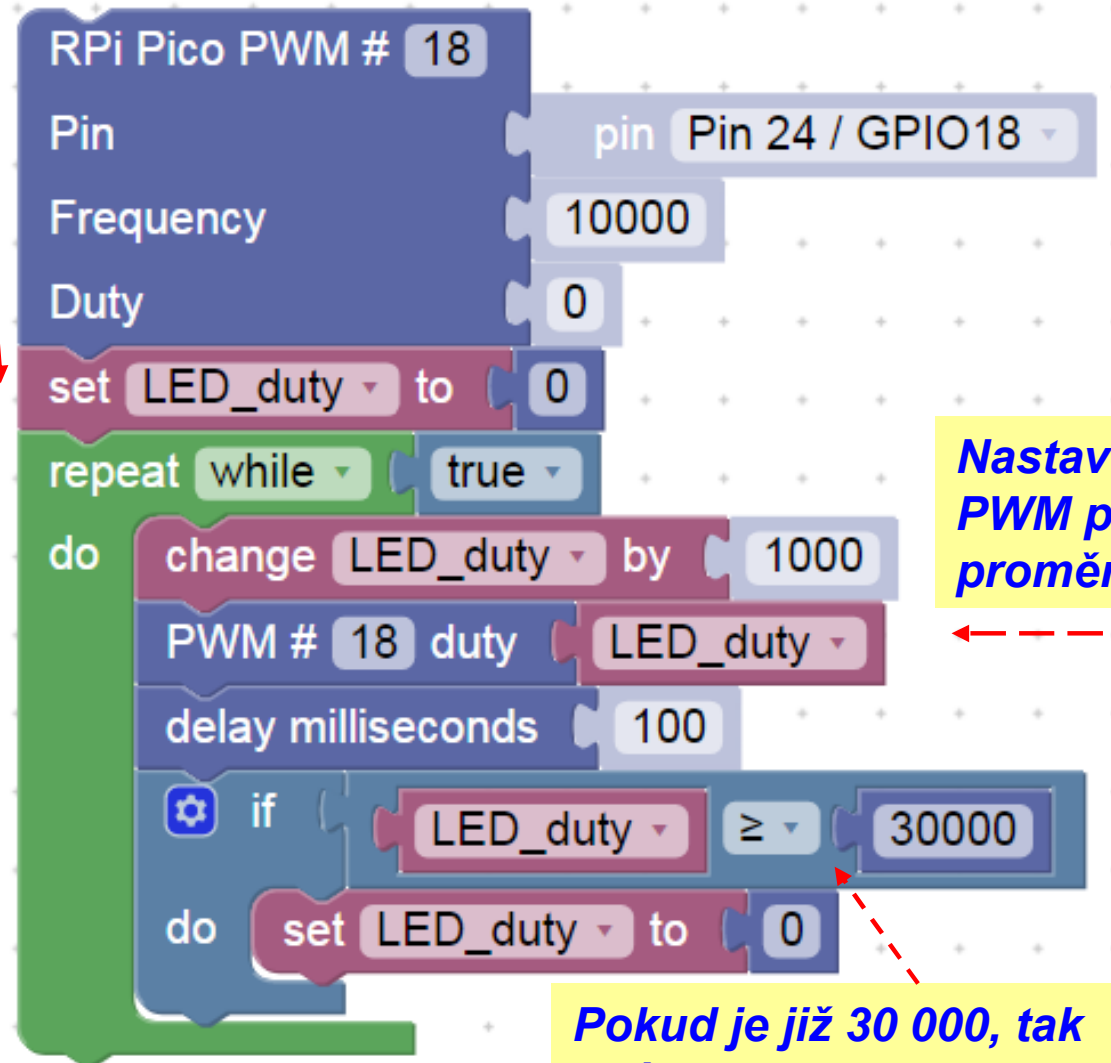
Dále – další příklady pro inspiraci

Postupná změna jasu LED na pinu GPIO18

Proměnná **LED_duty** nese informaci o požadované střídě

Postupné rozsvěování

LED a naráz pohasnutí

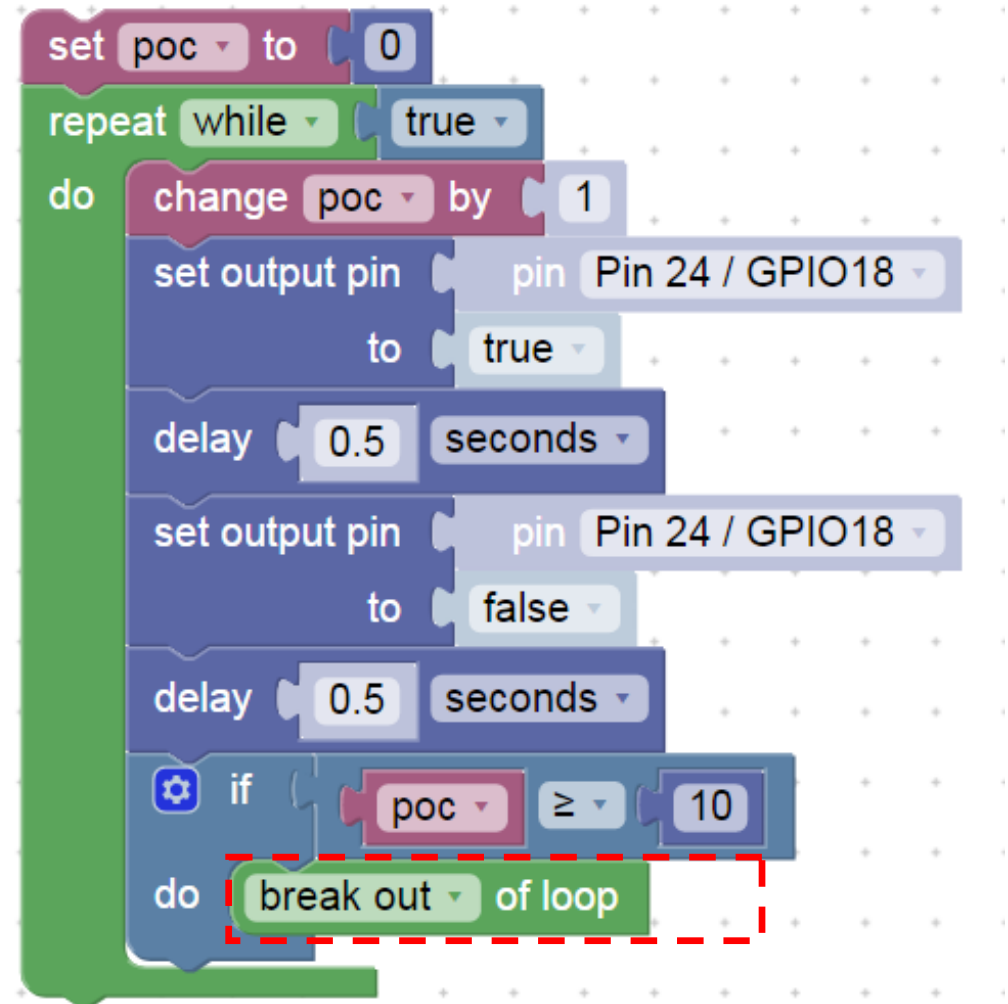


Nastavení PWM podle proměnné

Pokud je již 30 000, tak zpátky na nulu

Ukončení smyčky podle podmínky

Při splnění podmínky se ukončí
smyčka i celý program



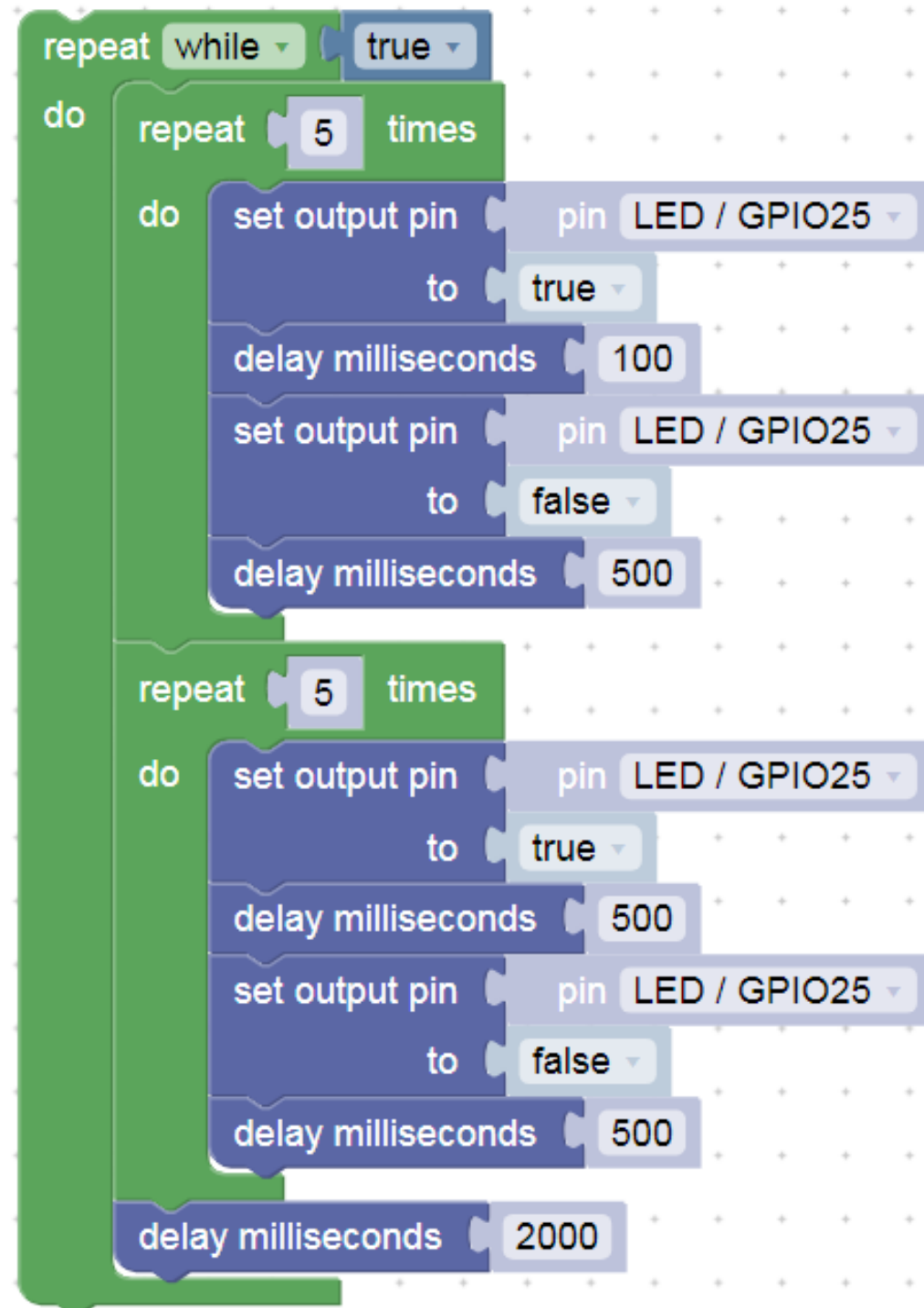
Postupná změna frekvence PWM pro bzučák (buzzer)

Frekvence 100 Hz až 2000 Hz

The image shows a Scratch-style block diagram for an RPi Pico. The main block is 'RPi Pico PWM # 22'. It has four sub-blocks: 'Pin' set to 'pin Pin 29 / GPIO22', 'Frequency' set to '100', and 'Duty' set to '32000'. Below this is a 'set Frkvence to 100' block. A 'repeat while true' loop contains three blocks: 'change Frkvence by 50', 'PWM # 22 frequency Frkvence', and 'delay milliseconds 200'. An 'if Frkvence ≥ 2000' block is connected to the loop, with a 'do set Frkvence to 100' block inside it. A yellow callout box with blue text 'Zvětšovat frek. po 50 Hz' and a red arrow points to the 'change Frkvence by 50' block.

```
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
```

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

