RPi Pico

Verze 1.00

Informace

RPi Pico Cčko 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 5

Související dokumentace 5

1. Úvod 6

2. Technické parametry 7

2.1. Rozměry 9

2.2. Doporučená schematická značka 9

3. Napájení 10

4. Rozmístění LED 11

4.1. Rozmístění LED 11

4.2. Konfigurace 11

5. Programové vybavení 12

6. Rozmístění konektorů 13

7. Montáž 14

7.1. Zásady instalace 14

8. Údržba 15

9. Likvidace odpadu 16

10. !! Instalace pomocí samoinstalačního programu !! 17

10.1. !!! Použijte tuto část, další části si můžete přečíst !!! 17

10.2. Nejdříve instalace VS Code 17

10.2.1 Mejdříve je NUTNÉ si stáhnout a nainstalovat program VSCode pro Windows !!! 17

10.3. Upravit Proměnné Prostředí 18

10.4. Nakopírování adresářů pico-examples a pico-sdk 19

10.5. Nainstalovat všechny potřebné programy a úpravy 19

10.6. „Naprogramování“ RPi Pico 27

10.7. Sestavení vlastního projektu 28

10.7.1 Umístění vlastního projektu 28

10.7.2 Obsah CMakeLists.txt 29

11. Instalace 31

11.1. RPi Pico programovani v Ccku – Windows 31

11.1.1 Ad 1) Instalace GCC: 31

11.1.2 Ad 2) Instalace CMake: 32

11.1.3 Ad 3) Instalace Build Tools for Visual Studio 2019: 33

11.1.4 Ad 4) Instalace Windows 10 SDK: 34

11.1.5 Ad 5) InstalacePathon 3.7: 35

11.1.6 Ad 6) InstalaceGit: 36

12. Provést před začátkem práce s SDK a Examples v CMD (spuštěném jako Správce) 38

12.1. Instalace VSCode 38

13. Start s C/C++ a Raspberry Pi Pico 40

13.1. Getting started with C/C++ and Raspberry Pi Pico 40

13.2. Finále 54

#### Historie revizí

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

Autor: Josef Havlíček

|  |  |  |
| --- | --- | --- |
| Verze | Datum | Změny |
| 100 | 11.02. 2021 | Nový dokument |
| 100 | 17.02.2021 | Vložení Automatické instalace programů – oddíl 10 |
|  |  |  |
|  |  |  |
|  |  |  |

#### 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

|  |
| --- |
|  |
|  |

## Rozměry

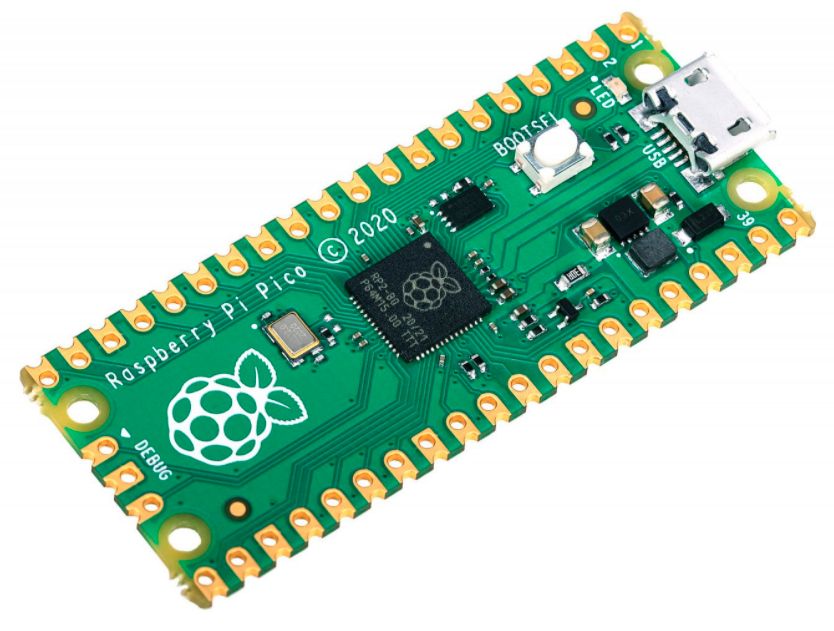
1. Vnější rozměry desky RPu Pico

## 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.



5V

GND

1. Umístění napájecího konektoru

# Rozmístění LED

## Rozmístění LED

1. Rozmístění LED

## Konfigurace

# Programové vybavení

Programové vybavení jednotky RPi Pico je proměnné.

# 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.

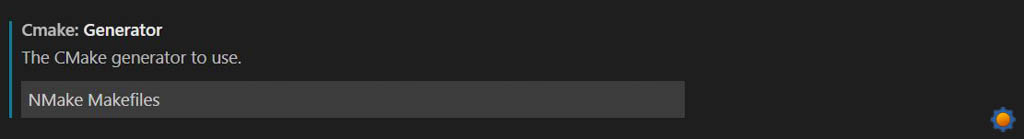
# !! Instalace pomocí samoinstalačního programu !!

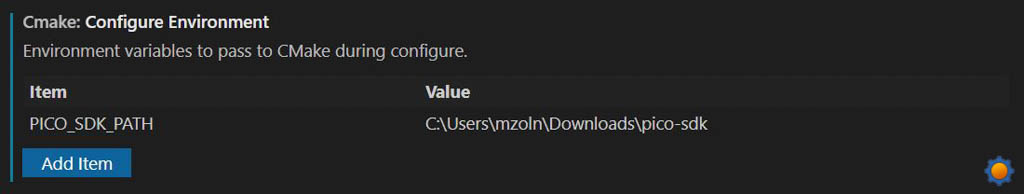
## !!! Použijte tuto část, další části si můžete přečíst !!!

## Nejdříve instalace VS Code

### Mejdříve je NUTNÉ si stáhnout a nainstalovat program VSCode pro Windows !!!

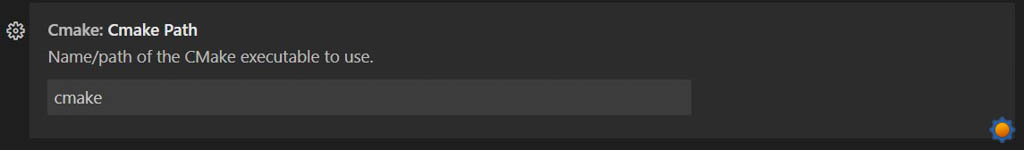
* Po nainstalování spusťte program VSCode
* V levé části zvolte přidání instalačních souborů **C/C++**  a **CMake Tools**, oboje pro **Microsoft**
* Po přídání stisknutím PTM (Pravé Tlačítko Mouse) zvolte **Setting**
* Uprostřed se zobrazí okno s možností nastavení prostředí **CMake Extensions**
* Zde najděte následující 3 volby (postupnýmpostumem dolů a nastavte jak je ukázáno





* V položce **Value** zadejte správnou cestu. Já zapř. Zadal:

C:\Users\Admin\Downloads\pico-sdk



* Ukončete **křížkem Settings**
* Proveďte vypnurí, ukončení VSCode

## Upravit Proměnné Prostředí

* Na ikoně **Tento počítač** (jak přidat na plochu je popsáno v části 13.1. pomocí PTM otevřete nabídku a zvolte **Vlastnosti.**
* Otevře se nové okno a vlevo zvolte nabídku **Upřesnit nastavení systému**
* Otevře se nové okno, kde zvolte nabídku **Proměnné prostředí**
* Otevře se nové okno, kde zvolte nabídky **Nová…** (pro PICO\_SDK\_PATH)a **Upravit…** (pro označenou **Proměnná Path**)
* Vložte následující:

|  |  |
| --- | --- |
| Proměnná | Hodnota |
| PICO\_SDK\_PATH | C:\Users\Pico\Downloads\pico-sdk |
| Path | C:\Users\mzoln\AppData\Local\Programs\Microsoft VS Code\bin |

V Hodnota jsem zadal:

C:\Users\Admin\AppData\Local\Programs\Microsoft VS Code\bin

* Pro  **Systémové proměnné** si označte  **Proměnnou Path** a zvolte tlačítko **Upravit…**
* Vložte jako **Nová** následující:

|  |  |
| --- | --- |
| **Systémové proměnné** | **Hodnota** |
| Path | C:\Program Files\Python39\ |
| Path | C:\Program Files\Python39\Scripts\ |
| Path | C:\Program Files\CMake\bin |
| Path | C:\Program Files (x86)\GNU Arm Embedded Toolchain\10 2020-q4-major\bin |
| Path | C:\Users\mzoln\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Visual Studio Code |

* C:\Program Files (x86)\GNU Arm Embedded Toolchain\10 2020-q4-major\bin

Já jsem vložil svojí cestu:

C:\Users\Admin\AppData\Local\Programs\Microsoft VS Code

* Vše potvrdit tlačítky  **OK**

## Nakopírování adresářů pico-examples a pico-sdk

Nakopírujte potřebné adresáře z Webu nebo z mé stránky do adresáře – C:/Users/Pico/Admin/Downloads/:

C: …/Admin/… je moje označení, změňte podle svého pojmenování počítače !

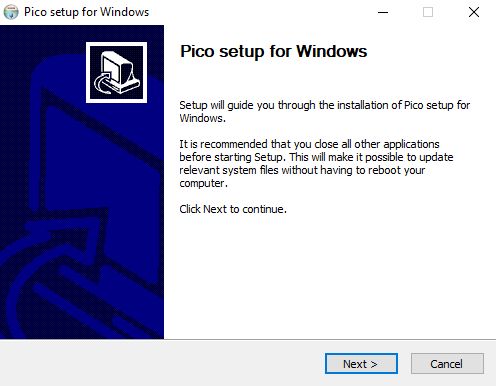
[http:/www.mikroHW.wz.cz](http://www.mikroHW.wz.cz/)

## Nainstalovat všechny potřebné programy a úpravy

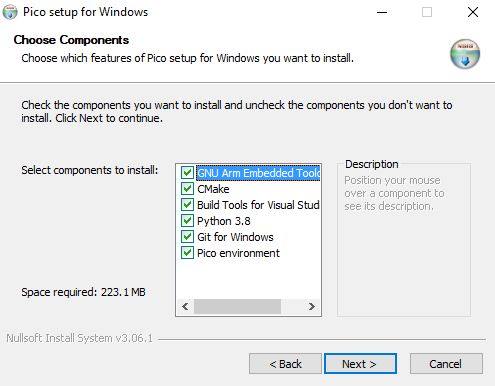
Spusťte program **Setup-Pico-Win-x64.exe.** Stažený z mých Web stránek:

[http:/www.mikroHW.wz.cz](http://www.mikroHW.wz.cz/)

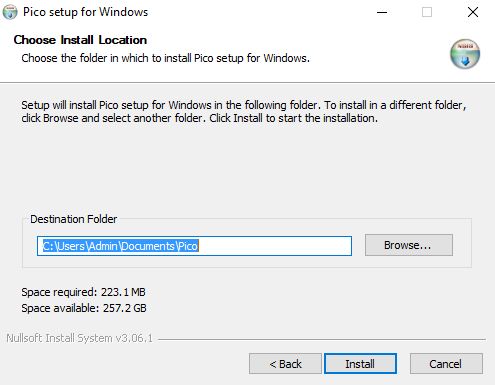
Program spusťte:



Potvrďte  **Next:**

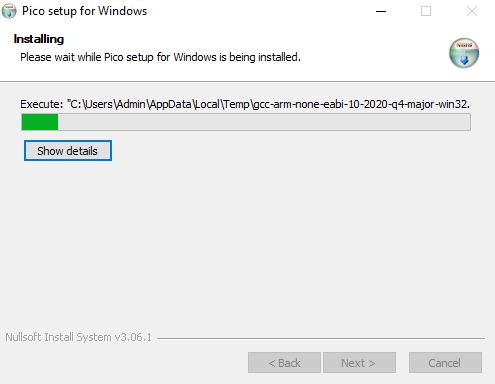


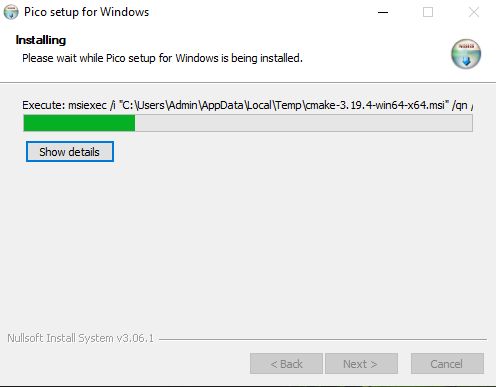
Ponechte **všechny zatržené volby** a stiskněte **Next:**

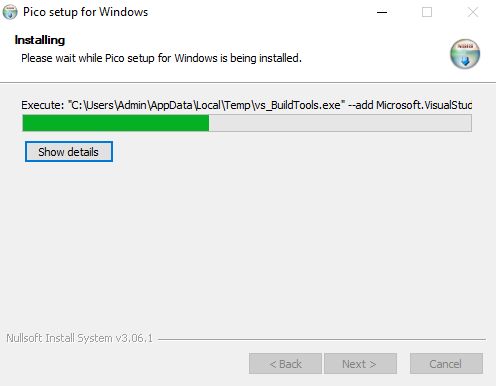


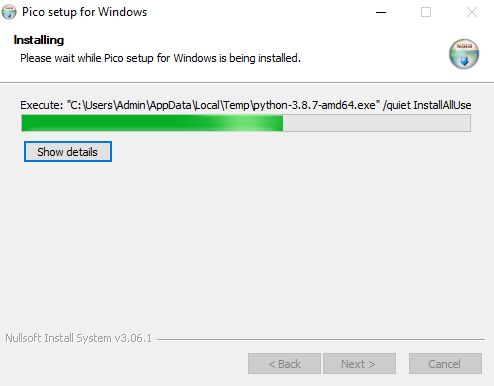
Ponechal jsem **navrženou cestu** pro instalaci a stiskněte **Next.**

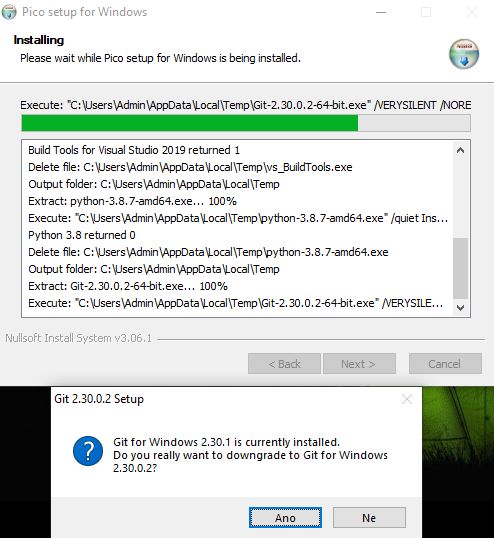
Probíhá instalace:



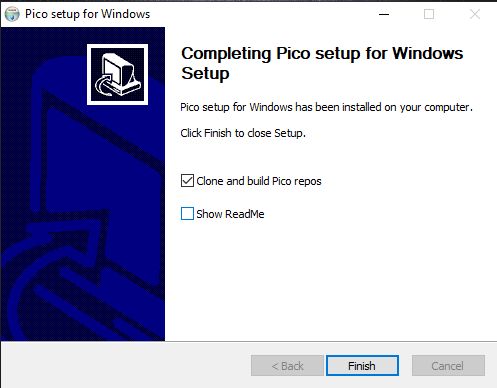




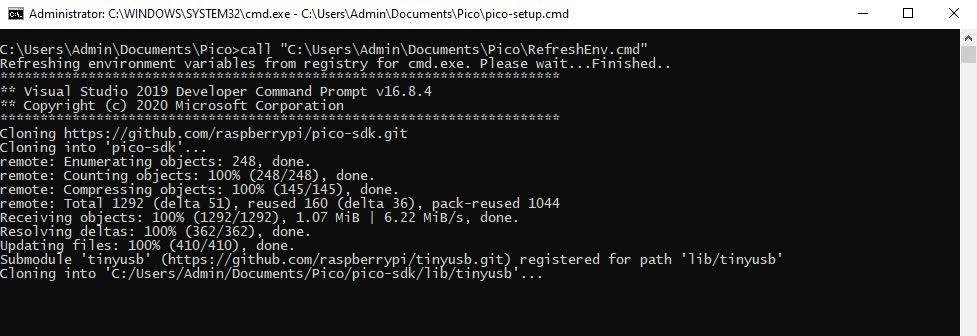


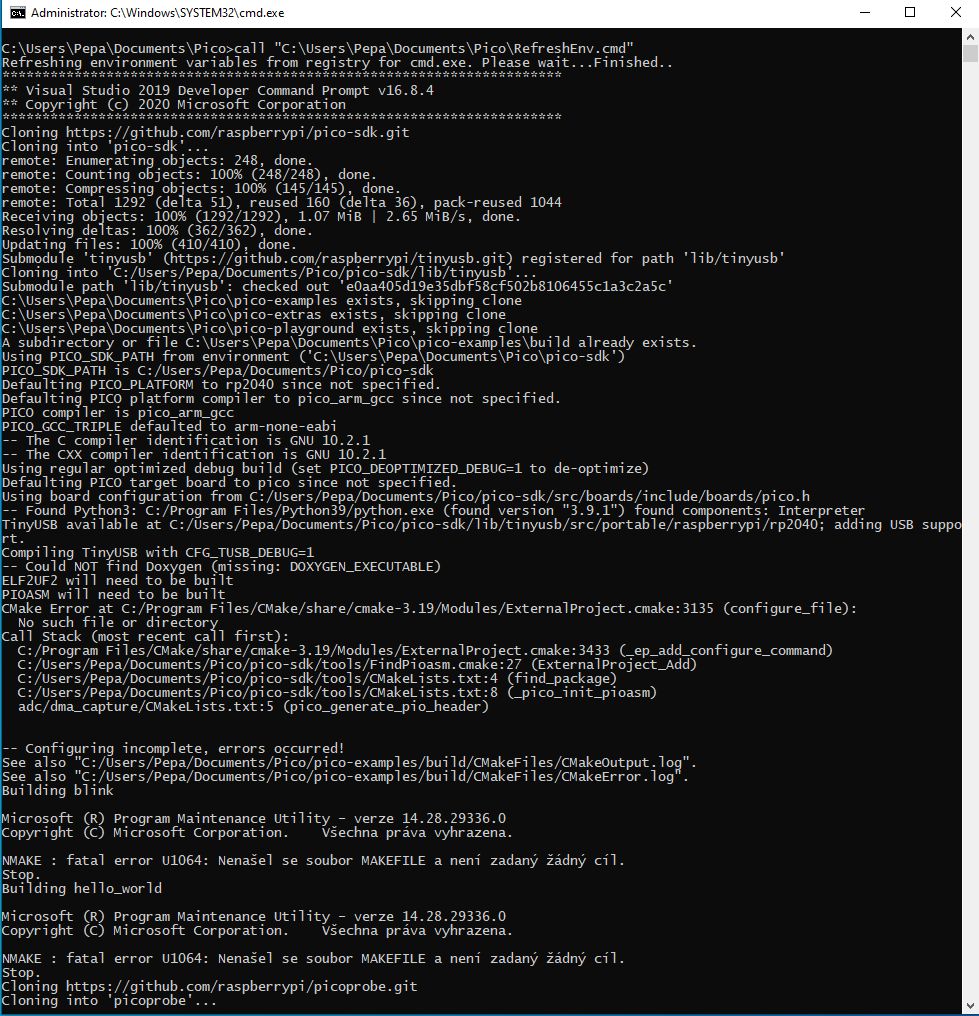
Během instalace jste požádáni o potvrzení, stiskněte tlačítko **Ano.**

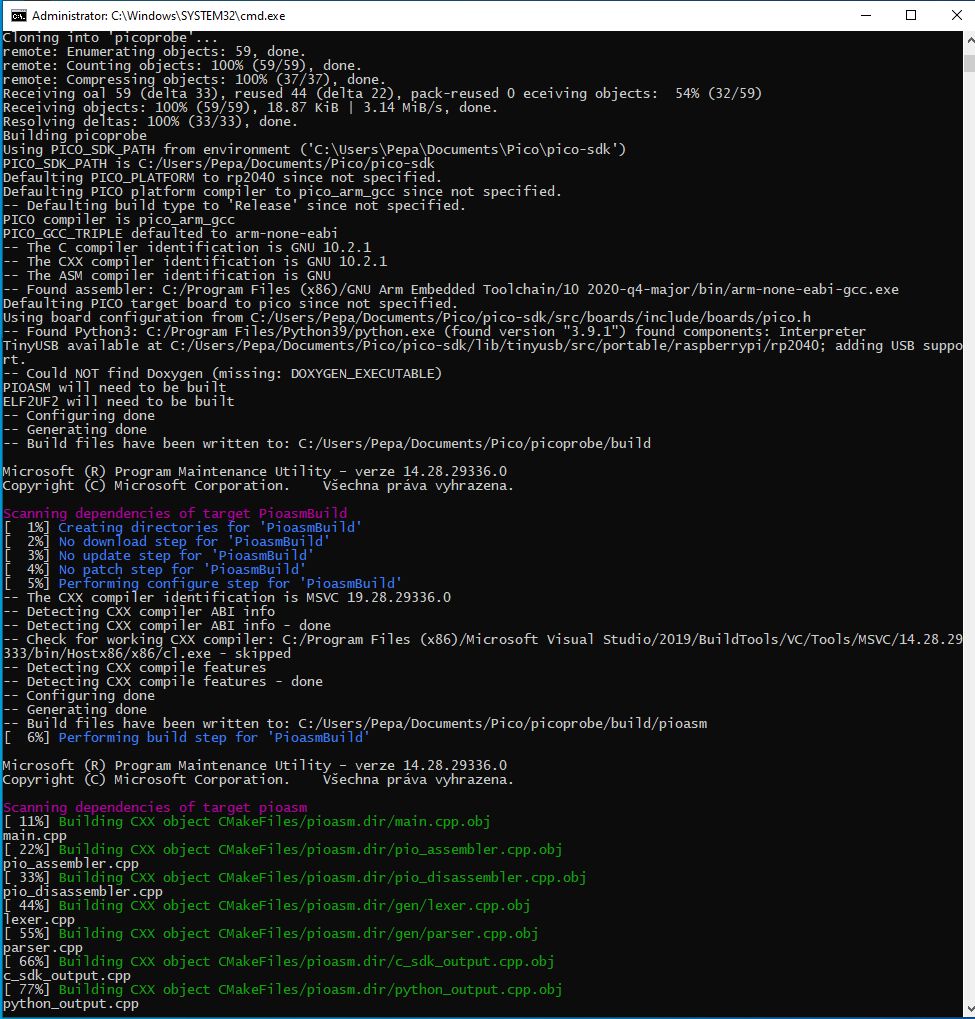
Můžete púonechast zaškrtnutý box **Show ReadMe**, aler v každém případě ponechte zaškrtnutý box **Clone and build Pico repos !!!**

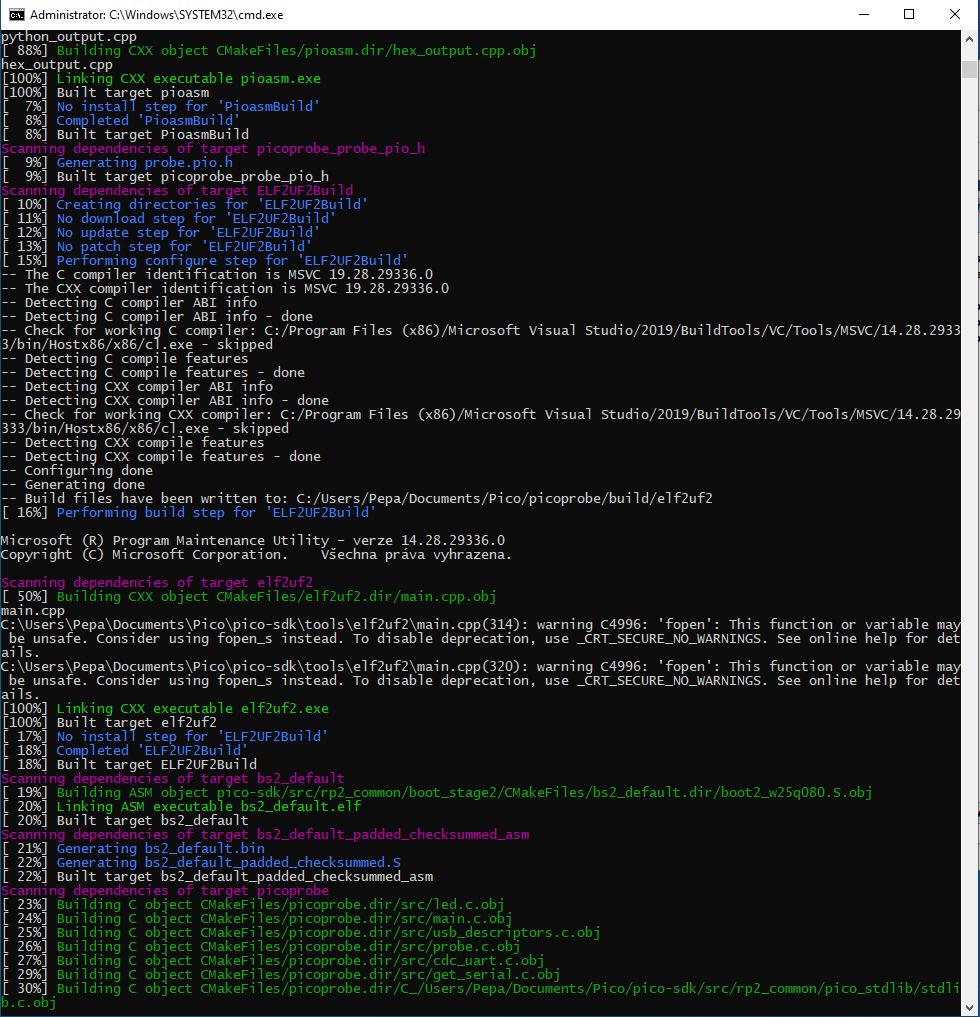


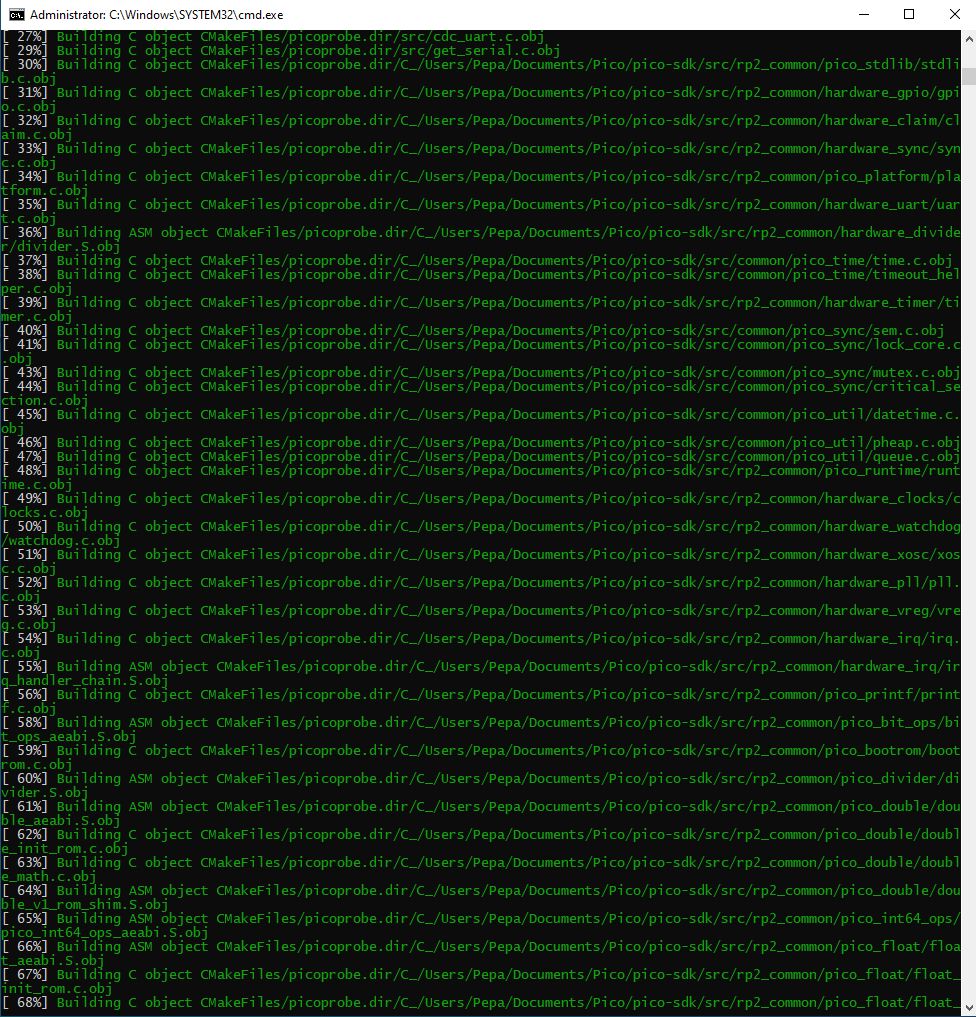
Spustí se okno **CMD Line (jako Administrator)** a poběží docela dlouhá instalace.

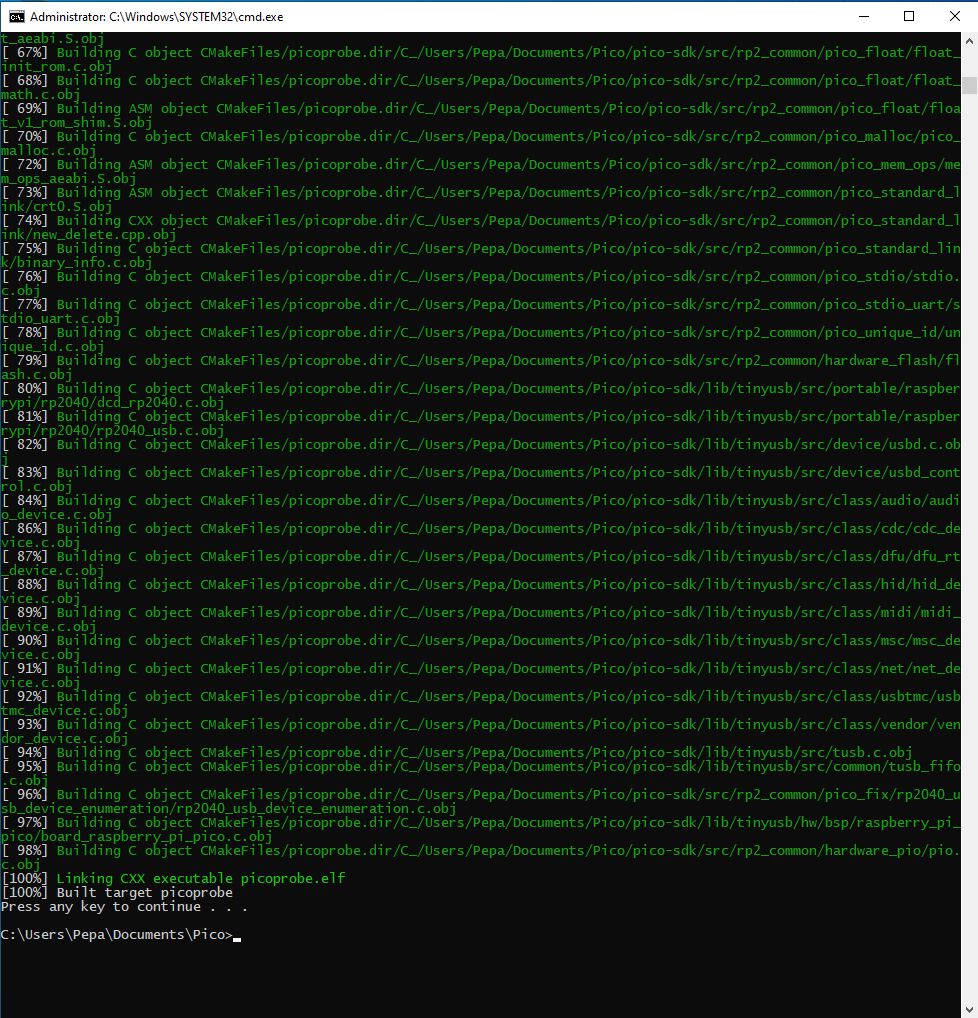




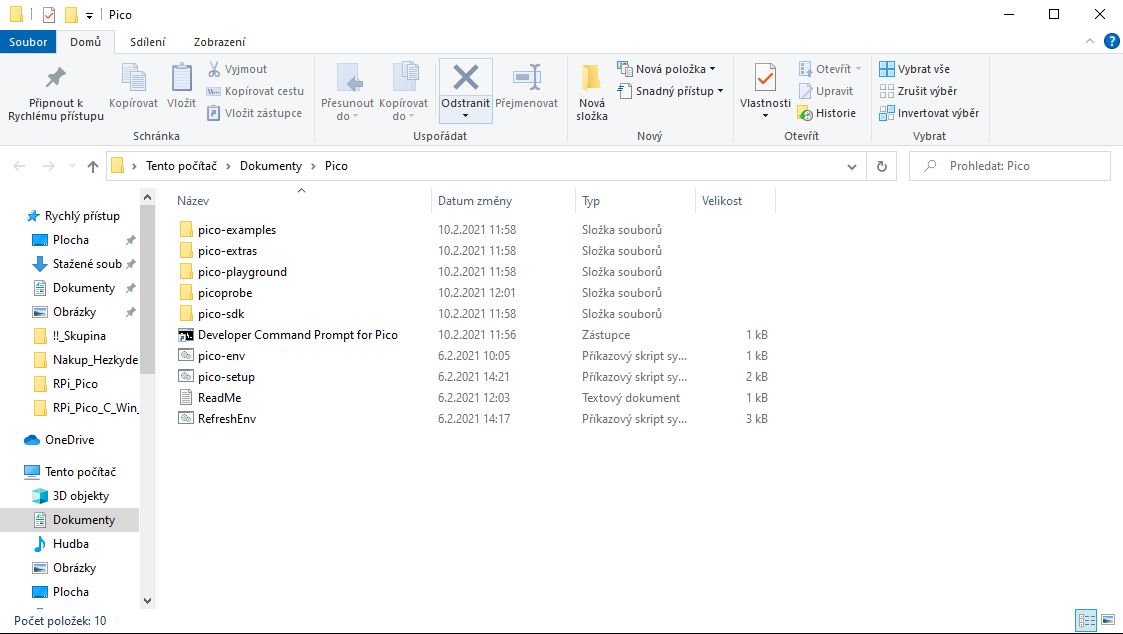




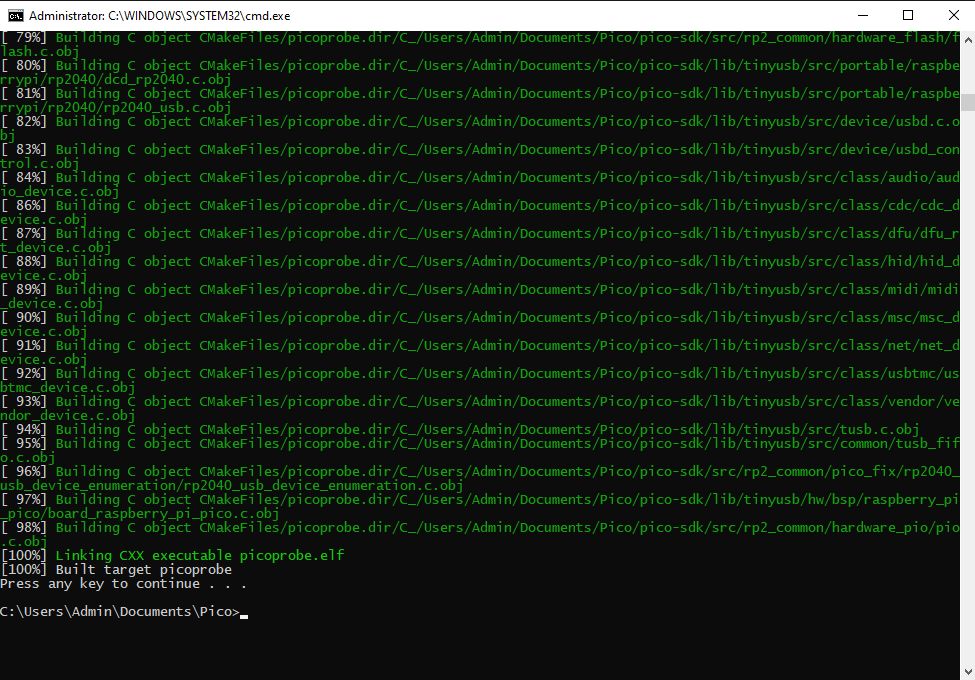




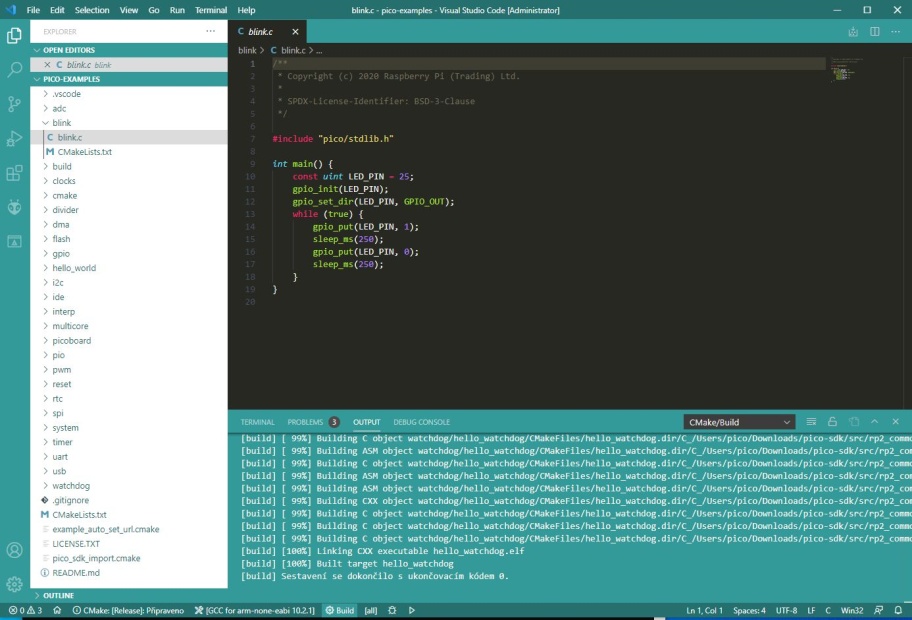
K závěru instalace se zobrazí nové okno, které můžete zavčít nebo ponechat otevřené. Podle toto s čím pracujete.



Na úplný závěr instalace se zobrazí následující informace:

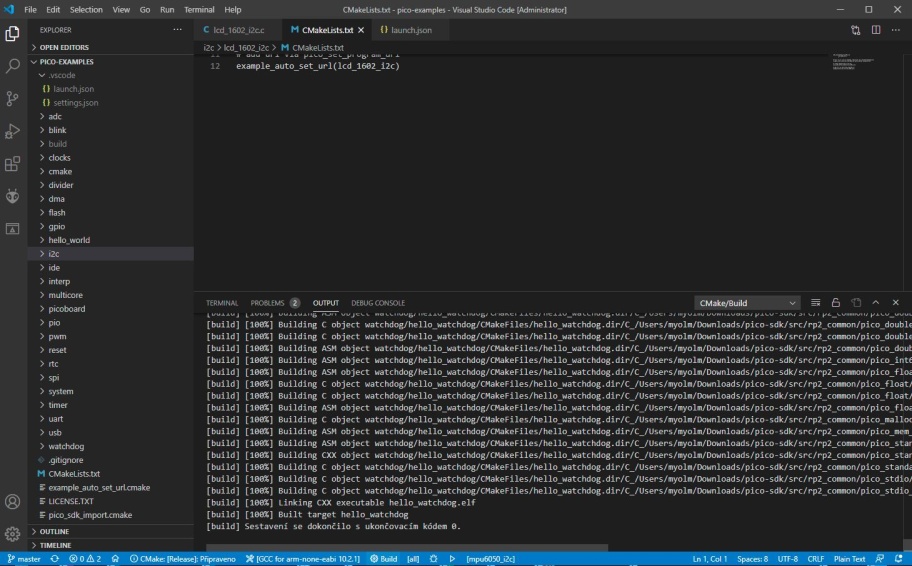


Nyní v **CMD Line (jako Správce)**  zadáme **code** a spustí se **VSCode.**



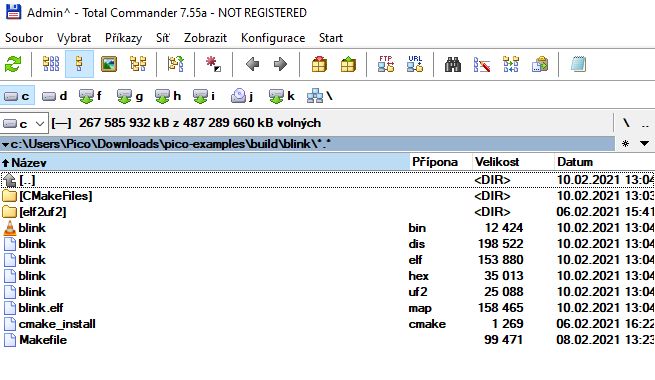
**A můžeme začít pracovat ve VSCode.**

* Ve **File** zvilit **Open Folder** a vybrat cestu, kde jsou **pico-examples**
* Já: **C:/Users/Pico/Downloads/pico-examples/**
* **Zibrazí se všechny v levém sloupci**
* Dole v modré liště kliknout na **GCC for arm-none-eabi 10.2.1** a pokud není, tak jej v horní, nově otevřené nabídce zvolit
* Dole v modré liště kliknout na CMake [….] a zvolit v horní, nově otevřené nabídce vybrat **Release** (nebo **Debug**) Podle potřeby.
* Pokud zvolíte **Release** pak můžete dole v modré liště kliknout na **Build,** vedle ozubeného kolečka.
* Začne **Sestavení a kompolece** všech předpřipravených souborů v **pico-example**
* A měl by skončit takto:

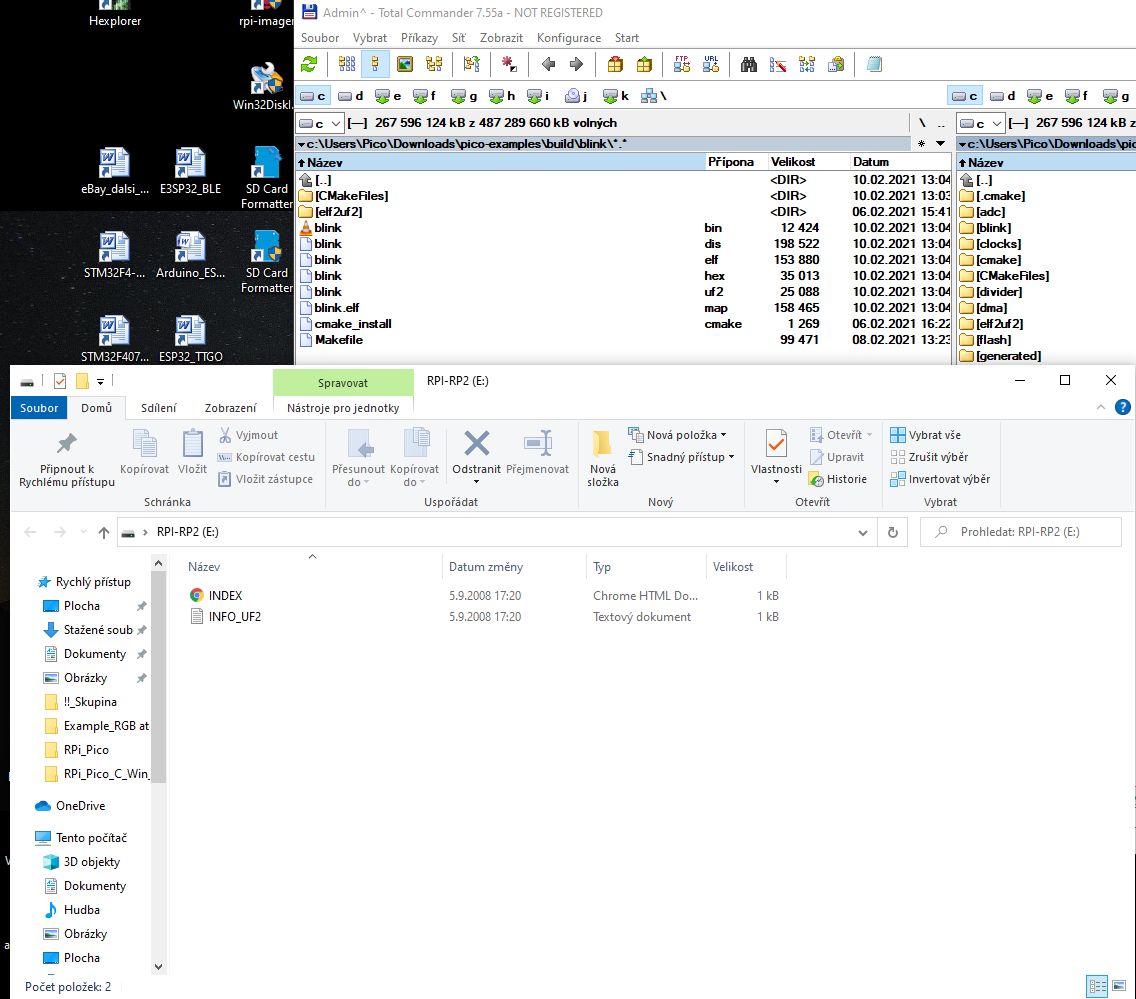


## „Naprogramování“ RPi Pico

* Otevřete v **pico-examples** adresář **build** a v něm např. **blink** třeba v TotalCMD.



* Připojte desku RPi Pico za současného držení tlačítka na desce. Zobrazí se nové okno. Nyní pusťte tlačítko na desce.



* Pomocí **Drag and Drop** přetáhněte soubor **blink.uf2** do okna **RPi POCO (disk)**
* Po přetažení se okno RPI PICO(disk) automaticky **zavře.** A LEDka na desce začne **blikat.**

## Sestavení vlastního projektu

### Umístění vlastního projektu

**C:\Users\Admin\Documents\Pico\my-project1\**

**Struktura:**

…\Documents\Pico\

…

my-project1 (adresář)

build (adresář)

„Prázdný, připravený adresář“

CMakeLists.txt

extrafunc.c

extrafunc.h

main.c

pico\_sdk\_import.cmake

pico-example (adresář)

picoprobe (adresář)

pico-sdk (adresář)

project1 (adresář)

Develop Command Prompt for Pico.lnk

pico-env.smd

pico-setup.cmd

ReadMe.txt

RefreshEnv.cmd

### Obsah CMakeLists.txt

Soubor je umístěn v adresáři:

**C:\Users\Admin\Documents\Pico\my-project1\**

cmake\_minimum\_required(VERSION 3.12)

include(pico\_sdk\_import.cmake)

set(projname "project1")

project(${projname} C CXX ASM)

set(CMAKE\_C\_STANDARD 11)

set(CMAKE\_CXX\_STANDARD 17)

pico\_sdk\_init()

#include(example\_auto\_set\_url.cmake)

add\_executable(${projname}

main.c

extrafunc.c

)

target\_link\_libraries(${projname} pico\_stdlib)

pico\_add\_extra\_outputs(${projname})

* Na řádce, obsahující kód:

**set(projname "project1")**

* změníme název projektu např. na **“my-project1“,** podle názvu našeho adresáře = názvu projektu.

# Instalace

## RPi Pico programovani v Ccku – Windows

Nejprve je potřeba nainstalovat potřebné programové vybaven. Tato instalace je odlišná od jiných platforem. Musíme také nainstalovat program pro sestavení (building code) programu.

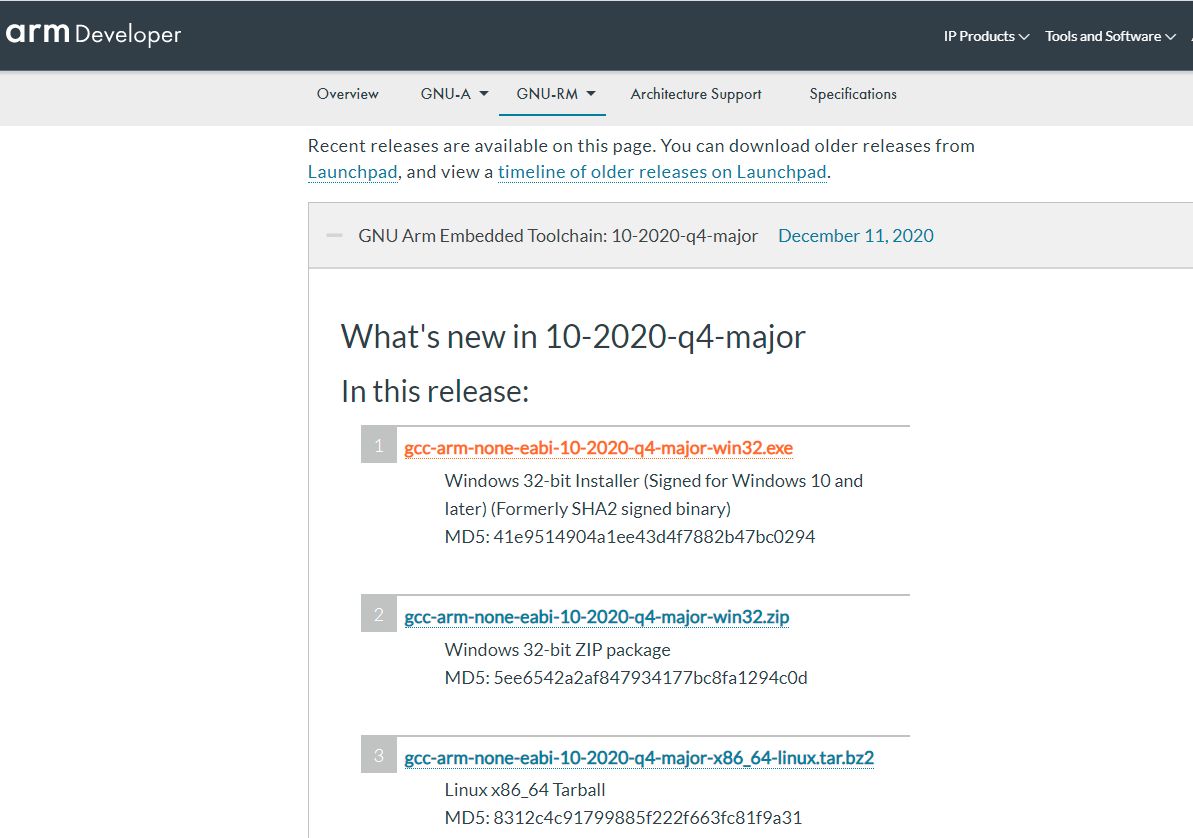
**Instalace sady nástrojů**

Pro buid je potřeba mít nainstalované tyto programy:

* [ARM GCC compiler](https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-rm/downloads)
* [CMake](https://cmake.org/download/)
* [Build Tools for Visual Studio 2019](https://visualstudio.microsoft.com/cs/downloads/)
* [Python 3.7](https://www.python.org/downloads/windows/)
* [Git](https://git-scm.com/download/win)

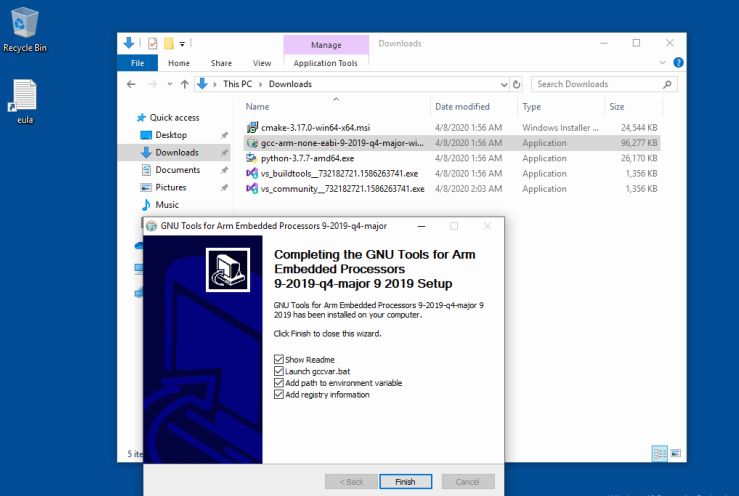
### Ad 1) Instalace GCC:

Otevřeme příslušnou stránku a zobrazí se nám okno, kde sjedeme dolů a vybereme ke stažení **gcc-arm-none-eabi-10-2020-q4-major-win32.exe** (02.2021):



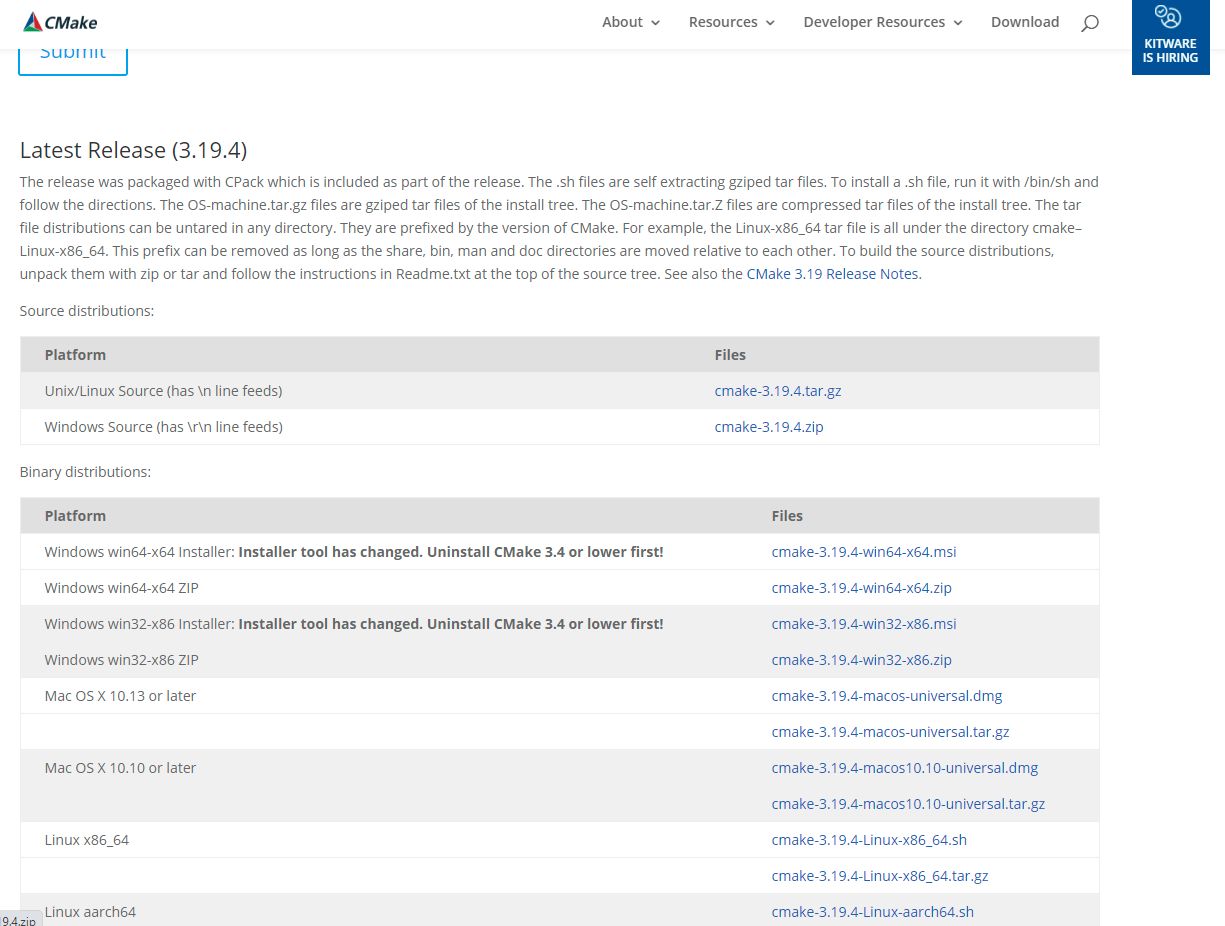
Po spuštění volíme:

* Česky a tlečítko Další
* Tlačítko Souhlas
* Tlačítko Instalovat Do zvolené cesty, já do - ProgramFiles(x86)/GNU …/10 2020-q4 …
* Během instalace zaškrtnout všechny 4 volby, zejména box pro Add path to envoroment variable ( duhý od spoda )
* Tlačítko Dokončit



### Ad 2) Instalace CMake:

Otevřeme příslušnou stránku a zobrazí se nám okno, kde sjedeme dolů a vybereme ke stažení **cmake-3.19.4-win64-x64.msi** (02.2021).



Během instalace přidat CMake to the systém **PATH** pro všechny uživatele ( … add CMake to the systém **PATH** for all users hen promted by the installer ).

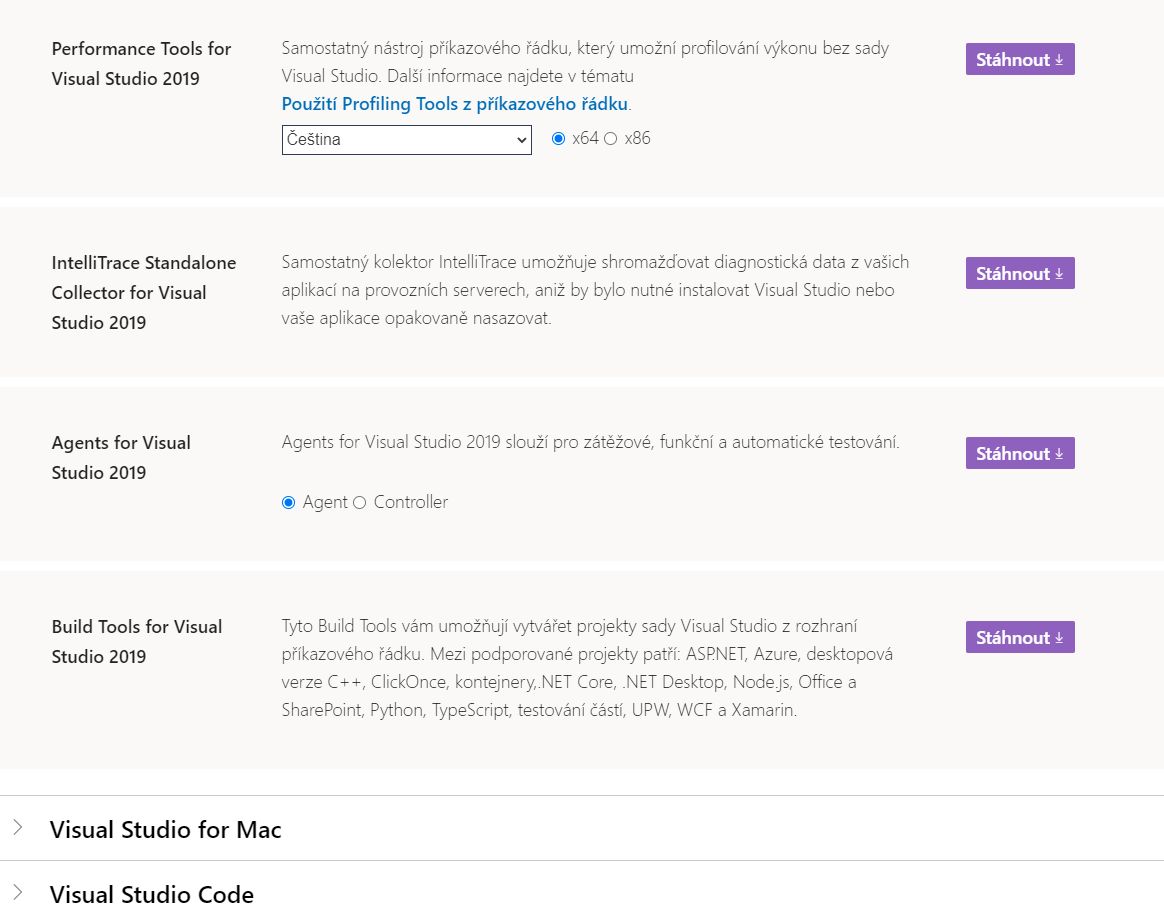
Po spuštění volíme:

* **Prepare to install** a tlačítko **Next**
* Tlačítka **Accept** a **Next**
* Zaškrtnout **3x volby + ikony** a tlačítko **Next**
* Nainstalovat do adresáře **ProgramFiles/CMake/** a tlačítko **Next**
* Běží **Instalace (Install)** (docela dlouhá)
* Nakonec stisknout tlačítko **Finish**

### Ad 3) Instalace Build Tools for Visual Studio 2019:

Otevřeme příslušnou stránku a zobrazí se nám okno, kde sjedeme dolů a vybereme ke stažení **Tools for Visual Studio 2019**.

Opět sjedeme dolů a vybereme **Build Tools for Visual Studio 2019**:



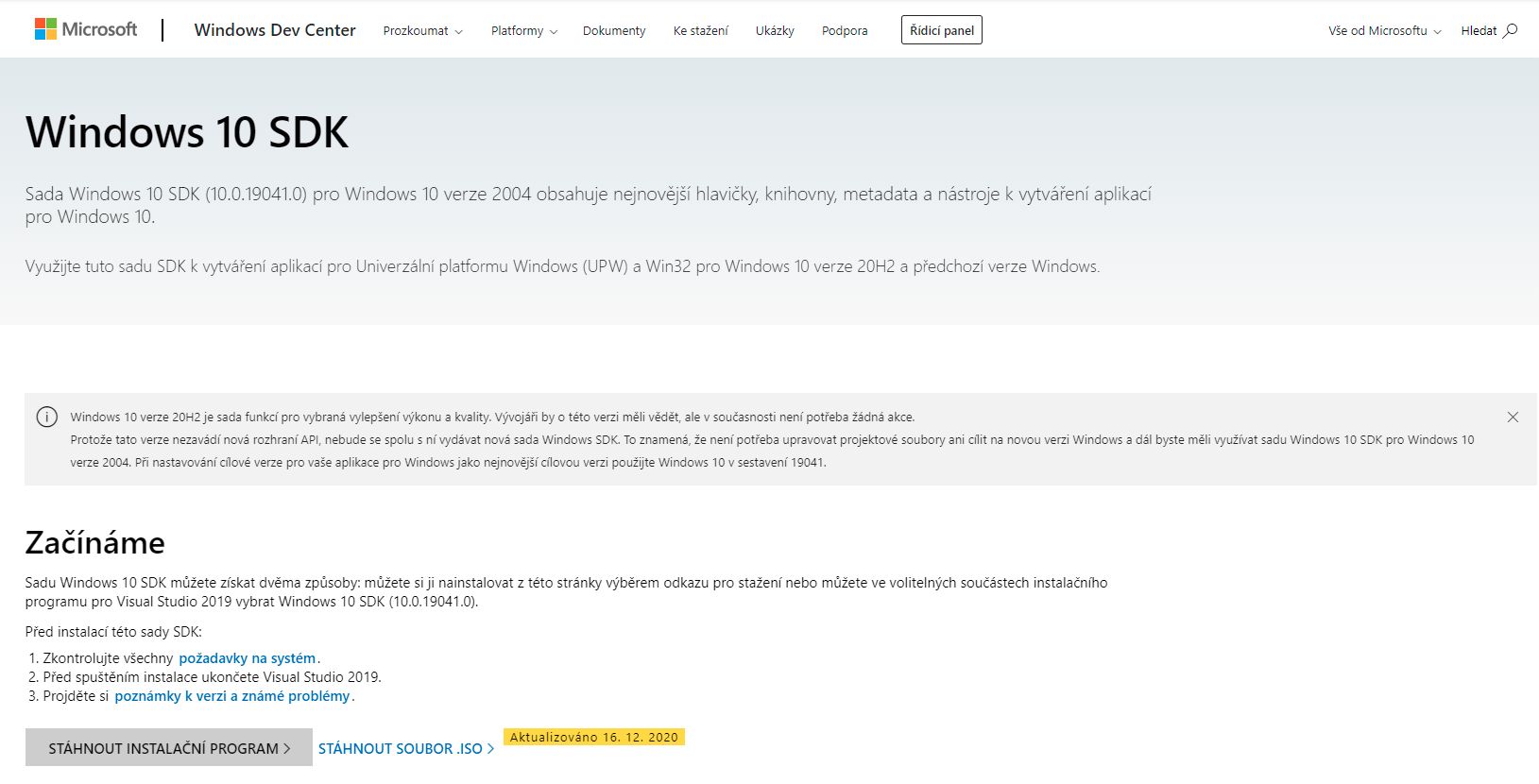
Spustíme soubor **vs\_buildtools\_\_354044476.1612321342.exe.**

* Zobrazí se okno s textem**: ….. nastavit několik věcí** a pracuje bez našeho zásahu.
* Stiskneme poté tlačítko **Pokračovat**
* Zobrazí se zaškrtnutý box  **Staženo** a **Probíhá instalace**
* Zobrazí se zaškrtnurý box **Naunstalováno**
* Zobrazí se nové okno
* Vlevo nahoře zaškrtneme box **C++ BuildTools (3,5GB)** a stiskneme tlačítko **Nainstalovat**
* Zobrazí se okno **Visual Studio Installer** a v **toolbar** s postupem instalace
* Informace **Stahuje** se pokračování instalace (919MB)
* Informace **Staženo**
* Instaluje se **: balíček xxx z 265**
* Zobrazí se informace **Hotovo** se zaškrtnutným boxem
* Více, jsou 4 volby

**Musí se nainstalovat PLNÉ INSTALACE „Windows 10 SDK“!**

### Ad 4) Instalace Windows 10 SDK:

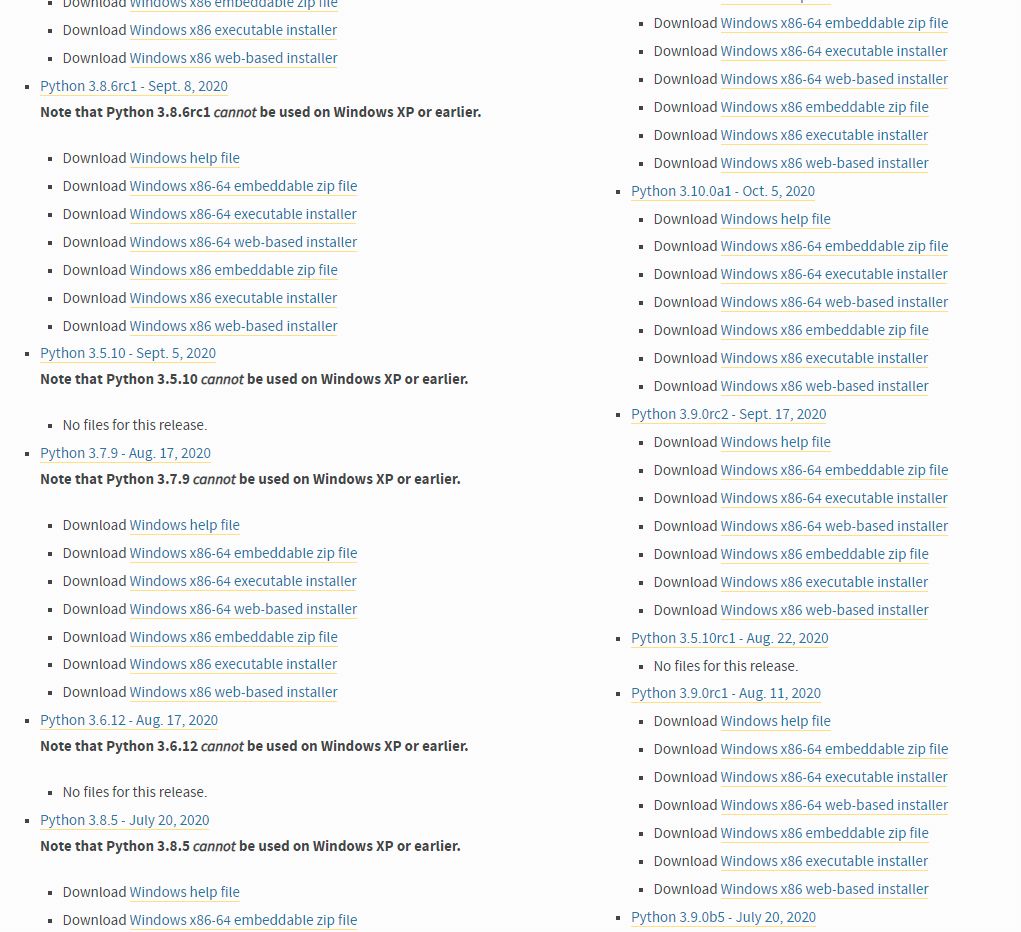
Stáhneme si **STÁHNOUT INSTALAČNÍ PROGRAM**, spustíme **winsdksetup.exe** a postupujeme podle pokynů na obrazovce.



Po spuštění volíme:

* Zvolíme **1. Volbu nahoře** a stiskneme tlačítko **Next**
* Zvolíme volbu **Neposílat informace MrkvoDoftu** a stiskneme tlačítko **Next**
* Zvolíme tlačítko **Accept**
* **Vše** zaskrtnuto a tlečítko **Install**
* Stiskneme tlačítko **Close**

### Ad 5) InstalacePathon 3.7:



Otevřeme příslušnou stránku a zobrazí se nám okno, kde sjedeme dolů na **Python 3.7.9 Aug 2020.**

Otevře se nám nové okno s nabídkou Python 3.7.9. **Sjedeme dolů a zvolíme** [**Windows x86-64 executable installer**](https://www.python.org/ftp/python/3.7.9/python-3.7.9-amd64.exe)**.** Stáhneme. A nainstalujeme. Já neinstaloval nyní, protože jej mám v PC již instalován!

Během instalace zvolte volbu **„ Customize instalation“** a klikněte na volnu **„Optional Features“** a následně zvolte **instalaci pro všechny uživatele** ( **„Install for all users“** ).

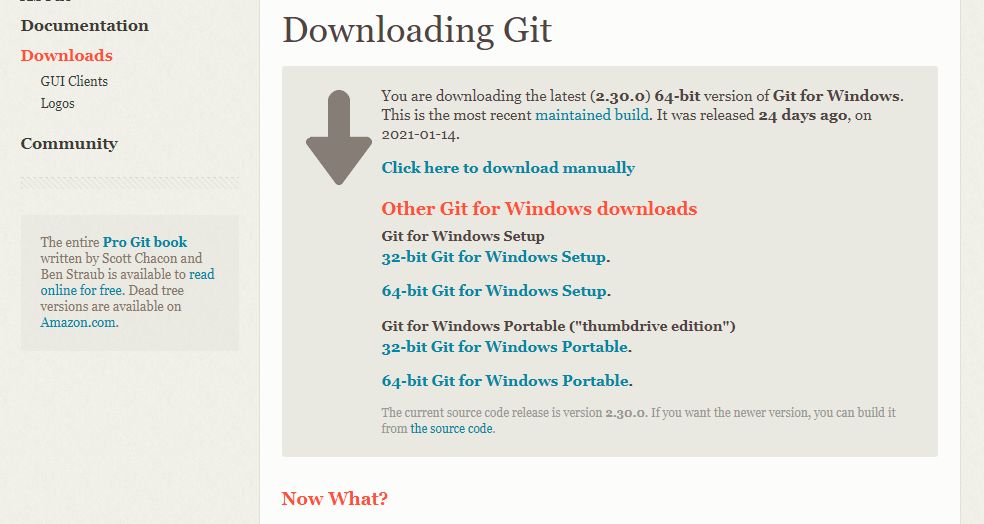
Možná budete muset vytvořit symbolický odkaz, aby Makefile mohl najít Python 3. Chcete-li tak učinit, zadejte do okna Spustit **cmd,** vedle nabídky Windows otevřete okno příkazového řádku pro vývojáře, ale v nabídce vyberte možnost **Spustit jako správce,** pravé podokno pro otevření okna s oprávněními správce. Poté přejděte V **cmd** do složky **C: \ Program Files \ Python37** a

vytvořit symbolický odkaz.:

C: \ Program Files \ Python37> mklink python3.exe python.exe

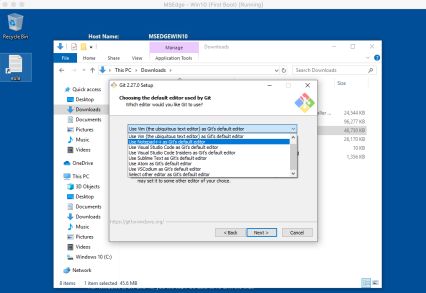
Nyní by to již nemělo být nutné. Pokud však vaše sestavení selže, protože **make** nemůže najít vaši instalaci Pythonu, tak by měl přidaný symbolický odkaz do spustitelného souboru pomoci. To může věci vyřešit.

### Ad 6) InstalaceGit:

Otevřeme příslušnou stránku a stáhneme soubor **Git-2.30.0.2-64-bit.exe.**

Ten spustíme a pokračujeme podle následujících pokynů:

* Vyberte volbu **Chose the default editor = Use Notepa …**  a stiskneme tlačítko **Next**
* Zaškrtnout kolečko **Let Git decide**  a stisknout tlačítko **Next**
* Zaškrtnuté kolečko **Default** a **Next**
* Zaškrtnout kolečko **1. Git Credential …** a stisknout tlačítko **Next**
* Zaškrtnout **Ebable a Install**
* **(** Zaškrtnout kolečko (uprostřed) **Git from command line …** a stisknout tlačítko **Next**
* Zaškrtnout kolečko **Use the Open SSL library** a stisknout tlačítko **Next**
* Zaškrtnout kolečko **Checkent Windows – style …** a stisknout tlačítko **Next**
* Zaškrtnout kolečko **Use Win’ s default konsole Windows** a stisknout tlačítko **Next**
* Zaškrtnout kolečko **Default** a stisknout tlačítko **Next**
* Zaškrtnout kolečko **1. Git Credential …** a stisknout tlačítko **Next**
* Zaškrtnout 2x **Enable …** a stisknout tlačítko **Next**
* Zaškrtnout **Enable …** a stisknout tlačítko **Install )**
* Zaškrtnou na závěr tlačítko **Finish**



# Provést před začátkem práce s SDK a Examples v CMD (spuštěném jako Správce)

C:\Users\pico\Downloads> git clone -b master https://github.com/raspberrypi/pico-sdk.git  
C:\Users\pico\Downloads> cd pico-sdk  
C:\Users\pico\Downloads\pico-sdk> git submodule update --init  
C:\Users\pico\Downloads\pico-sdk> cd ..  
C:\Users\pico\Downloads> git clone -b master https://github.com/raspberrypi/pico-examples.git

I toto:

C: \ Program Files \ Python37> mklink python3.exe python.exe

## Instalace VSCode

V **CMD Line**, spuštěné jako **SPRÁVCE** zadáme příkaz:

**C:>code**

Po chvilce se spustí VSCode.

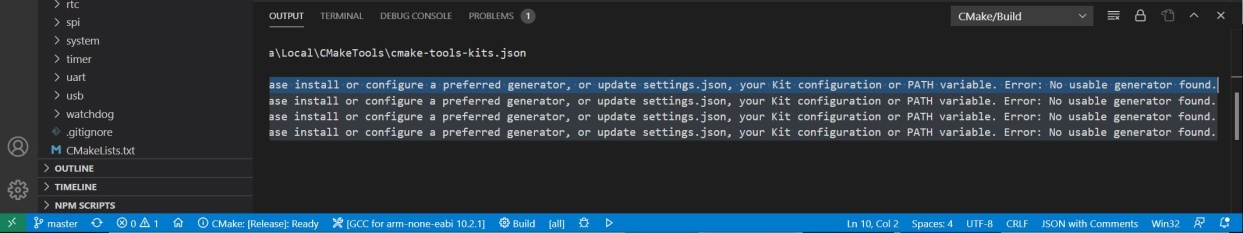
* V levém sloupci zvolíme **Extensions**
* Zvolíme **CMake** **Tools**
* Dále klikneme na **Install**  čímž se do VSCode nainstaluje, přidá CMake Tools
* V levém sloupci dole zvilíme **Manage** (ozubené kolečko)
* Vybereme **Settings**, dále **Upravit** dáleEdensiong a **CMake Tools config**
* V prostředním nově otevřeném okně sjedeme dolů a najdeme **Configure Enviroment**
* V něm klikneme na **Add Item** a vložíme **PICO\_SDK\_PATH** a ..\..\pico-sdk
* Stiskneme tlačítko **OK** = přidáno
* Dále sjedeme ještě dolů a najdeme volbu **Cmake Generator**
* V něm vložíme **NMale Makefiles**
* Vybereme **Chose Setting**
* Ukončíme, zavřením
* Přejdeme, vlevo nahoře do **File** zvolíme volbu **Open Folder** a zvolíme **pico-examples** v cestě: **C:/Users/Pico/Downloads/pico-sexamples**
* Stuskneme tlačítko **OK** a **Save zour workspace** a **Save** jako Pico do adresáře
* Otevře se okno s Examples, vpravo **Chcete nakonfigurovat projekt pico –examples** zvolit **Ano**
* Nahoře, v otevřené nabídce, vložit **GCC for arm-noéne …**
* Konfigureje se projekt , uprostřed dole se vypisují informační hlášky
* Zvolit **Povolit** a tlačítko **Ano** při otevření konfigurovat
* Dole, v **modré liště** je **Ozubené kolečko** s popisem **Build**
* Restart VSCode
* Po kliknutí, v modré liště, na Build se spustí sestavení

# Start s C/C++ a Raspberry Pi Pico

## Getting started with C/C++ and Raspberry Pi Pico

Strávil jsem pár hodin pokusem zjistit, jaký je správný proces nastavení SDK pro C / C ++ na mém počítači se systémem Windows. Pečlivě jsem dodržoval oficiální pokyny, ale narazil jsem na chyby. Aby byl proces komplikovanější, nejde o instalaci jedné věci a jen s ní válcování. Ve světě plném alternativních mikrokontrolérů, které jednoduše pracují s PlatformIO nebo Arduino IDE, je Raspberry Pi Pico SDK opravdu krátký.

Jediným světlem v tunelu je spolupráce s Arduinem. Jelikož také získají mikrokontrolér založený na RP2040, jsem si docela jistý, že programování v C / C ++ bude stejně snadné jako výběr správné desky od správce desky v Arduino IDE.

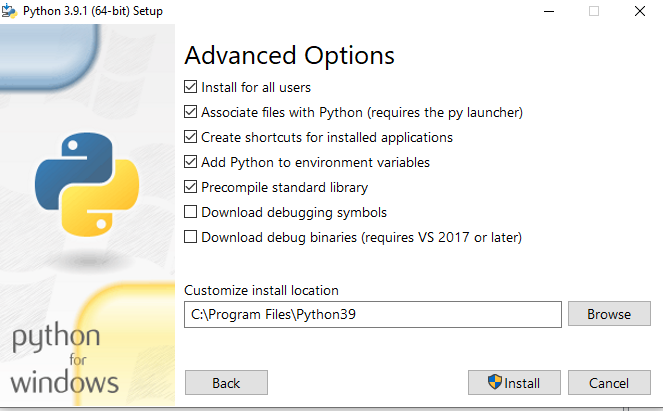


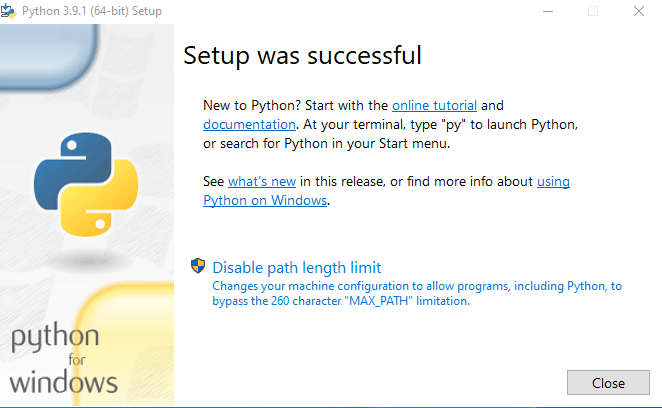
Chcete-li to zkusit, musíte udělat. Stáhněte si všechny níže uvedené programy (ano, to je důvod, proč na mě nástrojová řada příliš neudělala dojem):

* [Python 3.x](https://www.python.org/downloads/)
* [ARM GCC compiler](https://developer.arm.com/-/media/Files/downloads/gnu-rm/10-2020q4/gcc-arm-none-eabi-10-2020-q4-major-win32.exe?revision=9a4bce5a-7577-4b4f-910d-4585f55d35e8&la=en&hash=068C813EEFFB68060B5FB40E6541BDE7159AFAA0)
* [CMake](https://github.com/Kitware/CMake/releases/download/v3.19.3/cmake-3.19.3-win64-x64.msi)
* [Build Tools for Visual Studio 2019](https://visualstudio.microsoft.com/thank-you-downloading-visual-studio/?sku=BuildTools&rel=16) (C++ build tools only)
* [Visual Studio Code](https://code.visualstudio.com/download)
* [pico-sdk tools](https://github.com/raspberrypi/pico-sdk)

Během instalace věnujte pozornost možnostem instalačního programu a ujistěte se, že jsou vybrány následující položky:  
**Python:** Přidat cestu, Instalovat pro všechny uživatele

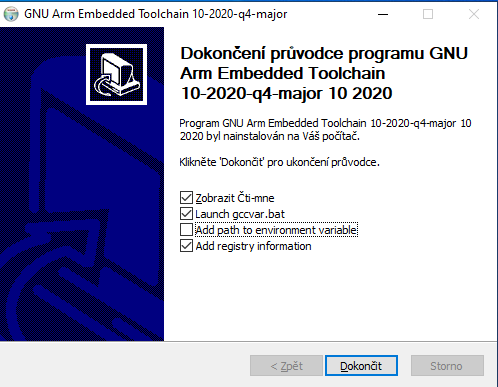


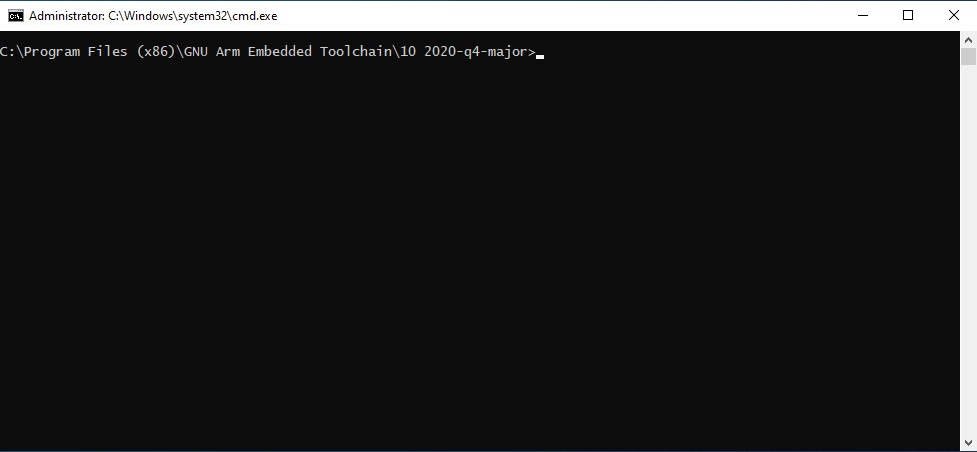




**Kompilátor AGM GCC:** Přidat cestu (všechny vybrané možnosti, instalační program může na konci viset, takže to udělejte ihned)

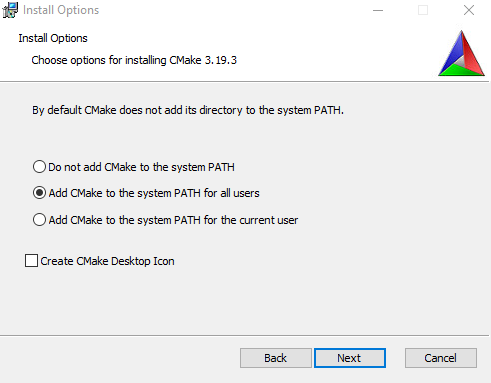
* Česky a OK
* Tlačítka Další a Souhlasím
* Tlačítko Instalovat
* Zaškrtnou: Add path to enviroment variable a tlačítko Dokončit



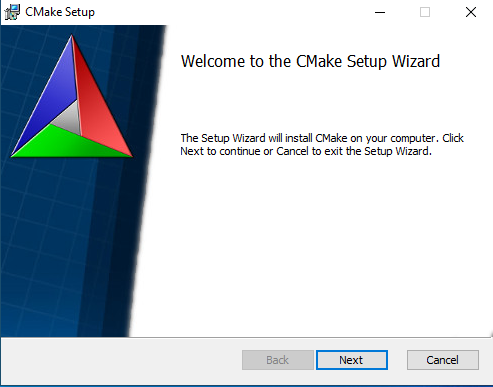


**cmake:** Přidat cestu

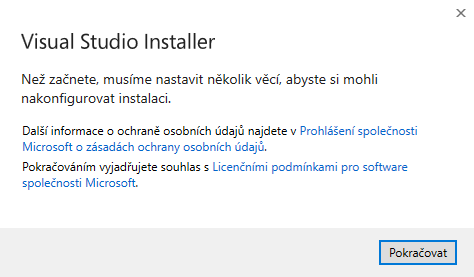
* Next
* Zaškrtnout I accept the terms in the Licence Agreement ¨
* Next
* Zaškrtnout: Add CMake to the systém PATH for all users



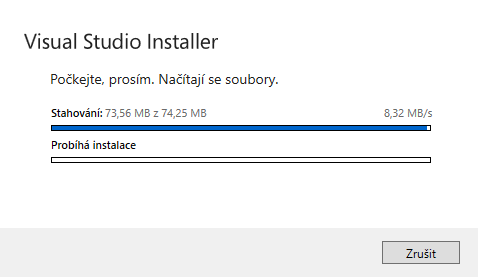
* Zaškrtnout Create CMake Desktpop Icon
* Next
* Next
* Install
* Finish

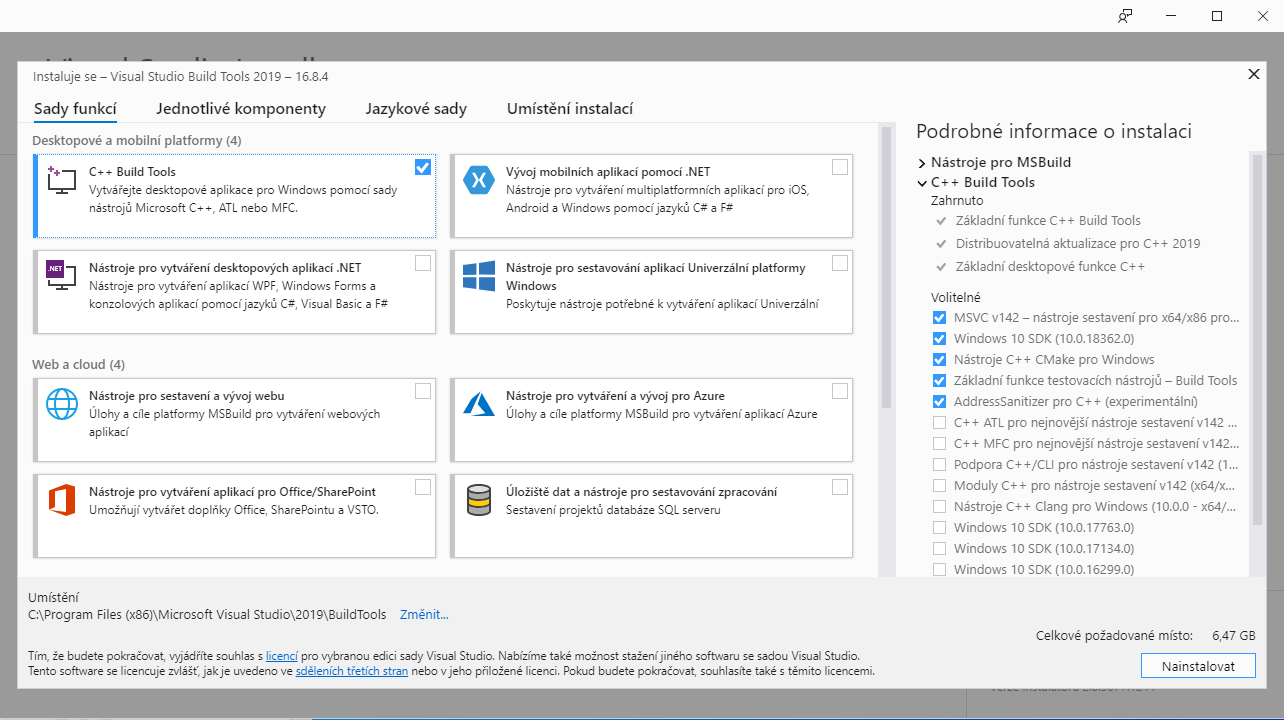


**vytvářet nástroje pro Visual Studio** : Instalace C ++ pouze s výchozí možnosti

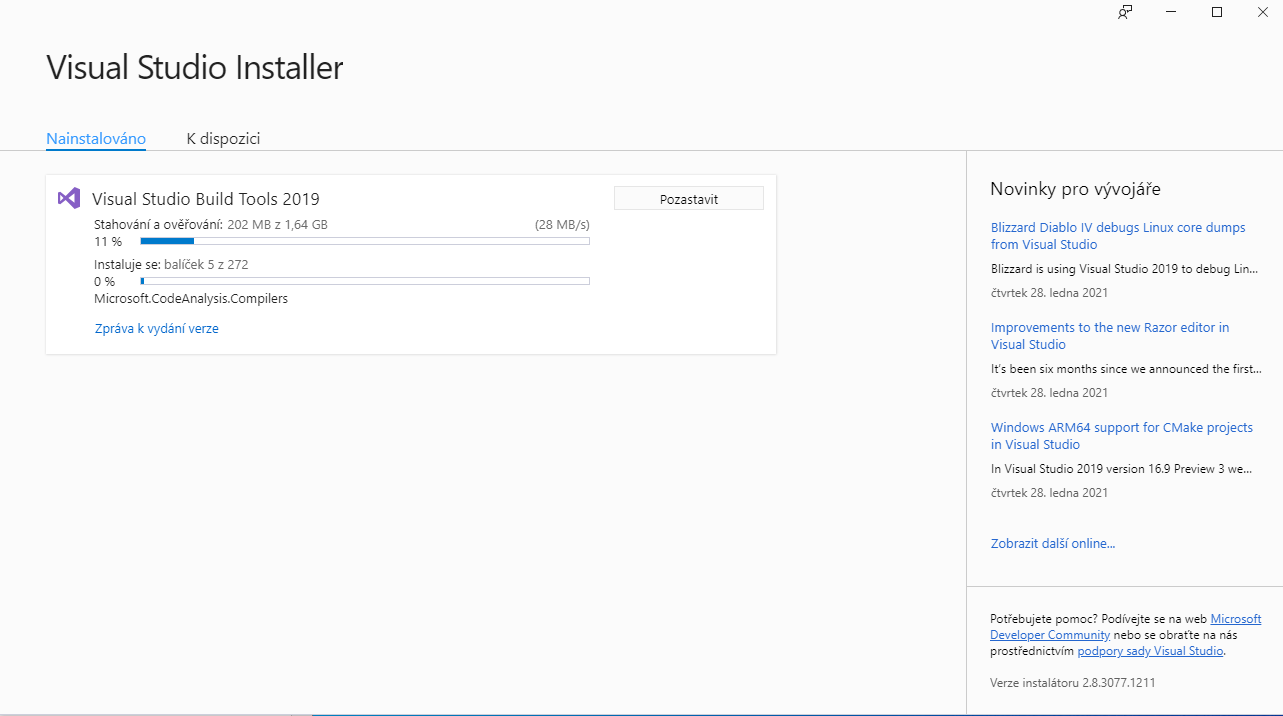


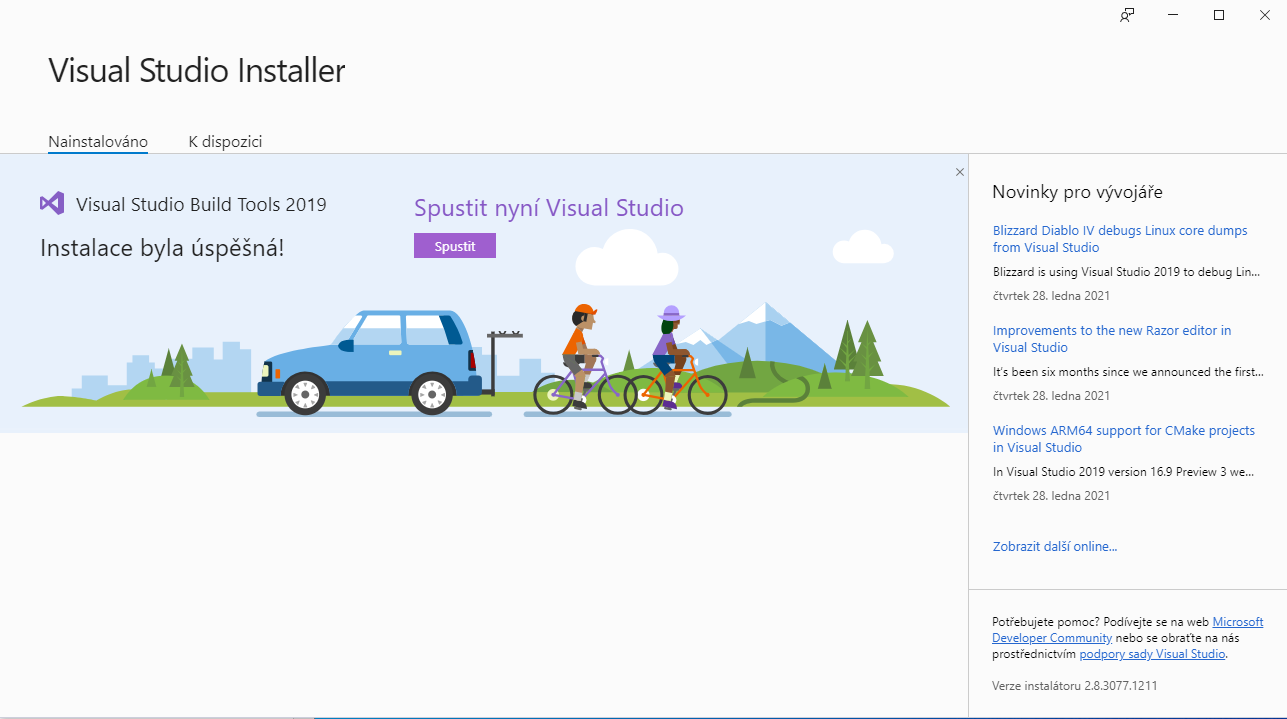
* Tlačítko Pokračovat





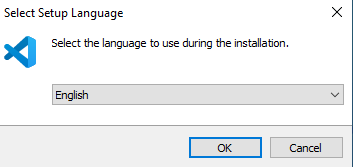
* Zaškrtnout vlevo nahoře
* Tlačítko Nainstalovat



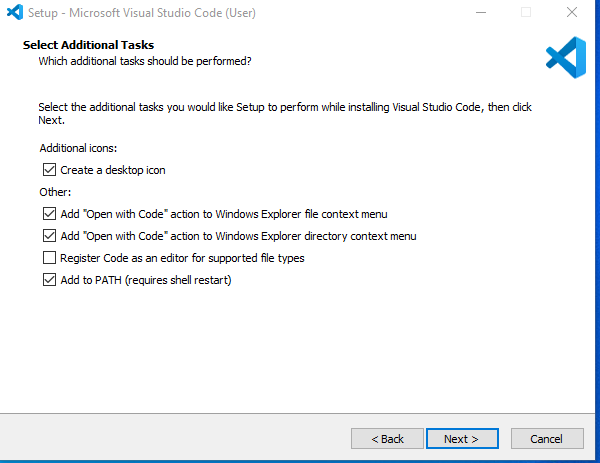


* Zavřít okno

**Visual Studio Code :**

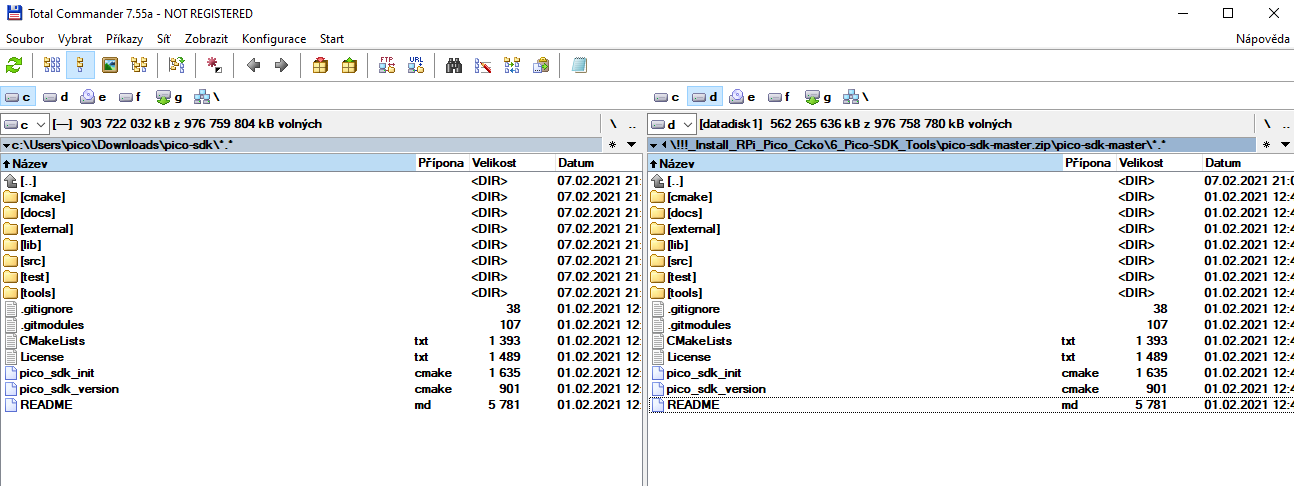


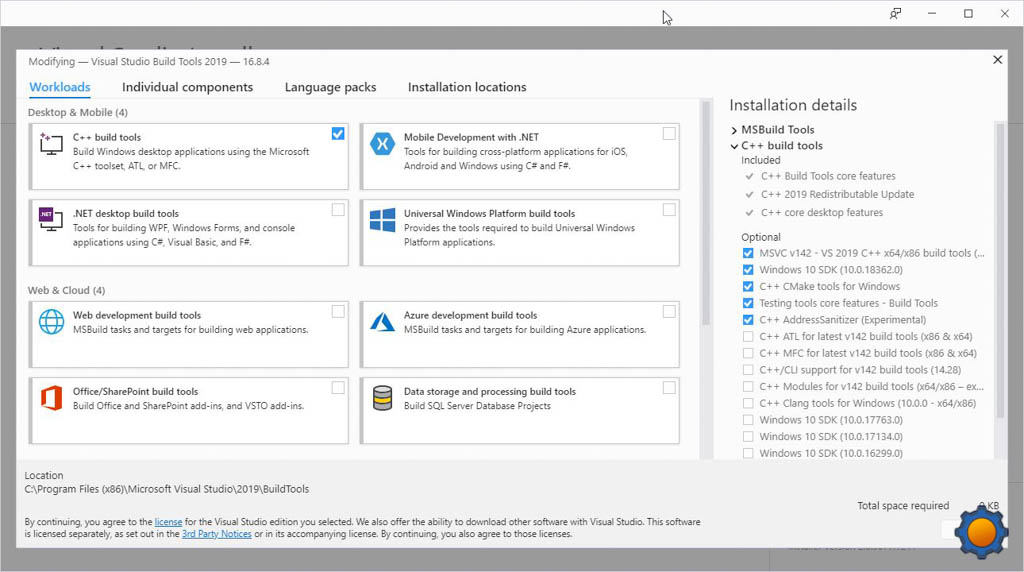
* Tlačítko OK
* Zaškrtnout: I accept the agreement a Next
* Zaškrtnout: Create a desktop icon a 2x první 2 volby v Other a Next



* Tlačítko Install
* Tlačítko Finish

**piko-SDK** : Stáhněte si zip a rozbalit složku do adresáře podle svého výběru

Vytvopřit adresář C:/Users/pico/Downloads/pico-sdk/ a nakopírovat sem rozbalená adresář.

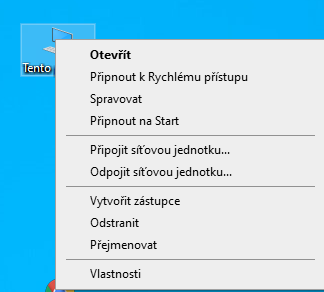


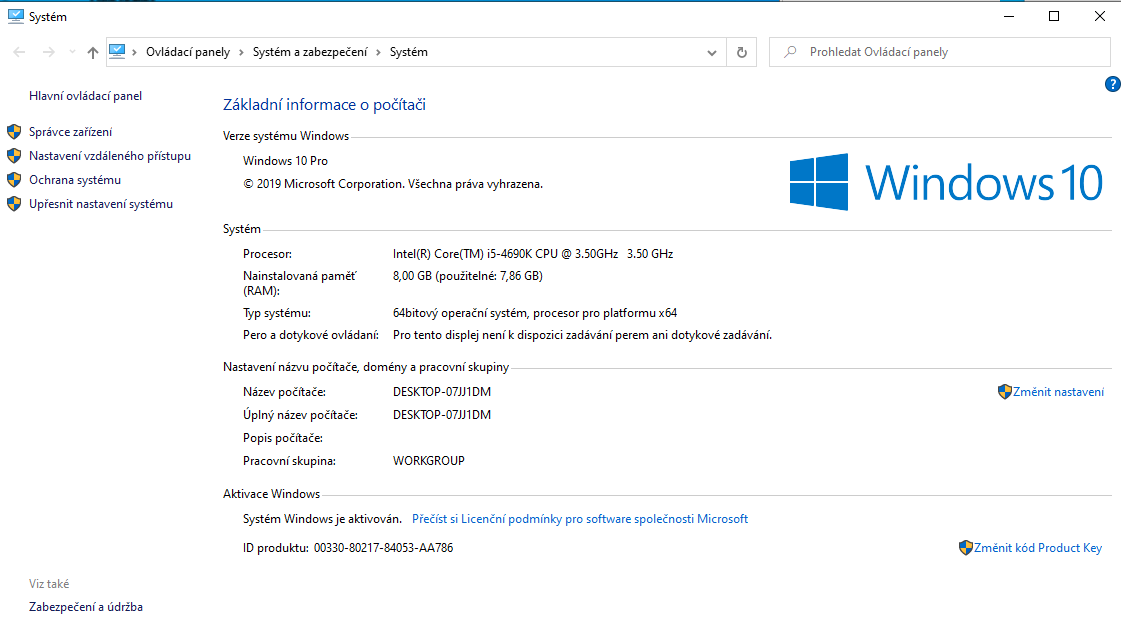
Po dokončení instalace se ujistěte, že jsou proměnné prostředí v systému Windows nastaveny správně. ( (Klávesa Win + zadejte „upravit systémové proměnné prostředí“). V **Uživatelské proměnné** přidejte: )

Ve **Windows 8 a Windows 10** se opět na ploše nezobrazuje ikona **Tento počítač**. Pokud jste na ní zvyklí z dřívějších dob, můžete ji snadno zobrazit.

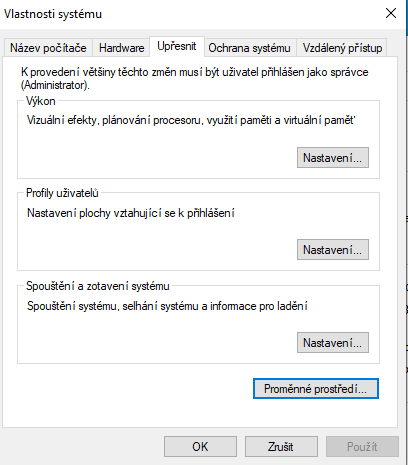
Klikněte na **Start** -> „**Nastavení“**, vyberte položku „**Přizpůsobení**„, vlevo zvolte „**Motivy**“ a vpravo vyberte klikněte na „**Nastavení ikon na ploše**„. Tam kliknutím označte „**Počítač**“ a potvrďte kliknutím na OK.

PTM na Tento počítač a zvolit **Vlastnosti**

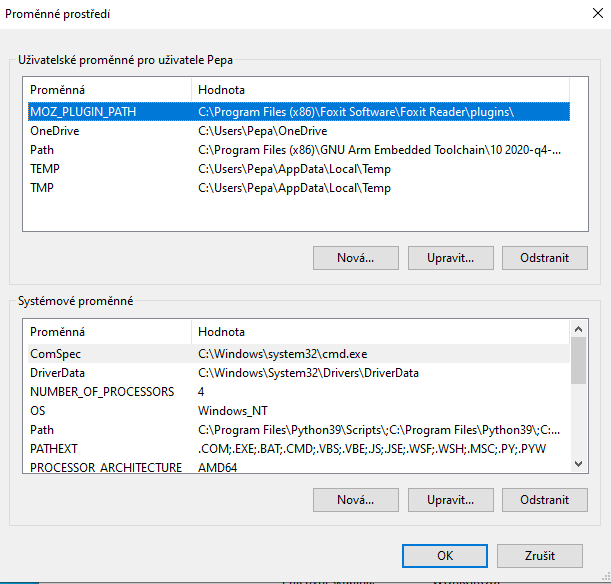




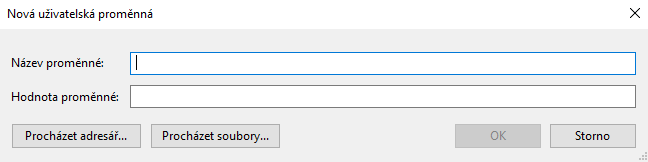
Vlevo klik na **Upřesnit nastavení systému**, otevře se nové okno:



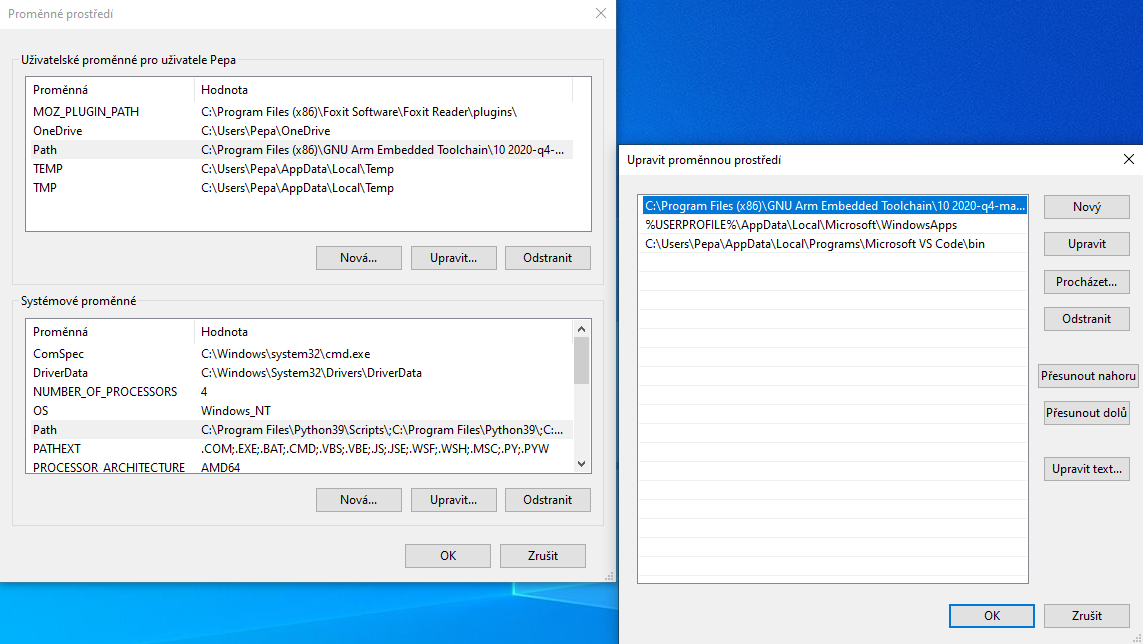
* Kliknout na **Proměnné prostředí** otevře se nové okno:



Přidáme pomocí **Nová… PICO\_SDK\_PATH** **.** V **Uživatelské proměnné pro uživatele xxxxxxx** přidáme:

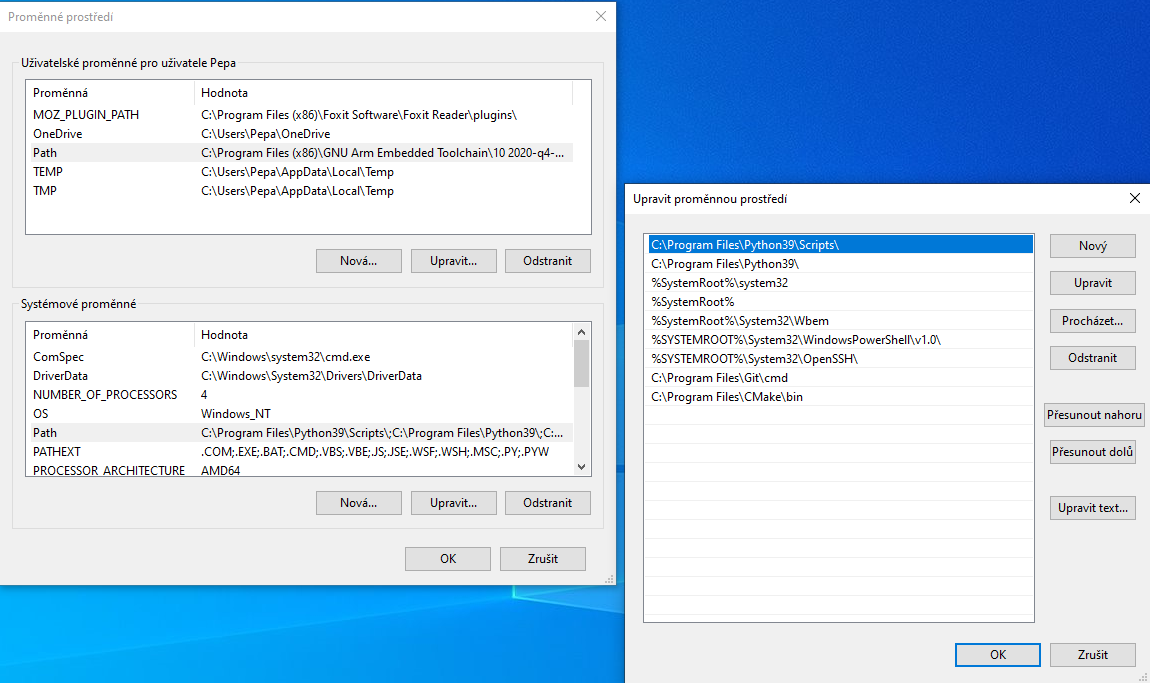


Kursor nastavit na **Path.** Přidáme pomocí **Upravit… Path.** V **Uživatelské proměnné pro uživatele xxxxxxx** přidáme:



|  |  |
| --- | --- |
| Proměnná | Hodnota |
| PICO\_SDK\_PATH | C:\path to downloaded folder\pico-sdk |
| Path | C:\Users\mzoln\AppData\Local\Programs\Microsoft VS Code\bin |

C:\Users\(mzoln)Admin\AppData\Local\Programs\Microsoft VS Code\bin

Kursor nastavit na **Path.** Přidáme pomocí **Upravit…. Path** V **Systémové proměnné** přidáme:

|  |  |
| --- | --- |
| **Systémové proměnné** | **Hodnota** |
| Path | C:\Program Files\Python39\ |
| Path | C:\Program Files\Python39\Scripts\ |
| Path | C:\Program Files\CMake\bin |
| Path | C:\Program Files (x86)\GNU Arm Embedded Toolchain\10 2020-q4-major\bin |
| Path | C:\Users\mzoln\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Visual Studio Code |

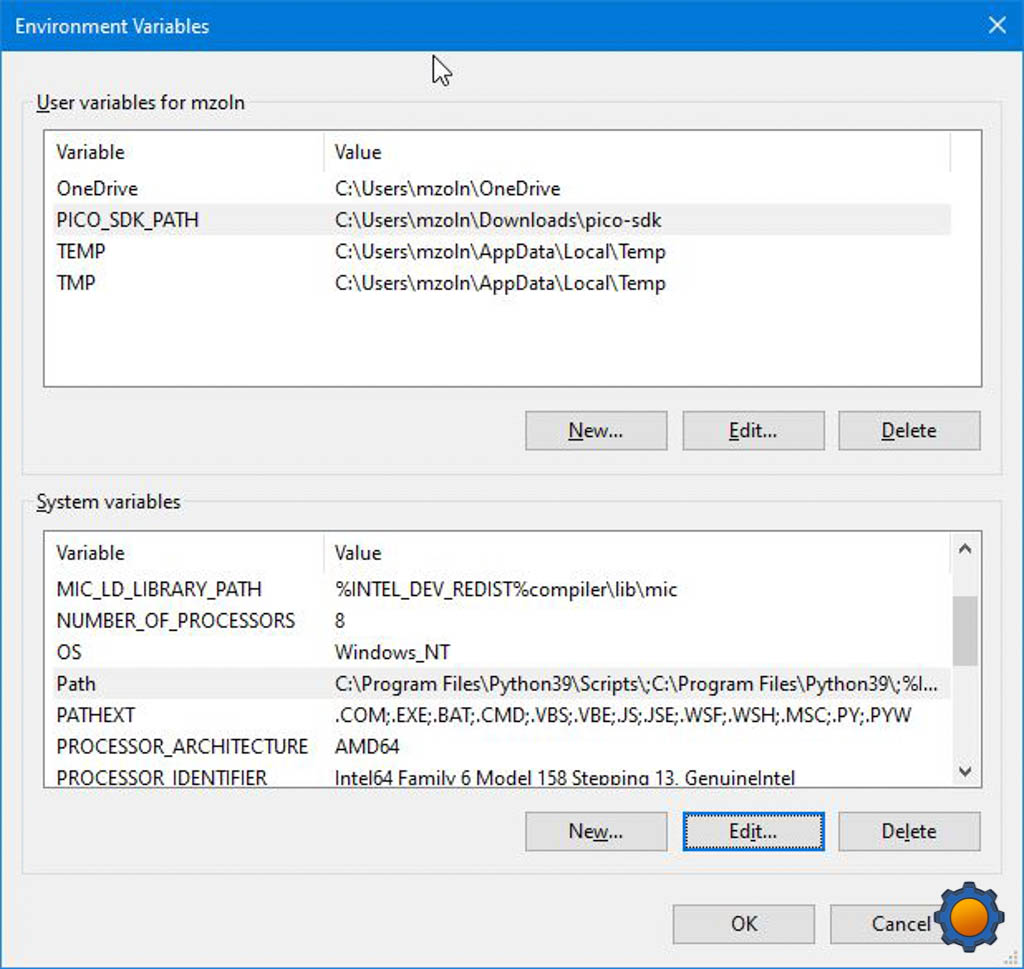
C:\Program Files (x86)\GNU Arm Embedded Toolchain\10 2020-q4-major\bin

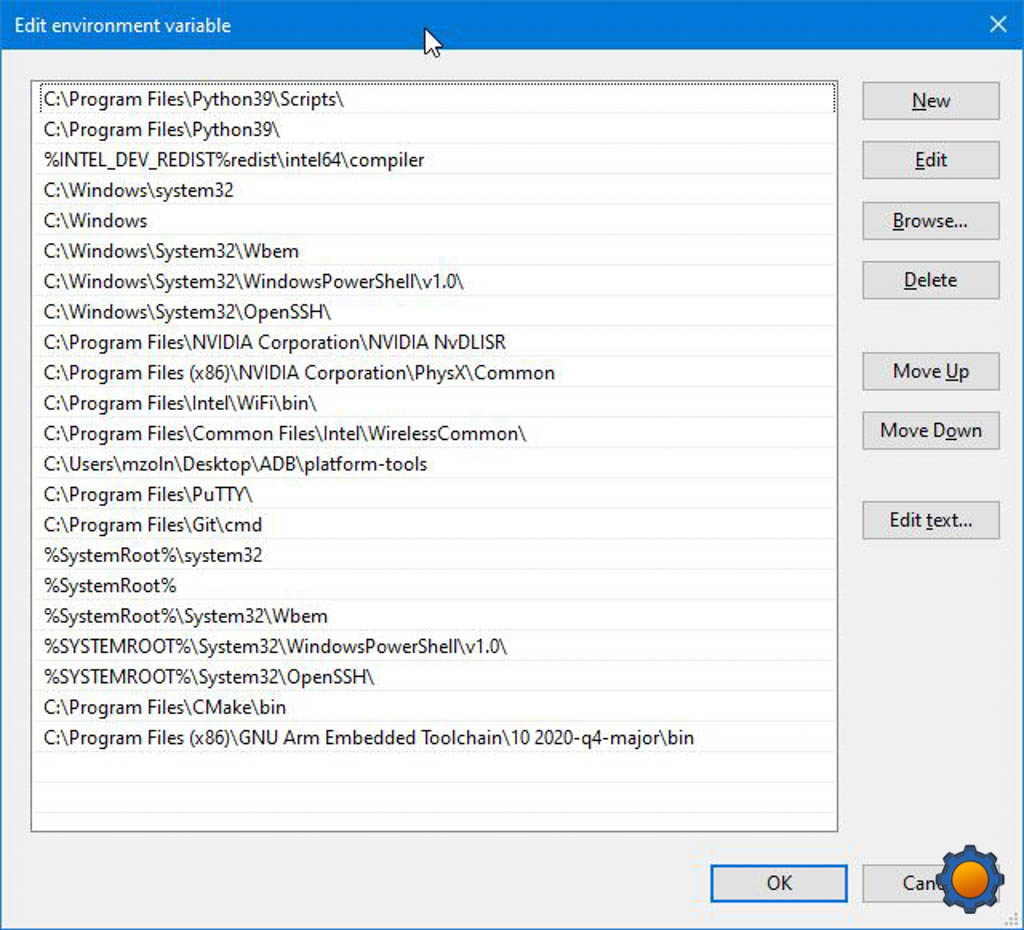
C:\Users\mzoln\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Visual Studio Code

|  |  |
| --- | --- |
| Variable | Value |
| PICO\_SDK\_PATH | C:\path to downloaded folder\pico-sdk |
| Path | C:\Users\mzoln\AppData\Local\Programs\Microsoft VS Code\bin |

In System, Variables make sure you have the following

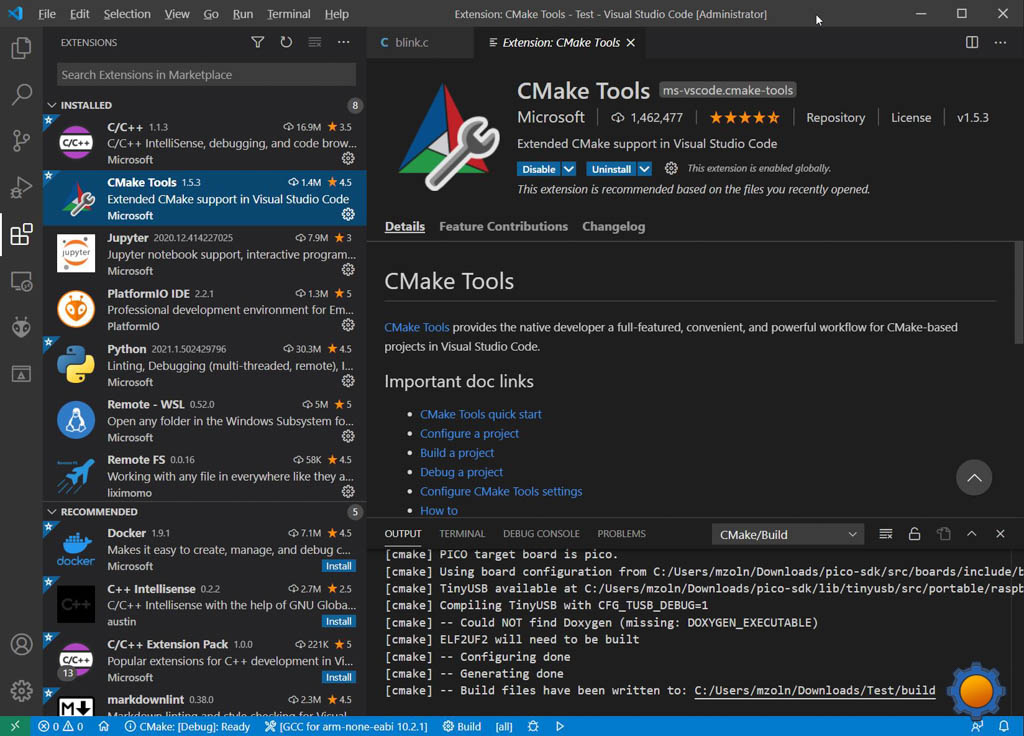
|  |  |
| --- | --- |
| **Variable** | **Value** |
| Path | C:\Program Files\Python39\ |
| Path | C:\Program Files\Python39\Scripts\ |
| Path | C:\Program Files\CMake\bin |
| Path | C:\Program Files (x86)\GNU Arm Embedded Toolchain\10 2020-q4-major\bin |
| Path | C:\Users\mzoln\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Visual Studio Code |



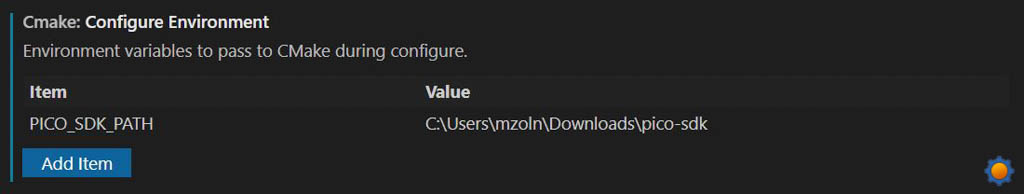


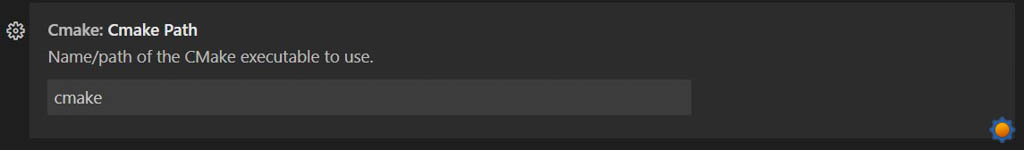
Dále vyhledejte nainstalované programy pro Visual Studio 2019 - a pomocí **příkazového řádku vývojáře** ( otevřete jako správce kliknutím pravým tlačítkem) otevřete VSC (pokud otevřete VSC normálně, nebude to fungovat a po 2 hodinách hraní s proměnnými jsem se vzdal ). Zadejte codeVSC se správnými nástroji pro sestavení.

**Konfigurace VSC**

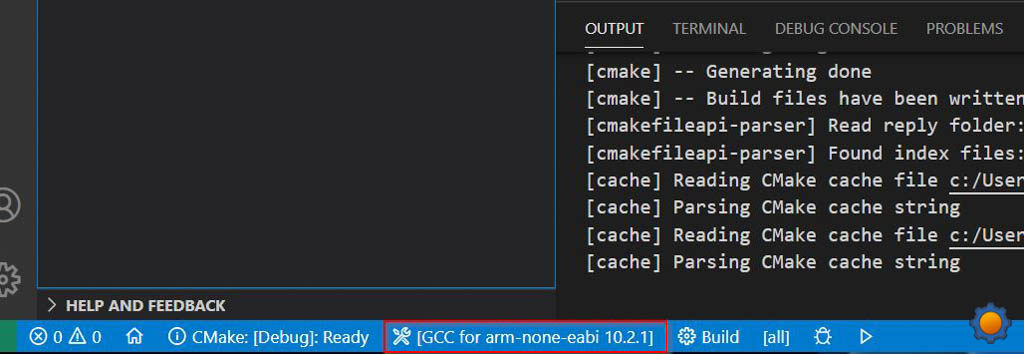


Na panelu rozšíření vlevo vyhledejte **C / C ++** a **nástroje CMake** od společnosti Microsoft. Po instalaci přejděte do nastavení rozšíření pro nástroje CMake a změňte následující:



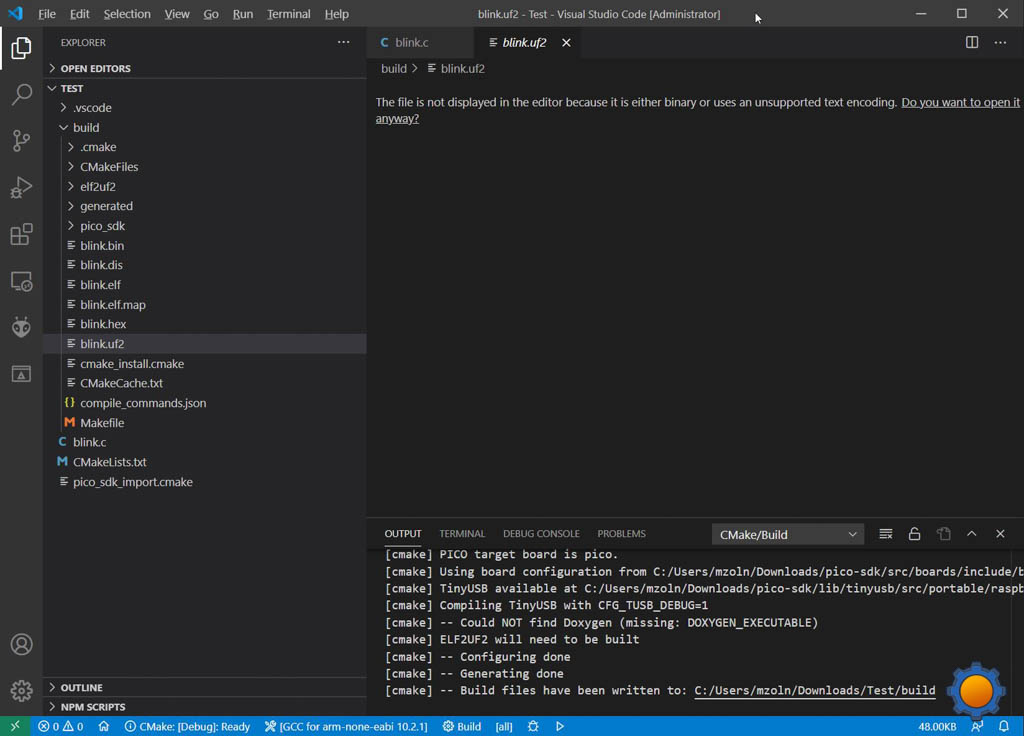


Vytvořte ve svém počítači novou složku pro testovací projekt. Pomocí VSC otevřete tuto složku, bude to váš pracovní prostor pro projekt. V dolní části okna VSC vyberte kompilátor (možná budete muset použít možnost skenování) do **GCC pro arm-none-eabi** .



**Testovací program (bliká)**

Místo toho, abych vás zmátl seznamem všeho dostupného pro Raspberry Pi Pico, jsem zeštíhlil soubory potřebné pro kompilace. Brzy vám vysvětlím soubory a funkce. Každý projekt bude ke kompilaci potřebovat alespoň 3 soubory. Váš programový soubor ( ) pokyny pro import SDK pro CMake ( ) a CMake build pokyny ( ).program.cpico\_sdk\_import.cmakeCMakeLists.txt



**blink.c**

Toto je váš programový soubor. Věnujte pozornost jménu, protože pro správné sestavení budete muset tento název použít v jiných souborech. Veškeré vaše programování by mělo jít do souboru, všechny odkazované soubory by měly být také umístěny do stejné složky projektu.name.c

**pico\_sdk\_import.cmake**

Toto je sada instrukcí pro CMake, která obsahuje informace o tom, kde je sdk a jak pro vás kód zkompilovat. Toto nebudete muset upravovat, ale soubor musí být přítomen pro každý pracovní prostor. V pracovním prostoru můžete mít několik příkladů svého programu, ale budete potřebovat pouze jeden soubor pico\_sdk\_import.cmake

**blink.c**

Tento soubor budete muset upravit podle svého projektu. CMake potřebuje vědět, jaké soubory má zkompilovat, a tento seznam to říká. Ke každému projektu byste měli připojit následující:

**pico\_sdk\_import.cmake**

**pico\_sdk\_import.cmake**

##### CMakeLists.txt

Tento soubor budete muset upravit podle svého projektu. CMake potřebuje vědět, jaké soubory má zkompilovat, a tento seznam to říká. Ke každému projektu byste měli připojit následující:

project(blink)

add\_executable(blink blink.c)

# Pull in our pico\_stdlib which pulls in commonly used features

target\_link\_libraries(blink pico\_stdlib)

# create map/bin/hex file etc.

pico\_add\_extra\_outputs(blink)

Zvýrazněné sekce musí být změněny na název vašeho souboru. Pokud má váš projekt více souborů a adresářů, bude muset váš soubor CMake odrážet také jeho. Za předpokladu, že vaše struktura projektu vypadá takto:program.c

main.c

-FolderA

--fileA.h

--fileB.c

Váš CMake oddíl, který chcete přidat, by musel vypadat takto:

project(test)

include\_directories(FolderA)

file(GLOB SOURCES "FolderA/\*.c")

add\_executable(test main.c ${SOURCES})

# Pull in our pico\_stdlib which pulls in commonly used features

target\_link\_libraries(test pico\_stdlib)

# create map/bin/hex file etc.

pico\_add\_extra\_outputs(test)

Kompilátor vygeneruje soubor k přenosu na Raspberry Pi Pico.build/test.uf2

**CMakeLists.txt**

**Build**

Naštěstí blikání programů na Raspberry Pi Pico je snadné. Vytvořte a zkompilujte program pomocí tlačítka VSB ve spodní části obrazovky (modrá část) a kompilovaný soubor .uf2 bude k dispozici ve složce sestavení.

Při zapnutí podržte tlačítko Raspberry Pi Pico a otevře se jako přenosná jednotka. Přetáhněte soubor .uf2 a jste připraveni!

## Finále

C / C ++ je trochu nepořádek. Překvapuje mě, že neexistuje integrace PlatformIO ani nic efektivnějšího. Koneckonců, všechny počítače se systémem Windows jsou nejoblíbenějšími stroji, takže má smysl strávit nějaký čas optimalizací tohoto procesu. Strávil jsem příliš mnoho času pokusem přijít na perfektní kombinace nastavení s mým přítelem (Ahoj Dmytro!) A dokážu si jen představit frustrace, které musí průměrný nadšenec Pi projít, aby udělal trochu o C / C ++ ve Windows . Doufám, že se zkušenost zlepší a nemohu se dočkat integrace Arduino IDE. Problémy nebo komentáře? Dejte mi vědět v tomto [vlákně Reddit](https://www.reddit.com/r/Not_Enough_Tech/comments/l48z0f/cc_and_micropython_sdk_for_raspberry_pi_pico_on/) .