

Raspberry Pi Pico

ladenie pomocou bežných ladiacich
nástrojov

Eduard Drusa

nopisonnope@gmail.com

February 23, 2022

Motivácia

Každý program obsahuje chyby

Motivácia

Každý program obsahuje chyby

Debuggery sa pri vývoji bežne používajú

Motivácia

Každý program obsahuje chyby

Debuggery sa pri vývoji bežne používajú

Arduino tento aspekt veľmi dlho ignorovalo

Motivácia

Každý program obsahuje chyby

Debuggery sa pri vývoji bežne používajú

Arduino tento aspekt veľmi dlho ignorovalo

Prezentovaný postup je prenositeľný

O mne

Absolvent KPI FEI TUKE

13 rokov profesiou programátor (C, C++)

18 rokov na Linuxe

ex-Autodesk, ex-Webasto

LADENIE

Prvý zaznamenaný počítačový bug na systéme Harvard Mark II

História

Prvý zaznamenaný počítačový bug na systéme Harvard Mark II

V mechanickom relé sa zasekla moľa

Prvý zaznamenaný počítačový bug na systéme Harvard Mark II

V mechanickom relé sa zasekla mola

Nejednalo sa o SW bug, ale podľa Grace Hopper sa odvtedy všetky SW problémy označovali pojmom "bug"

Breakpoint umožňuje automaticky zastaviť beh programu, ak sa vykoná špecifická časť kódu

Breakpoint umožňuje automaticky zastaviť beh programu, ak sa vykoná špecifická časť kódu

Stepping je proces behu programu po malých krokoch

Breakpoint umožňuje automaticky zastaviť beh programu, ak sa vykoná špecifická časť kódu

Stepping je proces behu programu po malých krokoch

Watch zobrazí hodnotu výrazu v čase, keď je beh programu zastavený

Breakpoint umožňuje automaticky zastaviť beh programu, ak sa vykoná špecifická časť kódu

Stepping je proces behu programu po malých krokoch

Watch zobrazí hodnotu výrazu v čase, keď je beh programu zastavený

Stack (trace) sprístupňuje cestu, ktorou sa program dostal k aktuálnemu miestu vykonávania

Bonusové vlastnosti

Pád programu je zachytený debuggerom. Vývojár môže skúmať stav programu v čase pádu.

Bonusové vlastnosti

Pád programu je zachytený debuggerom. Vývojár môže skúmať stav programu v čase pádu.

Core dump uloží stav procesu, ktorý nebeží v debuggeri v čase pádu. Umožňuje sa tým tzv. posmrtná analýza (post-mortem)



EXTENSIONS



Search Extensions in Marketplace

INSTALLED

3



C/C++

C/C++ IntelliSense, debugg...

Microsoft



CMake Tools

Extended CMake support i...

Microsoft

45ms



RECOMMENDED

4

Ubuntu 20.04 LTS Visual Studio Code



ventyl@ghost: ~



```
ventyl@ghost:~$ sudo apt-get install build-essential cmake code
```

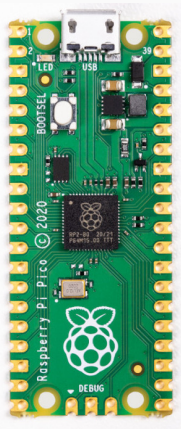
SHOWTIME

RASPBERRY PI PICO

A ladenie embedded zariadení

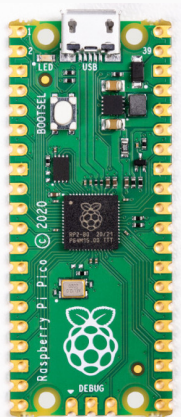
Raspberry Pi Pico

32-bitový jednočipový mikropočítač na báze ARM Cortex-M0+ a 256kB RAM



Raspberry Pi Pico

32-bitový jednočipový mikropočítač na báze ARM Cortex-M0+ a 256kB RAM
Externá 2MB flash

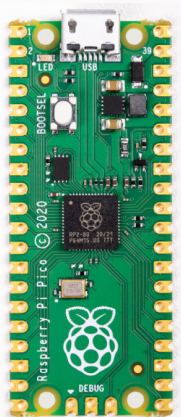


Raspberry Pi Pico

32-bitový jednočipový mikropočítač na báze ARM Cortex-M0+ a 256kB RAM

Externá 2MB flash

Pestrá paleta periférii vrátane USB hosta



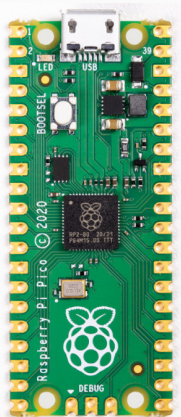
Raspberry Pi Pico

32-bitový jednočipový mikropočítač na báze ARM Cortex-M0+ a 256kB RAM

Externá 2MB flash

Pestrá paleta periférii vrátane USB hosta

PIO - Programmable I/O peripheral



Ladenie embedded zariadení

Softwareový **Debugger** potrebuje asistenciu hardwareoveho debuggera (ST-Link, Atmel-ICE, J-Link, Renesas E1, ...)

Ladenie embedded zariadení

Softwareový **Debugger** potrebuje asistenciu hardwareoveho debuggera (ST-Link, Atmel-ICE, J-Link, Renesas E1, ...)

Na PC zabezpečí načítanie, spustenie, zastavenie a prístup k RAM operačný systém, embedded spravidla nič také nemá



EXTENSIONS



Search Extensions in Marketplace

INSTALLED

3

**C/C++**

C/C++ IntelliSense, debugg...

Microsoft

**CMake Tools**

Extended CMake support i...

Microsoft

73ms

**Cortex-Debug**

ARM Cortex-M GDB Debug...

marus25

67ms



RECOMMENDED

4

Prekladač pre ARM
Debugger pre rôzne
architektúry
Git a nástroje na build
OpenOCD
Plugin pre VS Code na
podporu HW debuggerov



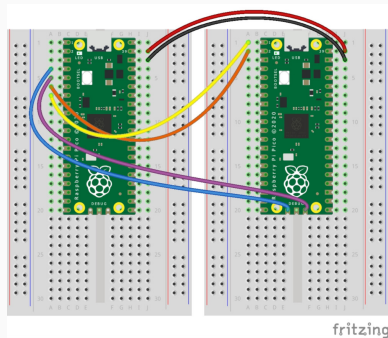
ventyl@ghost: ~



```
ventyl@ghost:~$ sudo apt-get install gcc-arm-none-eabi gdb-multiarch git libtool  
libusb-1.0-0 libusb-1.0-0-dev git
```

HW debugger pre Raspberry Pi Pico

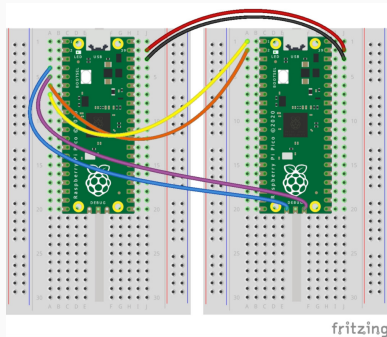
RPi Pico sa da použiť ako HW debugger - Picoprobe



HW debugger pre Raspberry Pi Pico

RPi Pico sa da použiť ako HW debugger - Picoprobe

OpenOCD slúži ako rozhranie medzi SW debuggerom a HW debuggerom

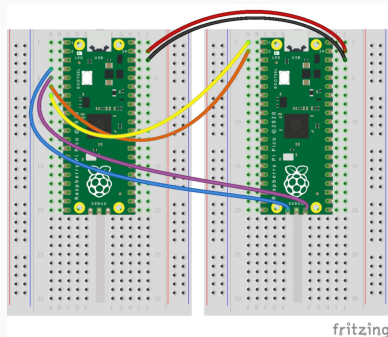


HW debugger pre Raspberry Pi Pico

RPi Pico sa da použiť ako HW debugger - Picoprobe

OpenOCD slúži ako rozhranie medzi SW debuggerom a HW debuggerom

Nanešťastie posledné vydanie OpenOCD vyšlo tesne po ohlásení RPi Pico



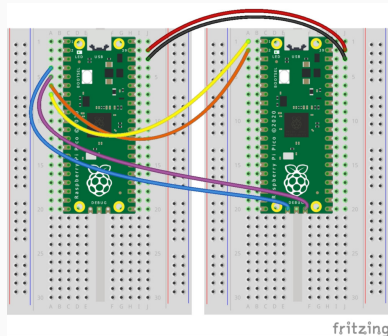
HW debugger pre Raspberry Pi Pico

RPi Pico sa da použiť ako HW debugger - Picoprobe

OpenOCD slúži ako rozhranie medzi SW debuggerom a HW debuggerom

Nanešťastie posledné vydanie OpenOCD vyšlo tesne po ohlásení RPi Pico

Čo znamená, že si ho musíme zostaviť zo zdrojových kódov

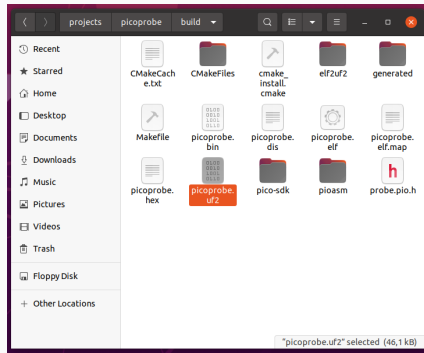


Zostavenie Picoprobe

```
mkdir projects && cd projects  
git clone https://github.com/raspberrypi/pico-sdk.git  
git clone https://github.com/raspberrypi/picoprobe.git  
cd pico-sdk && git submodule update --init  
cd ../picoprobe && mkdir build && cd build  
cmake -DPICO_SDK_PATH=../../pico-sdk ..  
make -j
```

Upload Picoprobe

Kompilácia porodí súbor
picoprobe.uf2



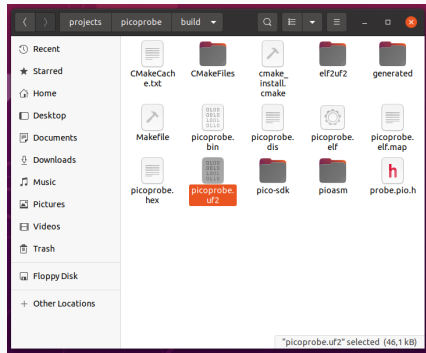
Upload Picoprobe

Kompilácia porodí súbor

picoprobe.uf2

Na RPi Pico stlačiť tlačidlo

BOOTSEL a pripojiť do USB



Upload Picoprobe

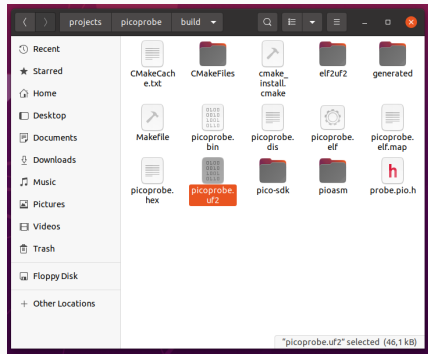
Kompilácia porodí súbor

picoprobe.uf2

Na RPi Pico stlačiť tlačidlo

BOOTSEL a pripojiť do USB

Pico sa bude chovať ako USB kľúč



Upload Picoprobe

Kompilácia porodí súbor
picoprobe.uf2

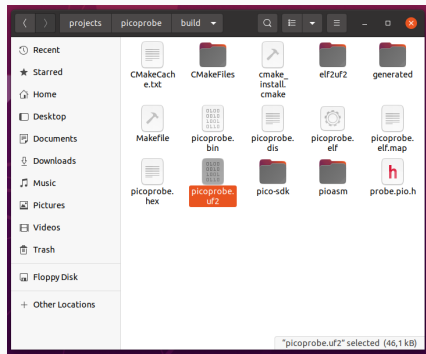
Na RPi Pico stlačiť tlačidlo

BOOTSEL a pripojiť do USB

Pico sa bude chovať ako USB kľúč

Súbor picoprobe.uf2 skopírovať

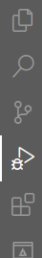
na zväzok na RPi Pico



Zostavenie OpenOCD

```
cd projects
git clone https://github.com/raspberrypi/openocd.git \
--branch picoprobe --depth 1
cd openocd
./bootstrap
./configure --enable-picoprobe
make -j && sudo make install
```

File Edit Selection View Go Run Terminal Help



RUN AN...

Pico Debug



{ } launch.json X

VARIABLES

WATCH

.vscode > { } launch.json > ...

```
1 {
2   "version": "0.2.0",
3   "configurations": [
4     {
5       "name": "Pico Debug",
6       "cwd": "${workspaceRoot}",
7       "executable": "${command:cmake.launchTargetPath}",
8       "request": "launch",
9       "type": "cortex-debug",
10      "serverType": "openocd",
11      // This may need to be arm-none-eabi-gdb depending on your system
12      "gdbPath": "gdb-multiarch",
13      "device": "RP2040",
14      "configFiles": [
15        "interface/picoprobe.cfg",
16        "target/rp2040.cfg"
17      ],
18      "svdFile": "/home/ventyl/projects/pico-sdk/src/rp2040/hardware_regs/rp2040.svd",
19      "runToMain": true,
20      // Work around for stopping at main on restart
21      "postRestartCommands": [
22        "break main",
23        "continue"
24      ],
25      "searchDir": ["/usr/local/share/openocd/scripts"],
26    }
27  ]
28 }
```

SHOWTIME

ZHRNUTIE

A ďalšie možnosti

Zhrnutie

Vytvorili sme si vývojové prostredie "od nuly"

Zhrnutie

Vytvorili sme si vývojové prostredie "od nuly"

Použité nástroje sú univerzálne

Zhrnutie

Vytvorili sme si vývojové prostredie "od nuly"

Použité nástroje sú univerzálne

Prostredie sa dá použiť s inými ARM zariadeniami

Zhrnutie

Vytvorili sme si vývojové prostredie "od nuly"

Použité nástroje sú univerzálne

Prostredie sa dá použiť s inými ARM zariadeniami

Minimálne zmeny umožnia ladiť prakticky akékoľvek zariadenie

Zdroje

Repozitár s príkladmi a konfiguráciou

<https://github.com/ventZl/debugging-pico>