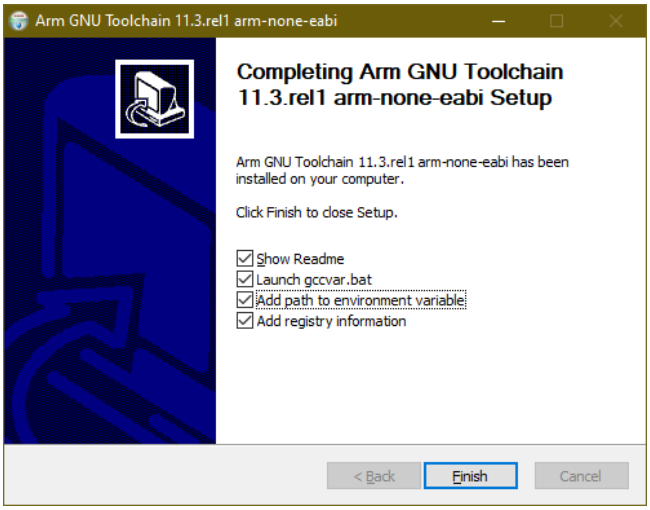
Install Pico (and PicoW) SDK 1.50 in Windows 10x64 May 2023

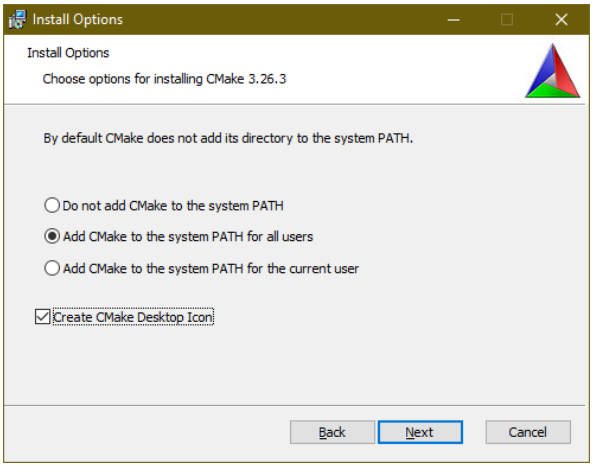
Partly based on RP2040 Development Setup on Windows https://len42.github.io/rp2040-dev-setup.html

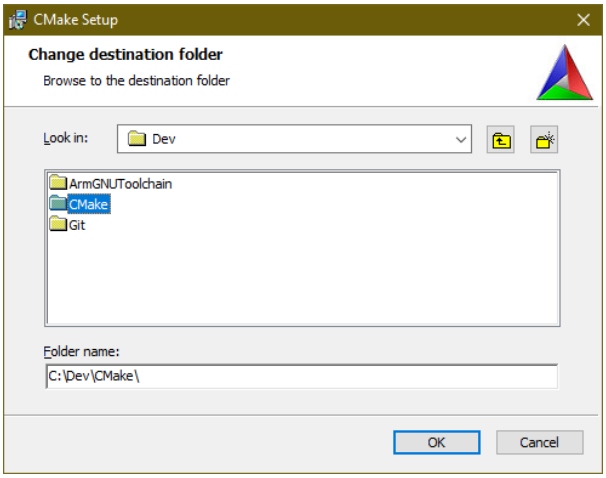
1. Make two new folders (such as C:\Dev and W:\Pico). *I used another drive for the Pico development folders, because it is then easier to make separate backups of the operating system and user data.*

2. Install arm-gnu-toolchain-11.3.rel1-mingw-w64-i686-arm-none-eabi.exe from: https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/downloads to C:\Dev\ArmGNUToolchain - add path to environment variable during install.
https://developer.arm.com/-/media/Files/downloads/gnu/11.3.rel1/binrel/arm-gnu-toolchain-11.3.rel1-mingw-w64-i686-arm-none-eabi.exe

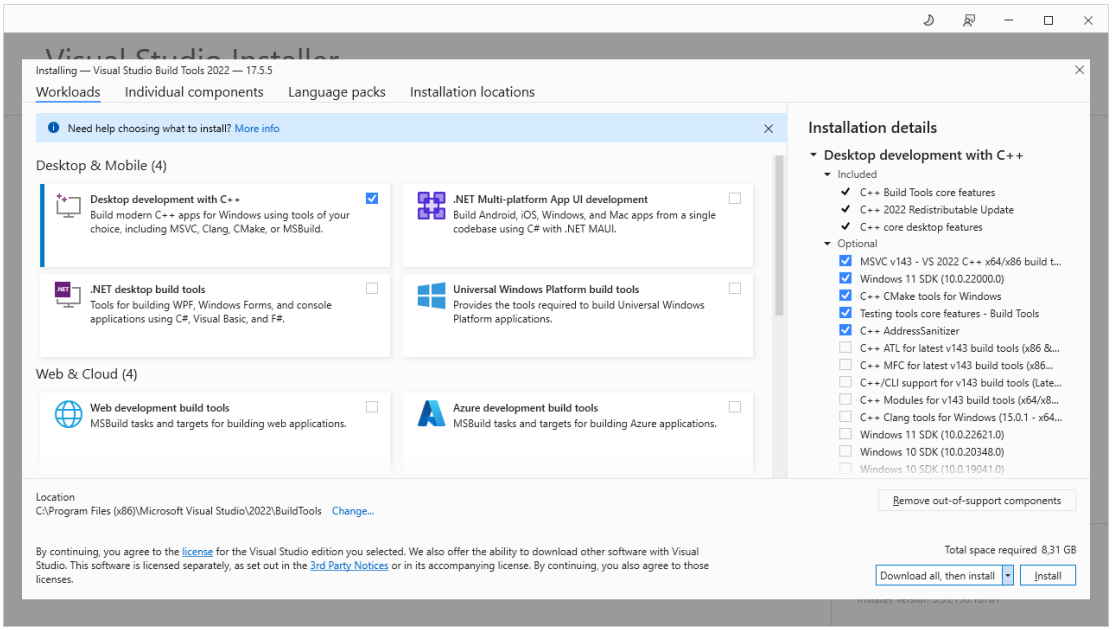


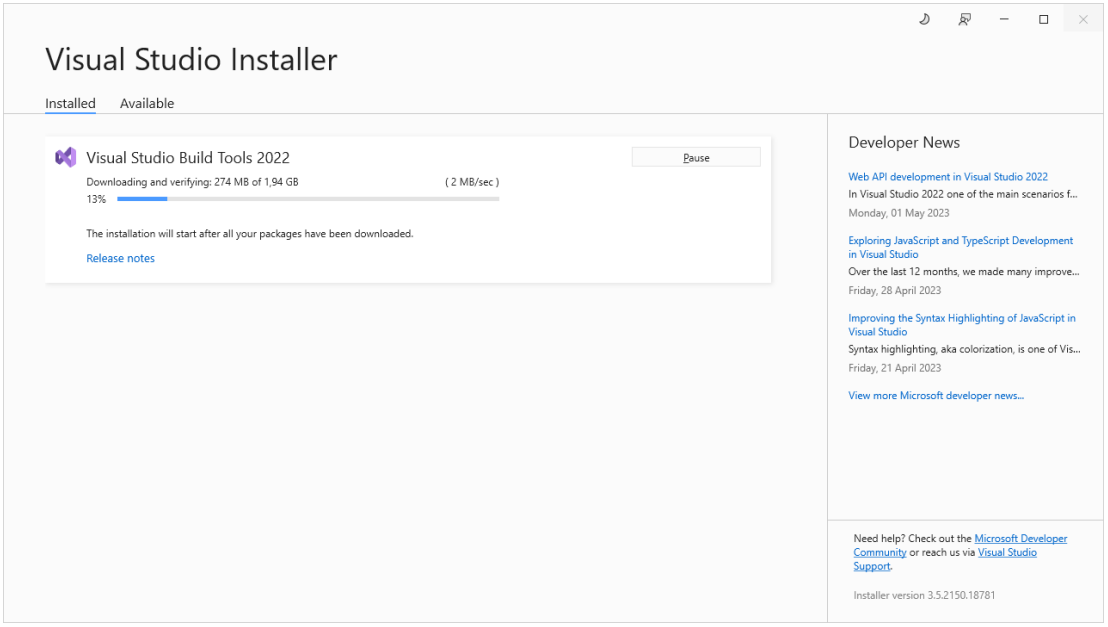
3. Install cmake-3.26.3-windows-x86_64.msi from https://cmake.org/download/ to C:\Dev\CMake\ - add cmake to the system PATH for all users.
https://github.com/Kitware/CMake/releases/download/v3.26.3/cmake-3.26.3-windows-x86_64.msi



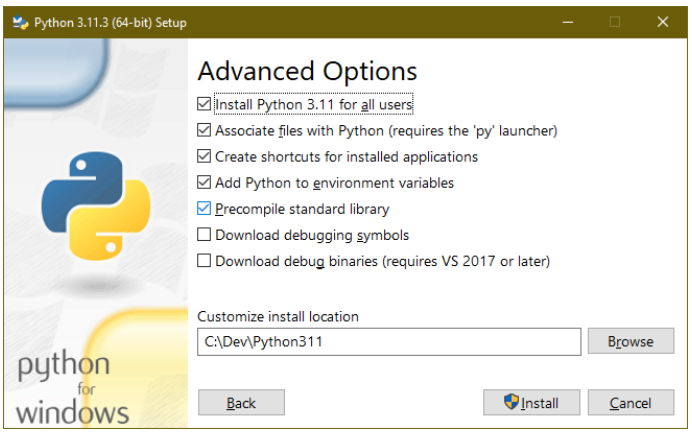


4. Install vs_BuildTools.exe from https://aka.ms/vs/17/release/vs_BuildTools.exe to the default folder - select C++ development tools. It was a 1.93 GB download.



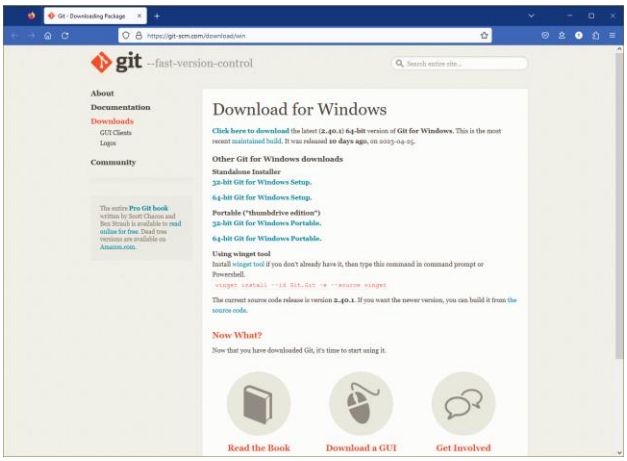


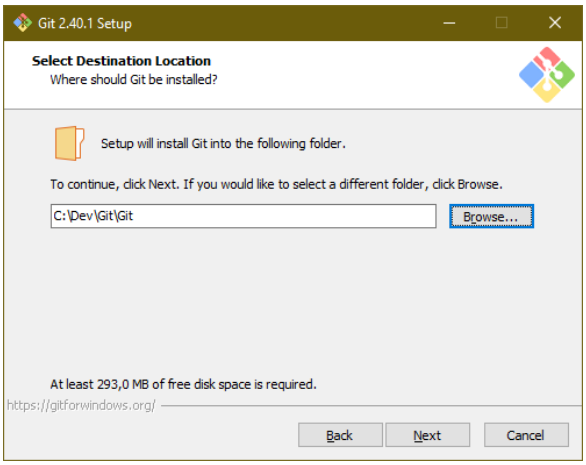
5. Python 3.11.3 - Install python-3.11.3-amd64.exe from https://www.python.org/ftp/python/3.11.3/ to C:\Dev\Python311 - select Add Python to PATH and also select to remove the max path length.
https://www.python.org/ftp/python/3.11.3/python-3.11.3-amd64.exe

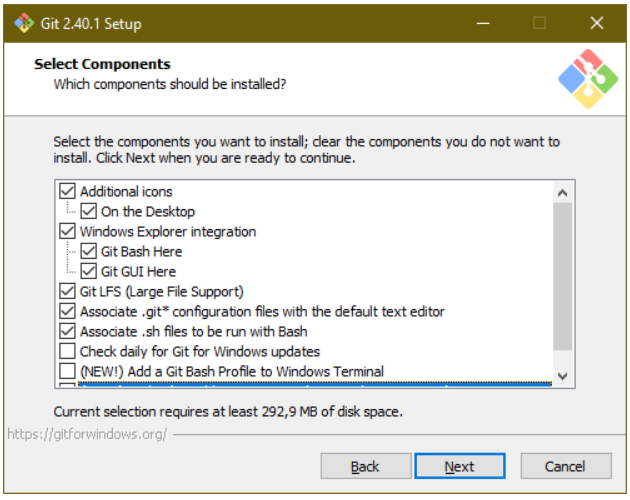


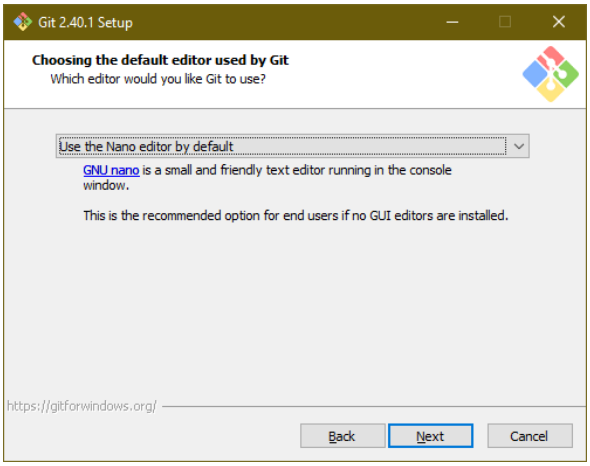
6. Install Git-2.40.1-64-bit.exe from https://git-scm.com/download/win to C:\Dev\Git - follow the instructions as below (from https://len42.github.io/rp2040-dev-setup.html).

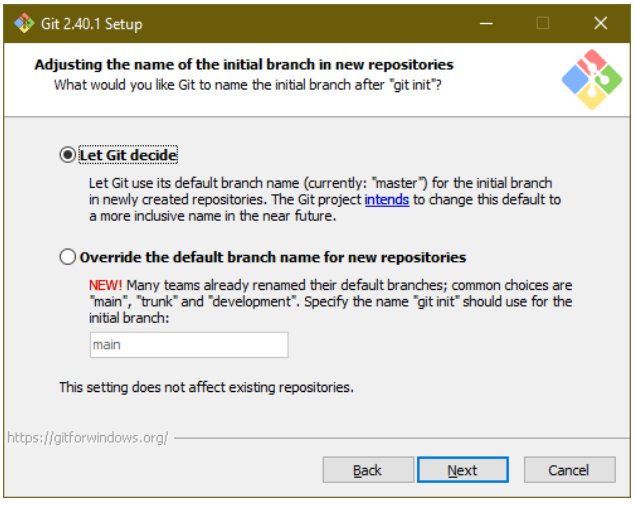
Destination Location: Default (or not)
Select Components: Default
Default editor: Select one you like.
Name of the initial branch: Let Git decide
PATH environment: Git from the command line and also from 3rd-party software
SSH executable: Use bundled OpenSSH
HTTPS transport backend: Use the OpenSSL library
Line ending conversion: Checkout as-is, commit as-is
Terminal emulator for Git Bash: Select either option
Default behavior of "git pull": Default (f-f or merge)
Credential helper: Default (Git Credential Manager Core)
Extra options: Default (Enable file system caching on, Enable symbolic links off)
Experimental options: Select "Enable experimental support for pseudo consoles"

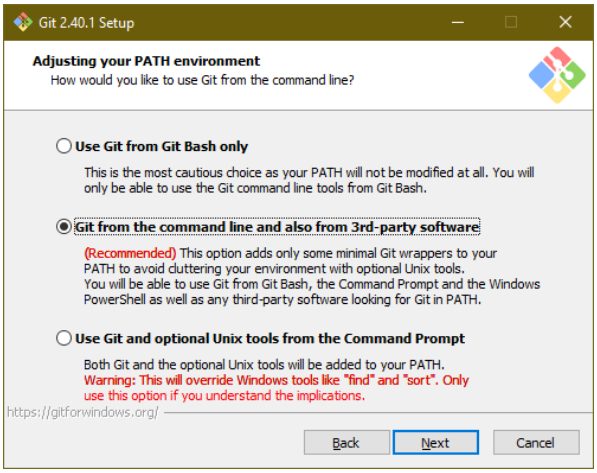


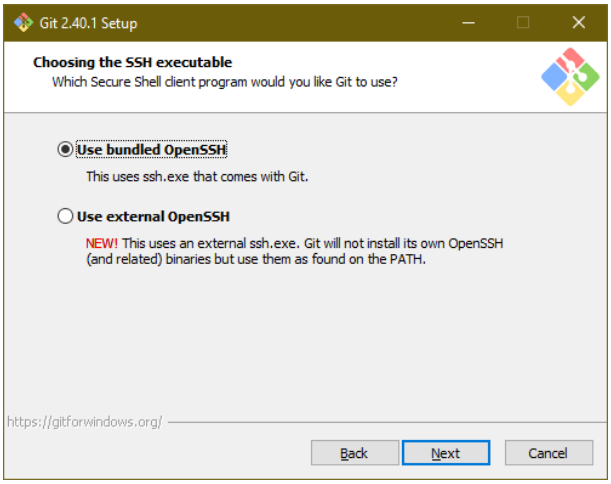


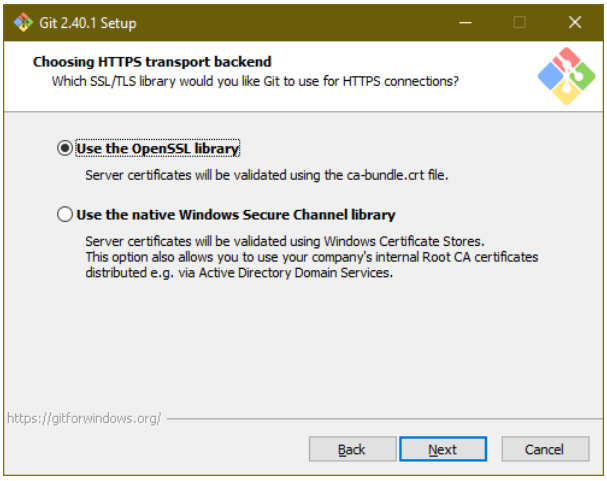


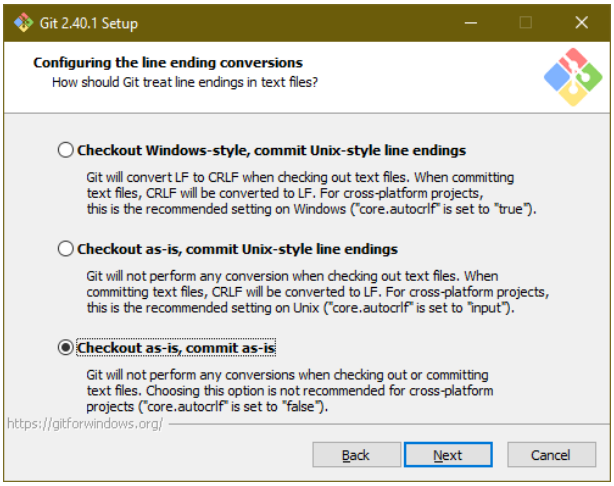


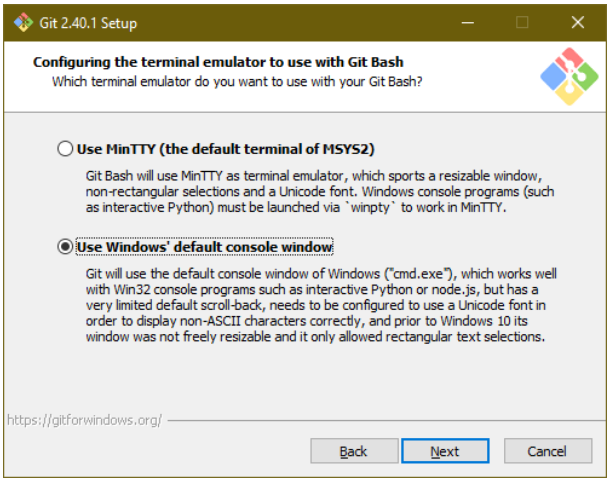


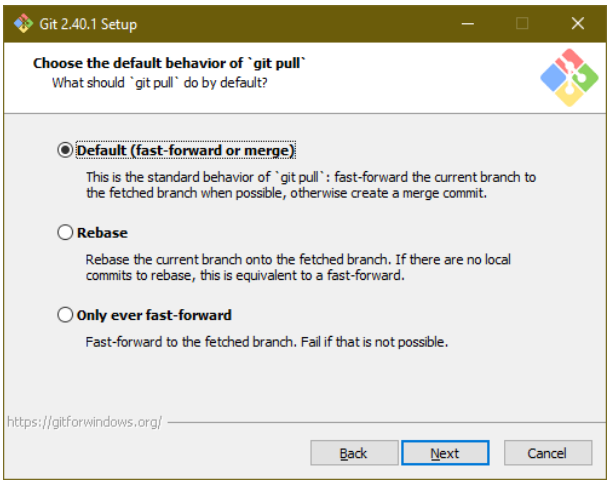


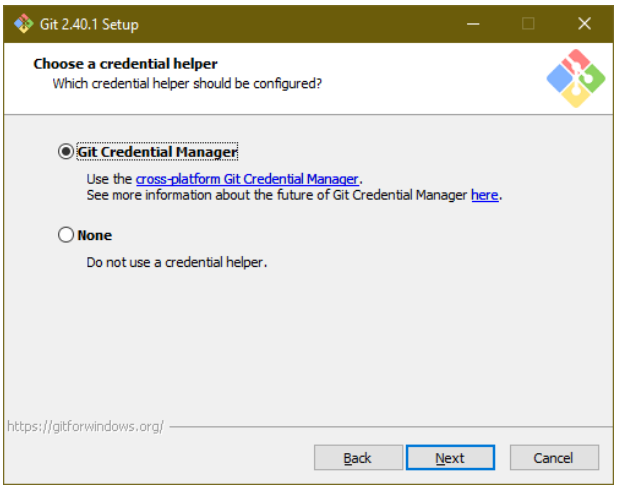

Git 2.40.1 Setup
Choosing the SSH executable
Which Secure Shell client program would you like Git to use?
Use bundled OpenSSH
This uses ssh.exe that comes with Git.
Use external OpenSSH
NEW! This uses an external ssh.exe. Git will not install its own OpenSSH (and related) binaries but use them as found on the PATH.
https://gitforwindows.org/
Back
Next
Cancel

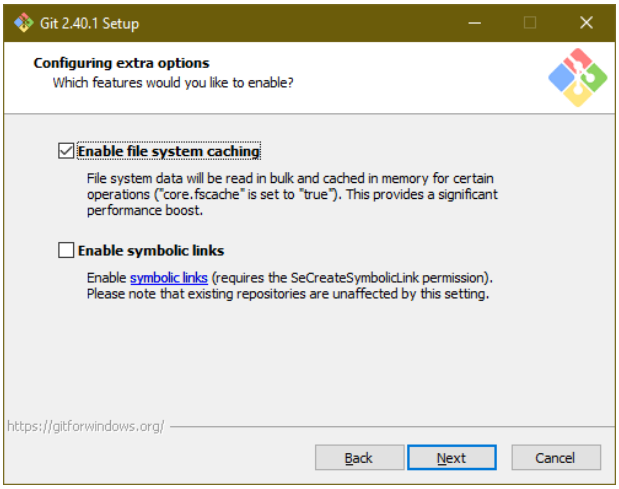

Git 2.40.1 Setup
Choosing HTTPS transport backend
Which SSL/TLS library would you like Git to use for HTTPS connections?
Use the OpenSSL library
Server certificates will be validated using the ca-bundle.crt file.
Use the native Windows Secure Channel library
Server certificates will be validated using Windows Certificate Stores.
This option also allows you to use your company's internal Root CA certificates distributed e.g. via Active Directory Domain Services.
https://gitforwindows.org/
Back
Next
Cancel

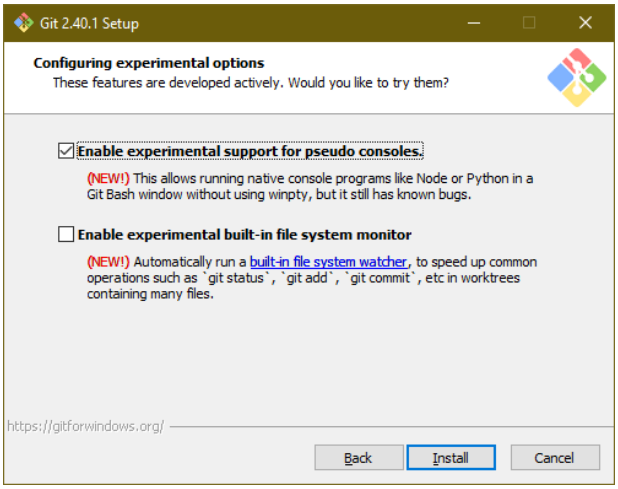

Git 2.40.1 Setup
Configuring the line ending conversions
How should Git treat line endings in text files?
Checkout Windows-style, commit Unix-style line endings
Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").
Checkout as-is, commit Unix-style line endings
Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").
Checkout as-is, commit as-is
Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").
https://gitforwindows.org/
Back
Next
Cancel


Git 2.40.1 Setup
Configuring the terminal emulator to use with Git Bash
Which terminal emulator do you want to use with your Git Bash?
Use MinTTY (the default terminal of MSYS2)
Git Bash will use MinTTY as terminal emulator, which sports a resizable window, non-rectangular selections and a Unicode font. Windows console programs (such as interactive Python) must be launched via `winpty` to work in MinTTY.
Use Windows' default console window
Git will use the default console window of Windows ("cmd.exe"), which works well with Win32 console programs such as interactive Python or node.js, but has a very limited default scroll-back, needs to be configured to use a Unicode font in order to display non-ASCII characters correctly, and prior to Windows 10 its window was not freely resizable and it only allowed rectangular text selections.
https://gitforwindows.org/
Back
Next
Cancel


Git 2.40.1 Setup
Choose the default behavior of `git pull`
What should `git pull` do by default?
Default (fast-forward or merge)
This is the standard behavior of `git pull`: fast-forward the current branch to the fetched branch when possible, otherwise create a merge commit.
Rebase
Rebase the current branch onto the fetched branch. If there are no local commits to rebase, this is equivalent to a fast-forward.
Only ever fast-forward
Fast-forward to the fetched branch. Fail if that is not possible.
https://gitforwindows.org/
Back
Next
Cancel

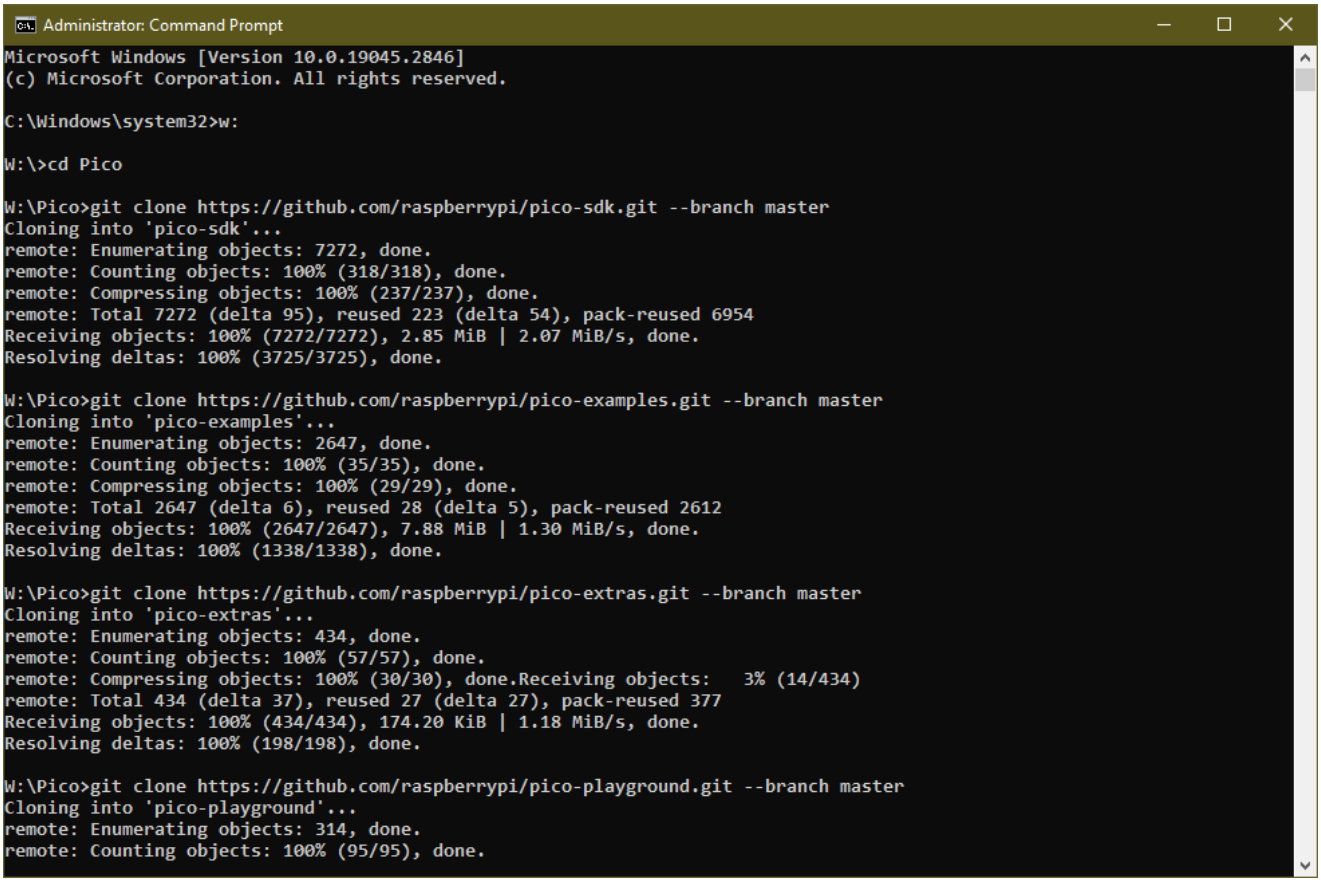

Git 2.40.1 Setup
Choose a credential helper
Which credential helper should be configured?
Git Credential Manager
Use the cross-platform Git Credential Manager.
See more information about the future of Git Credential Manager here.
None
Do not use a credential helper.
https://gitforwindows.org/
Back
Next
Cancel


Git 2.40.1 Setup
Configuring extra options
Which features would you like to enable?
Enable file system caching
File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.
Enable symbolic links
Enable symbolic links (requires the SeCreateSymbolicLink permission).
Please note that existing repositories are unaffected by this setting.
https://gitforwindows.org/
Back
Next
Cancel

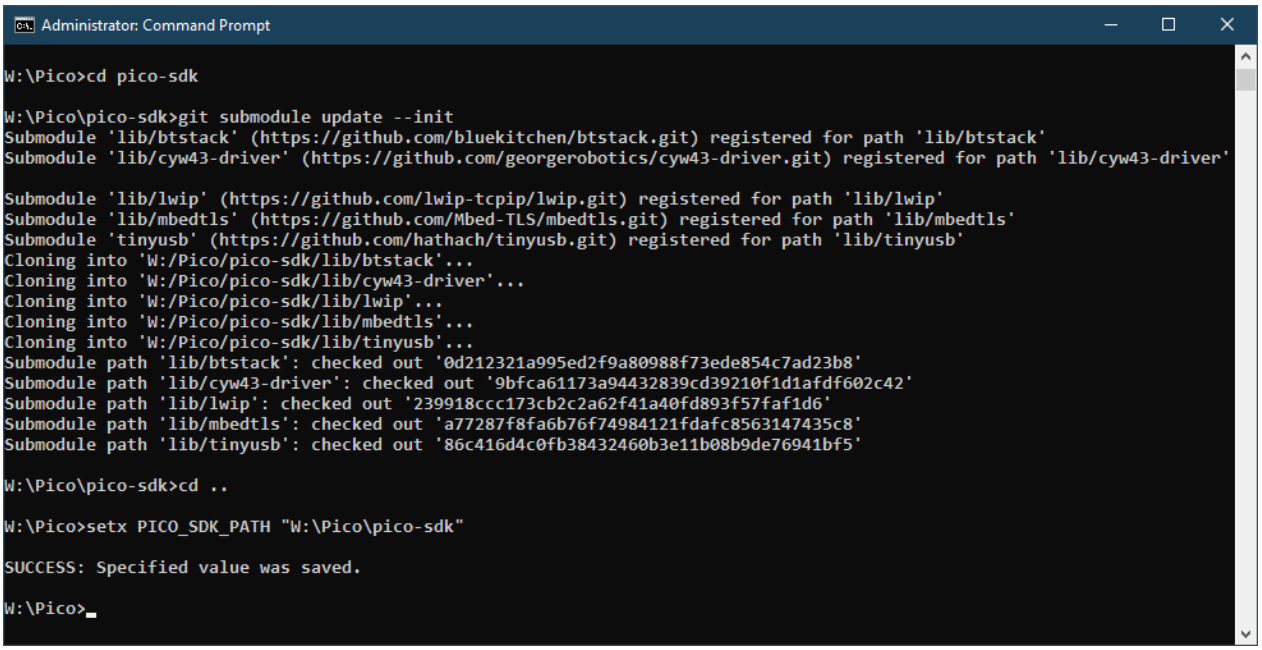

Git 2.40.1 Setup
Configuring experimental options
These features are developed actively. Would you like to try them?
Enable experimental support for pseudo consoles.
(NEW!) This allows running native console programs like Node or Python in a Git Bash window without using winpty, but it still has known bugs.
Enable experimental built-in file system monitor
(NEW!) Automatically run a built-in file system watcher, to speed up common operations such as `git status`, `git add`, `git commit`, etc in worktrees containing many files.
https://gitforwindows.org/
Back
Install
Cancel

7. Use the windows admin cmd prompt to install the Pico SDK

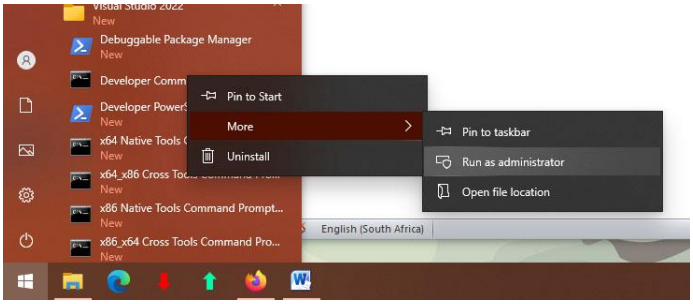
```
Microsoft Windows [Version 10.0.19045.2846]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>w:

W:\>cd Pico

W:\Pico>git clone https://github.com/raspberrypi/pico-sdk.git --branch master
Cloning into 'pico-sdk'...
remote: Enumerating objects: 7272, done.
remote: Counting objects: 100% (318/318), done.
remote: Compressing objects: 100% (237/237), done.
remote: Total 7272 (delta 95), reused 223 (delta 54), pack-reused 6954
Receiving objects: 100% (7272/7272), 2.85 MiB | 2.07 MiB/s, done.
Resolving deltas: 100% (3725/3725), done.

W:\Pico>git clone https://github.com/raspberrypi/pico-examples.git --branch master
Cloning into 'pico-examples'...
remote: Enumerating objects: 2647, done.
remote: Counting objects: 100% (35/35), done.
remote: Compressing objects: 100% (29/29), done.
remote: Total 2647 (delta 6), reused 28 (delta 5), pack-reused 2612
Receiving objects: 100% (2647/2647), 7.88 MiB | 1.30 MiB/s, done.
Resolving deltas: 100% (1338/1338), done.

W:\Pico>git clone https://github.com/raspberrypi/pico-extras.git --branch master
Cloning into 'pico-extras'...
remote: Enumerating objects: 434, done.
remote: Counting objects: 100% (57/57), done.
remote: Compressing objects: 100% (30/30), done.Receiving objects:   3% (14/434)
remote: Total 434 (delta 37), reused 27 (delta 27), pack-reused 377
Receiving objects: 100% (434/434), 174.20 KiB | 1.18 MiB/s, done.
Resolving deltas: 100% (198/198), done.

W:\Pico>git clone https://github.com/raspberrypi/pico-playground.git --branch master
Cloning into 'pico-playground'...
remote: Enumerating objects: 314, done.
remote: Counting objects: 100% (95/95), done.
remote: Compressing objects: 100% (54/54), done.
remote: Total 314 (delta 56), reused 54 (delta 41), pack-reused 219
Receiving objects: 100% (314/314), 2.28 MiB | 2.25 MiB/s, done.
Resolving deltas: 100% (123/123), done.

W:\Pico>
```



```
W:\Pico>cd pico-sdk

W:\Pico\pico-sdk>git submodule update --init
Submodule 'lib/btstack' (https://github.com/bluekitchen/btstack.git) registered for path 'lib/btstack'
Submodule 'lib/cyw43-driver' (https://github.com/georgerobotics/cyw43-driver.git) registered for path
'lib/cyw43-driver'
Submodule 'lib/lwip' (https://github.com/lwip-tcpip/lwip.git) registered for path 'lib/lwip'
Submodule 'lib/mbedtls' (https://github.com/Mbed-TLS/mbedtls.git) registered for path 'lib/mbedtls'
Submodule 'tinyusb' (https://github.com/hathach/tinyusb.git) registered for path 'lib/tinyusb'
Cloning into 'W:/Pico/pico-sdk/lib/btstack'...
Cloning into 'W:/Pico/pico-sdk/lib/cyw43-driver'...
Cloning into 'W:/Pico/pico-sdk/lib/lwip'...
Cloning into 'W:/Pico/pico-sdk/lib/mbedtls'...
Cloning into 'W:/Pico/pico-sdk/lib/tinyusb'...
Submodule path 'lib/btstack': checked out '0d212321a995ed2f9a80988f73ede854c7ad23b8'
Submodule path 'lib/cyw43-driver': checked out '9bfca61173a94432839cd39210f1d1afdf602c42'
Submodule path 'lib/lwip': checked out '239918ccc173cb2c2a62f41a40fd893f57faf1d6'
Submodule path 'lib/mbedtls': checked out 'a77287f8fa6b76f74984121fdafc8563147435c8'
Submodule path 'lib/tinyusb': checked out '86c416d4c0fb38432460b3e11b08b9de76941bf5'

W:\Pico\pico-sdk>cd ..

W:\Pico>setx PICO_SDK_PATH "W:\Pico\pico-sdk"

SUCCESS: Specified value was saved.

W:\Pico>
```

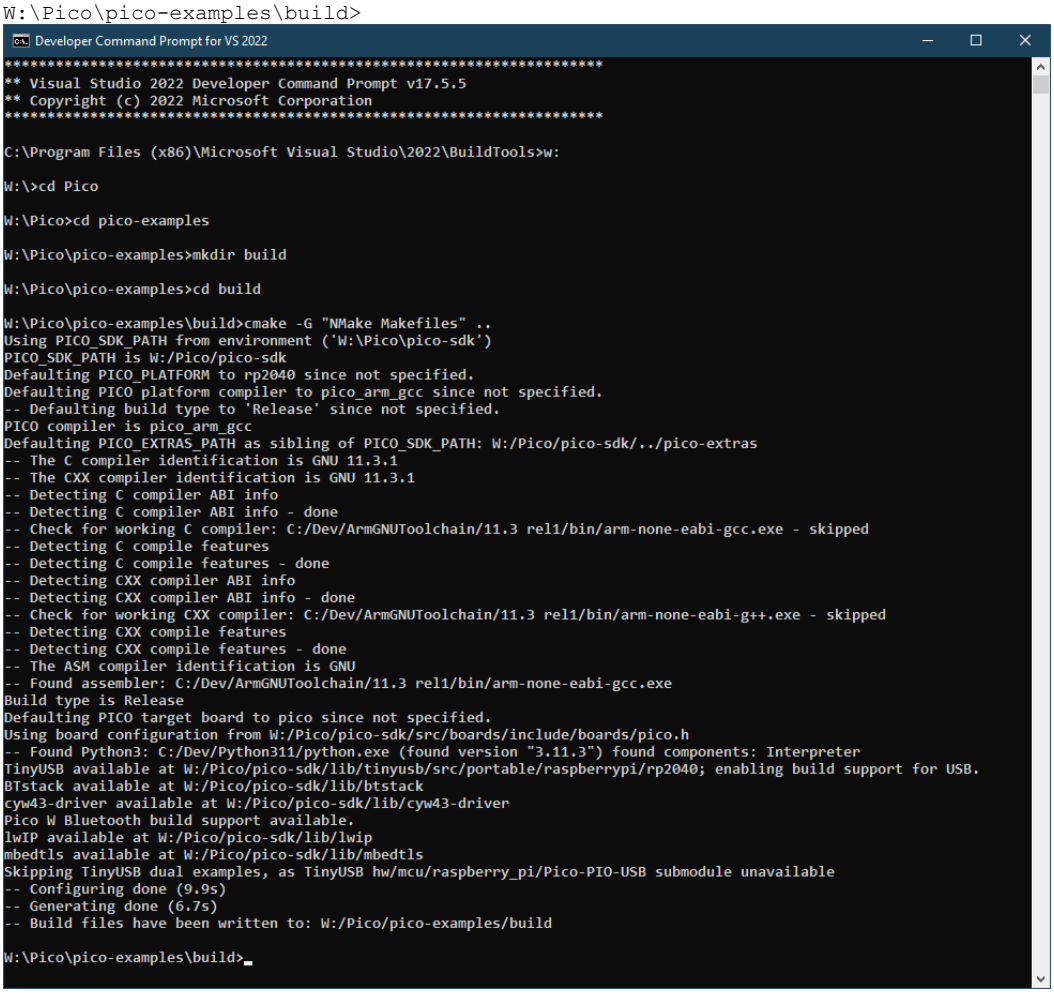


9. Close the cmd window and run the VS Developer Command Prompt (optionally as admin), to start the build on all (non-W) the examples in pico-examples.

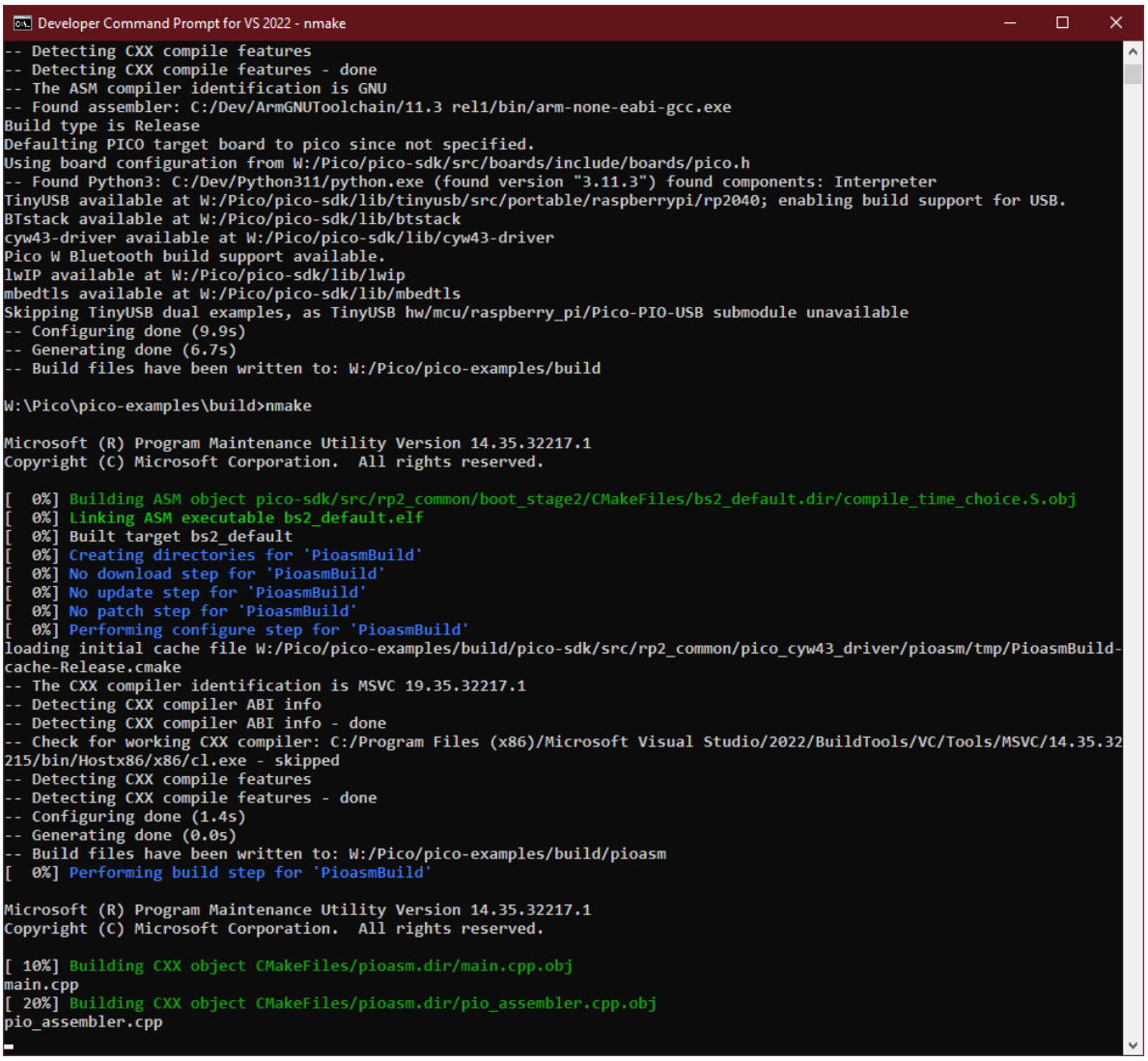


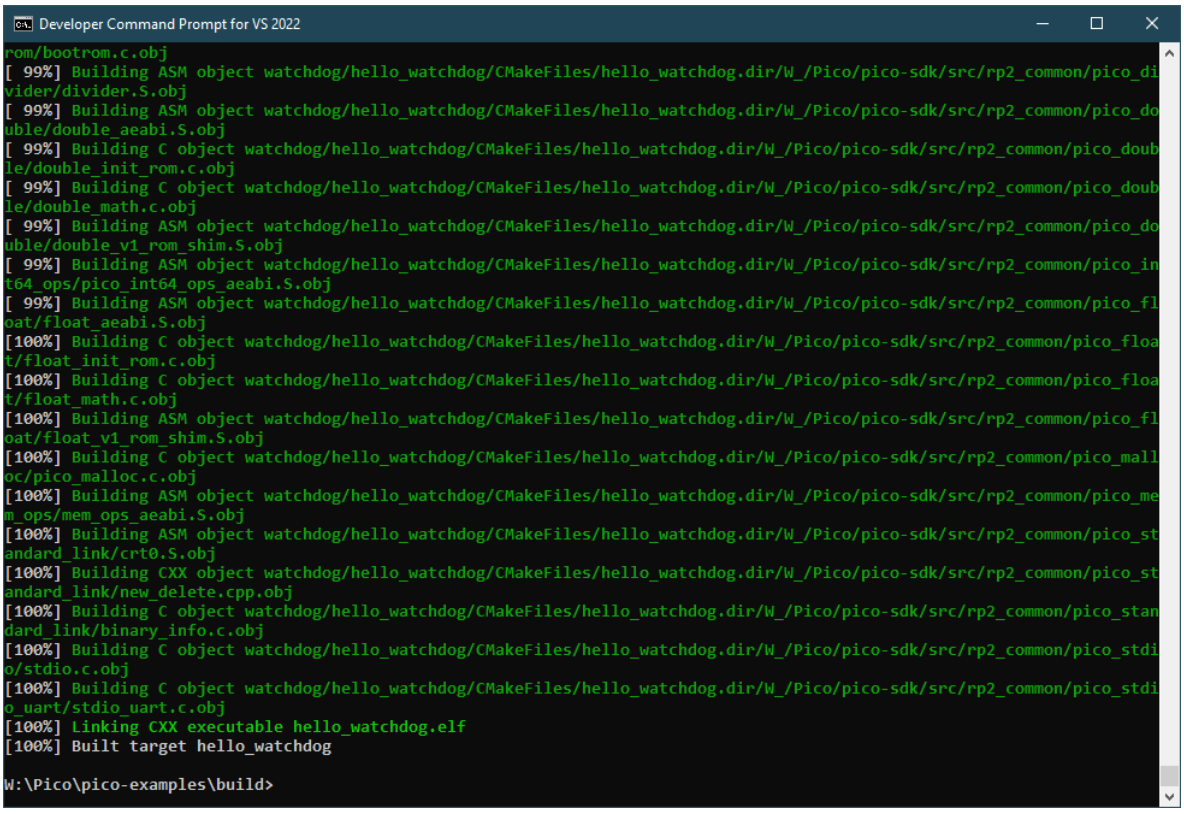

```
**********************************************************************
** Visual Studio 2022 Developer Command Prompt v17.5.5
** Copyright (c) 2022 Microsoft Corporation
**********************************************************************

W:\Pico>cd pico-examples
```

```
W:\Pico\pico-examples>mkdir build

W:\Pico\pico-examples>cd build

W:\Pico\pico-examples\build>cmake -G "NMake Makefiles" ..
Using PICO_SDK_PATH from environment ('W:\Pico\pico-sdk')
PICO_SDK_PATH is W:/Pico/pico-sdk
Defaulting PICO_PLATFORM to rp2040 since not specified.
Defaulting PICO platform compiler to pico_arm_gcc since not specified.
-- Defaulting build type to 'Release' since not specified.
PICO compiler is pico_arm_gcc
Defaulting PICO_EXTRAS_PATH as sibling of PICO_SDK_PATH: W:/Pico/pico-sdk/../pico-extras
-- The C compiler identification is GNU 11.3.1
-- The CXX compiler identification is GNU 11.3.1
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: C:/Dev/ArmGNUToolchain/11.3 rel1/bin/arm-none-eabi-gcc.exe - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: C:/Dev/ArmGNUToolchain/11.3 rel1/bin/arm-none-eabi-g++.exe - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- The ASM compiler identification is GNU
-- Found assembler: C:/Dev/ArmGNUToolchain/11.3 rel1/bin/arm-none-eabi-gcc.exe
Build type is Release
Defaulting PICO target board to pico since not specified.
Using board configuration from W:/Pico/pico-sdk/src/boards/include/boards/pico.h
-- Found Python3: C:/Dev/Python311/python.exe (found version "3.11.3") found components: Interpreter
TinyUSB available at W:/Pico/pico-sdk/lib/tinyusb/src/portable/raspberrypi/rp2040; enabling build support for
USB.
BTstack available at W:/Pico/pico-sdk/lib/btstack
cyw43-driver available at W:/Pico/pico-sdk/lib/cyw43-driver
Pico W Bluetooth build support available.
lwIP available at W:/Pico/pico-sdk/lib/lwip
mbedtls available at W:/Pico/pico-sdk/lib/mbedtls
Skipping TinyUSB dual examples, as TinyUSB hw/mcu/raspberry_pi/Pico-PIO-USB submodule unavailable
-- Configuring done (9.9s)
-- Generating done (6.7s)
-- Build files have been written to: W:/Pico/pico-examples/build

W:\Pico\pico-examples\build>
```

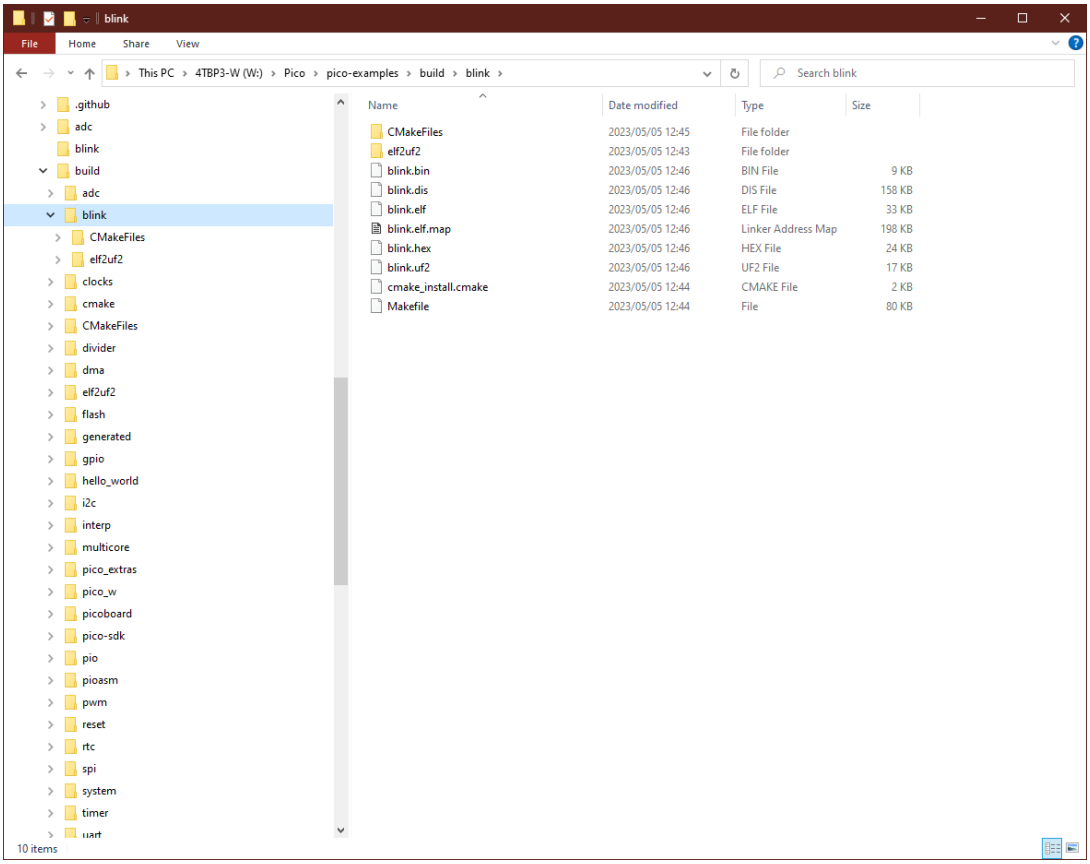


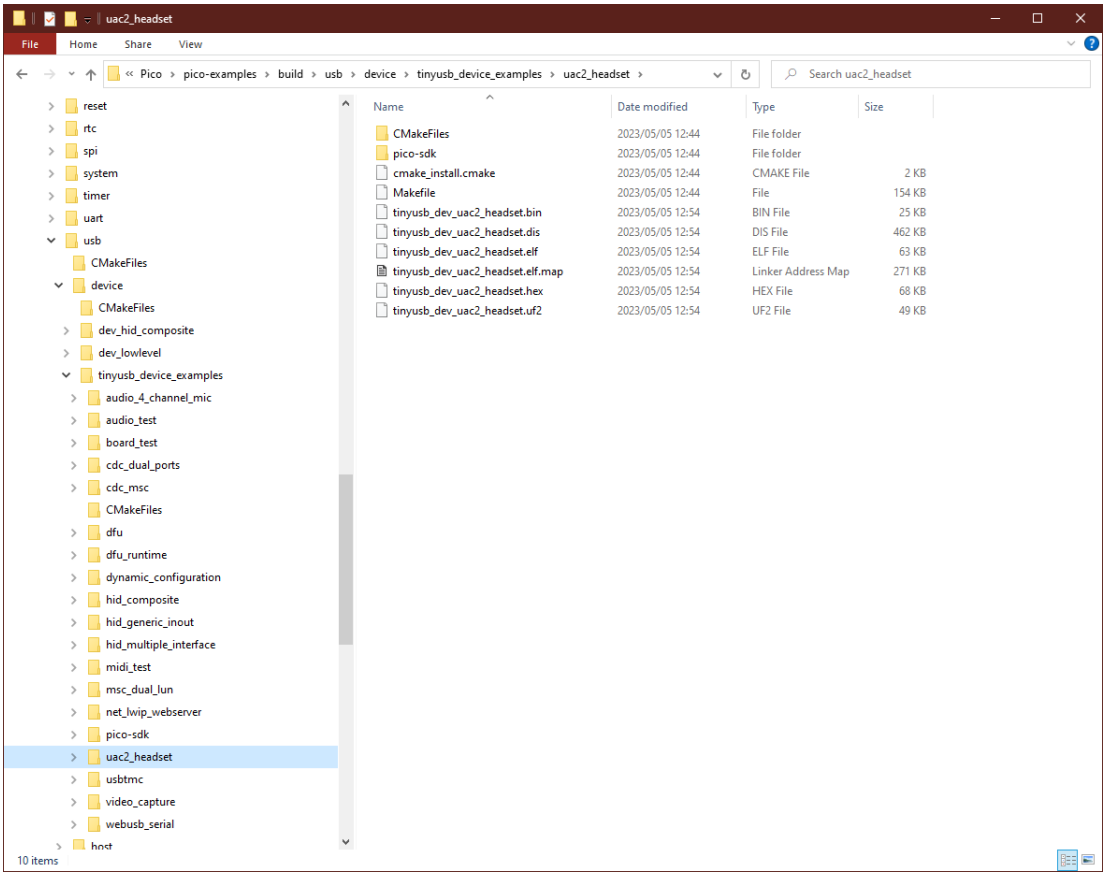
```
W:\Pico\pico-examples\build>nmake
```





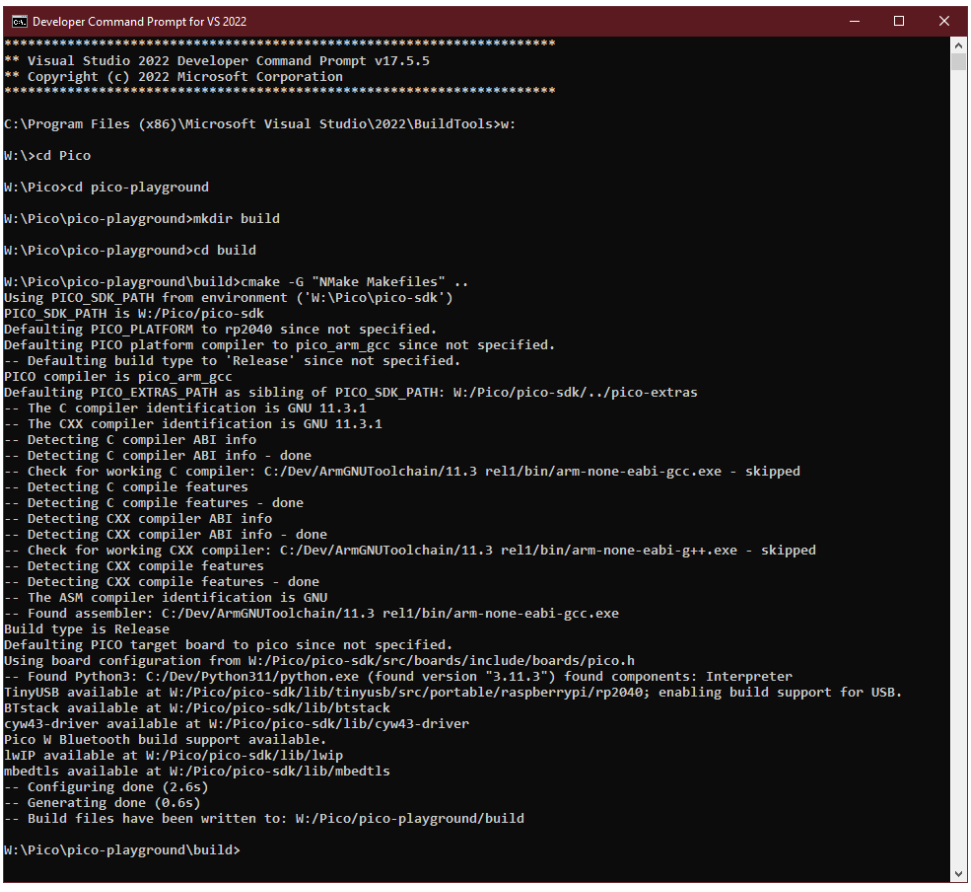
The uf2 files are in the pico-examples\build\sub-folders – including the tinyusb examples.



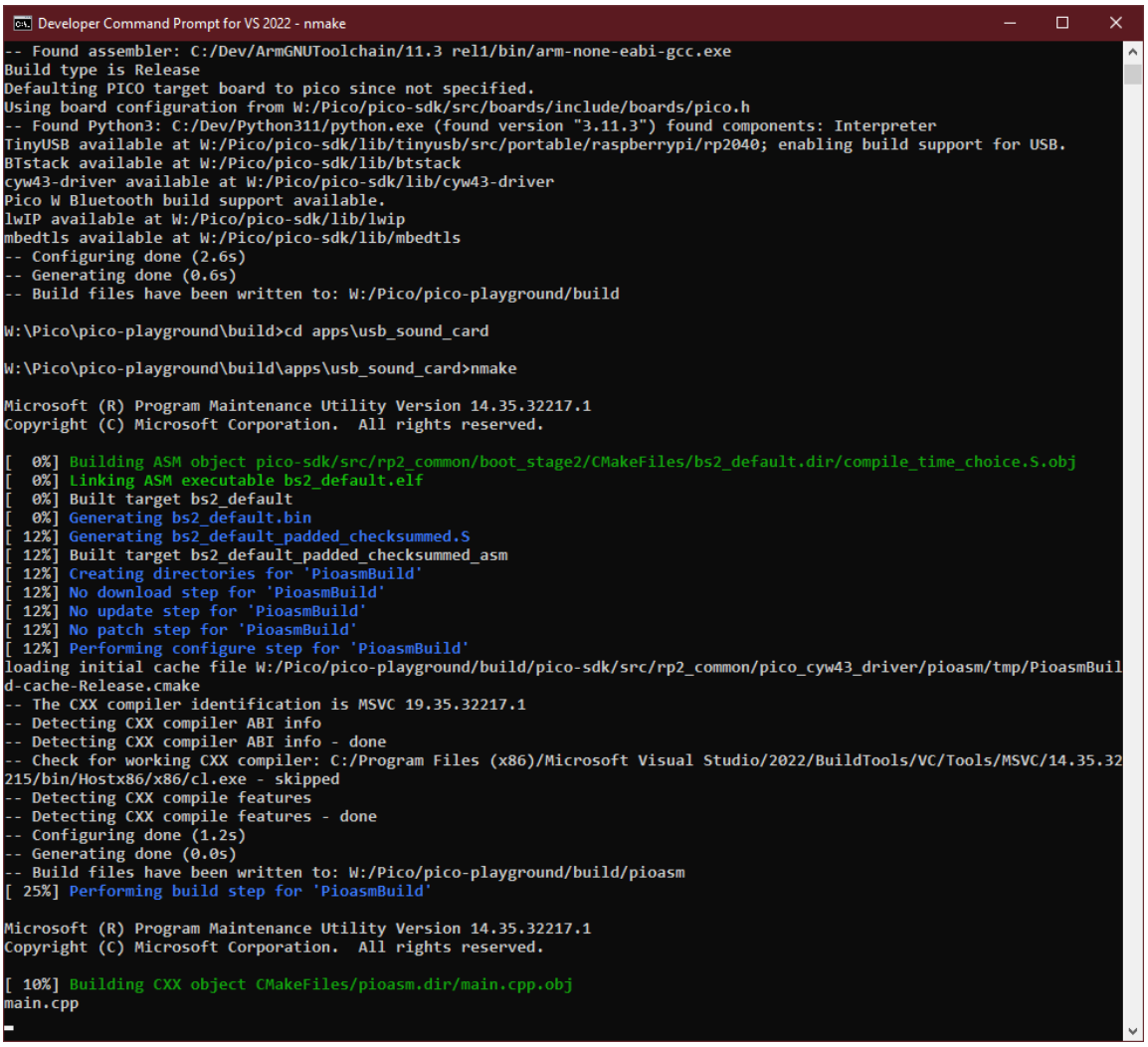


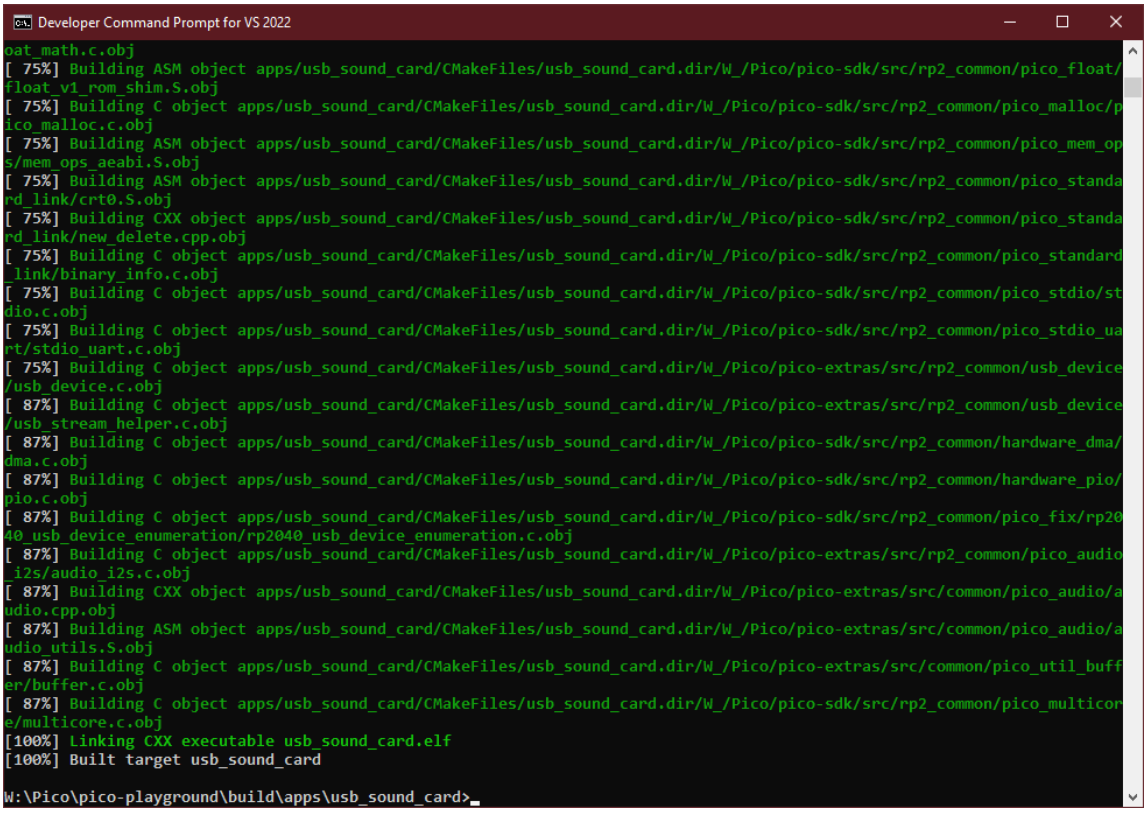
Building the usb_sound_card only:

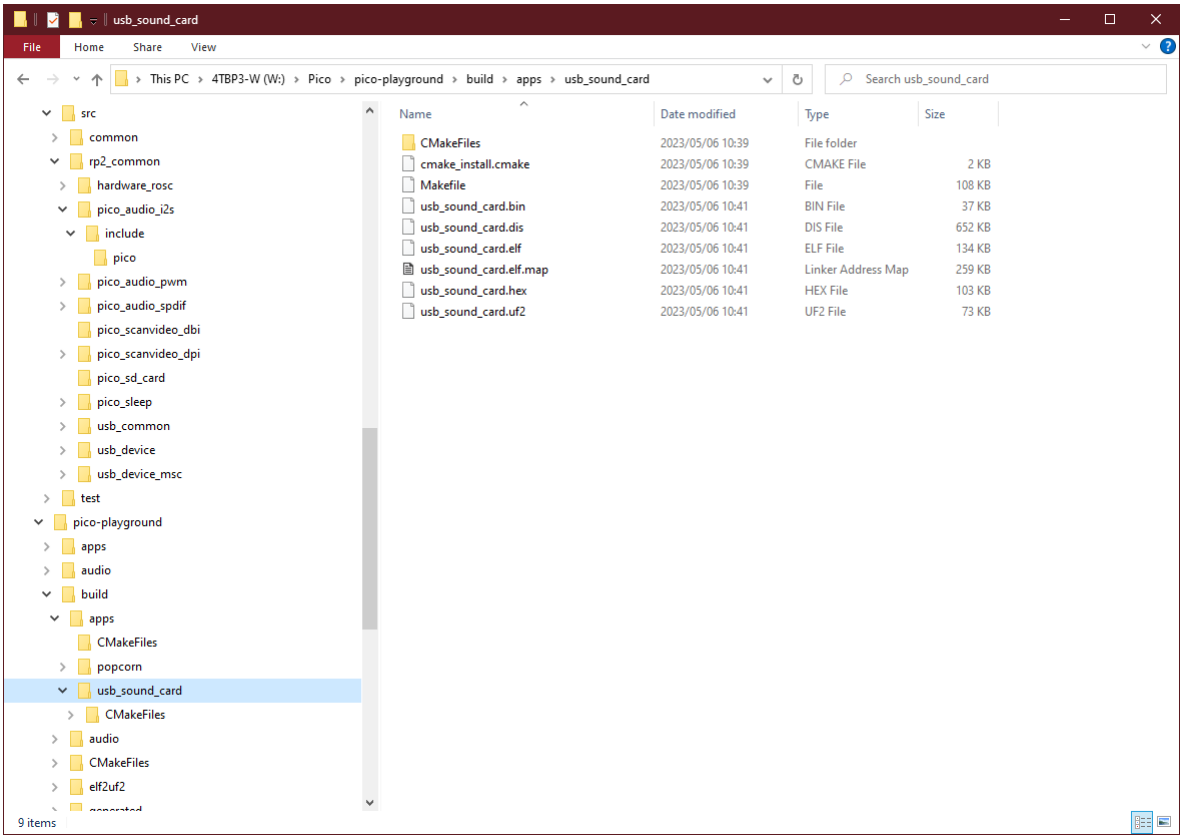
```
**********************************************************************
** Visual Studio 2022 Developer Command Prompt v17.5.5
** Copyright (c) 2022 Microsoft Corporation
**********************************************************************

W:\Pico>cd pico-playground

W:\Pico\pico-playground>mkdir build

W:\Pico\pico-playground>cd build

W:\Pico\pico-playground\build>cmake -G "NMake Makefiles" ..
Using PICO_SDK_PATH from environment ('W:\Pico\pico-sdk')
PICO_SDK_PATH is W:/Pico/pico-sdk
Defaulting PICO_PLATFORM to rp2040 since not specified.
Defaulting PICO platform compiler to pico_arm_gcc since not specified.
-- Defaulting build type to 'Release' since not specified.
PICO compiler is pico_arm_gcc
Defaulting PICO_EXTRAS_PATH as sibling of PICO_SDK_PATH: W:/Pico/pico-sdk/../pico-extras
-- The C compiler identification is GNU 11.3.1
-- The CXX compiler identification is GNU 11.3.1
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: C:/Dev/ArmGNUToolchain/11.3 rel1/bin/arm-none-eabi-gcc.exe - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: C:/Dev/ArmGNUToolchain/11.3 rel1/bin/arm-none-eabi-g++.exe - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- The ASM compiler identification is GNU
-- Found assembler: C:/Dev/ArmGNUToolchain/11.3 rel1/bin/arm-none-eabi-gcc.exe
Build type is Release
Defaulting PICO target board to pico since not specified.
Using board configuration from W:/Pico/pico-sdk/src/boards/include/boards/pico.h
-- Found Python3: C:/Dev/Python311/python.exe (found version "3.11.3") found components: Interpreter
TinyUSB available at W:/Pico/pico-sdk/lib/tinyusb/src/portable/raspberrypi/rp2040; enabling build support for
USB.
BTstack available at W:/Pico/pico-sdk/lib/btstack
cyw43-driver available at W:/Pico/pico-sdk/lib/cyw43-driver
Pico W Bluetooth build support available.
lwIP available at W:/Pico/pico-sdk/lib/lwip
mbedtls available at W:/Pico/pico-sdk/lib/mbedtls
-- Configuring done (2.6s)
-- Generating done (0.6s)
-- Build files have been written to: W:/Pico/pico-playground/build
```



```
W:\Pico\pico-playground\build>
W:\Pico\pico-playground\build>cd apps\usb_sound_card
W:\Pico\pico-playground\build\apps\usb_sound_card>nmake
```






usb_sound_card
File
Home
Share
View
This PC › 4TBP3-W (W:) › Pico › pico-playground › build › apps › usb_sound_card
Search usb_sound_card
src
common
rp2_common
hardware_rosc
pico_audio_i2s
include
pico
pico_audio_pwm
pico_audio_spdif
pico_scanvideo_dbi
pico_scanvideo_dpi
pico_sd_card
pico_sleep
usb_common
usb_device
usb_device_msc
test
pico-playground
apps
audio
build
apps
CMakeFiles
popcorn
usb_sound_card
CMakeFiles
audio
CMakeFiles
elf2uf2
Name
Date modified
Type
Size
CMakeFiles 2023/05/06 10:39 File folder
cmake_install.cmake 2023/05/06 10:39 CMAKE File 2 KB
Makefile 2023/05/06 10:39 File 108 KB
usb_sound_card.bin 2023/05/06 10:41 BIN File 37 KB
usb_sound_card.dis 2023/05/06 10:41 DIS File 652 KB
usb_sound_card.elf 2023/05/06 10:41 ELF File 134 KB
usb_sound_card.elf.map 2023/05/06 10:41 Linker Address Map 259 KB
usb_sound_card.hex 2023/05/06 10:41 HEX File 103 KB
usb_sound_card.uf2 2023/05/06 10:41 UF2 File 73 KB
9 items