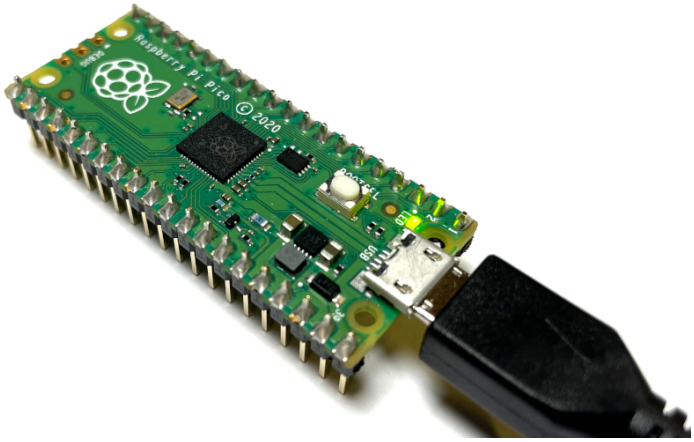
SHAWN HYMEL

How to Set Up Raspberry Pi Pico C/C++ Toolchain on Windows with VS Code

APRIL 9, 2021 / TUTORIAL / 120 COMMENTS



This tutorial will show you how to install the Raspberry Pi Pico toolchain on Windows 10 for C and C++ development.

Raspberry Pi has a fantastic getting started guide for the Pico that covers installation steps for the major operating systems.

CATEGORIES

- Classroom

started guide shows you how to do this, but I have issues with two parts: you need to install Build Tools for Visual Studio (around 6 GB) and you must run VS Code from within a Developer Command Prompt every time.

This guide walks you through an alternative way of installing the C/C++ toolchain for the Pico, using MinGW in place of the Build Tools for Visual Studio.

Table of Contents

Directory Setup

With the exception of CMake and Python, we will want all of our tools and SDK files to exist in one folder on Windows. This setup will make it easy to configure the Path and find things later on.

Create a folder named **VSARM** in the top level of your C drive.

In C:\VSARM, create the following folders:

- C:\VSARM\armcc
- C:\VSARM\lib

TAGS

ARDUINO ARM BLE BLUETOOTH C C++ CAD CLASSROOM CUDA DEBIAN EASYEDA EDISON ELECTRONICS ESP32 GAMEDEV HTML INTEL IOT ISR JAVASCRIPT LED LINUX MACHINE LEARNING MRAA NVIDIA OLIMEX OLINUXINO PCB PHASER PROGRAMMING PYTHON QUICK TIPS RASPBERRY PI PICO SAMD21 SD CARD SERVICE SPARKFUN

SHAWN HYMEL

coexist with other tools and SDKs.

We use these short, one-word folder names to keep paths short and to avoid spaces. Some Unix/Linux tools (that have been ported to Windows) do not work well with paths that have spaces in them.

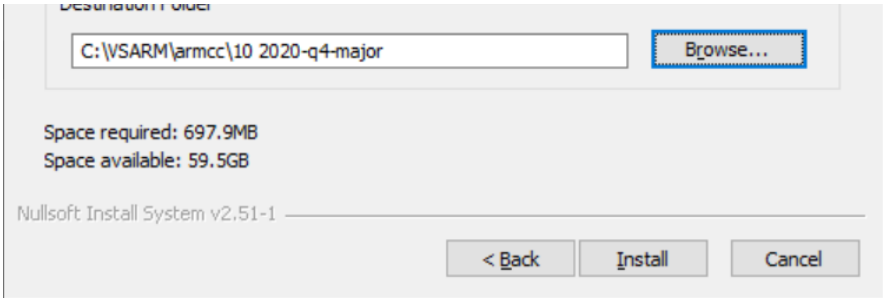
Install GNU Arm Embedded Toolchain

The GNU Arm Embedded Toolchain contains the Arm GCC compiler that we need to compile C and C++ code for the RP2040.

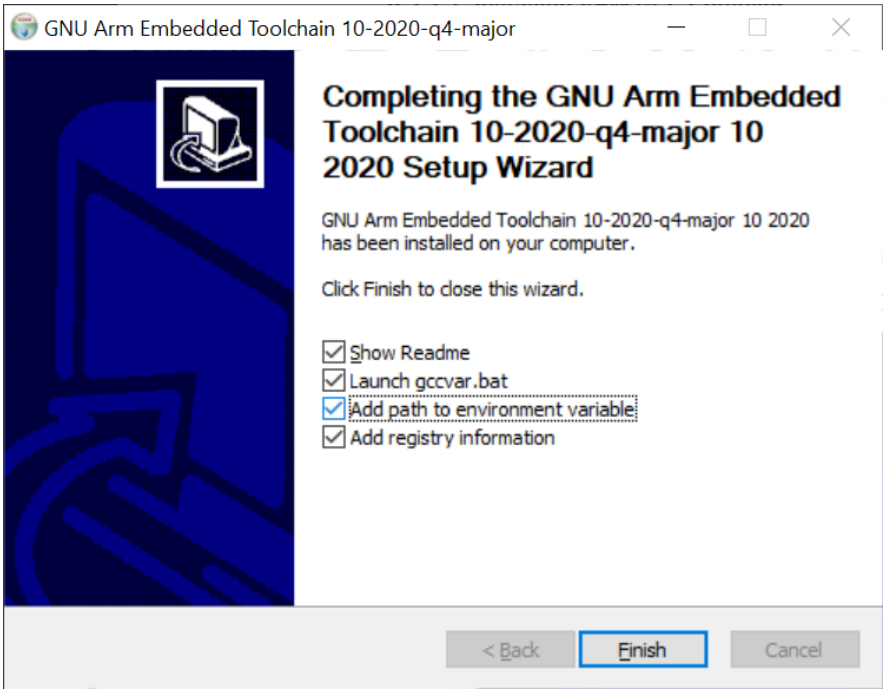
Head to the GNU Arm Embedded Toolchain download page and download the latest installer for Windows. For me, this was *gcc-arm-none-eabi-10-2020-q4-major-win32.exe*.

[Update Apr 19, 2022] I have verified that Arm GNU Embedded Toolchain 11.2-2022.02 works.

Run the installer. When asked, change the installation location to **C:\VSARM\armcc**. It will auto-fill the destination directory to the name of the current toolchain release.



Continue with the installation process. When it is complete, select the option to **Add path to environment variable**.



At this point, you should be able to call any of the Arm compiler tools in a new command prompt, such as *arm-none-eabi-gcc.exe*.

Install MinGW-w64 GCC Tools

MinGW (short for Minimalist GNU for Windows) is a collection of open-source utilities, such as compilers and linkers, that allow us to build applications for Windows.

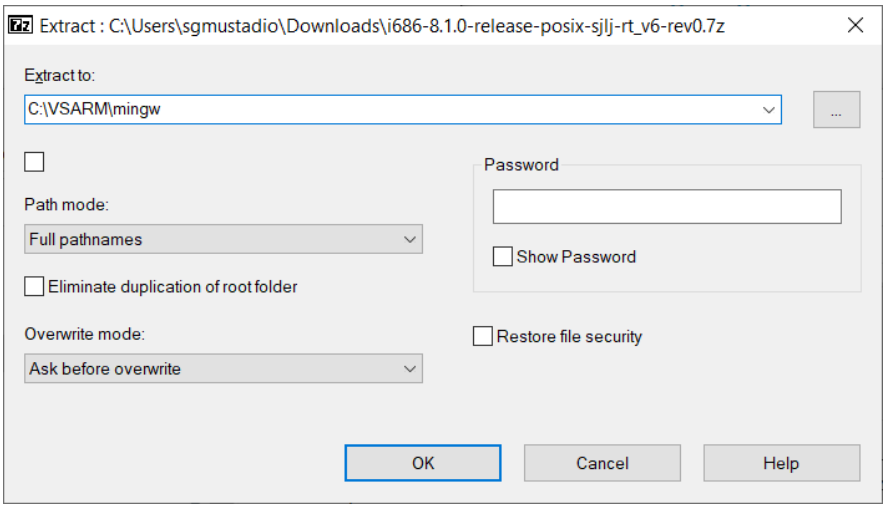
SHAWN HYMEL

[Updated Apr 19, 2022] Note: at this time, the MinGW (.exe) installer appears to be broken. You will likely see the error message "The file has been downloaded incorrectly!" As a result, I have updated the portion below to download only the MinGW-W64 GCC files (tested with v8.1.0), as that's all we need to compile Pico programs.

Head to the MinGW-W64 files page on SourceForge: https://sourceforge.net/projects/mingw-w64/files/.

Download the **i686-posix-sjlj** zip file for your desired MinGW-W64 GCC version (e.g. 8.1.0).

Use 7-Zip to unzip it into the **C:\VSARM\mingw** directory. Uncheck the named unzip directory so that when everything unzips, you should have C:\VSARM\mingw\mingw32.



When it's done, open a Windows Command Prompt and enter the following into the terminal:

mingw32\bin (along with this .bat file) in a later step.

Install CMake

CMake is a tool that helps you automate the build process of programs. It does not build/compile (like Make does), but rather, it can generate the directory structures and files needed for any build system (Make, Qt Creator, Ninja, etc.). The Raspberry Pi Pico SDK relies on CMake to help create these build files.

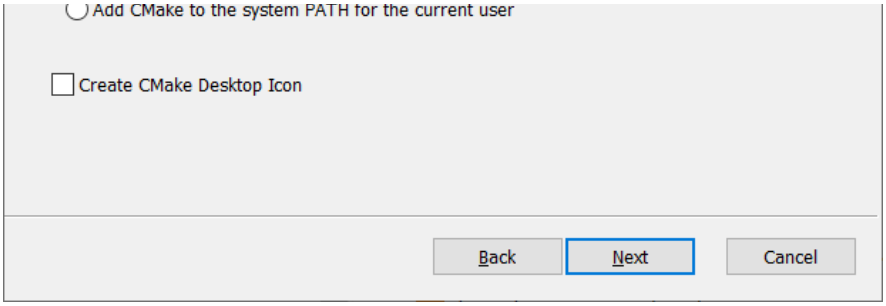
Head to the download page on CMake's site.

Important! There is a bug in CMake version 3.20 (at the time of writing). On the second run of make or nmake (after running cmake), the process will fail. If you're using nmake, you'll get an error like `fatal error U1033: syntax error : ':' unexpected`, or if you're using mingw32-make, something like `*** multiple target patterns. Stop.` To prevent this, install **CMake version 3.19**. Future versions of CMake may fix this bug, but for now, know that version 3.19 worked for me.

[Update Apr 19, 2022] I have verified that CMake 3.23.1 now works. The bug has been fixed.

Download the version 3.19.8 installer for Windows (*cmake-3.19.8-win64-x64.msi*).

Run the installer and accept the user license. On *Install Options*, select **Add CMake to the system PATH for all users**.



Continue the installation process, accepting all the defaults. Note that this will install CMake to C:\Program Files\CMake, which is fine, as it will be used as a system-wide tool (not just for VSARM projects).

Install Python

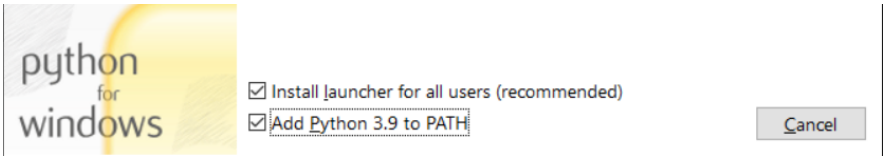
The Pico SDK relies on Python to script and automate some of the build functions.

Important! At the time of writing, the Pico SDK recommends **Python version 3.9**. I do not know if other versions will work.

[Update Apr 19, 2022] I have verified that Python 3.10.2 works.

Head to the downloads page on the Python site. Download the latest Python 3.9 (or 3.10) installer.

Run the installer. On the first screen, make sure that **Install launcher for all users (recommended)** is checked and check **Add Python 3.9 to PATH**.

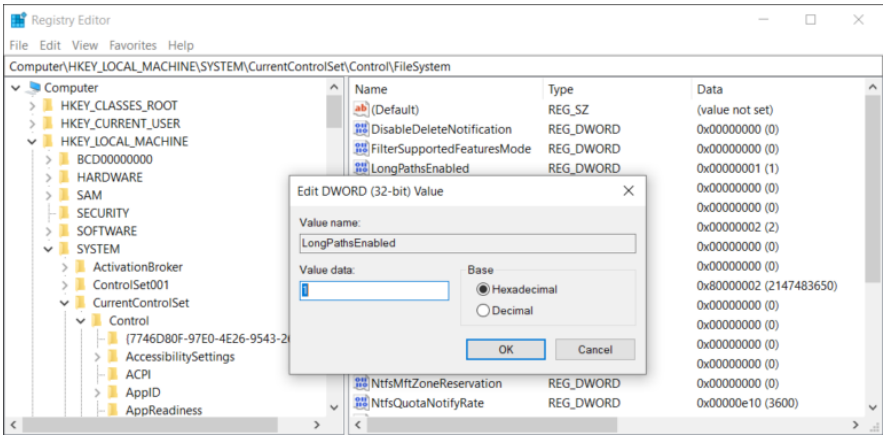


Click **Install Now** and wait while it installs Python.

At the end of the installation process, select the option to **disable the MAX_PATH length limit**.

Important! If you were not asked to disable the MAX_PATH length limit, you will want to make sure that long paths are enabled. The Pico SDK (and many other SDKs) often have long, nested folders, resulting in pathnames that exceed the original Windows limit (260 characters).

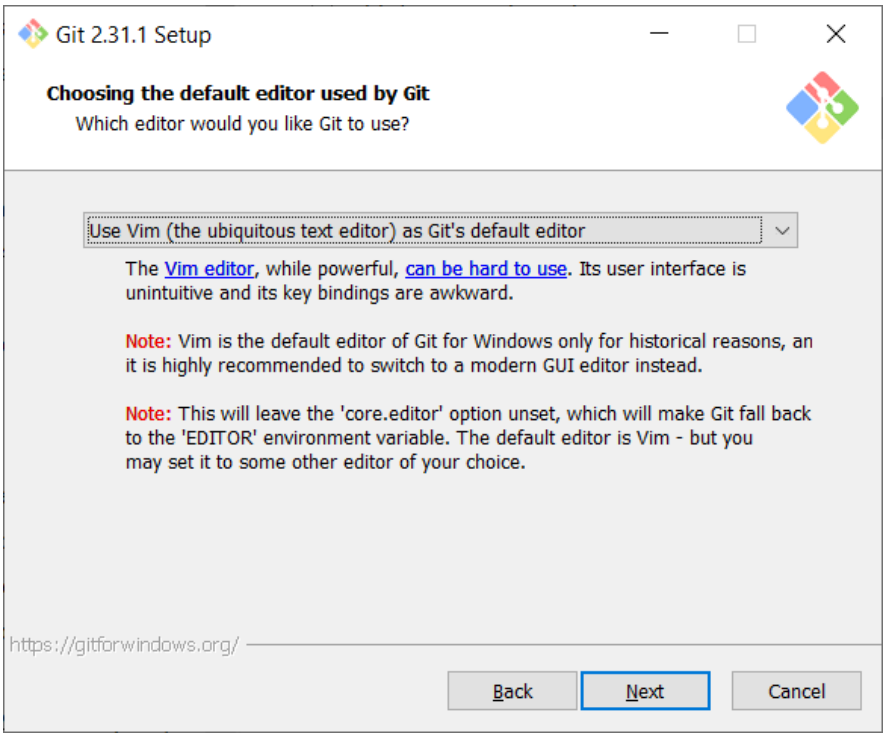
To enable long paths, search for **regedit** in the Windows search bar and run the **Registry Editor** program. Navigate to `Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem` and add an entry (if one is not already there) for `LongPathsEnabled`. Change *Value data* to **1** and *Base* to **Hexadecimal**.



Here is a full guide if you need help modifying the registry to disable the pathname limit.

installer for Windows (e.g. *32-bit Git for Windows Setup*).

Run the installer. When you get to the screen asking you to choose a default editor, feel free to pick whatever you want. I kept *vim* because I know it (barely) well enough to edit git comments.



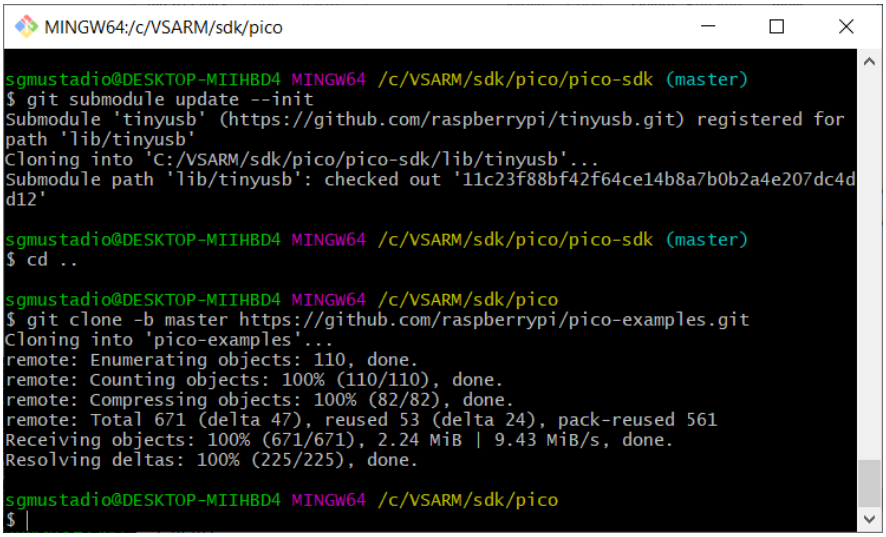
Continue the installation process, accepting all the defaults.

Download Pico SDK and Examples

The Pico SDK conains a collection of tools and libraries used to make developing on the Pico (and RP2040) much easier. We can also download a set of C/C++ examples that are useful demonstrations of how to use the SDK.

To start, create a new folder named **pico** in C:\VSARM\sdk.

```
cd /c/VSARM/sdk/pico
git clone -b master https://github.com/raspber
cd pico-sdk
git submodule update --init
cd ..
git clone -b master https://github.com/raspber
```



At this point, you should have all of the necessary build tools, SDK, and examples installed to start developing programs for the Raspberry Pi Pico and RP2040.

Update Environment Variables

Some of the tools we installed automatically updated the Windows environment variables (specifically, the Path). However, a few (like MinGW and the SDK) did not. We need to update the environment variables so that the shell and various build tools know where to find things in our filesystem.

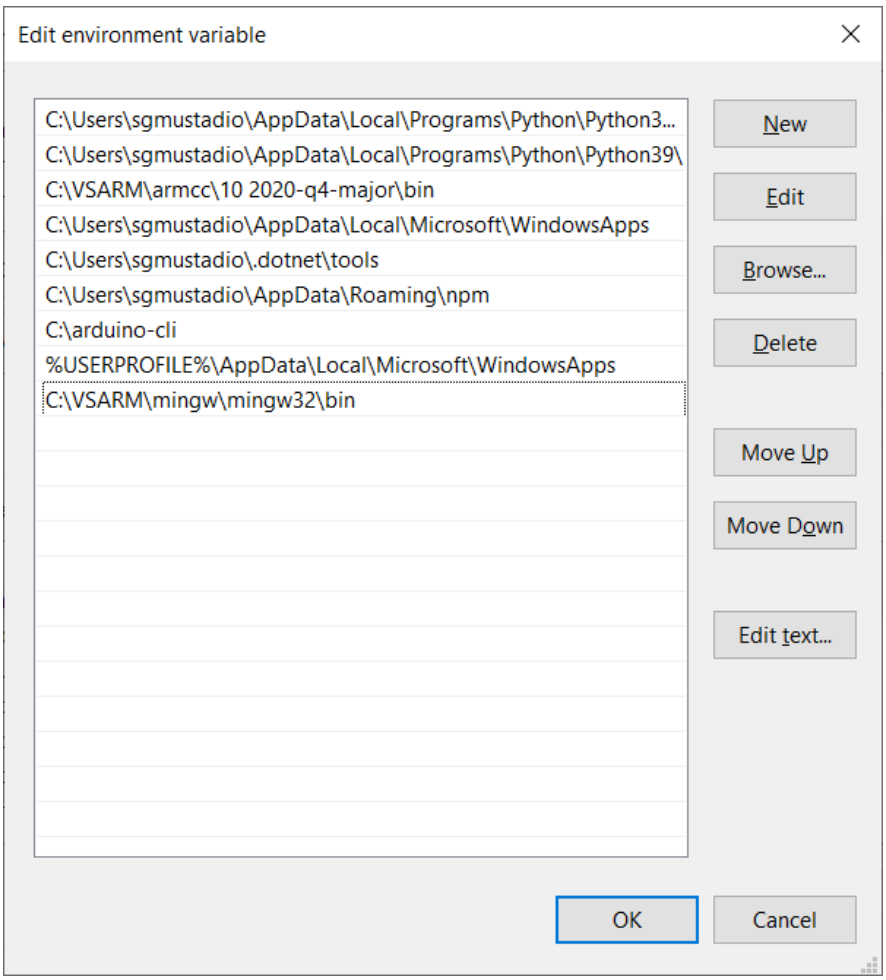
SHAWN HYMEL

Add **C:\VSARM\mingw\mingw32\bin** as a new entry. This will allow us to use things like *gcc* and *ld* to build C and C++ programs for Windows.

Make sure you see the following entries listed:

- C:\VSARM\armcc\<release version>\bin
- C:\VSARM\mingw\mingw32\bin

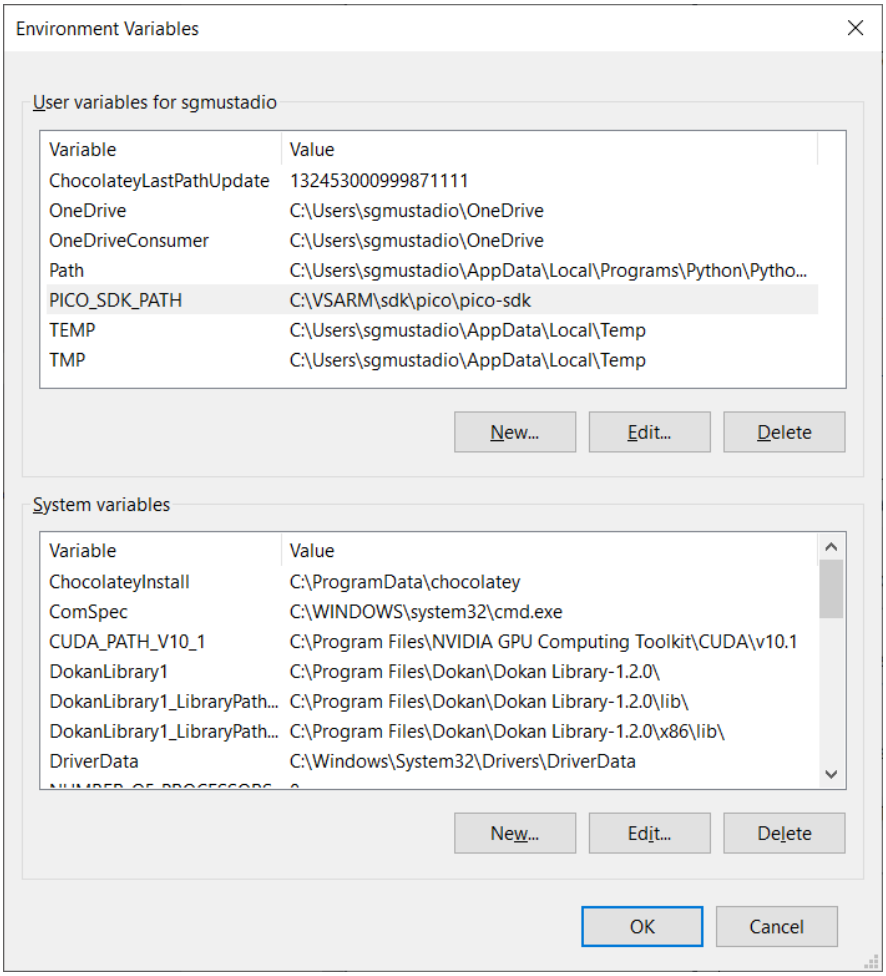
You might see an entry for Python39 if you chose to install Python for the current user (as I did).



Click **OK** to exit the user Path window.

SHAWN HYMEL

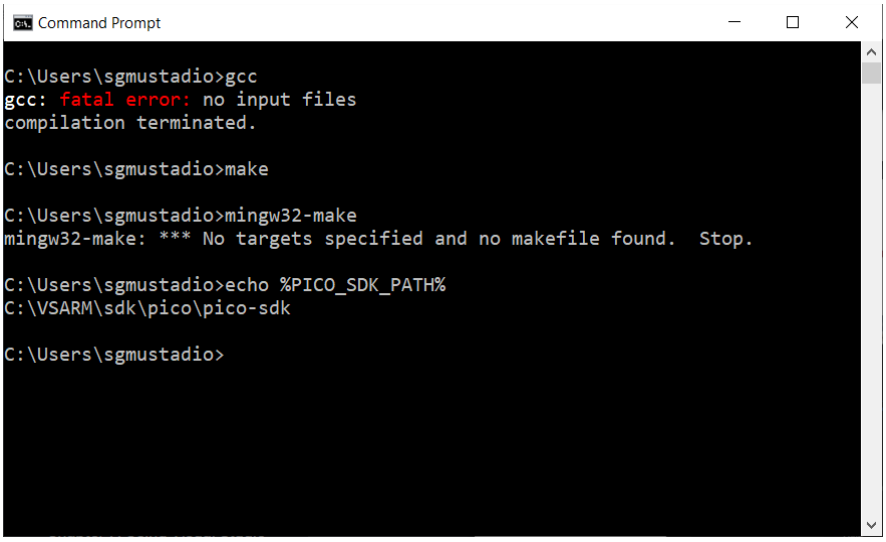
User variables should have an updated Path as well as PICO_SDK_PATH variables.



Under *System variables*, select **Path** and click **Edit**. Check to make sure you see the following entries (add them if you do not see them):

- C:\Program Files\CMake\bin
- C:\Program Files\Git\cmd

These entries should have been automatically added to your system-wide Path when you ran their installers. You might also see an entry for Python39 (or Python310) if you chose to install Python for all users.

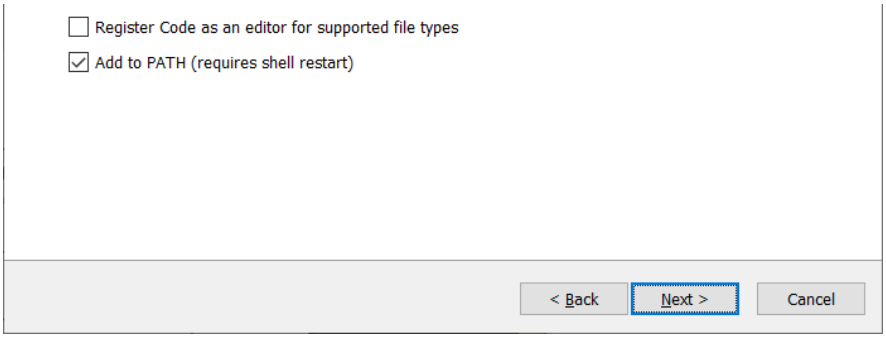


Install VS Code

Visual Studio Code (VS Code) is a great, cross-platform text editor that offers highly configurable plugins. With the right plugins, you can essentially turn VS Code into a full-fledged integrated development environment (IDE) with step-through debugging! I'll only show the basics for now so you can get your VS Code on Windows to act like VS Code on other operating systems when working with the Pico.

Head to code.visualstudio.com and download the latest release for Windows.

Run the installer and accept all of the defaults. You're welcome to create a desktop icon and add "Open with Code" to the Windows Explorer context menu. I like to enable these options, but they're not necessary.



When the installer is done, it's time to test building a program for the Pico.

Build Blink Example

I got errors when I tried to build. These were resolved by modifying the path variable. See comments by "Footleg" in comments at the end.

You should be able to build Pico code from any command prompt at this point. However, we're going to do so in VS Code to show how it might be done during development.

Open VS Code. Click **Terminal > New Terminal**. The default terminal that opens is likely Windows PowerShell. I recommend changing this to Git Bash by clicking on the drop-down menu and selecting **New Git Bash**.

Just like we did with the *make.bat* file (for Windows terminals), I recommend creating an *alias* for *mingw32-make.exe* in Git Bash. Enter the following commands (you will only need to do this once):

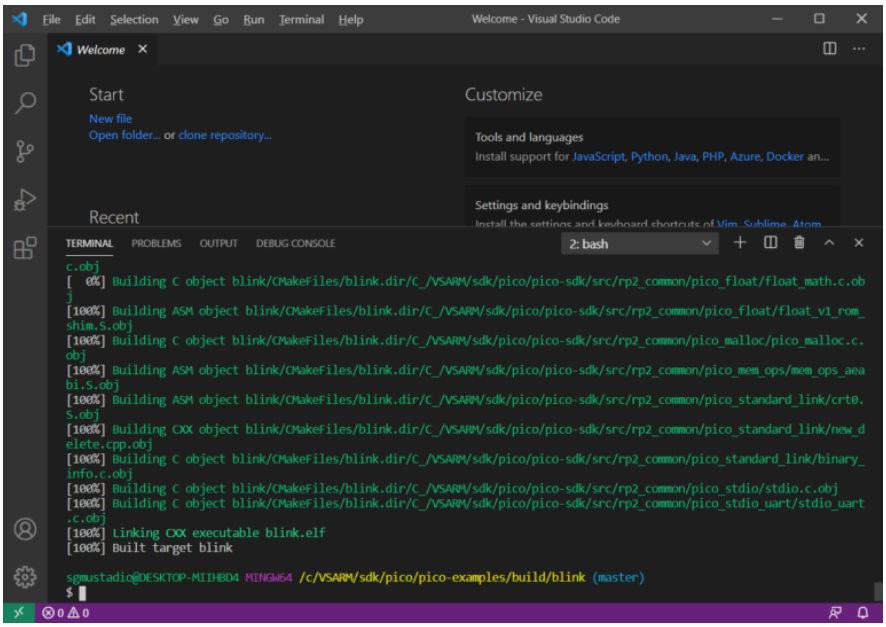
```
echo "alias make=mingw32-make.exe" >> ~/.bashr
source ~/.bashrc
```

Build the blink example by entering the following commands into your terminal:

SHAWN HYMEL

```
cd /c/VSARM/sdk/pico/pico-examples/
mkdir build
cd build
cmake -G "MinGW Makefiles" ..
cd blink
make
```

Important! Any time you call CMake from a terminal like this, you will need to specify "MinGW Makefiles" as the build system generator (-G option).



This should build the blink example and generate the .elf, .hex, and .uf2 binary files. We can easily upload the .uf2 file to our Pico without any special tools.

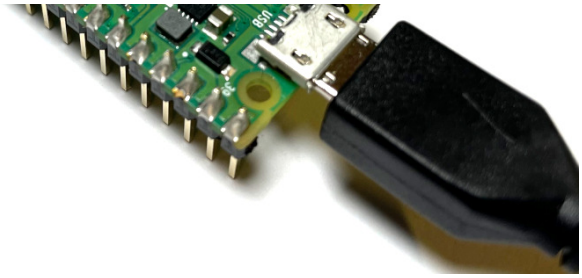
Put the Pico board into bootloader mode (press and hold the BOOTSEL button while plugging a USB cable into the Pico).

Find which drive letter the RPI-RP2 drive is mounted to (e.g. it was G: for me). Enter the following into Git Bash (change the drive letter as necessary):

```
cp blink.uf2 /g/
```

Your Pico board should be blinking away!

SHAWN HYMEL



I hope this has helped you get started using the Raspberry Pi Pico SDK on Windows! I personally found that relying on MinGW for the compiler tools was easier than using Build Tools for Visual Studio.

From here, you can set up VS Code any way you like with various plugins like CMake Tools. I did not cover setting up debugging and other tools (picotool, gdb, OpenOCD, etc.), which are topics for another time. For now, I recommend referring to the Pico C/C++ Getting Started Guide to see how to configure those tools in VS Code.

Happy hacking!

[Update May 23, 2021] Since posting this, I have made a video (for Digi-Key) that shows you how to use VS Code plugins for one-click CMake and building. You can see that video here.

C C++ RASPBERRY PI PICO TUTORIAL VS CODE WINDOWS

Share this Article

SHAWN HYMEL

Author

ShawnHymel

120 thoughts on “How to Set Up Raspberry Pi Pico C/C++ Toolchain on Windows with VS Code”

1

Avinash on May 20, 2021 at 10:32 am REPLY

Excellent documentation.

1

gautom Bose on April 5, 2022 at 1:11 am REPLY

Excellent.
Is this process suitable for windows 10, x86, 32 bit
Pc.
Can you please confirm if it will work on a 32 bit windows 10 pc ?

1

ShawnHymel on April 15, 2022 at 4:53 pm REPLY

I don’t have a 32-bit version of Windows to try, but I have to imagine it will work.

SHAWN HYMEL

KennyD on August 31, 2022 at 3:29 pm

REPLY

Excellent guide, very detailed and love the screenshots. You must have spent an age writing this guide, so many thanks for your hard work.
Like many, I had no success following the "Getting Started with Raspberry Pi Pico" guide for Windows but with your instructions I have succeeded!

vidal on May 23, 2021 at 10:06 am REPLY

Hello and thank you for this documentation. Using the VS-Code and Cmake-Tools tools reduces the use of the command windows and therefore the risk of errors! It's a shame to use Windows tools like those of a Linux! There is Simpler in VS-Code and CMake parameters to use CCG automatically, which is not the case here. Maybe in another videos?

ShawnHymel on May 23, 2021 at 3:33 pm

REPLY

Yes, I cover using plugins for one-click CMake and building in this video: https://www.youtube.com/watch?v=B5rQSoOmR5w. This post was made

SHAWN HYMEL

details and the screenshots.

1.

ShawnHymel on May 27, 2021 at 2:03 pm

REPLY

I'm glad to hear it helped!

4.

dmytro on June 1, 2021 at 11:28 pm REPLY

is there a reason to prefer mingw on window rather then wsl? it seems that setup instructions for the linux work fine under the wsl. the only thing – i could not build picotool, because the script could not find libusb. otherwise, i've compiled blink pretty easily, and uploaded it in regular way from my windows laptop.

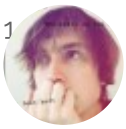
5.

Pena on June 13, 2021 at 8:50 am REPLY

Thanks Shawn. After couple installation rounds I finally managed to compile and load blink.uf2 on my picoboard. I got this error : "bash: $'\377\376alias': command not found" when I opened Git Bash from Desktop. I had to clean all files from my home direcotory. Yes I had previous VSCode installations and others too before starting your project. But now everything works fine and I am going to proceed with your tutorials.

SHAWN HYMEL

Thanks!



Ben on May 7, 2022 at 8:26 am REPLY

If either of you two managed to get a solution to this I'd be very thankful!

Fab on July 8, 2022 at 11:47 am REPLY

I have also this error message : "bash: $'\377\376alias" . I have reinstalled VSCODE and try to delete all settings before (code folder) but I have still the same error;
How can I solve this issue ? What other parameters or settings I have to delete to clean other VSCODE install I have to do ?

Stan on June 13, 2021 at 11:11 pm REPLY

Thank you. It is very helpful and clear.

bitluni on June 24, 2021 at 11:03 pm REPLY

I cant find make only mingw32-make.exe after the path configuration

It works… this is the first tutorial that works for me. Thanks!

Max on June 26, 2021 at 5:48 pm REPLY

Thank you – this worked for me.

zoot on July 6, 2021 at 8:56 pm REPLY

Wanted to put a comment on here — my make was dying on the elf2uf2 step of the makefile. If the same thing is happening to you, you will get a code '-1073741511' in this scenario.

The fix was a direct copy of the mingw32 libstdc++-6.dll into the created build/elf2uf2 directory, alongside elf2uf2.exe

There was a silent dll failure buried in there, which will open up a dialog only if you run the tool on its own whiel this error is present.. This is a fairly mature system. I don't know which libstdc++-6.dll it was /trying/ to use, but the quickest way to get rid of it is to move the library into that same location.

ShawnHymel on July 22, 2021 at 2:53 pm

REPLY

SHAWN HYMEL

C:\VSARM\mingw\minggw64\bin).

Douglas Gilliland on May 28, 2022 at 5:43 pm REPLY

Very helpful. I had two issues.

One was the “Error -1073741511” issue Zoot mentioned above and his fix worked. I think I ended up copying libstdc++-6.dll to C:\VSARM\sdk\pico\pico-examples\build\ .

The other issue was setting up VS Code to run git bash in the terminal. Apparently, the “new” version of VS Code I’m running (2022-05-17718…) has changed that method and the instructions above no longer work. I fumbled around with a couple of suggestions on stackoverflow and gave up.

The chain runs fine by launching git bash directly from start menu and typing make.

I’m happy. You got me to a place that works after trying several other site instructions. I think it’s a mistake for the RPi Foundation to make so much effort getting the Pico SDK instructions tied around the Pi, especially with the total lack of Pi’s in the market now.

Thanks!

SHAWN HYMEL

associated…". For me, it was something I didn't need in my PATH, so I just removed it.

Footleg on September 4, 2022 at 2:25 pm

REPLY

I also ran into the Error -1073741511 issue. I wasn't happy having to copy the libstdc++-6.dll into every build directory, and found an alternative solution. I added the following line into my .bashrc file:
export PATH="C:\VSARM\mingw64\bin":$PATH

(Change to point to your mingw bin folder).

Now it finds the libstdc++-6.dll on my path before any other versions in my git bash terminal windows.

Mike on April 5, 2023 at 4:47 pm

REPLY

Your comment is awaiting moderation.
This solved the libstdcc++-6.dll issue. I tried the copying across above but all that happened is it started complaining about more missing dlls. Adding this to my path meant they were all found.

SHAWN HYMEL

ShawnHymel on July 22, 2021 at 2:51 pm

REPLY

I don't have that board, so I won't be able to help. Your best bet would be to ask for help from SparkFun on the product page (https://www.sparkfun.com/products/16985) or the forums (https://forum.sparkfun.com/).

Jonnyboy on July 22, 2021 at 10:46 am REPLY

Have followed your tutorial to the letter, but with VS Code already previously installed. When I try to build the 'blink' example, I get the following error:

mingw32-make[2]: *** [blink\CMakeFiles\blink.dir\build.make:812: blink/blink.elf] Error -1073741511
mingw32-make[2]: *** Deleting file 'blink/blink.elf'
mingw32-make[1]: *** [CMakeFiles\Makefile2:3723: blink/CMakeFiles/blink.dir/all] Error 2
mingw32-make: *** [Makefile:90: all] Error 2

Any ideas what the problem could be?

ShawnHymel on July 22, 2021 at 2:49 pm

REPLY

SHAWN HYMEL

ShawnHymel on July 22, 2021 at 2:54 pm

REPLY

I just noticed a comment from someone else with the same issue. Here's the proposed fix:

"The fix was a direct copy of the mingw32 libstdc++-6.dll into the created build/elf2uf2 directory, alongside elf2uf2.exe"

Jonnyboy on July 23, 2021 at 6:32 am

REPLY

Aha! Perfect! Thank you – that fixes the issue, I can build the examples now. Thanks for an excellent tutorial too. 🙂

Lucas on March 23, 2022 at 11:54 am

REPLY

This worked for me too. Had a few problems locating libstdc++-6.ddl but found it under C:\VSARM\mingw\mingw64\x86_64-w64-mingw32\lib32. I also had a few problems with the sourceforge installer as it currently doesn't work and presents me with the message "the file has been downloaded incorrectly". This is why the path to this file is slightly different as I has to find a workaround by downloading the x86_64-posix-sjlj

SHAWN HYMEL

An alternative when you want to de Arduino development on windows for the Pi Pico you could consider start using the VsCode / PlatformIO. This combination does also support the Pi Pico. All toolchain stuff is done for you by PlatformIO.

Kind regards, Luc Looman

Luc Looman on August 4, 2021 at 6:14 am

REPLY

A little rectification needed on my Rely.
Indeed installing and running blink example in win10/vscode/platformio worked very quick.
Unfortuneatly this is a Arduino-mbed implementation which means you can not use the full Pico C++ SDK.
There is hope it will be supported soon with earlephilhower/arduino-pico.

Tarik Bell on August 13, 2021 at 7:11 pm REPLY

Hi Shawn,

Please could help to fix this issue:
$ cmake -G “MinGW Makefiles” ..
bash: cmake: command not found

SHAWN HYMEL

(https://cmake.org/download/), did you click the option to "Add CMake to system PATH for all users"? If not, check to make sure "C:\Program Files\CMake\bin" is included in the Path variable under Environment Variables > System Variables.

Tarik Bell on August 16, 2021 at 3:51 pm

REPLY

It works now after I installed the last version of cmake-3.21.1-windows-x86_64.msi.
My Pico is blinking Thank you.

ShawnHymel on August 16, 2021 at 5:58 pm REPLY

Glad to hear it works! And it's good to know that it seems CMake 3.21 fixed the syntax error bug.

tortik92 on September 17, 2021 at 9:00 pm

REPLY

Hi Shawn!

After executing "cmake -G "MinGW Makefiles" .." in Git Bash Terminal of VS Code I get CMake Error:

SHAWN HYMEL

CMakeLists.txt:4 (include)
— Configuring incomplete, errors occurred!"

This path is existing and I can see it in explorer….

Thank you!

ShawnHymel on September 18, 2021 at 2:33 pm REPLY

I can't seem to replicate this issue. Could you tell me the output of the following commands (from, say, Git Bash):

```
echo $PICO_SDK_PATH
ls $PICO_SDK_PATH
cmake --version
```

tortik92 on September 18, 2021 at 9:30 am REPLY

Hi Shawn!

After executing "cmake -G "MinGW Makefiles" .." in Bash in VS Code I recieve an error:

CMake Error at pico_sdk_import.cmake:52 (message):
Directory 'C:/VSARM/sdk/pico/pico-examples/build/ C:/VSARM/sdk/pico/pico-sdk' not found
Call Stack (most recent call first):

SHAWN HYMEL

Patrik Källback on September 23, 2021 at 6:42 am REPLY

Thank you Shawn for the guide to this toolchain! Now I'm able to compile and build my pico c++ programs.
Something wierd though (this might be to another version of VS code) I was not able to select Bash in VS Code in the terminal selection. After shoe horning Bash as a default terminal and was running make, I got the error that mingw32-make.exe was not present. Then I realized that when I was using the windows command prompt as the terminal, than the make was working perfectly. Wieard, but then I decided not to think more about it… just running make in the windows command prompt.

David Marshall on November 2, 2021 at 1:50 am REPLY

The link to "ming-w64" is broken.

…..Head to the downloads page on the Mingw-w64 project site, and go to the Mingw-w64 Builds page for Windows: http://mingw-w64.org/doku.php/download/mingw-builds.

ShawnHymel on November 2, 2021 at 3:44 am REPLY

SHAWN HYMEL

https://sourceforge.net/projects/mingw-w64/files/Toolchains%20targetting%20Win32/Personal%20Builds/mingw-builds/installer/mingw-w64-install.exe/download

ShawnHymel on November 2, 2021 at 2:16 pm REPLY

Yes, I believe that is the correct one. It looks like they changed the installer slightly, but if you go through the steps, it should install ming-w64 as shown in the tutorial.

David Marshall on November 5, 2021 at 6:58 pm

The tutorial works for me, I am having an issue with an older win7 machine.

Przemysław Kurzak on November 12, 2021 at 9:35 pm REPLY

Hi guys, I got this issue after calling make:

mingw32-make[2]: *** [pico-sdk\src\rp2_common\boot_stage2\CMakeFiles\bs2_default_padded_checksummed_asm.dir\b

mingw32-make: *** [Makefile:90: all] Error 2

I tried copy of the mingw32 libstdc++-6.dll into the created build/elf2uf2 directory, alongside elf2uf2.exe but it won't help :(. Any idea?

John on November 17, 2021 at 5:32 pm REPLY

Thank you, it is really helpful for the installing! But there is a problem, the build is already complete but when I was trying to flash it to rp2040, it didnt blink. Can you help me?

ShawnHymel on November 17, 2021 at 6:39 pm REPLY

I'm not sure why it's not blinking for you. Are you seeing any errors? Did you copy the .uf2 file? Can you verify that the LED will blink using other methods (e.g. MicroPython)?

Alexander Kirillov on November 20, 2021 at 3:50 pm REPLY

Thanks a lot – great tutorial!!
Two comments:
1. In section "installing Mingw", where you say "head to the downloads page on the Mingw-w64 project site, and go to the Mingw-w64 Builds page for Windows", your link is broken – link text is correct, but link URL is a placeholder:

SHAWN HYMEL

VScode didn't. However, after I relaunched VScode, deleted Git Bash terminal in VScode and launched it again, everything worked. Apparently VScode didn't read all env variables on first launch...

ShawnHymel on November 20, 2021 at 9:36 pm REPLY

Thanks for the heads up! I really don't know why the link didn't take in WordPress...I updated it, and it should work now. That's weird about VSCode not reading the environment variables, but good to know about needing to relaunch it.

Serhii on November 22, 2021 at 8:27 pm REPLY

Thank you for the excellent guide
Everything works

Only one note that I'd add

echo mingw32-make %* > C:\VSARM\mingw\mingw32\bin\make.bat

This line created make.bat with utf-16 encoding on my machine. Which windows did not like. Updated it to utf-8 with notepad++. And that made it work

REPLY

Thank you very much, I've been struggling for days trying to use the extensions in VScode, your method worked seamlessly!

David Marshall on December 15, 2021 at 6:29 pm

REPLY

For some reason "GIT BASH" is not available in the terminal. I've removed VS code and followed your method.

Robbok on December 18, 2021 at 4:39 pm

REPLY

Hi there, I know you spent a ton of time making this page and being thorough in showing every aspect of what needed to be done to get these tools up and working. I just wanted to say THANK YOU. You saved me earlier this week when I needed to compile some files for my Raspberry Pi Pico– I was completely lost, and quite frustrated that 95% of everything written online not only assumes familiarity with Linux, but assumes you are running Linux. As a Windows user, I was lost. This page is now bookmarked because it's one of the best how-to write-ups I've ever seen online. Again, thank you!

SHAWN HYMEL

Dave on January 12, 2022 at 8:33 pm REPLY

Jan 12 2022

I've made it to the step where visual studio code is installed. When visual studio code first runs, I encounter a dialog box as follows:

"Unable to determine what CMake generator to use. Please install or configure a preferred generator, or update settings.json, your Kit configuration or PATH variable."

In a command prompt, cmake can be started:

Microsoft Windows [Version 10.0.19044.1415]
(c) Microsoft Corporation. All rights reserved.
C:\Users\user>cmake
Usage
cmake [options]
cmake [options]
cmake [options] -S -B

In the user's path environment variable, the following value is set:

C:\Program Files\CMake\bin

Would be very grateful for advice as to how to proceed.

Thank you!

SHAWN HYMEL

https://github.com/microsoft/vscode-cmake-tools/issues/880

James on January 14, 2022 at 5:11 am REPLY

Thank you Hymel.
However, I needed to restart my computer twice, for some reason I guess, the environmental variables didn't apply immediately.

Dave on January 14, 2022 at 5:13 pm REPLY

Shawn, thank you for acknowledging my post about "Unable to determine what CMake generator to use. Please install or configure a preferred generator, or update settings.json, your Kit configuration or PATH variable."

I was able to resolve this by uninstalling visual studio code, and then (important) removing two directories which had retained state information from visual studio code. I then re-installed visual studio code and the start-up error messages were not present.

I continued in the installation procedure to this point:

echo "alias make=mingw32-make.exe" >>
~/.bashrc
source ~/.bashrc

I was able to complete the rest of the demonstration resulting in a working “blink LED” example.

1. Do you have any idea how I could fix “source ~/.bashrc” so it completes?

2. Can the selection of bash be made in permanent or persistent in visual studio code? Each time it starts power shell, but I can then switch to bash. I would prefer bash permanently.

Thank you again. After many hours this week with failed installation attempts using (apparently untested) documentation from raspberry pi foundation, your procedure produced my first end-to-end working result.

ShawnHymel on January 15, 2022 at 3:11 pm

REPLY

Hi Dave,

1. Check the spelling and syntax of your alias command with `$ cat ~/.bashrc` Make sure that alias is spelled correctly, and that there are no spaces around the equals sign (see here for more info: https://askubuntu.com/questions/391518/bash-alias-alias-not-found).
2. You can make Git Bash the default terminal. Press `F1` and type “Terminal: Select Default Profile.” From the drop-down list, select Git Bash to be your default

SHAWN HYMEL

BAVEK MISTRI on January 23, 2022 at 8:44 pm

REPLY

Hi, I get the following error . Any suggestions for remediation ?
***
CMake Warning at D:/users/vivek/work/RPi-PICO/projects/pico-sdk/cmake/preload/toolchains/find_compiler.cmake:22 (message
):
PICO_TOOLCHAIN_PATH specified (D:\"Program Files (x86)\GNU Tools ARM Embedded\10 2021.10''\bin), but arm-none-eabi-gcc not found there
...
CMake Error at D:/users/vivek/work/RPi-PICO/projects/pico-sdk/cmake/preload/toolchains/find_compiler.cmake:28 (message)

Compiler 'arm-none-eabi-gcc' not found, you can specify search path with "PICO_TOOLCHAIN_PATH".
***

1

Oke Petersen on March 24, 2022 at 10:22 pm

REPLY

Hi I encountered the same issue. The solution forme was to add another System variable:
Name :PICO_TOOLCHAIN_PATH

SHAWN HYMEL

Therefore you only have to define it

Best regards
Oke

William on January 26, 2022 at 10:36 pm

REPLY

Fantastic tutorial! I'm so looking forward to this working, but I get the following error….

CMake Error at C:/Program Files/CMake/share/cmake-3.22/Modules/CMakeTestCCompiler.cmake:69 (message):
The C compiler "C:/build/17.0/arm/sysroots/i686-nilrtsdk-mingw32/usr/libexec/arm-nilrt-linux-gnueabi/gcc/arm-nilrt-linux-gnueabi/4.9.2/gcc.exe"
is not able to compile a simple test program.

I've followed you instructions and all went well up to the point of compiling blink… any help appreciated ☝️

Sebastian Abril on January 31, 2022 at 4:54 am

REPLY

Thanks for the tutorial!!! I have a question that I still can't solve, how can I set in visual code a different RP2040 based board, such as the adafruit feather or custom board.

SHAWN HYMEL

REPLY

It appears that there is a very recent issue with the Ming-w online installer. The error says “The file has been downloaded incorrectly!” It seems to be very recent and many have commented on Source Forge. Is there any solutions/ways around this?

1

Juan David Medina Tobon on March 4, 2022 at 5:46 pm REPLY

Hi,

I followed the answer from this post: https://stackoverflow.com/questions/46455927/mingw-w64-installer-the-file-has-been-downloaded-incorrectly. Apparently, there’s a problem with the installer.

Download the MinGW archive from https://sourceforge.net/projects/mingw-w64/files/mingw-w64/. I downloaded the one named “x86_64-posix-sjlj”. Extract the file and copy the folder to the mingw folder created inside VSARM. Worked for me.

Also, make sure to change mingw32 to mingw64 when following the instructions, if that’s the version you downloaded.

1

SHAWN HYMEL

example dir: blink and the path for mingw32. Using the new version ? just confused me. blink worked. moving forward to other and RTOS.
more BEER.

Lucas on March 24, 2022 at 4:02 pm

REPLY

This workaround worked for me. Only had one more minor setback after this while running make for the blink example. I was prompted with the following error:

mingw32-make[2]: *** [blink\CMakeFiles\blink.dir\build.make:812: blink/blink.elf] Error -1073741511
mingw32-make[2]: *** Deleting file 'blink/blink.elf'
mingw32-make[1]: *** [CMakeFiles\Makefile2:3723: blink/CMakeFiles/blink.dir/all] Error 2
mingw32-make: *** [Makefile:90: all] Error 2

This was solved after I included a file called libstdc++-6.ddl (which, among another few directories, can be found in C:\VSARM\mingw\mingw64\x86_64-w64-mingw32\lib32) into the build/elf2uf2 folder.

SHAWN HYMEL

My PATH variable.

JK_student on March 21, 2022 at 9:04 am

REPLY

Hello,

I have question. I would like write permamently program on my RP2040.
I wrote code in circuitpython. I need RP2040 which when it will be plug in to computer, It send data from RP2040 to computer by UART.
How Can I do it?

I think it's the same work as https://www.youtube.com/watch?v=IMZUZuytt7o , but in circuitpython and to RP20404.

Juan Manuel on March 31, 2022 at 5:51 pm

REPLY

Hi shawn, may you help me with this issue? after writing this comand on VS code " cmake -G "MinGW Makefiles" .." terminal appears the following error

$ cmake -G "MinGW Makefiles" ..
CMake Error at pico_sdk_import.cmake:44 (message):
SDK location was not specified. Please set PICO_SDK_PATH or set PICO_SDK_FETCH_FROM_GIT to on to fetch

SHAWN HYMEL

I have this same issue and it seems that this has not been addressed yet.
Any suggestions?

ShawnHymel on June 21, 2022 at 4:38 pm REPLY

Hi Juan and Kelju,

Please run the following:

echo $PICO_SDK_PATH

If you do not see anything printed, it means you did not set PICO_SDK_PATH in your environment variables. That needs to be set to run cmake with the Pico SDK.

Nino on April 1, 2022 at 11:55 am REPLY

Hey Shawn

Followed your tutorial… all worked seemlessly but as i try to run the command (cmake -G “MinGW Makefiles” ..) i get these errors:
CMake Error: CMAKE_C_COMPILER not set, after EnableLanguage
CMake Error: CMAKE_CXX_COMPILER not set, after EnableLanguage
CMake Error: CMAKE_ASM_COMPILER not set,

SHAWN HYMEL

Mustafa on April 2, 2022 at 12:00 pm REPLY

H' Shawn\

I tracked and used step that you mentioned. I get error about below. Could you please help me
[cmake] PICO_SDK_PATH is C:/Pico/pico-sdk
[cmake] PICO platform is rp2040.
[cmake] CMake Error at CMakeLists.txt:8 (project):
[cmake] Running
[cmake]
[cmake] 'nmake' '-?'
[cmake]
[cmake] failed with:
[cmake]
[cmake] The system cannot find the file specified
[cmake]
[cmake]
[cmake] — Configuring incomplete, errors occurred!
[cmake] See also "C:/Pico/blink/build/CMakeFiles/CMakeOutput.log".

my cmakelists.txt

```
# Set minimum required version of CMake
cmake_minimum_required(VERSION 3.12)

# Include build functions from Pico SDK
include($ENV{PICO_SDK_PATH}/external/pico_sdk_import.cmake)
```

SHAWN HYMEL

```
pico_sdk_init()

# Tell CMake where to find the executable
source file
add_executable(${PROJECT_NAME}
main.c
)

# Create map/bin/hex/uf2 files
pico_add_extra_outputs(${PROJECT_NAME})

# Link to pico_stdlib (gpio, time, etc. functions)
target_link_libraries(${PROJECT_NAME}
pico_stdlib
)

# Enable usb output, disable uart output
pico_enable_stdio_usb(${PROJECT_NAME} 1)
pico_enable_stdio_uart(${PROJECT_NAME} 0)
```

Mustafa on April 2, 2022 at 12:02 pm REPLY

H' Shawn,

I followed and aplyed step that you mentioned. I get error about below. Could you please help me
[cmake] PICO_SDK_PATH is C:/Pico/pico-sdk
[cmake] PICO platform is rp2040.
[cmake] CMake Error at CMakeLists.txt:8 (project):
[cmake] Running
[cmake]
[cmake] 'nmake' '-?'
[cmake]
[cmake] failed with:
[cmake]

my cmakelists.txt

```
# Set minimum required version of CMake
cmake_minimum_required(VERSION 3.12)

# Include build functions from Pico SDK
include($ENV{PICO_SDK_PATH}/external/pico_s
dk_import.cmake)

# Set name of project (as PROJECT_NAME) and
C/C++ standards
project(blink C CXX ASM)
set(CMAKE_C_STANDARD 11)
set(CMAKE_CXX_STANDARD 17)

# Creates a pico-sdk subdirectory in our project
for the libraries
pico_sdk_init()

# Tell CMake where to find the executable
source file
add_executable(${PROJECT_NAME}
main.c
)

# Create map/bin/hex/uf2 files
pico_add_extra_outputs(${PROJECT_NAME})

# Link to pico_stdlib (gpio, time, etc. functions)
target_link_libraries(${PROJECT_NAME}
pico_stdlib
)
```

SHAWN HYMEL

[blink\CMakeFiles\blink.dir\build.make:812: blink/blink.elf] Error -1073741511
mingw32-make[2]: *** Deleting file 'blink/blink.elf'
mingw32-make[1]: *** [CMakeFiles\Makefile2:3723: blink/CMakeFiles/blink.dir/all] Error 2
mingw32-make: *** [Makefile:90: all] Error 2

I had to download "x86_64-8.1.0-release-posix-seh-rt_v6-rev0" (instead of i686-8.1.0-release-posix-dwarf-rt_v6-rev0) and unpack it in "C:\VSARM\mingw". There you will have a folder called "mingw64". You either set the environment variable to "C:\VSARM\mingw\mingw64\bin", or keep it as is "C:\VSARM\mingw\mingw32\bin" but rename unpacked folder to mingw32 (a hack !)
It took me many days before I found a solution. Thanks God !

Ibrahim on April 5, 2022 at 8:24 pm REPLY

You will need to append the following line to your ./.vscode/settings.json file:
"cmake.generator": "MinGW Makefiles",

HuyLE on April 27, 2022 at 5:24 pm REPLY

I tested arm-non-eabi-gdb.exe of the latest version Arm GNU Embedded Toolchain 11.2-2022.02 and it doesn't work.

SHAWN HYMEL

Mohamed on May 1, 2022 at 5:42 pm REPLY

Hi
Very good tutorial, I did like it… But I couldn't succeed in trying it.
Here is the the error output I get:

moham@DESKTOP-OTSLFLL MINGW64 /c/VSARM/sdk/pico/pico-examples (master)
$ cd build

moham@DESKTOP-OTSLFLL MINGW64 /c/VSARM/sdk/pico/pico-examples/build (master)
$ cmake -G "MinGW Makefiles" ..
PICO_SDK_PATH is C:/VSARM/sdk/pico/pico-sdk
PICO platform is rp2040.
PICO target board is pico.
Using board configuration from C:/VSARM/sdk/pico/pico-sdk/src/boards/include/boards/pico.h
TinyUSB available at C:/VSARM/sdk/pico/pico-sdk/lib/tinyusb/src/portable/raspberrypi/rp2040; enabling build support for USB.
— Configuring done
— Generating done
— Build files have been written to: C:/VSARM/sdk/pico/pico-examples/build

moham@DESKTOP-OTSLFLL MINGW64 /c/VSARM/sdk/pico/pico-examples/build (master)
$ cd blink

468-1.bat, …) failed.
make (e=2): The system cannot find the file specified.
mingw32-make: *** [Makefile:1211: cmake_check_build_system] Error 2

moham@DESKTOP-OTSLFLL MINGW64 /c/VSARM/sdk/pico/pico-examples/build/blink (master)
$

Of course the folders structure and the variables environment are set as described by you.

Any advice on the issue?

ShawnHymel on May 2, 2022 at 2:00 am

REPLY

Hi Mohamed,

It looks like your system cannot find mingw32-make. I recommend checking that C:\VSARM\mingw\mingw32\bin is on your PATH environment variable and that there is a mingw32-make.exe file in that directory.

Shubham on May 23, 2022 at 5:15 am

REPLY

I deleted the mingw path from user and saved the path in system variables

SHAWN HYMEL

then I restarted my system
and 'make' started working

Ade on May 10, 2022 at 4:43 pm REPLY

With the current (3.23.1, as of May 2022) version of CMake, on Win10, I had to add
cmake_policy(SET CMP0057 NEW)
at the top of my CMakeLists.txt file to get it to work.

Hope this might help others....

apfelschorle on May 27, 2022 at 2:19 pm REPLY

Hi Shawn,

thanks for the great detailed tutorial.

I have an problem. After typing typing the last command or step with "make" into the git bash terminal of VS i get the following message:

@user_pc /c/VSARM/sdk/pico/pico-examples/build/blink $ make
bash: make: command not found

Do you have an idea whats wrong with this? I followed every step from your tutorial, I think. Would be greatful if you help me here out

SHAWN HYMEL

PATH in environment variables. You might want to try the “make” command in other terminals to see if Windows really does see it on your path.

apfelschorle on May 27, 2022 at 4:10 pm

REPLY

Hey Shawn, Then I misunderstood this step:

Enter the following commands (you will only need to do this once):

echo “alias make=mingw32-make.exe” >> ~/.bashrc
source ~/.bashrc

Started directly with next steps

apfelschorle on May 27, 2022 at 9:01 pm

REPLY

What have I to do if I want to add a new project into th pico-examples folder? I created a folder into the pico-examples folder. But git bash cant find the folder.

bash: cd: test: No such file or directory

SHAWN HYMEL

Martin F on June 2, 2022 at 7:16 pm REPLY

Hello.
I followed these directions, and things up to and including the “cmake” command worked fine.
But then when I tried to run “make”, I got the following:

Marty F on June 2, 2022 at 7:50 pm REPLY

Sorry! Clicked Submit too soon!)
Again, up to and including running cmake worked fine.
But the output of make is
undefined reference to ‘_exit’

(And lots more undefined reference linker errors, to things like: _exit, _fstat, _open, _sbrk, “kill, _getpid, _unlink, _write, …)

I would appreciate any help with this!

Thanks.

Marty

Marty F on June 2, 2022 at 9:02 pm REPLY

I tried to post the entire output of the make command, but the website says:
Not Acceptable!
An appropriate representation of the

SHAWN HYMEL

ShawnHymel on June 2, 2022 at 9:59 pm REPLY

I think WordPress comments limit the number of characters (I don't know the exact amount). You can always put your code or debug info in gist.github.com or pastebin.com and provide the link in your comment.

My guess for the error is that you have a path error somewhere (see this thread: https://forums.raspberrypi.com/viewtopic.php?p=1994291). Check to make sure that the C:\VSARM\sdk\pico\pico-sdk exists and that you initialized all the submodules.

Wire on June 7, 2022 at 5:27 am REPLY

Has anyone seen this error when building the blink example after setting up the tools per above?
In file included from D:\VSARM\sdk\pico\pico-sdk\tools\elf2uf2\main.cpp:14:
D:/VSARM/sdk/pico/pico-sdk/src/common/boot_uf2/include/boot/uf2.h:44: error: expected constructor, destructor, or type conversion before '(' token
In file included from D:\VSARM\sdk\pico\pico-sdk\tools\elf2uf2\main.cpp:15:
D:\VSARM\sdk\pico\pico-sdk\tools\elf2uf2\elf.h:60:7: warning: no newline

SHAWN HYMEL

Christopher on July 1, 2022 at 2:18 pm REPLY

Thanks Shawn, I tested this on Windows 11 on (July 1 2022), it worked great!

Bill on July 22, 2022 at 7:31 am REPLY

Having a weird issue I can't seem to resolve it. During Make I get:
Scanning dependencies of target bs2_default
[0%] Linking ASM executable bs2_default.elf
arm-none-eabi-gcc.exe: error: nosys.specs: No such file or directory
mingw32-make[2]: *** [pico-sdk\src\rp2_common\boot_stage2\CMakeFiles\bs2_default.dir\build.make:96: pico-sdk/src/rp2_common/boot_stage2/bs2_default.elf] Error 1
mingw32-make[1]: *** [CMakeFiles\Makefile2:4216: pico-sdk/src/rp2_common/boot_stage2/CMakeFiles/bs2_default.dir/all] Error 2
mingw32-make: *** [Makefile:90: all] Error 2

Alexander on August 6, 2022 at 4:49 am REPLY

Do you expect this to work using Cygwin?

I followed basically the same steps, but there are a couple changes:

SHAWN HYMEL

cc1.exe: fatal error: /cygdrive/c/VSARM/pico-sdk/src/rp2_common/boot_stage2/compile_time_choice.S: No such file or directory

I have confirmed that compile_time_choice.S does exist at the stated path. Any ideas?

Buli on August 6, 2022 at 9:03 pm REPLY

Failed in the step of “make” in the Blink example. Somehow, it uses cl.exe to compile. Please help.

GBXXET1@SVP-LDGVST0F3 MINGW64 ~
$ cd /c/VSARM/sdk/pico/pico-examples/

GBXXET1@SVP-LDGVST0F3 MINGW64 /c/VSARM/sdk/pico/pico-examples (master)
$ mkdir build

GBXXET1@SVP-LDGVST0F3 MINGW64 /c/VSARM/sdk/pico/pico-examples (master)
$ cd build

GBXXET1@SVP-LDGVST0F3 MINGW64 /c/VSARM/sdk/pico/pico-examples/build (master)
$ cmake -G “MinGW Makefiles” ..
Using PICO_SDK_PATH from environment (‘C:\VSARM\sdk\pico\pico-sdk’)
PICO_SDK_PATH is C:/VSARM/sdk/pico/pico-sdk
Defaulting PICO_PLATFORM to rp2040 since not specified.
Defaulting PICO platform compiler to pico_arm_gcc since not specified.

```
Build type is Release
Defaulting PICO target board to pico since not specified.
Using board configuration from C:/VSARM/sdk/pico/pico-sdk/src/boards/include/boards/pico.h
— Found Python3: C:/Program Files/Python310/python3.exe (found version "3.10.6") found components: Interpreter
TinyUSB available at C:/VSARM/sdk/pico/pico-sdk/lib/tinyusb/src/portable/raspberrypi/rp2040; enabling build support for USB.
cyw43-driver available at C:/VSARM/sdk/pico/pico-sdk/lib/cyw43-driver
lwIP available at C:/VSARM/sdk/pico/pico-sdk/lib/lwip
— Configuring done
— Generating done
— Build files have been written to: C:/VSARM/sdk/pico/pico-examples/build

GBXXET1@SVP-LDGVST0F3 MINGW64 /c/VSARM/sdk/pico/pico-examples/build (master)
$ cd blink

GBXXET1@SVP-LDGVST0F3 MINGW64 /c/VSARM/sdk/pico/pico-examples/build/blink (master)
$ make
[ 0%] Creating directories for 'ELF2UF2Build'
[ 0%] No download step for 'ELF2UF2Build'
[ 0%] No update step for 'ELF2UF2Build'
[ 0%] No patch step for 'ELF2UF2Build'
```

— Check for working C compiler: C:/Program Files/Microsoft Visual Studio/2022/Community/VC/Tools/MSVC/14.32.31326/bin/Hostx64/x64/cl.exe
— Check for working C compiler: C:/Program Files/Microsoft Visual Studio/2022/Community/VC/Tools/MSVC/14.32.31326/bin/Hostx64/x64/cl.exe – broken
CMake Error at C:/Program Files/CMake/share/cmake-3.24/Modules/CMakeTestCCompiler.cmake:69 (message):
The C compiler

"C:/Program Files/Microsoft Visual Studio/2022/Community/VC/Tools/MSVC/14.32.31326/bin/Hostx64/x64/cl.exe"

is not able to compile a simple test program.

It fails with the following output:

Change Dir: C:/VSARM/sdk/pico/pico-examples/build/elf2uf2/CMakeFiles/CMakeTmp

Run Build Command(s):C:/VSARM/mingw/mingw32/bin/mingw32-make.exe -f Makefile cmTC_188ce/fast && mingw32-make.exe[3]: Entering directory 'C:/VSARM/sdk/pico/pico-examples/build/elf2uf2/CMakeFiles/CMakeTmp'
C:/VSARM/mingw/mingw32/bin/mingw32-make.exe -f

SHAWN HYMEL

```
"C:\Program Files\CMake\bin\cmake.exe" -E
cmake_cl_compile_depends –dep-
file=CMakeFiles\cmTC_188ce.dir\testCCompiler.
c.obj.d –working-dir=C:\VSARM\sdk\pico\pico-
examples\build\elf2uf2\CMakeFiles\CMakeTm
p –filter-prefix="Note: including file: " —
C:\PROGRA~1\MICROS~4\2022\COMMUN~1\VC
\Tools\MSVC\1432~1.313\bin\Hostx64\x64\cl.ex
e /nologo /DWIN32 /D_WINDOWS /W3 /MDd
/Zi /Ob0 /Od /RTC1 /showIncludes
/FoCMakeFiles\cmTC_188ce.dir\testCCompiler.
c.obj /FdCMakeFiles\cmTC_188ce.dir/ /FS -c
C:\VSARM\sdk\pico\pico-
examples\build\elf2uf2\CMakeFiles\CMakeTm
p\testCCompiler.c
testCCompiler.c
Linking C executable cmTC_188ce.exe
"C:\Program Files\CMake\bin\cmake.exe" -E
cmake_link_script
CMakeFiles\cmTC_188ce.dir\link.txt –verbose=1
"C:\Program Files\CMake\bin\cmake.exe" -E
vs_link_exe –intdir=CMakeFiles\cmTC_188ce.dir
–rc=rc –mt=CMAKE_MT-NOTFOUND –manifests
—
C:\PROGRA~1\MICROS~4\2022\COMMUN~1\VC
\Tools\MSVC\1432~1.313\bin\Hostx64\x64\link.
exe /nologo
@CMakeFiles\cmTC_188ce.dir\objects1.rsp
/out:cmTC_188ce.exe /implib:cmTC_188ce.lib
/pdb:C:\VSARM\sdk\pico\pico-
examples\build\elf2uf2\CMakeFiles\CMakeTm
p\cmTC_188ce.pdb /version:0.0 /machine:x64
/debug /INCREMENTAL /subsystem:console
kernel32.lib user32.lib gdi32.lib winspool.lib
shell32.lib ole32.lib oleaut32.lib uuid.lib
```

cmTC_188ce.exe] Error -1
mingw32-make.exe[4]: Leaving directory 'C:/VSARM/sdk/pico/pico-examples/build/elf2uf2/CMakeFiles/CMakeTmp'
mingw32-make.exe[3]: *** [Makefile:126: cmTC_188ce/fast] Error 2
mingw32-make.exe[3]: Leaving directory 'C:/VSARM/sdk/pico/pico-examples/build/elf2uf2/CMakeFiles/CMakeTmp'

CMake will not be able to correctly generate this project.
Call Stack (most recent call first):
CMakeLists.txt:2 (project)

— Configuring incomplete, errors occurred!
See also "C:/VSARM/sdk/pico/pico-examples/build/elf2uf2/CMakeFiles/CMakeOutput.log".
See also "C:/VSARM/sdk/pico/pico-examples/build/elf2uf2/CMakeFiles/CMakeError.log".
mingw32-make[2]: *** [blink\CMakeFiles\ELF2UF2Build.dir\build.make:90: blink/elf2uf2/src/ELF2UF2Build-stamp/ELF2UF2Build-configure] Error 1
mingw32-make[1]: *** [CMakeFiles\Makefile2:4399: blink/CMakeFiles/ELF2UF2Build.dir/all] Error 2
mingw32-make: *** [Makefile:90: all] Error 2

Buli on August 13, 2022 at 12:10 am REPLY

SHAWN HYMEL

EEM on August 14, 2022 at 2:10 pm REPLY

gcc-arm-none-eabi-10-2020-q4-major-win32.exe." is Signed for Windows 10 and later". What about the Windows 8 (not8.1 just 8)?

Amir on August 15, 2022 at 12:44 pm REPLY

Hello Shawn, I've followed all the steps but when I type "make" command in the bash terminal, it says

'C:/Program' is not recognized as an in internal or external command, operable program or bash file.
mingw32-make:* [Makefile:90 :all] Error 1

Footleg on August 29, 2022 at 4:42 pm REPLY

I ran into this same issue, and after a lot of trial and error I discovered it happened if I tried to run make from a Powershell command prompt, but it does not happen if you run make from a git bash command prompt. I don't understand why, but you should be able to get things to compile using a git bash terminal window.

Paulus on August 28, 2022 at 8:36 pm REPLY

SHAWN HYMEL

Hans on September 2, 2022 at 1:30 pm REPLY

I have at least 7 differrent make.exe files on my computer. When I open a command prompt and type 'make' it choses a wrong one. How can I fix this without messing up other installed software? Is using a virtual machine a solution?

Footleg on September 4, 2022 at 3:42 pm
REPLY

If you update your path to have the mingw bin directory as the first item if should find the make.bat file created in the steps about before any other make.exe on your system. See my comment above where I do this to solve the error caused by the wrong libstdc++-6.dll as that change should also solve your multiple make.exe files issue.

Footleg on September 4, 2022 at 3:46 pm
REPLY

Setting up my dev environment again today on a different PC, I tried the arm-gnu-toolchain-11.3.rel1-mingw-w64-i686-arm-none-eabi release and confirm it did NOT work for me. I reverted to the gcc-arm-11.2-2022.02-mingw-w64-i686-arm-none-eabi release and everything is working fine. I am using the

SHAWN HYMEL

for the detailed information, but I still get errors even after my 3rd try to make it work. My system is probably cluttered with junk from earlier attempts.

When I run 'make', I get this error (just like Buli):
— Check for working C compiler: C:/Program Files/Microsoft Visual Studio/2022/Community/VC/Tools/MSVC/14.32.31326/bin/Hostx86/x86/cl.exe – broken

silently crying in a corner

6

Jared on October 26, 2022 at 4:45 am REPLY

I keep getting bash: cmake: command not found . I've reinstalled cmake and made sure to add CMake to the system PATH for all users. make.bat is in C:\VSARM\mingw\mingw32\bin. I've tried reinstalling and installing an earlier version that you or others said worked for them.

1

Jared on October 27, 2022 at 1:10 am
REPLY

It worked when I tried the the next day. lol however I have a Pico W that does not respond im assuming this is because they have different ports associated with the led?

6

SHAWN HYMEL

examples/ build)
C:/VSARM/sdk/pico/pico-sdk' not found
Call Stack (most recent call first):
CMakeLists.txt:4 (include)

— Configuring incomplete, errors occurred!

It must be something simple…

$ echo $PICO_SDK_PATH
C:\VSARM\sdk\pico\pico-sdk\

$ ls $PICO_SDK_PATH
cmake/ CONTRIBUTING.md external/
LICENSE.TXT pico_sdk_version.cmake src/
tools/
CMakeLists.txt docs/ lib/ pico_sdk_init.cmake
README.md test/

$ cmake –version
cmake version 3.24.1

CMake suite maintained and supported by
Kitware (kitware.com/cmake).

Gavin on November 22, 2022 at 8:20 pm REPLY

Seem to have got past the cmake stage.

I noticed there was a CMakeCache.txt in the build directory.

SHAWN HYMEL

Girish on December 31, 2022 at 9:26 am REPLY

Thanks you very much.
This article made my life very easy. I tried and it worked in the first attempt itself

Serdar kayiskiran on January 17, 2023 at 8:31 pm
REPLY

Thanks you very much.
I tried and it worked in the first attempt itself

Matt on February 25, 2023 at 3:23 am REPLY

Thank you for the guide. For me it works till I try to enter this command “source ~/.bashrc”. When I enter that command it gives me this error “bash: $’\377\376alias’: command not found” After that nothing works and I can’t get it to compile the example.

Leave a Comment

Your email address will not be published. Marked fields are required.

SHAWN HYMEL

Email *

Website

I'm not a robot

reCAPTCHA
Privacy - Terms

SUBMIT