ZAČÍNÁME S HERNÍ KONZOLÍ PICOPAD

Rychlá příručka s popisem
a vzorovými kódy

Tato příručka Vás seznámí s herní konzolí Picopad, jejím hardwarovým vybavením a popíše vzorové kódy, které Vás uvedou do základní problematiky programování.

Programování lze provádět dvěma způsoby. Buď pomocí kompileru v rámci PicoLibSDK nebo pomocí PlatformIO pro Visual Studio Code. Pokusíme se ukázat oba způsoby, abyste si mohli vybrat ten, který Vám nejlépe vyhovuje.

Co budeme potřebovat:

HARDWARE:

- herní konzole Picopad, Picopad WiFi nebo Picopad Pro
  - v tomto tutoriálu budeme používat verzi Picopad Pro
  - příručku lze použít pro všechny verze herní konzole, pouze části zabývající se připojením k Wi-Fi je možno vyzkoušet jen u verzí Wi-Fi a Pro
- mikro USB kabel
- základní elektronické součástky, např. rezistory, LED diody, potenciometr apod.

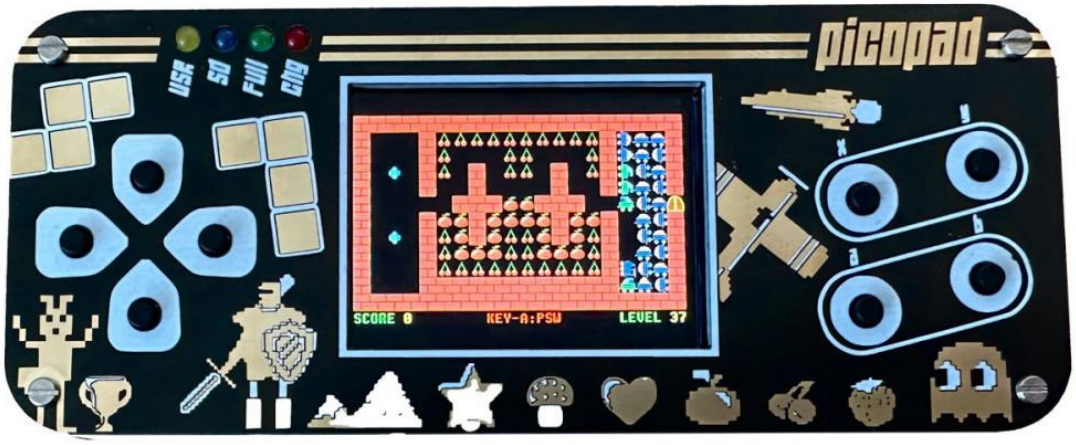
SOFTWARE:

- ARM GCC Compiler
- Git
- Visual Studio Code s nainstalovaným rozšířením PlatformIO, C/C++ Extension Pack, Batch Runner

1. Specifikace

1.1 Picopad

Herní konzole Picopad je založena na jednočipovém počítači Raspberry Pi Pico, který je postavený na mikročipu RP2040.



Specifikace:

- 2“ IPS displej s rozlišením 240x320 pixelů
- slot pro microSD kartu
- ovládací prvky v podobě osového kříže a čtyř ovládacích tlačítek
- přepínač POWER
- tlačítko Reset a BOOTSEL
- stavové LED diody
- reproduktor
- externí konektor
- baterie 500 mAh nabíjená přes modul s TP4056 (ochrana proti přepětí, podpětí a nadproudu)

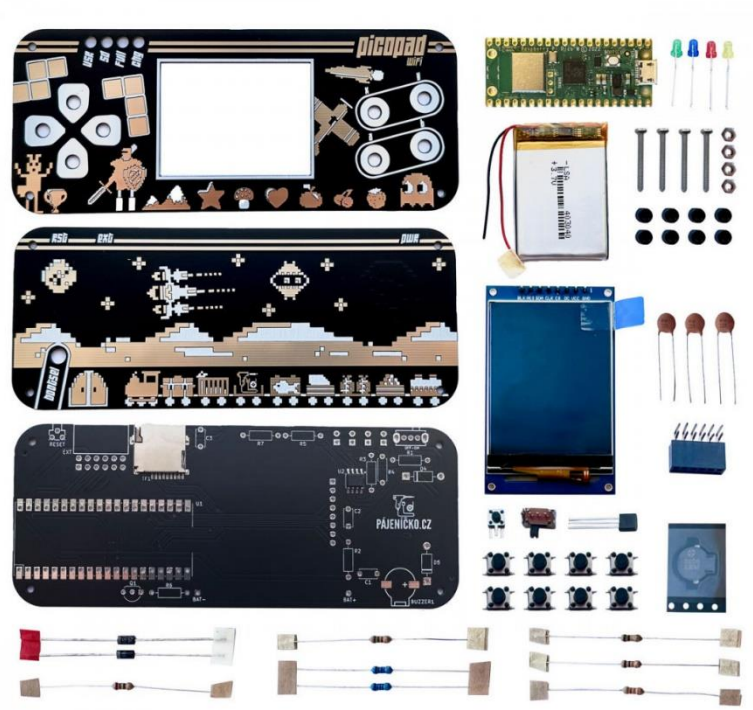
Externí konektor:

Příslušenství:

- USB adaptér
- rámeček

1.2 Picopad Wi-Fi

Herní konzole Picopad Wi-Fi je založena na jednočipovém počítači Raspberry Pi Pico W, který je postavený na mikročipu RP2040.



Specifikace:

- 2“ IPS displej s rozlišením 240x320 pixelů
- slot pro microSD kartu
- ovládací prvky v podobě osového kříže a čtyř ovládacích tlačítek
- přepínač POWER
- tlačítko Reset a BOOTSEL
- stavové LED diody
- reproduktor
- externí konektor
- baterie 500 mAh nabíjená přes modul s TP4056 (ochrana proti přepětí, podpětí a nadproudu)

Konektivita:

- Wi-Fi 802.11n, 2.4 GHz, WPA 3
- Bluetooth 5.2

Externí konektor:

Příslušenství:

- USB adaptér
- rámeček

1.3 Picopad Pro

Herní konzole Picopad Pro je založena na jednočipovém počítači Raspberry Pi Pico W, který je postavený na mikročipu RP2040.

Specifikace:

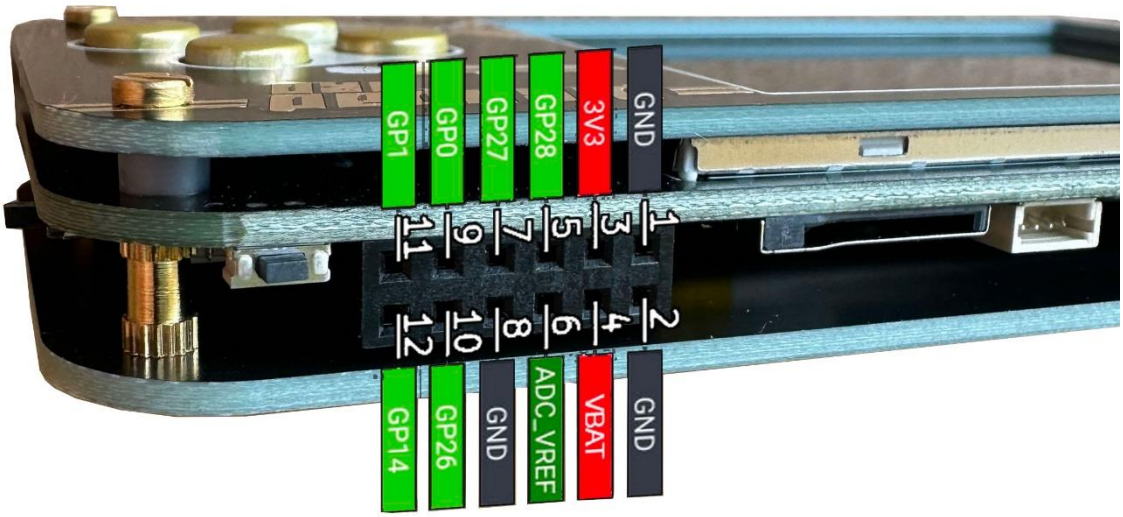
- 2,8“ IPS displej s rozlišením 240x320 pixelů
- slot pro microSD kartu
- ovládací prvky v podobě osového kříže a čtyř ovládacích tlačítek
- přepínač POWER a MUTE
- tlačítko Reset a BOOTSEL
- stavové LED diody
- reproduktor
- 3,5 mm konektor pro sluchátka
- sběrnice Picobus
- externí konektor
- I2C konektor
- baterie 600 mAh, nabíjena přes modul s TP4056 (ochrana proti přepětí, podpětí a nadproudu)

Konektivita:

- Wi-Fi 802.11n, 2.4 GHz, WPA 3
- Bluetooth 5.2

Externí konektor:

U této verze si dejte pozor na zapojení externího konektoru, oproti původní verzi je otočen o 180 °.



Příslušenství:

- videokarta
- USB adaptér
- sada externí baterie
- rámeček

2. Instalace vývojového prostředí

V první řadě si stáhneme potřebnou softwarovou výbavu:

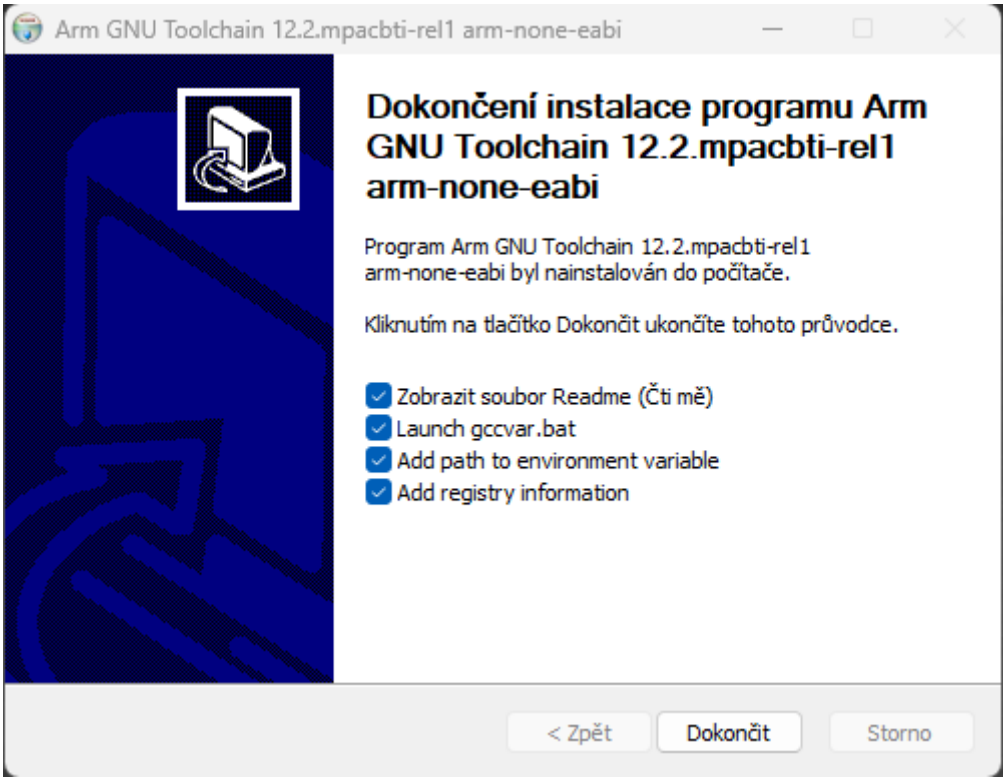
- ARM GCC Compiler
- git
- Visual Studio Code

Kompilátory spolu s programy vývojového prostředí doporučujeme instalovat buď přímo na systémový disk nebo do jednoho adresáře, např. C:\development. Instalace do hlubší adresářové struktury by mohlo vést k nesprávnému chování vývojového prostředí. Pro repositáře a knihovny lze vytvořit např. adresář sdk v cestě C:\development. Zachováme tím větší přehlednost celého prostředí.

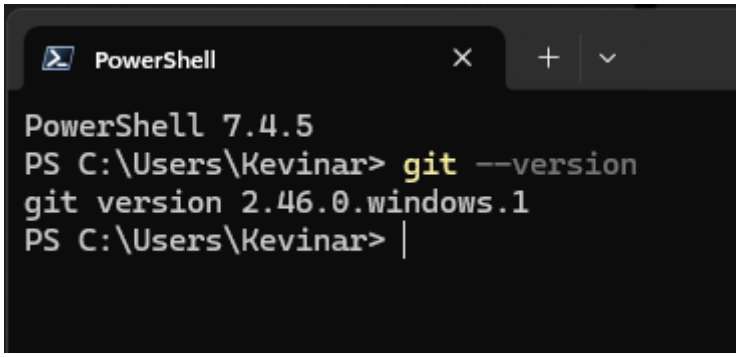
Máme-li vše staženo, začneme instalací:

1) ARM GCC Compiler:
   - během instalace nezapomeňte zaškrtnout možnost *Add path to environment variable*
   - zkontrolujte přidání *C:\Development\arm-none-eabi\bin* do uživatelských proměnných PATH (cestu upravte dle svých potřeb)
   - ověření správné instalace provedete z příkazového řádku zadáním příkazu

     arm-none-eabi-gcc –version



2) Git:
   - při výběru komponent zkontrolujte zaškrtnutí u *Check daily for Git for Windows Update*, *(NEW!) Add a Git bash Profile to Windows Terminal* a *(NEW!) Scalar (Git add-on to manage large- scale repositories)*
   - jako výchozí editor Git zvolte vámi preferovanou možnost
   - u výběru názvů pro nové branch doporučujeme nastavit možnost *Override the default branch name for new repositories*
   - pro nastavení PATH prostředí nechte předvolenou volbu *Git from the command line and also from 3-rd party software*
   - SSH klienta zvolte dle svých preferencí, stejně tak v dalším kroku knihovnu SSL/TSL
   - konfiguraci převodu konců řádku ponechte v systémech Windows ve výchozím nastavení *Checkout Windows-style, commit Unix-style line ending*
   - emulátor terminálu ponechte ve výchozím nastavení *Use MinTTY (the default terminal of MSYS2)*
   - chování příkazu 'git pull' je doporučeno nastavit na *Fast-foward or merge*
   - správce hesel nechte nastaven na *Git Credencial Manager*
   - při výběru extra možností zkontrolujte zaškrtnutí volby *Enable file system caching*
   - u experimentálních možností nezaškrtávejte žádnou volbu
   - po dokončení instalace ověřte její správný průběh příkazem *git --version*



   - příkazem *git config --system core.longpaths true* povolte dlouhé cesty k souborům
   - restartujte počítač

3) Visual Studio Code:
   - instaluje se jako běžný program
   - nainstalujte rozšíření PlatformIO
   - nainstalujte rozšíření C/C++ Extension Pack
   - nainstalujete rozšíření Batch Runner, umožňuje spouštět dávkové soubory v terminálu Visual Studio Code

4) Otevřeme PowerShell a zadáme příkaz

winget install Microsoft.VisualStudio.2022.BuildTools --force --override "--wait --passive --add Microsoft.VisualStudio.Workload.VCTools --add Microsoft.VisualStudio.Component.VC.Tools.x86.x64 --add Microsoft.VisualStudio.Component.Windows11SDK.22000"

Příkazem spustíme instalaci Microsoft Visual Studio Tools. Ten je potřeba pro správnou funkci IntelliSense ve Visual Studio Code.

5) Klonování potřebných úložišť:

a) pico-sdk

Pro tento repositář doporučujeme vytvořit adresář pico, do kterého budeme klonovat. Přejděte do něj, spusťte PowerShell a zadejte příkaz

git clone --recurse-submodules https://github.com/raspberrypi/pico-sdk.git

Přejděte do nově vytvořeného adresáře pico-sdk příkazem *cd pico-sdk* a proveďte kontrolu všech stažených submodulů pomocí

git submodule update --init --recursive

Abychom nemuseli knihovnu kopírovat do každého nového projektu, nastavíme její cestu do proměnného prostředí PATH ve Windows. Spusťte PowerShell a zadejte příkaz

setx PICO_SDK_PATH "C:\development\sdk\pico\pico-sdk"

Cestu ke knihovně Pico SDK samozřejmě upravte podle své potřeby.

b) PicoLibSDK

Přejděte do adresáře, kam chcete repositář klonovat, spusťte PowerShell a zadejte příkaz

git clone --recurse-submodules https://github.com/Panda381/PicoLibSDK.git

c) Picopad

Přejděte do adresáře, kam chcete repositář klonovat, spusťte PowerShell a zadejte příkaz

git clone --recurse-submodules https://github.com/Pajenicko/Picopad.git

Nyní máme naklonovány všechny repositáře, které budeme potřebovat.

3. Pracujeme s knihovnou PicoLibSDK

V knihovně PicoLibSDK najdete několik adresářů, které mají tento význam:

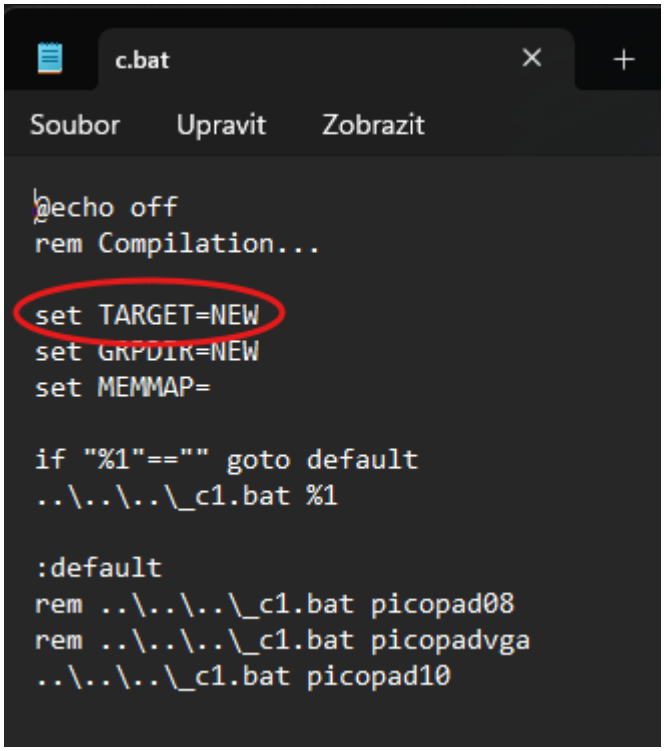
!<název adresáře>	kompilované ukázkové programy
_<název adresáře>	interní soubory knihovny
<název adresáře>	zdrojové kódy ukázkových programů

Adresář v cestě PicoLibSDK\PicoPad\NEW\NEW je tzv. vzorový a slouží pro vytváření nových projektů. Pokud budeme chtít začít s novým projektem, stačí vytvořit její kopii a pojmenovat ji dle potřeby.

Uvnitř adresáře nalezneme několik dávkových souborů:

c.bat	kompilace programu
d.bat	smazání dočasných souborů kompilace
e.bat	odeslání kompilovaného programu do zařízení
a.bat	provede všechny předešlé popsané činnosti

Před samotnou kompilací nového programu je potřeba upravit název projektu i v samotných dávkových souborech. Stačí je otevřít v jakémkoliv editačním programu (např. Poznámkový blok) a upravit část za rovnítkem u položky *set TARGET*. Název projektu nesmí obsahovat více jak 8 znaků.

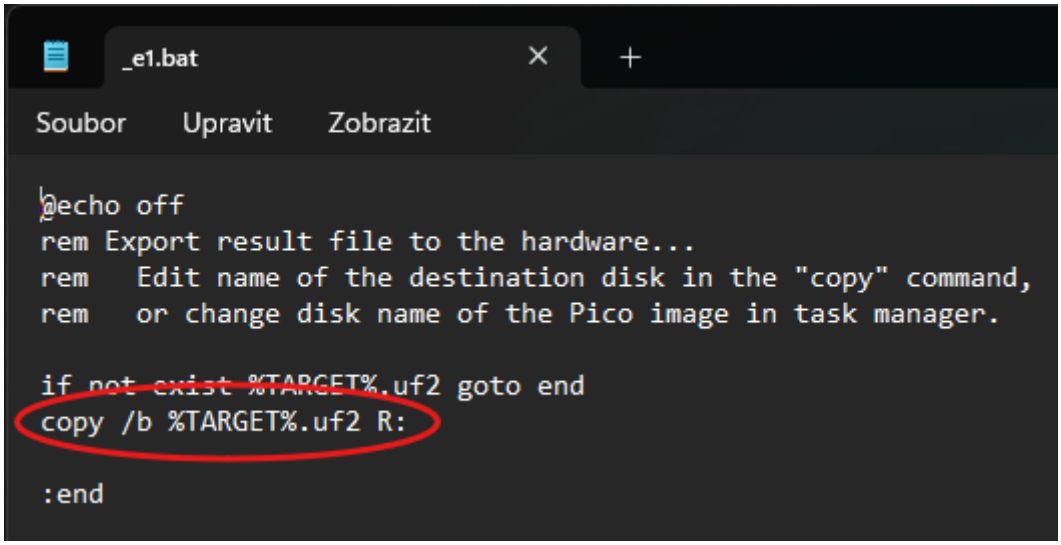


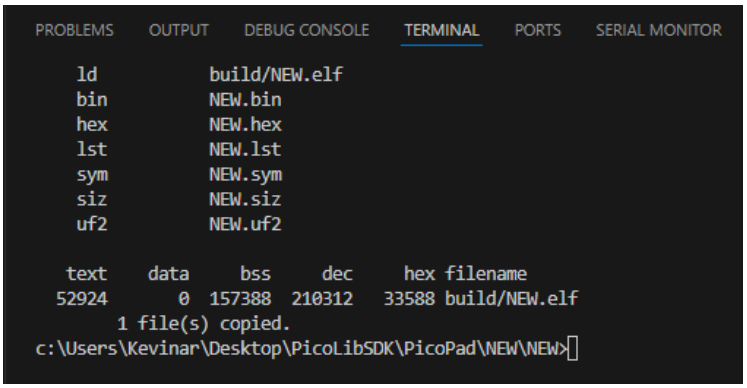
Dále uvnitř adresáře najdeme složku *src*. V ní jsou vloženy soubory *main.cpp* a *main.h*, soubor zdrojového kódu a hlavičkový soubor. Do těchto souborů budeme zapisovat zdrojový C/C++ kód programu.

Nahrání kompilovaného programu do herní konzole Picopad:

V režimu BOOTSEL se herní konzole připojuje k počítači jako USB Mass Storage. K nahrání programu stačí kompilovaný soubor *.UF2 na toto úložiště kopírovat. Po zkopírování se herní konzole restartuje a dojde k nahrání programu. Ten následně spustíme stiskem klávesy Y.

Pro automatizaci procesu kopírování slouží dávkový soubor *_e1.bat*, který se nachází v kořenovém adresáři PicoLibSDK. V něm je přednastaven parametr kopírování pomocí příkazu *copy /b %TARGET%.uf2 R:*, kde R je systémem Windows přiřazené písmeno disku. Upravte písmeno disku dle vašich potřeb.





3.1 Nástroje knihovny PicoLibSDK

Nástroje knihovny nalezneme ve složce *_tools*. Pro činnosti spjaté s herní konzolí Picopad nás budou zajímat především nástroje:

PicoPadImg	převod *.bmp obrázků do C pole
RaspPicoSnd	převod zvukových souborů do C pole

PicoPadImg:

- vstupní soubor musí být ve formátu Windows BMP s bitovou hloubkou 4-bit, 8-bit nebo 24-bit
- konverze se spouští dávkovým souborem *test.bat*, který je potřeba před spuštěním editovat dle potřeb

Generování nekomprimovaného obrázku:

PicoPadImg input.bmp output.c name

Generování komprimovaného RLE obrázku:

PicoPadImg input.bmp output.c name r

RaspPicoSnd:

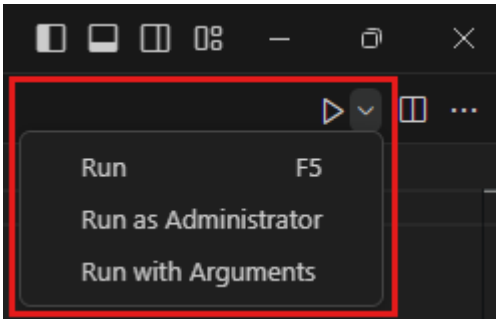
- vstupní soubor musí být ve formátu PCM, mono, 8-bit unsigned, vzorkovací frekvence 22 050 Hz nebo komprimovaný IMA ADPCM, vzorkovací frekvence 22 050 Hz
- konverze se spouští dávkovým souborem *test.bat*, který je před spuštěním potřeba editovat podle svých potřeb

Generování zvukového pole:

RaspPicoSnd input.bmp output.c name

3.2 Ověření funkčnosti prostředí

1) Připojte Picopad k počítači v režimu BOOTSEL
2) Otevřete složku *NEW* v editoru zdrojového kódu Visual Studio Code
   Pokud jste již nainstalovali rozšíření Batch Runner, měli byste po označení jednoho z dávkových souborů v pravém horním roku vidět ikonu *"Play"*



3) Po levé straně, v stromové struktuře souborů a adresářů, označte dávkový soubor s názvem *a.bat*
4) Klikněte na ikonu *"Play"*

Provede se kompilace programu s následným nahráním do konzole Picopad. Program nyní můžete spustit stiskem klávesy Y.

4. Pracujeme s PlatformIO

PlatformIO je multiplatformní nástroj určený pro vývoj softwaru embedded systémů a instaluje se jako rozšíření editoru zdrojového kódu Visual Studio Code.

Pomocí PlatformIO můžeme Picopad programovat v Arduino kódu. Pro lepší pochopení práce s PlatformIO doporučujeme pročíst příručku PlatformIO for VSCode.

Vzorový projekt s názvem *template* nalezneme v dříve naklonovaném repositáři v cestě *Picopad\arduino\template*. Projekt obsahuje kód Hello World a je připraven k okamžité kompilaci.

Před samotnou kompilací si popíšeme adresářovou strukturu projektu:

assets	soubory v něm obsažené budou po kompilaci vloženy do adresáře *build*
build	po kompilaci obsahuje *.UF2 soubor a soubory z adresáře *assets*
include	hlavičkové soubory projektu
lib	knihovny projektu
patches	specifické patch soubory pro Picopad
src	zdrojový kód programu
tools	specifické nástroje potřebné ke kompilaci programu

Pro nastavení vývojového prostření projektu se používá soubor *platformio.ini*. Ten lze použít ke konfiguraci více platforem a architektur najednou.

Vzorový obsah souboru platformio.ini:

```
[env]
name = TEMPLATE
platform = https://github.com/maxgerhardt/platform-raspberrypi.git
board = rpipicow
framework = arduino
board_build.core = earlephilhower
board_build.filesystem_size = 0m
monitor_speed = 115200

[env:picopad]
extra_scripts =
  pre:tools/patch_memmap.py
  post:tools/copy_build.py

[env:pico]
extra_scripts =
  pre:tools/revert_patch_memmap.py
  post:tools/copy_build.py
```

Pokud máte herní konzoly verze Picopad, budete mít vždy nakonfigurovaný typ vývojové desky na *pico* (Raspberry Pi Pico). Používáte-li herní konzoly verze Wi-Fi nebo Pro, můžete nakonfigurovat typ vývojové desky na *rpipicow* (Raspberry Pi Pico W). Poté budete moci ve svých projektech využívat i Wi-Fi připojení.

Chcete-li ve svém projektu použít externí knihovnu, můžete ji do projektu zahrnout nastavením závislosti pomocí parametru lib_deps. Je-li závislost společná pro všechna vývojová prostředí konfigurujte ji v sekci *[env]*. Je-li závislost potřebná pouze v jednom z několika vývojových prostředí, konfigurujte ji v sekci *[env]* daného vývojového prostředí např: *[env:picopad]*.

Ovládací prvky rozhraní:

1. zobrazení počtu chyb či varování
2. PlatformIO: Home — zobrazení domovské stránky PlatformIO
3. PlatformIO: Build — kompilace programu
4. PlatformIO: Upload — nahrání programu do zařízení
5. PlatformIO: Clean — vyčištění kompilačních souborů
6. Serial Port Monitor — otevře Seriál Port Monitor
7. PlatformIO: New Terminal — otevře nový terminál
8. Switch PaltformIO Project Environment — výběr vývojového prostředí
9. Set upload/monitor/test port — nastavení COM portu zařízení

Tlačítka toolbaru je možno skrýt/zobrazit pomocí kontextového menu vyvolaného stiskem pravého tlačítka myši. Zároveň je možno jejich funkci editovat v nastavení rozšíření, čímž je umožněno vytvořit toolbar vlastní.

Nahrání kompilovaného programu do herní konzole Picopad:

V režimu BOOTSEL se herní konzole připojuje k počítači jako USB Mass Storage. K nahrání programu stačí kompilovaný soubor *.UF2 na toto úložiště kopírovat. Po zkopírování se herní konzole restartuje a dojde k nahrání programu. Ten následně spustíme stiskem klávesy Y.

Pro kopírování kompilovaného programu z prostředí Visual Studio Code použijeme příkazu copy. Zjistíme, jaké je v systému Windows velkokapacitnímu médiu přirazeno písmeno disku a poté do terminálu zadáme příkaz

copy <cesta k souboru>.uf2 <jednotka>:*

kde:

<cesta k souboru>	cesta ke kompilovanému souboru *.UF2
<jednotka>	systémem Windows přiřazené písmeno disku

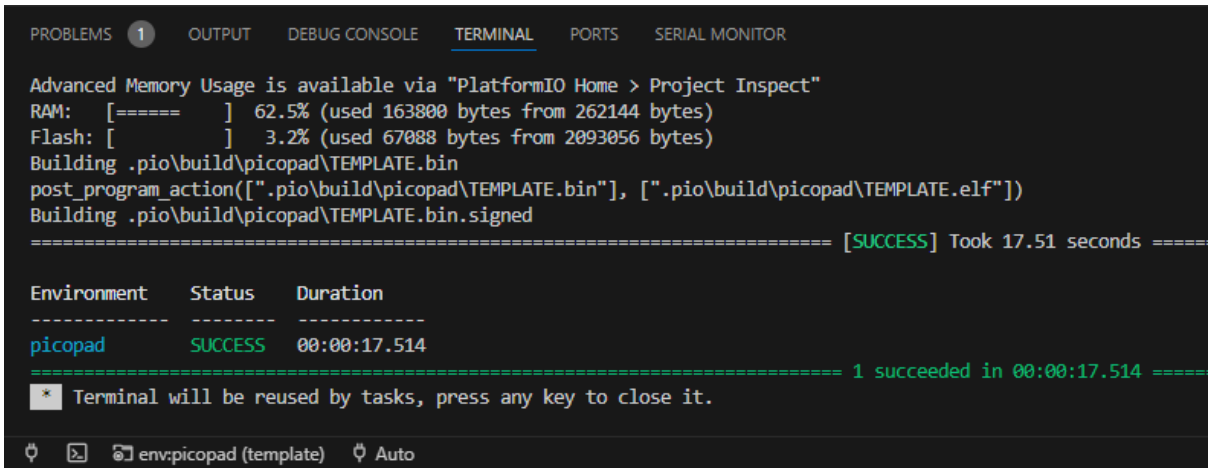
příklad tvaru příkazu

copy build.uf2 J:*

V případě, že bychom před dalším nahráním programu zadávali do terminálu příkazy jiné, není potřeba příkaz *copy* znovu psát, lze jej "nalistovat" pomocí kurzorových šipek nahoru a dolů.

4.1 Ověření funkčnosti prostředí

1) Otevřete složku *template* v editoru zdrojového kódu Visual Studio Code
2) Zvolíme vývojové prostředí projektu *env:picopad*
3) Stiskneme tlačítko *Build* k provedení kompilace programu



4) Připojíme herní konzoly Picopad k počítači v režimu BOOTSEL
5) V panelu rozhraní se přepneme do terminálu (*pwsh*)a spustíme příkaz pro kopírování souboru *copy build/*uf.2 J:*
6) Provede se zkopírování programu do konzole Picopad. Program nyní můžete spustit stiskem klávesy Y.