RPi Pico

Verze 1.00

Popis Informace Programování

RPi Pico microPython ve Win



mikroHW, Josef Havlíček nepřejímá žádné záruky, pokud se týče obsahu této publikace a vyhrazuje si právo měnit obsah dokumentace bez závazku tyto změny oznámit jakékoli osobě či organizaci.

Tento dokument může být kopírován a rozšiřován za následujících podmínek:

1. Celý text musí být kopírován bez úprav a se zahrnutím všech stránek.
2. Všechny kopie musí obsahovat označení autorského práva společnosti mikroHW, Josef Havlíček a veškerá další upozornění v dokumentu uvedená.
3. Tento dokument nesmí být distribuován za účelem dosažení zisku.

V publikaci použité názvy produktů, firem apod. mohou být ochrannými známkami nebo registrovanými ochrannými znám­kami příslušných vlastníků.

Copyright (c) 2019, mikroHW, Josef Havlíček

Výrobce: mikroHW, Josef Havlíček

Zrzavého 1085/16, 163 00 Praha 6

www.mikrohw.wz.cz

Technická podpora: mikrohw@email.cz

Obsah

Historie revizí 4

Související dokumentace 4

1. Úvod 5

2. Technické parametry 6

2.1. Rozměry 8

2.2. Doporučená schematická značka 9

3. Napájení 10

4. Rozmístění LED 11

4.1. Rozmístění LED 11

4.2. Konfigurace 11

5. Popis RPi Pico 12

5.1. Krok stranou k Arduinu 14

5.2. Takže proč? 14

6. Programování 16

6.1. Nyní již k vlastní práci s modulem RPi Pico, microPython, pod Windows10 16

7. Programové vybavení 19

7.1. Rozblikání LED 20

8. Rozmístění konektorů 22

9. Montáž 23

9.1. Zásady instalace 23

10. Údržba 24

11. Likvidace odpadu 25

12. Instalace 26

#### Historie revizí

Jméno dokumentu: ESP32\_Install\_Examples\_cz\_100.pdf

Autor: Josef Havlíček

|  |  |  |
| --- | --- | --- |
| Verze | Datum | Změny |
| 100 | 05.02. 2021 | Nový dokument |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

#### Související dokumentace

1. Aplikační poznámka

# Úvod

RPi Pico je embeded deska pro konstrikce s AEM.

##### Základní vlastnosti

* CPU
* Komunikace 1 × USB.

1 × RS232 pro servisní účely.

* Napájení 3V3 ss.
* Montáž na bezkontaktní pole, PCB atd.

Deska je konstruována tak, že v celém teplotním rozsahu pracuje ihned po

zapnutí bez nutnosti temperace.

# Technické parametry

##### CPU

|  |  |
| --- | --- |
| Procesor |  |
| Interní flash |  |

##### Ethernet

|  |  |
| --- | --- |
| Galvanické oddělení |  |
| Pevnost galvanického oddělení |  |
| Přenosová rychlost |  |
| Doporučená kabeláž |  |
| Maximální délka segmentu |  |
| Indikace funkce |  |
| Přípojné místo |  |

###### Poznámka

\*) Galvanické oddělení nesmí být použito pro oddělení nebezpečných napětí.

##### RS485

|  |  |
| --- | --- |
| Galvanické oddělení |  |
| Pevnost galvanického oddělení |  |
| Ochrana proti přepětí |  |
| Přípojné místo |  |
| Připojovací vodiče |  |

###### Poznámka

\*) Galvanické oddělení nesmí být použito pro oddělení nebezpečných napětí.

##### RS232

|  |  |
| --- | --- |
| Galvanické oddělení |  |
| Ochrana proti přepětí |  |
| Přípojné místo |  |

###### Poznámka

\*) Galvanická oddělení nesmí být použita pro oddělení nebezpečných napětí.

##### Mechanika

|  |  |
| --- | --- |
| Mechanické provedení |  |
| Montáž |  |
| Krytí |  |
| Hmotnost |  |
| Rozměry (š × v × h) |  |

##### Napájení

|  |  |
| --- | --- |
| Napájecí napětí |  |
| Maximální odběr \*) |  |
| Špičkový zapínací proud \*\*) |  |

###### Poznámka

\*) Bez zátěže.

\*\*) Doba trvání špičkového zapínacího proudu je kratší než 1 ms.

##### Teploty

|  |  |
| --- | --- |
| Pracovní teplota |  |
| Skladovací teplota |  |

##### Ostatní

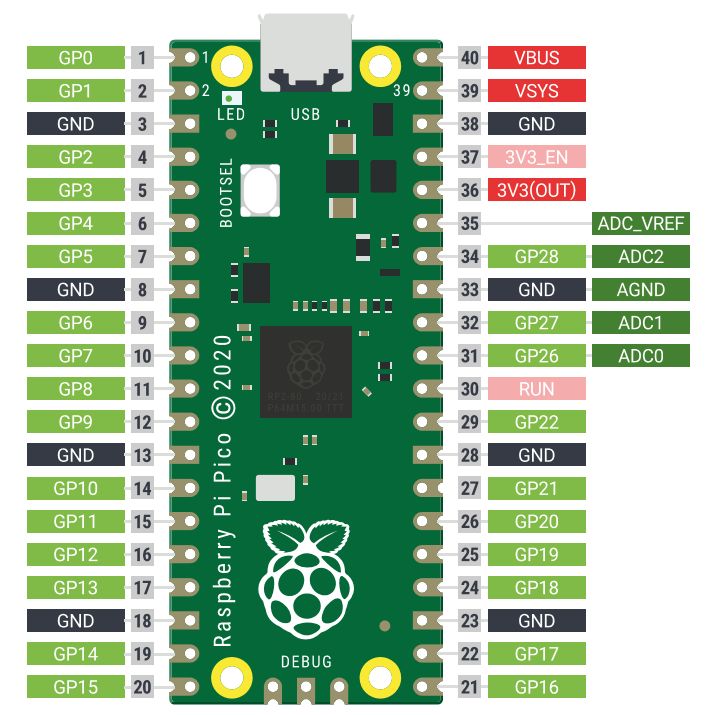
|  |  |
| --- | --- |
| Maximální vlhkost okolí | < 95 % nekondenzující |

##### Požadavky ČSN EN 50155 ed3

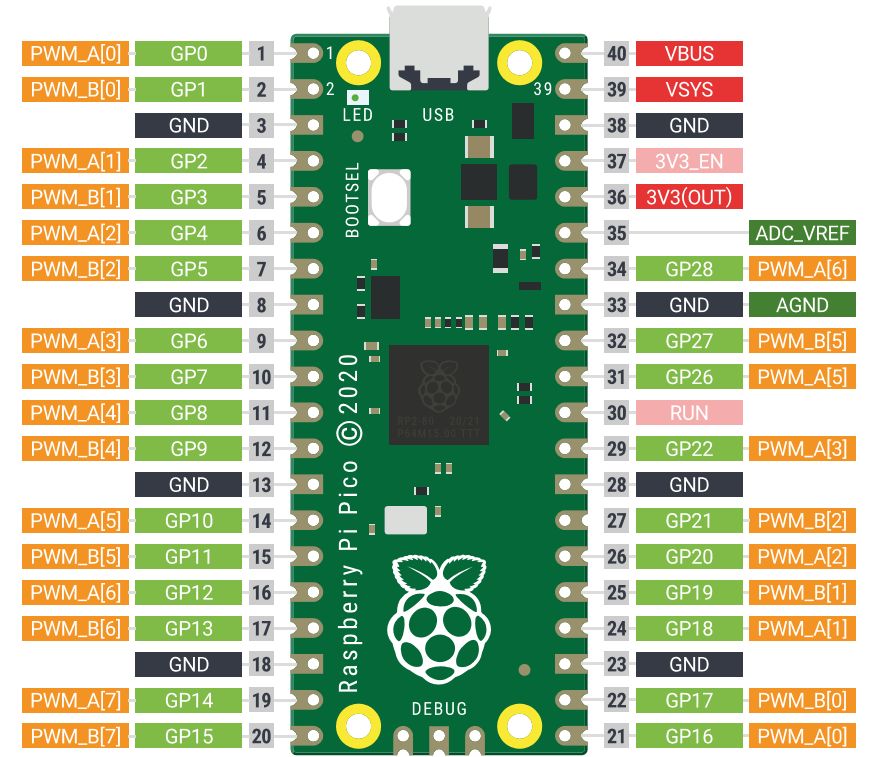
|  |
| --- |
| Zařízení je určeno k provozu do 1400 m n. m. |
| Zařízení je konstruováno dle teplotní třídy TX |
|  |
| U napájecího napětí jsou přípustná přerušení do 10 ms  (Třída S2 dle kap 3.1.1.2. normy ČSN EN 50155 ed.3). |
|  |

##### Klasifikace bezpečnosti

|  |
| --- |
|  |
|  |



1. Pins desky RPu Pico



1. Pins PWM, ADC desky RPu Pico

## Rozměry

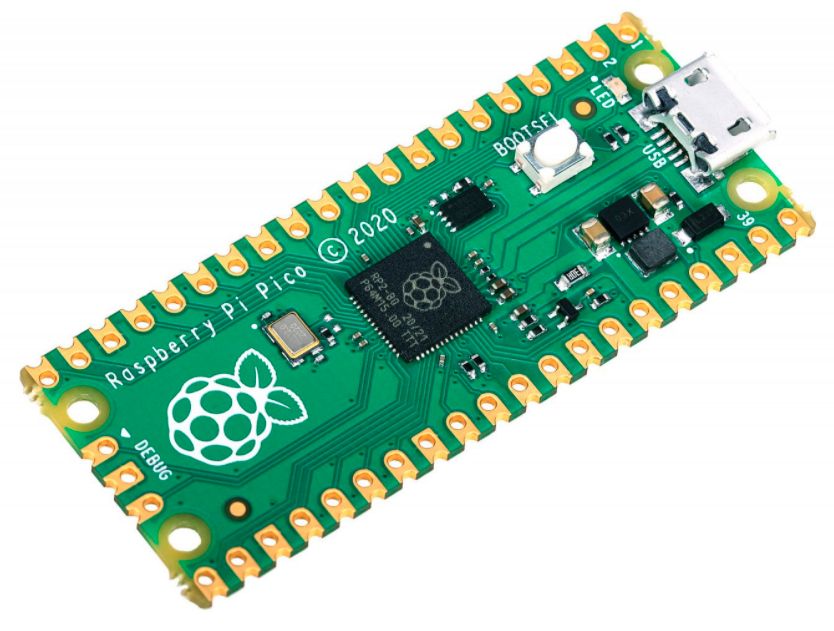
1. Vnější rozměry desky RPu Pico

## Doporučená schematická značka

Doporučená schematická značka pro RPi Pico

# Napájení

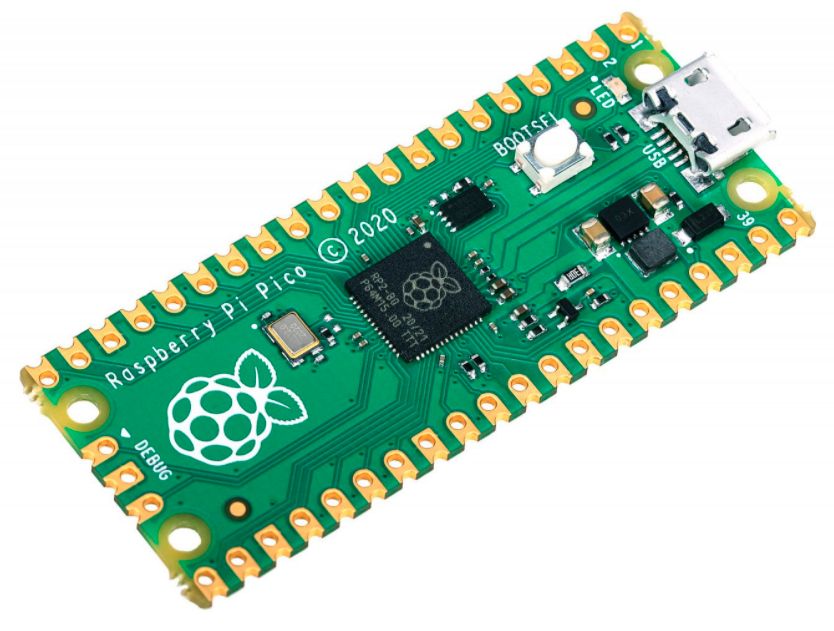
Jednotku **RPi Pico** je možné napájet pouze stejnosměrným napájecím napětím. Jednotka je navržena pro napájení 3 3V ss.



1. Umístění napájecího konektoru

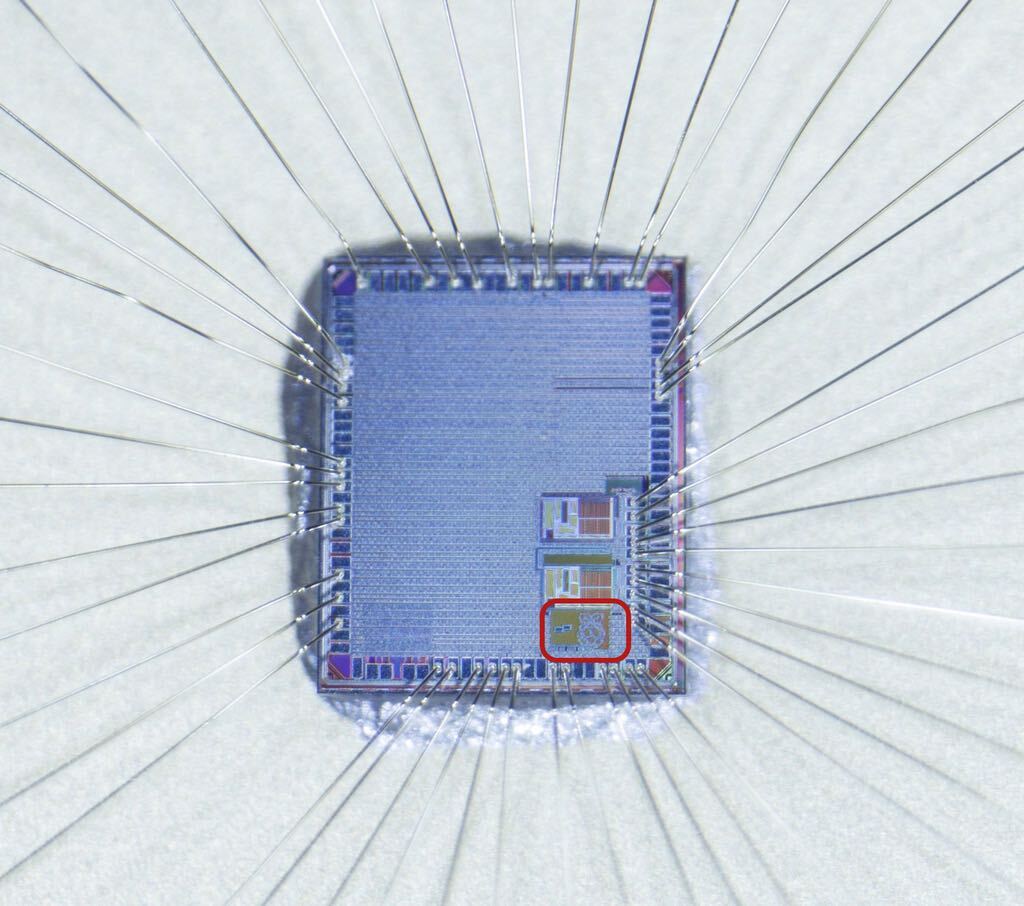
# Rozmístění LED

## Rozmístění LED



1. Rozmístění LED

## https://chiptron.cz/images/news/2021_MISC/RPi_pico2.JPGKonfigurace

Vnitřek čipu☺ :

# Popis RPi Pico

**Pico** je jednočipový počítač založený na mikrokontroléru **RP2040**. Oba, čip i celá deska, jsou od společnosti Raspberry Pi. Stejně jako jeho starší sourozenci z rodiny Raspberry Pi je i Pico snadno přístupný a široce dostupný kus výpočetní techniky, který se hodí ne jen ke vzdělávání, ale i domácím kutilům a do průmyslu.

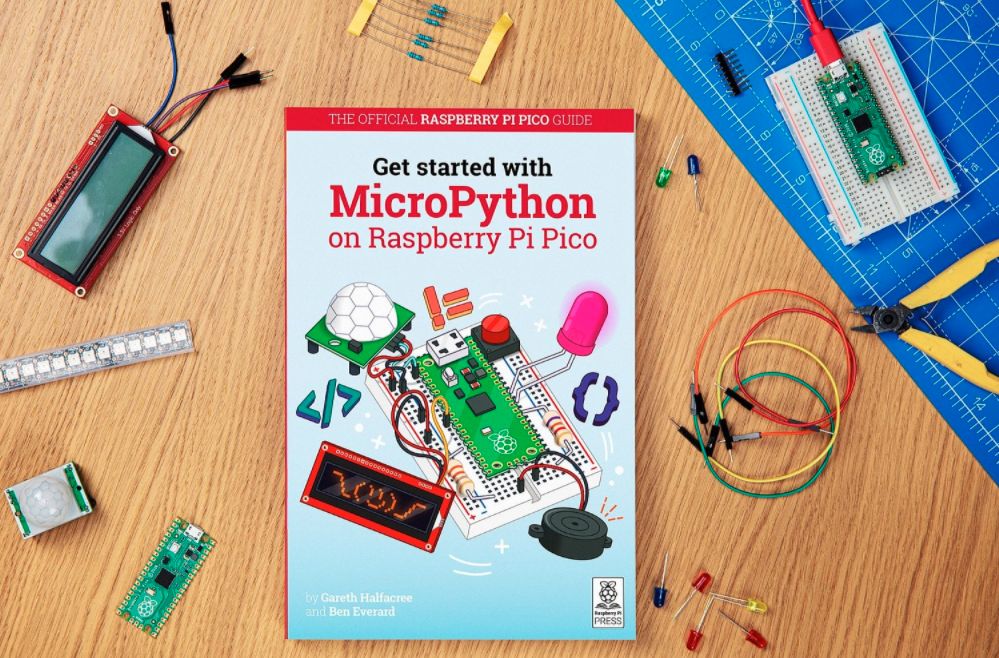
Nadace Raspberry Pi Foundation představila nového člena rodiny svých počítačů. Jedná se o [jednočipový počítač Raspberry Pi Pico](https://www.raspberrypi.org/blog/raspberry-pi-silicon-pico-now-on-sale/), který je určen pro vývoj elektroniky. Na rozdíl od „velkých bratříčků“ má velmi malou spotřebu a je možné se na něm obejít bez operačního systému. Tím se vývojář dostane blíže k hardware a může tak například snížit latenci. Velmi sympatická je i cena, která byla stanovena na čtyři eura.

Pico obsahuje vlastní procesor RP2040, který je vyrobený 40nm technologií, má povrch dva čtvereční milimetry a vešel se do pouzdra QFN-56 o rozměrech 7×7 mm. Najdete v něm:

* dvoujádrový ARM Cortex-M0+ @ 133MHz
* 264 KB RAM
* podporu až 16 MB externí paměti přes QSPI
* řadič DMA
* 30 GPIO pinů(4 mohou být analogovými vstupy)
* 2 × UART, 2 × SPI a 2 × I2C
* 16 × PWM
* 1 × USB 1.1
* 8 × Raspberry Pi Programmable I/O (PIO) state machines
* USB mass-storage boot

Počítač Raspberry Pi Pico nabízí kromě procesoru také 2 MB flashové paměti, zdroj schopný pracovat s napětím 1,8 – 5,5 V, jedno tlačítko, 26 vyvedených pinů včetně tří analogových. Programovat je možné v C nebo je k dispozici podpora pro MicroPython. Začátečníkům je určena kniha [Get Started with MicroPython on Raspberry Pi Pico](https://store.rpipress.cc/products/get-started-with-micropython-on-raspberry-pi-pico).

Pro další práci a připojování je již několik „destiček“ Jednu z nich, pico explorer je na obrázku.





Raspberry Pi Foundation představila nový model Rispberry Pi nazvaný Pico. Opustila svoji doménu jednodeskových linuxových počítačů s výkonným procesorem a pustilo se do vod malých embedded procesorů, na kterých Linux nepustíte.

To by ještě nebylo tak zajímavé. To, zajímavé na tom je, že se rozhodli na toto tažení vytvořit vlastní mikrokontroler — tedy dnes po vzoru Apple M1 bychom měli asi použít termín „own silicon“ — nazvaný **RP2040**.

Když se podíváte na parametry tak to není špatné. Má i několik celkem zajímavých periferií (PIO), má to dvě jádra, celkem hodně RAM a vysokou frekvenci (na to že to je ARM Cortex-M0+).

Obecně se ale nedá říct, že v porovnání s konkurencí jako je STM32, ESP32, SAM a mnoho, mnoho dalších je to něco, co by vám vyrazilo dech. V různých článcích a recenzích najdete často jako argument cenu. Ta je sice nižší, ale ne tak diametrálně.

Pokud by ST představilo novou řadu dvoujádrových ARM Cortex-M0+, tak to pro ně dává smysl a nikoho to moc nepřekvapí. Ale Raspberry Pi? Honila se mi hlavou jediná otázka a nemohl jsem se jí zbavit:

Proč se v Raspberry Pi do něčeho takového pustili?

Opravdu chtějí konkurovat velkým hráčům jako ST, Microchip nebo Espressif. V tom musí být něco, co mi uniká… Žádná z recenzí ani diskusí, co jsem četl, mi to nebyla schopna vysvětlit.

**Odpověď bude v datasheetu**

Večer jsem si otevřel datasheet, že tam bude určitě nějaká killer feature a dostaví se kýžený „aha“ moment. No a po třech hodinách čtení a procházení příkladů (v datasheetu jsou ukázky C-čka a odkazy rovnou na GitHub) stále nic… počkat, **tři hodiny?!**

V tu chvíli, jsem si uvědomil, jak čtivě je ten datasheet napsaný. Jak pěkně je to propojené s příklady, jejich kvalita, SDK a jeho dokumentace, quick start… prostě nádhera to studovat. A ještě jednu věc jsem si uvědomil. Periferie toho procesoru jsou takové přímočaré, snadno pochopitelné.

Mimochodem, všimli jste si, že Pico nemá žádné Wi-Fi nebo Bluetooth? Hodně lidí na to v diskusích nadává (v pořadí důležitosti je hned za malým počtem ADC pinů – proč zrovna tohle je pro tolik diskutujících strašný problém, prostě nechápu). Možná je to kvůli tomu, že RF by to neúnosně prodražilo. Nebo proto, že to je vyšší liga a prostě si na to u svého prvního křemíku netroufli. Ale možná je ten důvod úplně jiný. Wi-Fi i Bluetooth už jsou tak složité, že vysvětlit je stejně lehce jako ostatní části prostě nejde. Tak je raději vypustili.

## Krok stranou k Arduinu

Jedna z věcí, díky které je populární Arduino, je velké zjednodušení pro rychlý začátek. Všechna ta složitost a komplexita je před vámi schovaná šikovně napsanými knihovnami a zvoleným jazykem. Začátečník ani nemusí vědět, co je přerušení…

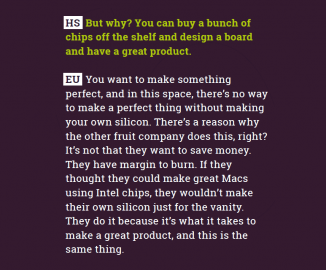
To funguje jen do chvíle, kdy vám dokumentace k Arduinu přestane stačit. To se dřív nebo později stane a vy musíte vzít do ruky datasheet. Najednou se ocitnete v detailní dokumentaci výrobce procesoru, která je na hony vzdálená od čehokoli, co jste viděli ve světě Arduina. Kazí to LX (Learning eXperience — doufám, že něco takového existuje, protože se mi to sem výborně hodí). Je to dané tím, že používají mikrokontroler třetí strany, který byl primárně určen pro profesionální nasazení, ne pro výuku. A těžko s tím jde něco udělat, ledaže…

## Takže proč?

Je třeba si uvědomit, že hlavním cílem Raspberry Pi Foundation není produkovat hardware. To je vlastně jen takový (příjemný) vedlejší efekt.

We engage millions of young people in learning computing and digital making skills. — [About RPi Foundation](https://www.raspberrypi.org/about/)

Když se ptali CEO, proč do něčeho takového šli, tak odpověděl:



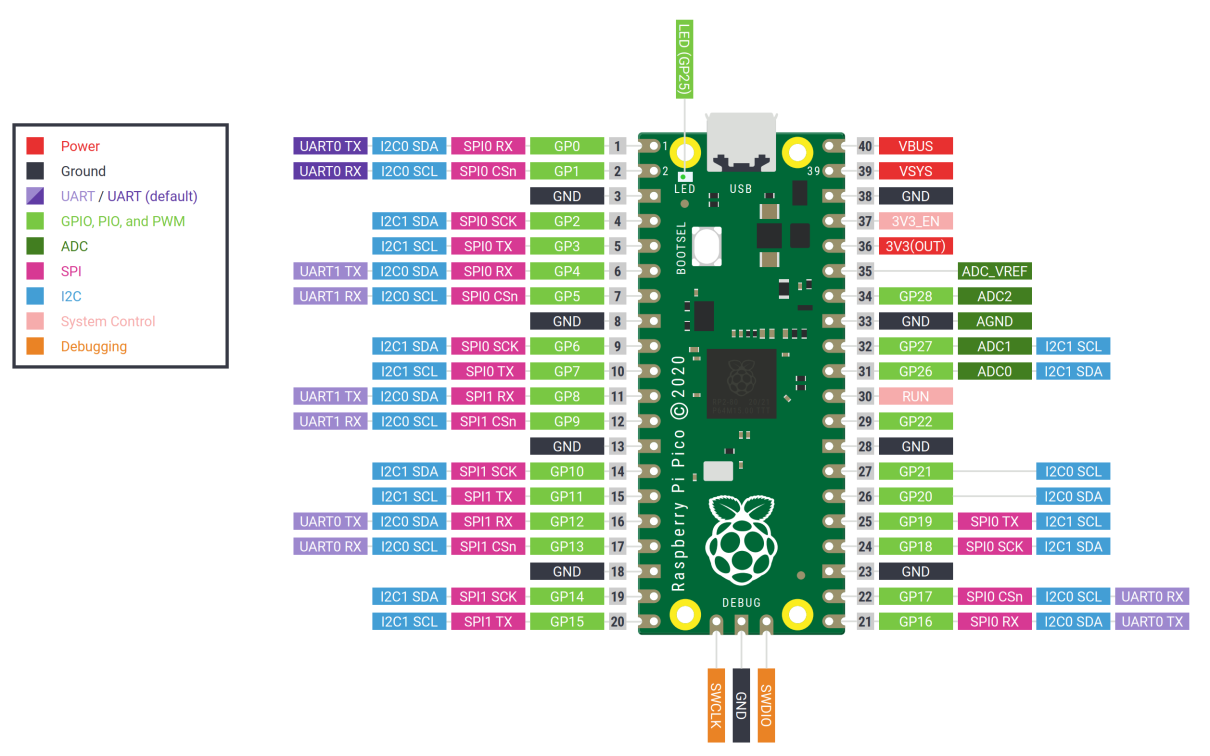
Ten perfektní produkt ale není Raspberry Pi Pico nebo samotný RP2040. Je to celý ekosystém od začátku do konce. Od ilustrované knížky, která vás nadchne do blikání LED v Micro Pythonu, přes čtivý datasheet, zdokumentované C-čkové SDK, až po pocit skutečného HW guru při programování PIO v assembleru (tedy až do chvíle, než vám někdo řekne, že existuje FPGA, ale pokud vám to dá odvahu zjistit, co to je, cíl splněn 👍).

Díval jsem se na RP2040 očima nadšeného elektroinženýra z nového mikrokontroleru, který okamžitě začne porovnávat velkost paměti, rychlost, počet periferií atd., a unikal mi celkový obrázek.

Takže pokud si říkáte, jako já na začátku, proč použít RP2040 místo ESP32 (doplňte svůj oblíbený mikrokontroler), tak pro vás mám odpověď, na kterou jsem nakonec narazil na Redditu, a která to dokonale vystihuje:

*If you’re looking at this and thinking an ESP32 or STM32 is a better choice then it probably is for you. For kids though, having a bundle of your own hardware and firmware makes writing your own teaching resources a lot easier. This is still in the Foundations remit as an educational tool, this isn’t an attempt to take on Microchip in the microcontroller market.*

*(Původně vyšlo na*[*Medium.com*](https://horcicaa.medium.com/raspberry-si-rp2040-pro%C4%8D-eac58d93626e)*)*



# Programování

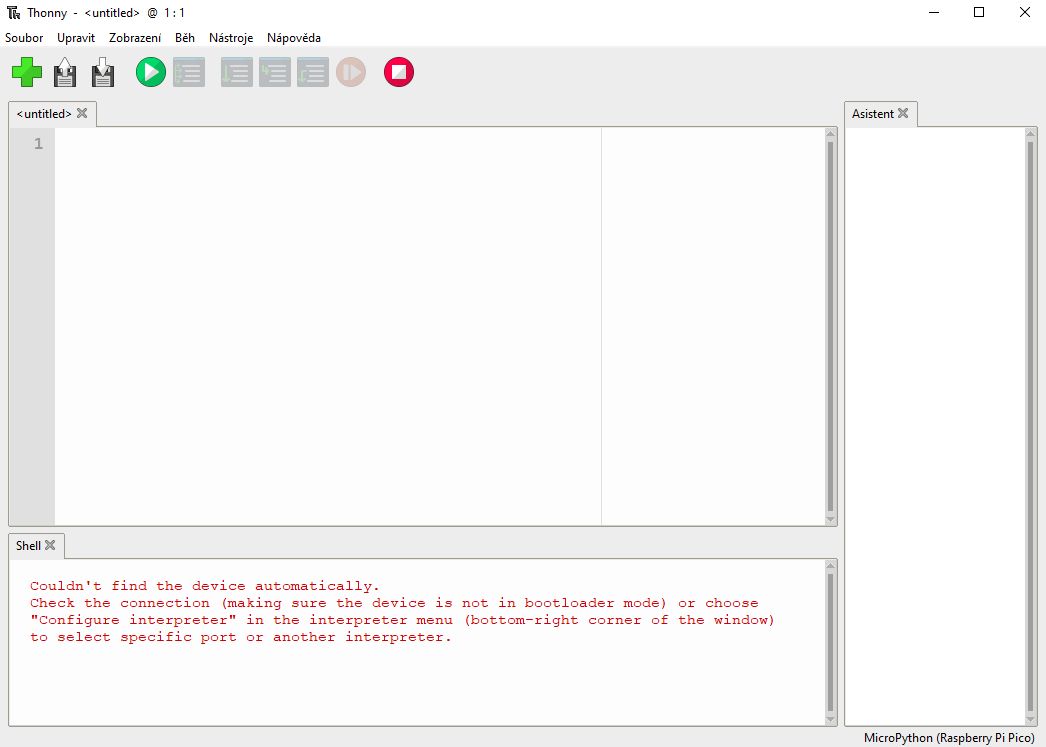
## Nyní již k vlastní práci s modulem RPi Pico, microPython, pod Windows10

Velice vhodným SW pro rychlou práci je **microPython interpreter.**

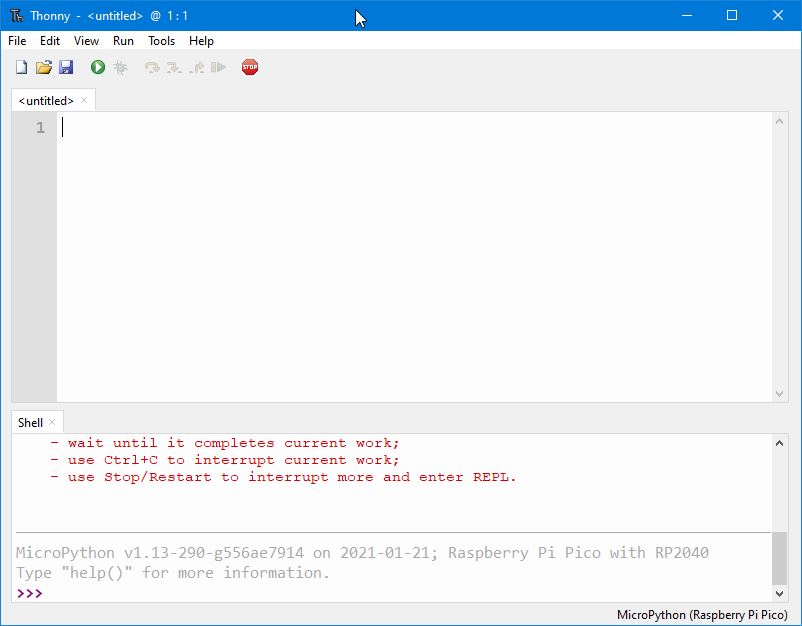
Chcete-li hrát s Raspberry Pi Pico ve Windows, budete si muset 1 SW stáhnout. Nadace doporučuje Thonny ( [Windows](https://github.com/thonny/thonny/releases/download/v3.3.3/thonny-3.3.3.exe) ) jako svůj IDE, protože je dodáván se vším, co potřebujete (včetně Pythonu 3.7), ale můžete si také nainstalovat nejnovější verzi [Pythonu 3.x](https://www.python.org/downloads/) jako instalační program pro celý systém v případě, že potřebujete nějaké doplňky pro vaše projekty.

**Pozn.:** Já jsem neinstaloval poslední verzi Pythonu 3.9.1 V počítači mám nainstalované verze 2.7 a 3.7.4. SW **Thonny** má v sobě **Python 3.7.9.**

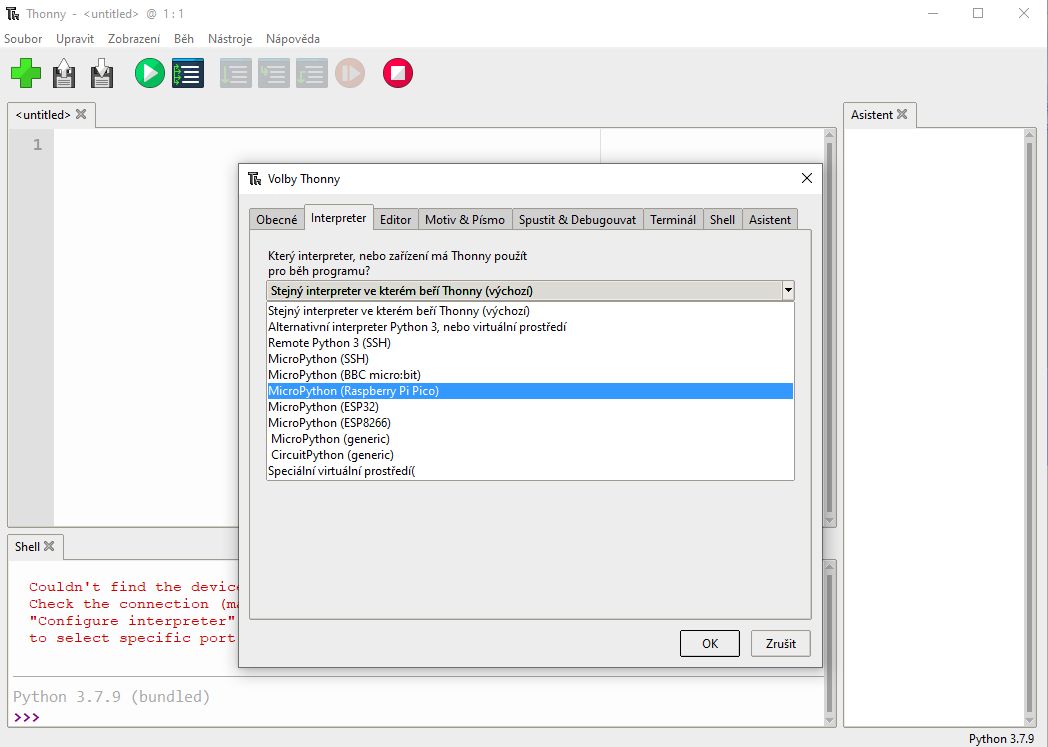
Po první instalaci a volbě – Čeština (Beta) – a dále: **Stejný interpretr ve kterém běží Thony,** se zobrazí následující obrázek, česky:

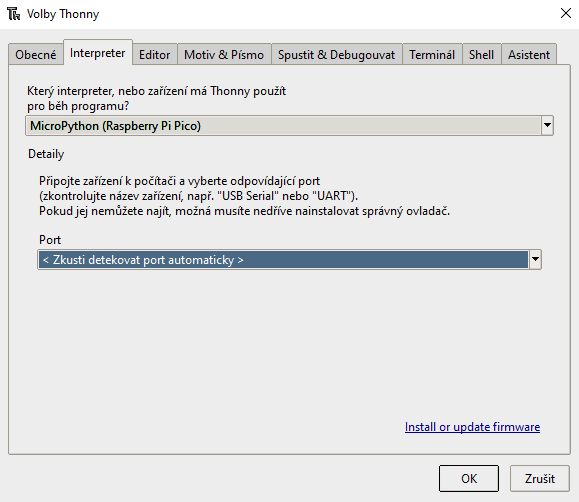


nebo Anglicky:



Otevřete Thonny a klikněte v pravém dolním rohu na nápis **python 3.7.9** a vyberte **Configure Interpreter…** Po otevření nového okna, v horním poli vyberte volbu: [MicroPython (Raspberry Pi Pico)](file:///C:\Users\mzoln\Desktop\RPIPICO\2021-01-23%2013_40_41-Thonny%20%20-%20%20_untitled_%20%20@%20%2052%20_%2026.jpg)

Okno se změní na:



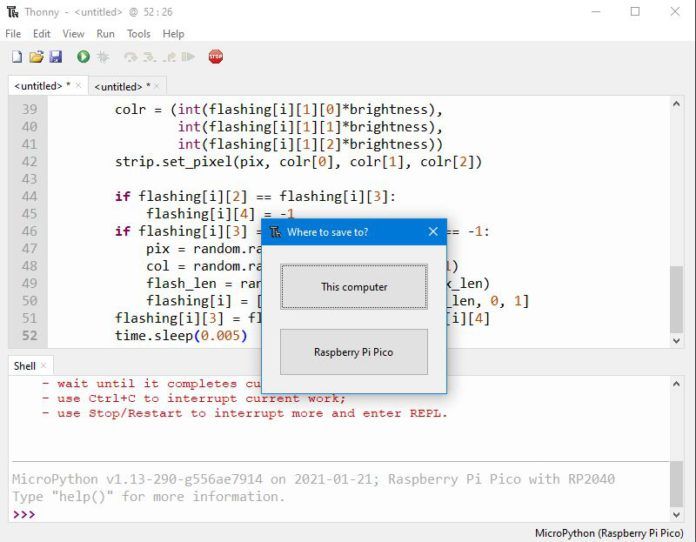
Konfiguraci portu můžete nechat automatickou. Poté klikněte na „ **Instalovat nebo aktualizovat firmware** “ a poté klikněte na OK.

**Postupujte podle pokynů v nově otevřeném okně!**

Podle pokynů musíte zapojit kabel USB a současně držet tlačítko na Raspberry Pi Pico.

Po ukončení instalace = **Done,** microPython do desky klikněte na tlačítko **OK.**

Napište nebo nahrajte nějaký program v microPythonu do Editoru. Kliknutím na tlačítko **Spustit** (zelené tlačítko přehrávání) ověřte, zda se můžete připojit k desce. Pokud vše proběhlo dobře, měli byste vidět něco takového:



Zvolíte si kam chcete program v microPythonu nahrát. Zda jej uložit do počítače – horní volba nebo jej nahrát do desky RPi Pico – spodní volba.

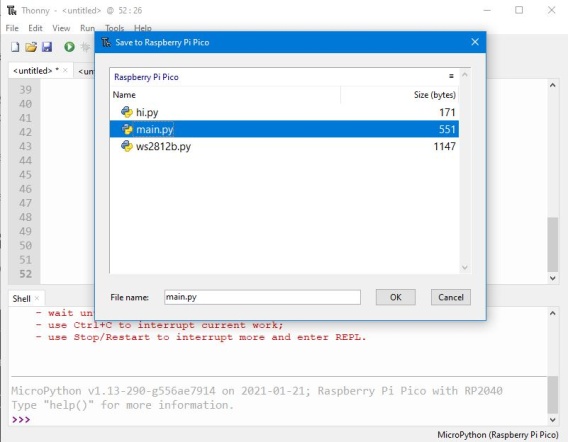
Jste připraveni programovat pomocí MicroPythonu. Jakákoli aplikace napsaná v tomto okně bude zpracována Raspberry Pi Pico a výsledky budou vráceny v okně shellu níže. Nezapomeňte uložit soubory.

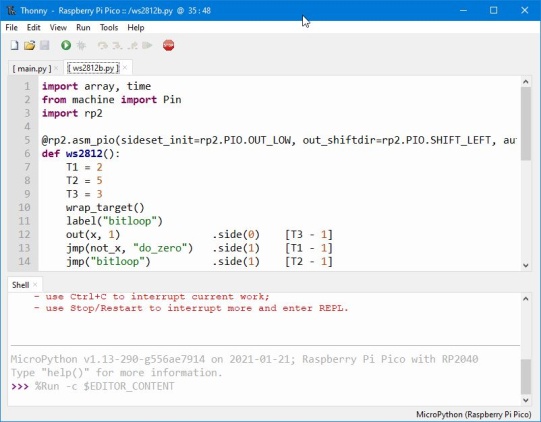
Uložení lokálně do svého PC, ale pro jejich následné spuštění byste je měli uložit jako:

JmenoSouboru.py.

I když můžete mít uloženo tolik souborů, kolik chcete, abyste je mohli spustit bez počítače, Raspberry Pi Pico musí vědět, kde začít. Soubor, který bude spuštěn při bootování, by měl být uložen jako . **main.py**Pokud tak neučiníte, váš skript se nespustí, když je připojen k napájecímu zdroji.

Pokud váš projekt obsahuje externí knihovny, musíte otevřít soubory a uložit je také na Raspberry Pi Pico. Zde je ukázka off LED knihovny z [benevpi / pico\_python\_ws2812b](https://github.com/benevpi/pico_python_ws2812b) :





A jste připraveni programovat s MicroPhyton a Raspberry Pi Pico!

# Programové vybavení

Programové vybavení jednotky RPi Pico je proměnné.

N **Názorně si tu ukážeme, jak Pico oživit, jak se do něj připojit, a jak rozblikat vestavěnou LED kontrolku**

**Programovat s jiným SW!**

**Instalace MicroPython**

**MicroPython** je programovací jazyk napsaný v jazyce C pro mikrokontroléry jako je čip v našem Raspberry Pi Pico. Používá stejná pravidla (syntaxi) jako jazyk Python, což je pro první krůčky velká výhoda.

MicroPython si nainstalujeme tak, že Pico připojíte k počítači přes USB a přetáhnete na něj soubor UFL s binárním kódem MicroPython.

* Stáhneme si soubor UF2 s MicroPythonem.
* Stiskeme a držíme tlačítko BOOTSEL. Tlačítko BOOTSEL přepnete Pico do režimu USB Mass Storage.
* Pomocí micro-USB kabelu připojíme Pico k počítači.
* Až se Pico objeví jako disk připojený k počítači, můžeme tlačítko BOOTSEL pustit.
* Připojený disk bude pojmenovaný RPI-RP2.
* Přetáhneme stažený soubor UF2 s MicroPythonem na svazek RPI-RP2.
* Pico se restartuje a spustí MicroPython.

[Stáhnout MicroPython UF2](http://blog.zonepi.cz/downloads/rp2-pico-20210202-v1.14.uf2)

**Připojení k textovému rozhraní**

Jakmile máme na Picu nainstalovaný MicroPython, můžeme se k němu připojit z řídícího počítače dvěma způsoby: přes USB, a když používáme Raspberry Pi, tak i přes GPIO.

Pro komunikaci přes USB si na řídící počítač potřebujeme nainstalujte program minicom:

$ sudo apt install minicom

a otevřít sériový port:

$ minicom -b 115200 -o -D /dev/ttyACM0

Po připojení Pico přes USB se dostaneme do textové rozhraní MicroPython, tzv. REPL:

MicroPython v1.12-725-g315e7f50c-dirty on 2020-08-21; Raspberry Pi PICO with cortex-m0plus

Type "help()" for more information.

>>>

Když v minicom stiskneme CTRL-A a následně U, přidá se do sériového výstupu odřádkování, takže každý výpis začne na nové řádku. Pro ukončení minicom slouží kombinace CTRL-A a pak X.

Ve výjimečných případech, kdy se nemůžeme k Raspberry Pi Pico připojit budeme možná muset napřed restartovat Raspberry Pi.

**Hello World**

Po připojení můžeme zkontrolovat, že vše funguje, zadáním jednoduchého „Hello World“ do REPL:

>>> print('hello pico!')

hello pico!

>>>

## Rozblikání LED

Integrovaná LED dioda na Raspberry Pi Pico je napojená na výstup GPIO č. 25. Můžeme jí rozsvítit a zhasnout zadáním správných příkazů do REPL:

>>> from machine import Pin

>>> led = Pin(25, Pin.OUT)

Dosazením 1 se LED zapne:

>>> led.value(1)

a dosazením 0 se LED zase vypne:

>>> led.value(0)

Alternativně můžeme k rozblikání LED použít vestavěný časovač:

1 from machine import Pin, Timer

2

3 led = Pin(25, Pin.OUT)

4 tim = Timer()

5 def tick(timer):

6 global led

7 led.toggle()

8

9 tim.init(freq=2.5, mode=Timer.PERIODIC, callback=tick)

**Dokumentace:**

<https://datasheets.raspberrypi.org/pico/raspberry-pi-pico-python-sdk.pdf>

# Rozmístění konektorů

1. Rozmístění konektorů

# Montáž

Jednotka RPi Pico určena k namontování na základovou desku.

1. Montážní otvory

## Zásady instalace

##### Konektory USB

Připojovací konektor USB je nutno po zasunutí zajistit proti vytažení.

akončení linek RS485 je z výroby zapnuto.

# Údržba

##### Upozornění

**Uvedenou údržbu může provádět pouze výrobce nebo po­vě­ře­ná autorizovaná firma**

##### Čištění

Podle způsobu použití zařízení je třeba čas od času z něj odstranit prach. Deska se čistí ve vypnutém a roze­bra­ném stavu suchým štětcem nebo jemným kartáčem případně vysavačem.

# Likvidace odpadu

##### Likvidace elektroniky

Likvidace zařízení je řízena předpisy o nakládání s elektroodpadem. Zařízení nesmí být likvidováno v běžném komunálním odpadu. Musí být odevzdáno na místech k tomu určených a recyklováno.

# Instalace