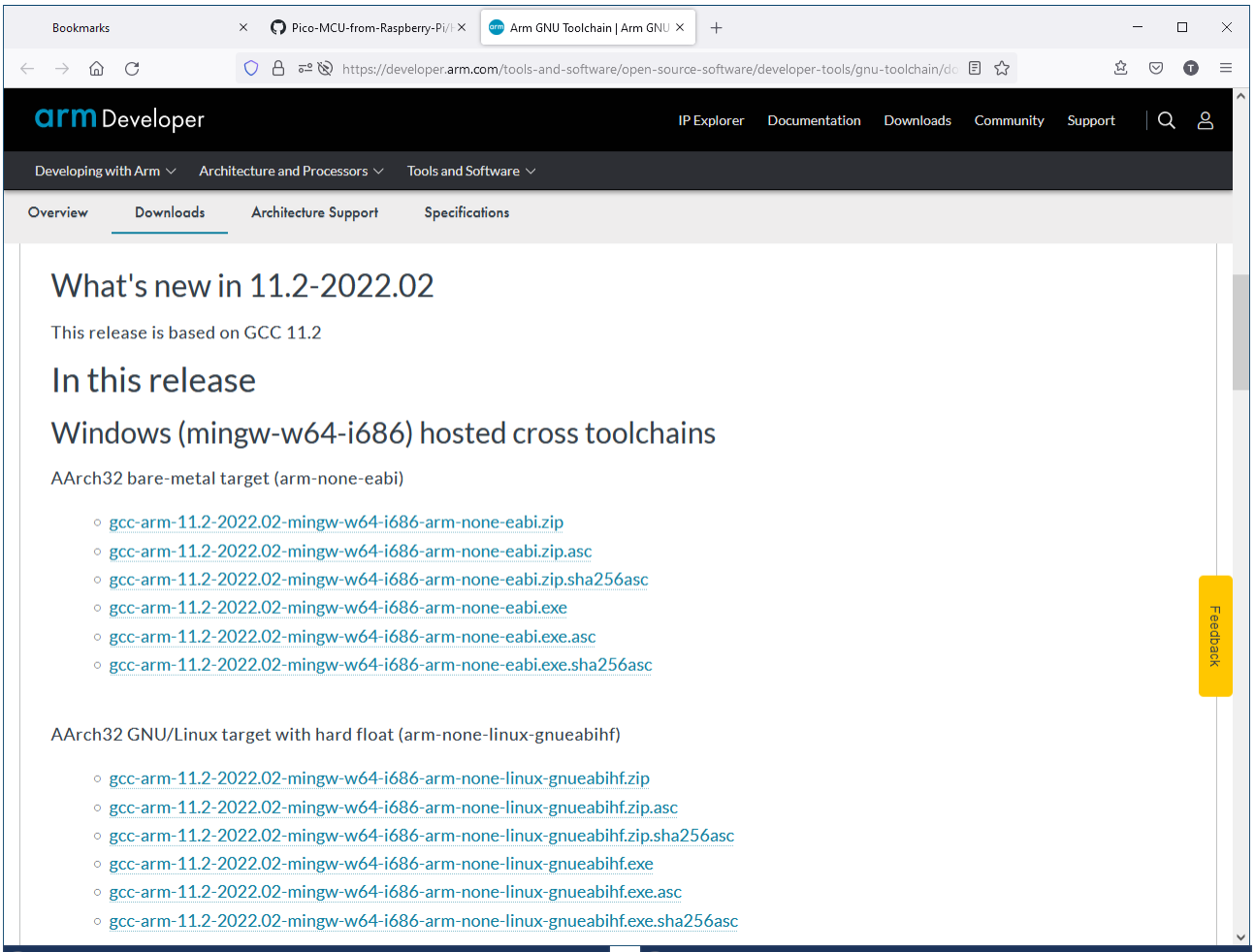
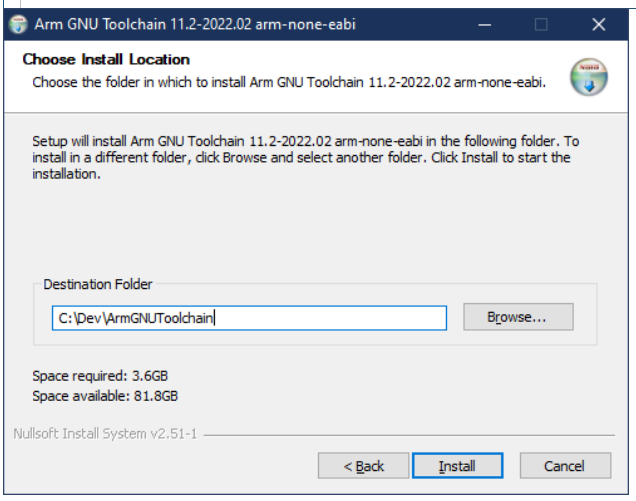
Install Pico (W) SDK 1.40 in Windows 10x64 July 2022

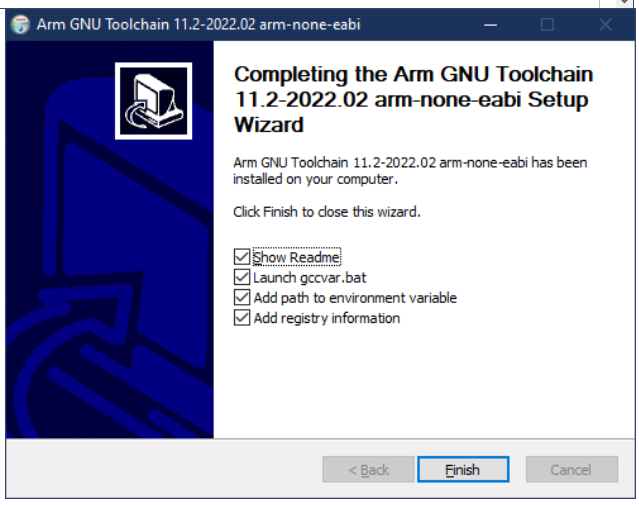
Partly based on RP2040 Development Setup on Windows https://len42.github.io/rp2040-dev-setup.html

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.*

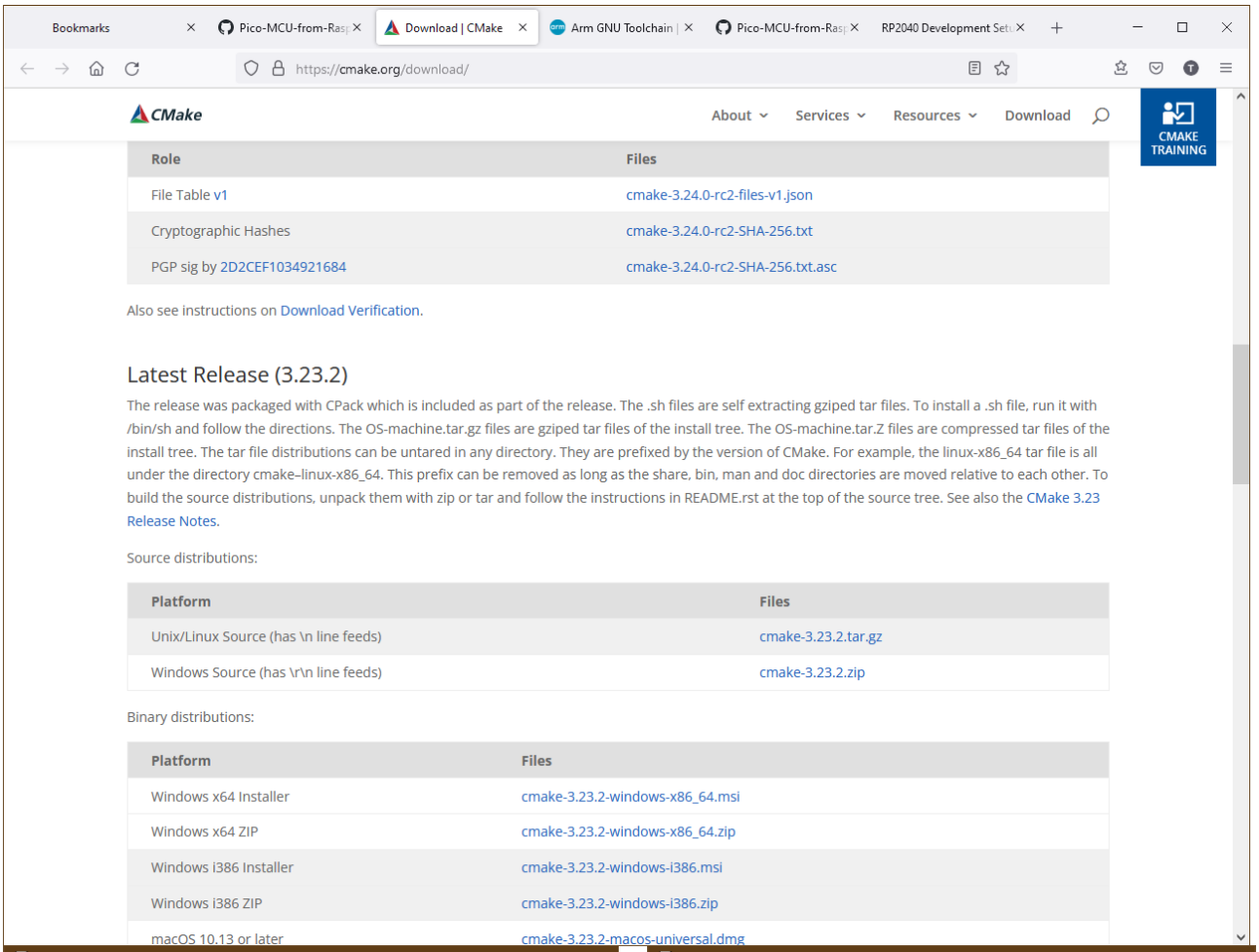
2. Install gcc-arm-11.2-2022.02-mingw-w64-i686-arm-none-eabi.exe from: https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/downloads to C:\Dev\ArmGNUToolchain - add path to environment variable during install.
https://developer.arm.com/-/media/Files/downloads/gnu/11.2-2022.02/binrel/gcc-arm-11.2-2022.02-mingw-w64-i686-arm-none-eabi.exe

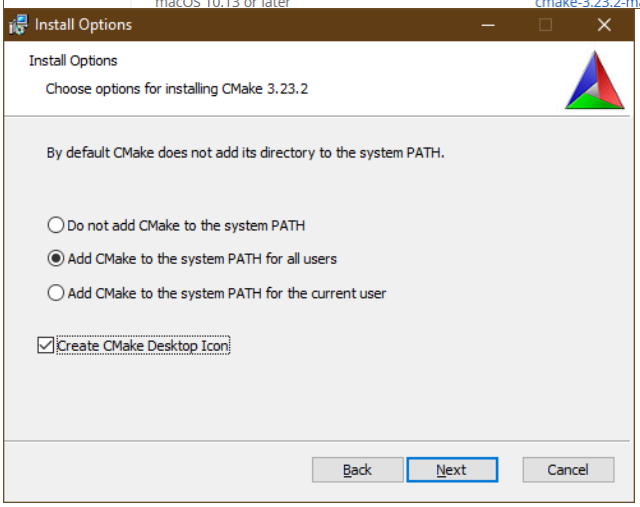


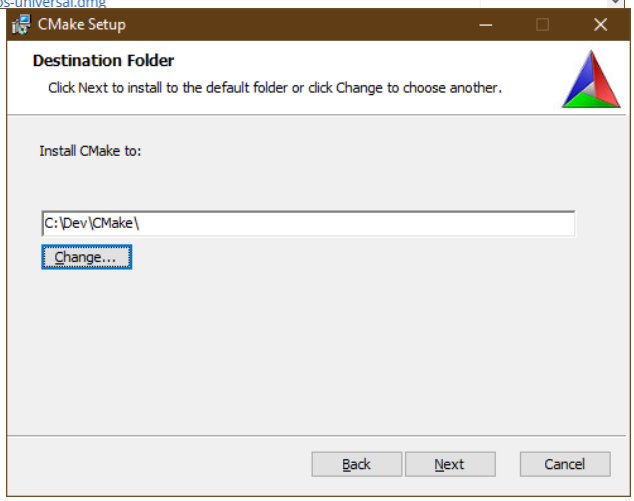




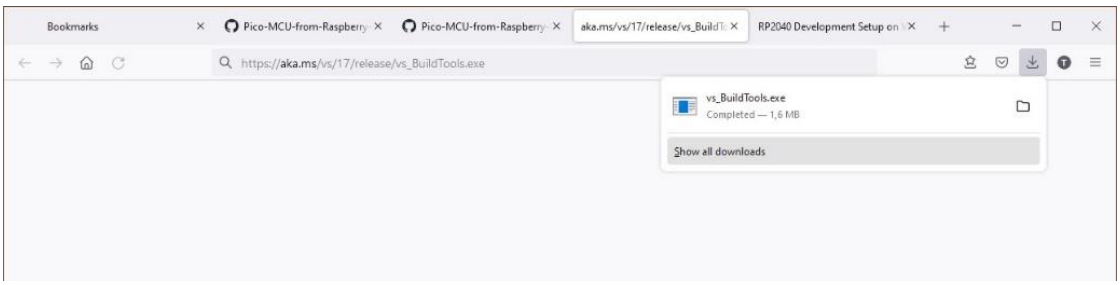
3. Install cmake-3.23.2-windows-x86_64.msi from https://cmake.org/download/ to C:\Dev\CMake\ - add Cmake to the system PATH for all users.
https://github.com/Kitware/CMake/releases/download/v3.23.2/cmake-3.23.2-windows-x86_64.msi

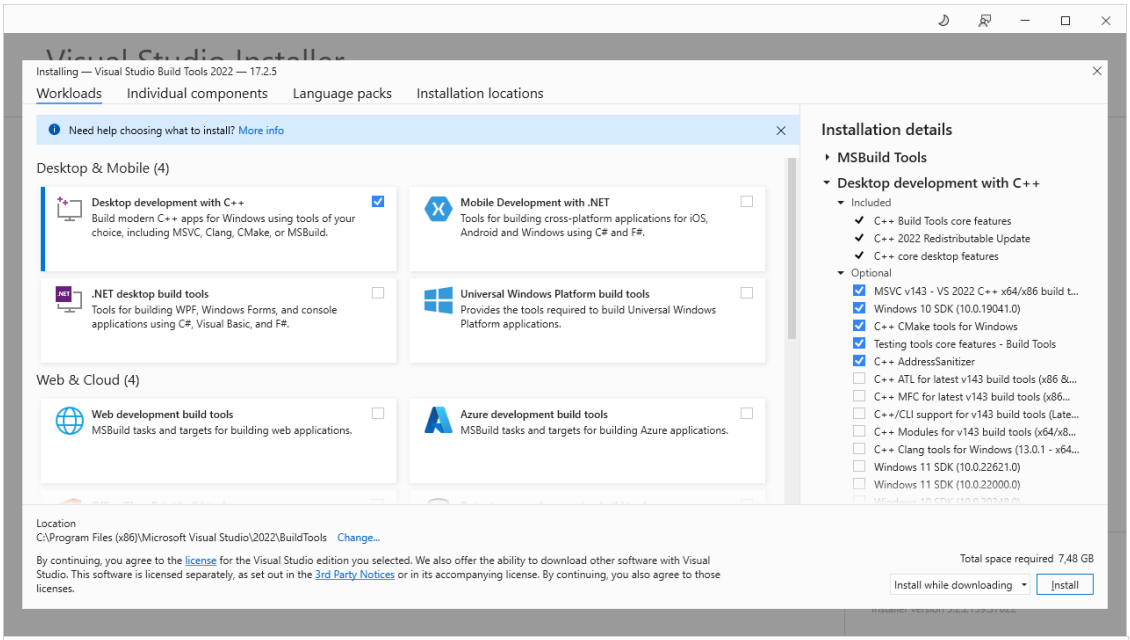


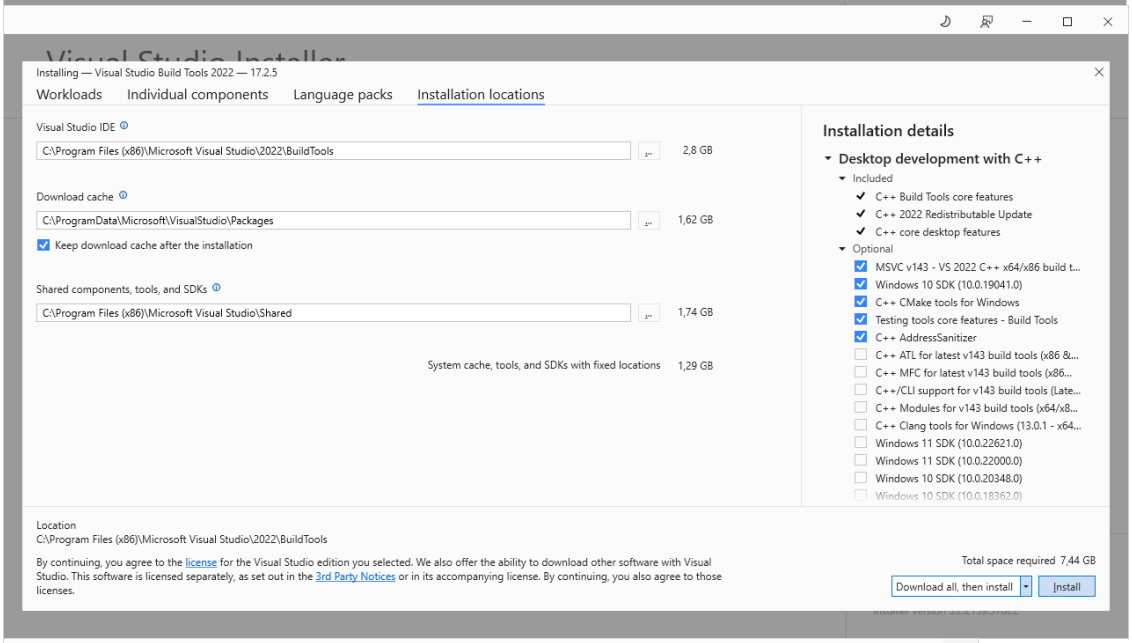


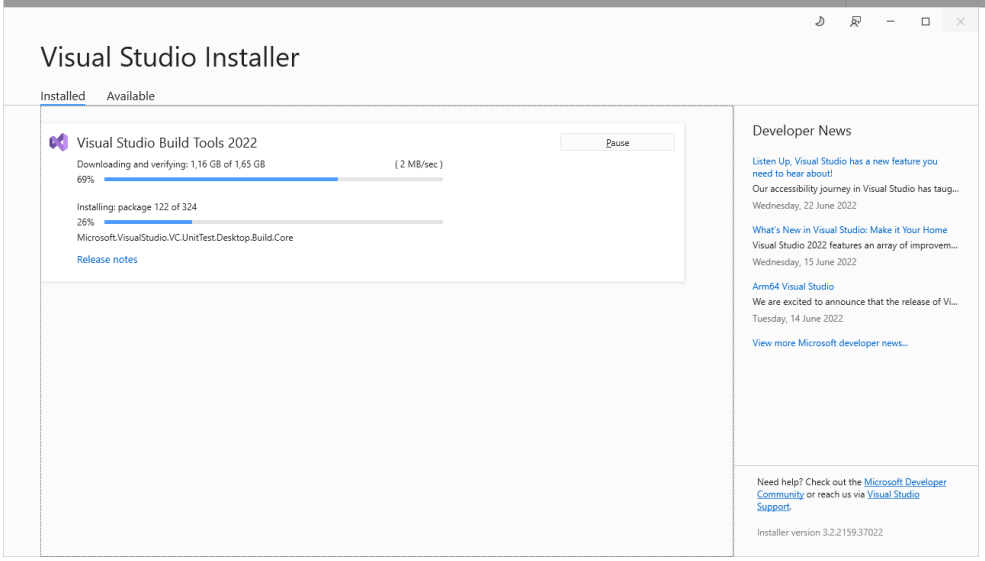


4. Install vs_BuildTools.exe from https://aka.ms/vs/17/release/vs_BuildTools.exe to the default folder - select C++ development tools. It was a 1.65 GB download.

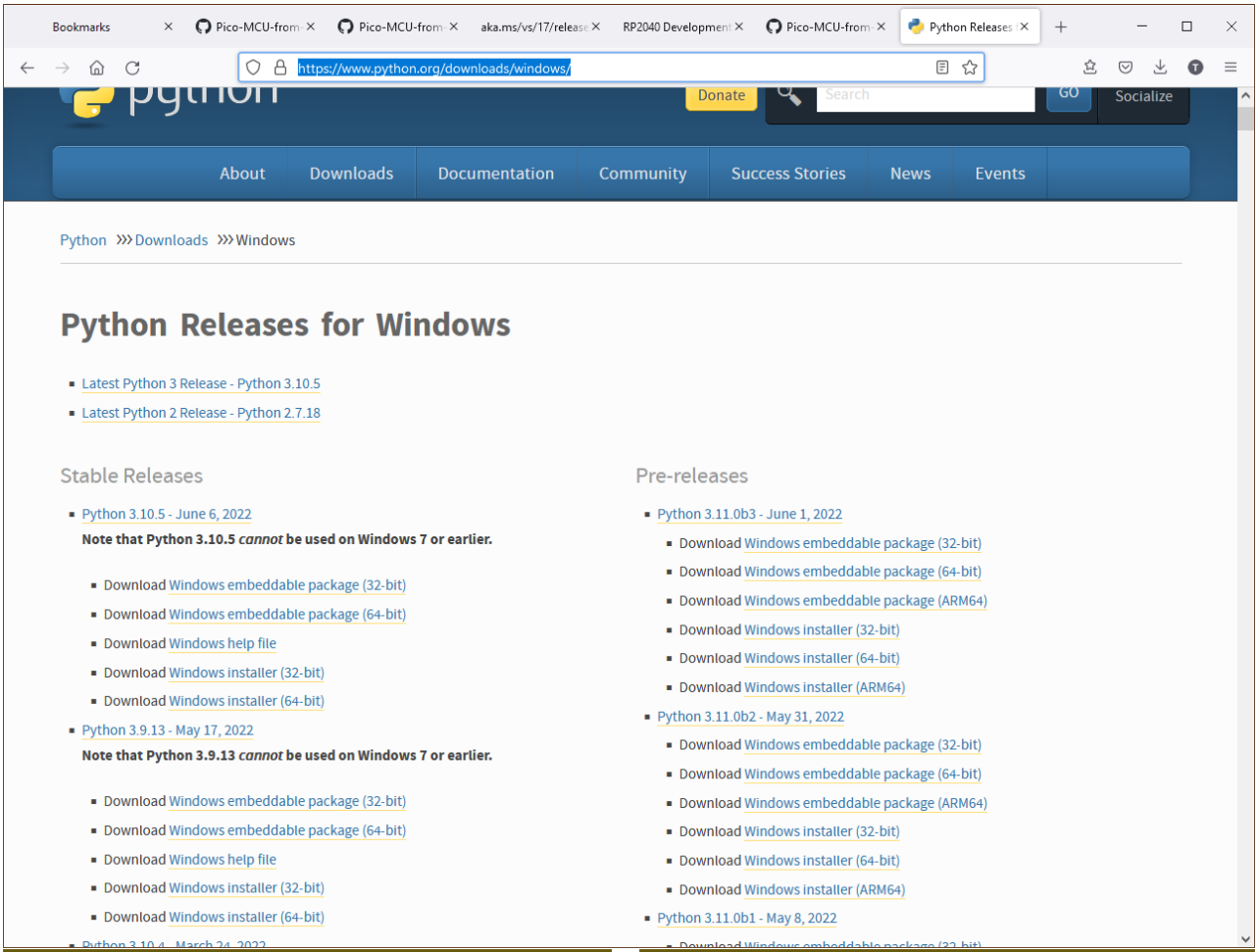


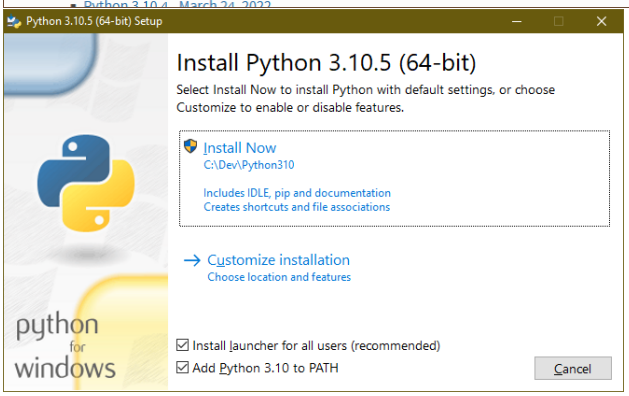


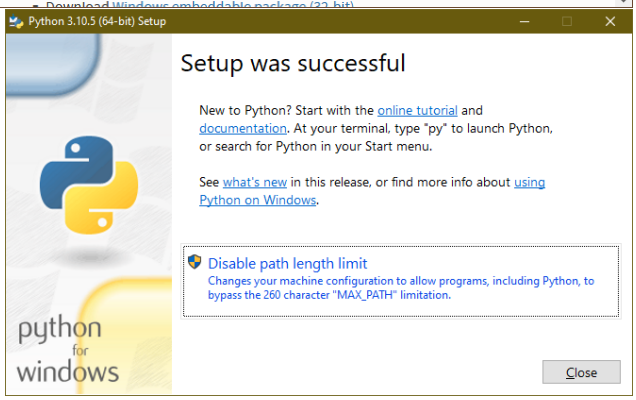




5. Install python-3.10.5-amd64.exe from https://www.python.org/downloads/windows/ to C:\Dev\Python310 - select Add Python to PATH and also select to remove the max path length.
https://www.python.org/ftp/python/3.10.5/python-3.10.5-amd64.exe

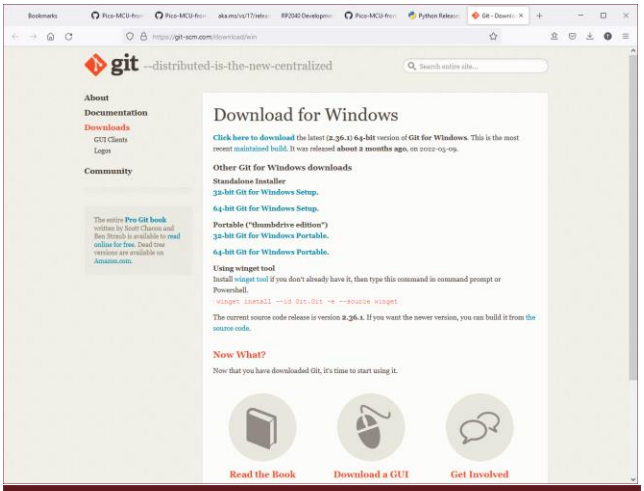


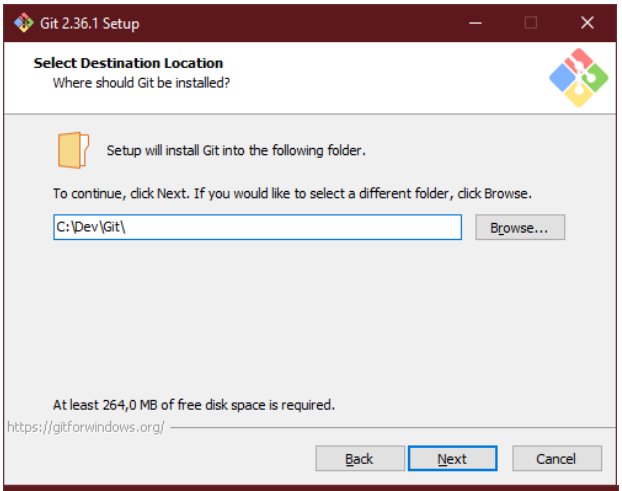


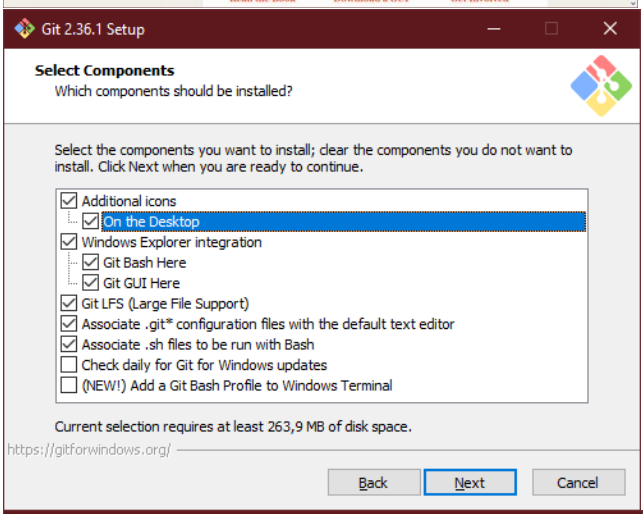


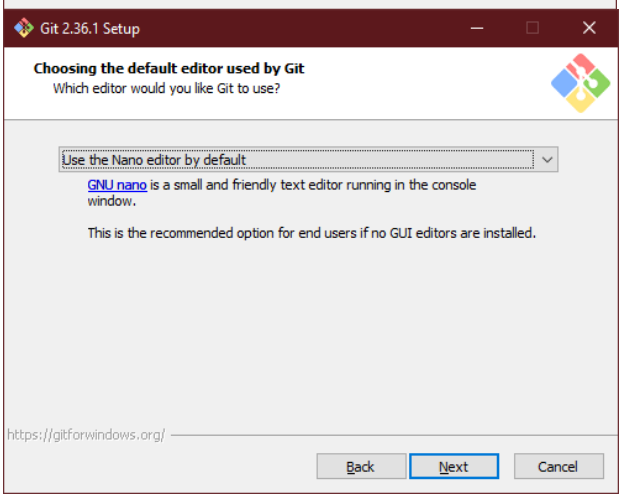
6. Install Git-2.36.1-64-bit.exe from https://git-scm.com/download/win to C:\Dev\Git - follow the instructions as below (from https://len42.github.io/rp2040-dev-setup.html).

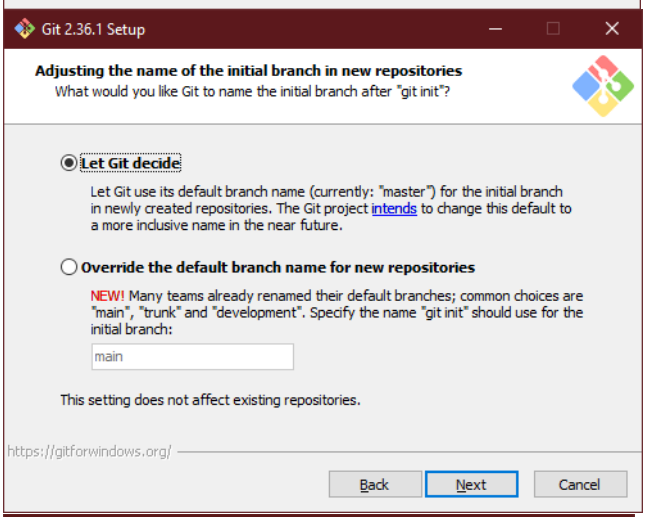
Destination Location: Default (or not)
Select Components: Default
Default editor: Select one you like.
Name of the initial branch: Let Git decide
PATH environment: Git from the command line and also from 3rd-party software
SSH executable: Use bundled OpenSSH
HTTPS transport backend: Use the OpenSSL library
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"

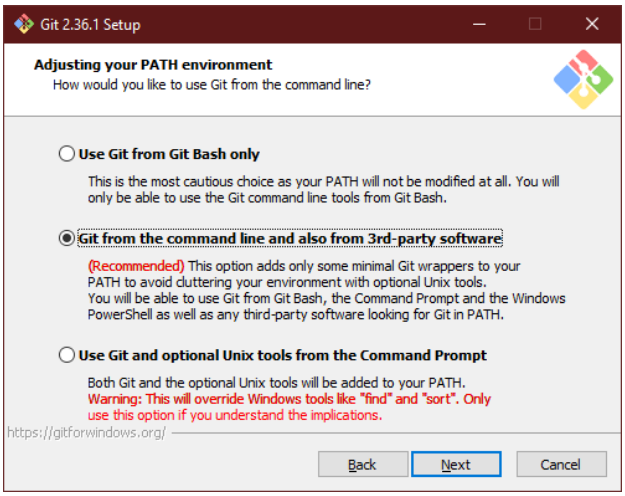

git --distributed-is-the-new-centralized
About
Documentation
Downloads
GUI Clients
Logos
Community
Download for Windows
Click here to download the latest (2.36.1) 64-bit version of Git for Windows. This is the most recent maintained build. It was released about 2 months ago, on 2022-05-09.
Other Git for Windows downloads
Standalone Installer
32-bit Git for Windows Setup.
64-bit Git for Windows Setup.
Portable ("thumbdrive edition")
32-bit Git for Windows Portable.
64-bit Git for Windows Portable.
Using winget tool
Now What?
Now that you have downloaded Git, it's time to start using it.
Read the Book
Download a GUI
Get Involved

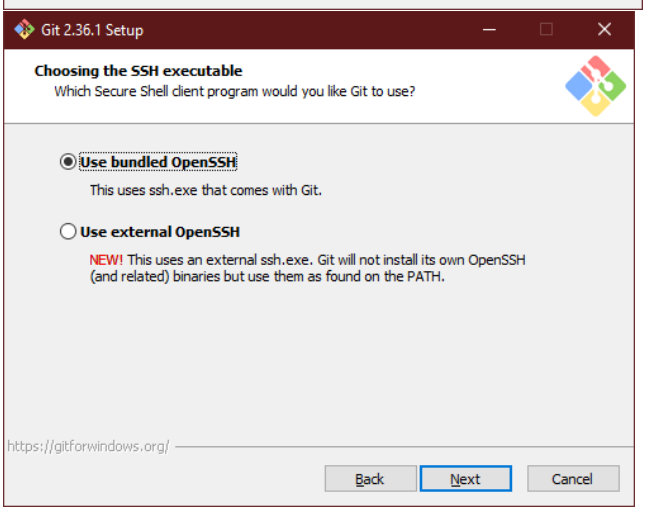

Git 2.36.1 Setup
Select Destination Location
Where should Git be installed?
Setup will install Git into the following folder.
To continue, click Next. If you would like to select a different folder, click Browse.
C:\Dev\Git\
Browse...
At least 264,0 MB of free disk space is required.
https://gitforwindows.org/
Back
Next
Cancel

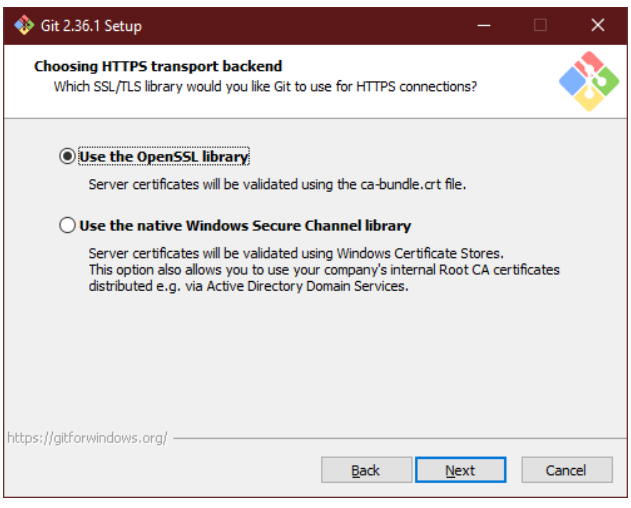

Git 2.36.1 Setup
Select Components
Which components should be installed?
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.
Additional icons
On the Desktop
Windows Explorer integration
Git Bash Here
Git GUI Here
Git LFS (Large File Support)
Associate .git* configuration files with the default text editor
Associate .sh files to be run with Bash
Check daily for Git for Windows updates
(NEW!) Add a Git Bash Profile to Windows Terminal
Current selection requires at least 263,9 MB of disk space.
https://gitforwindows.org/
Back
Next
Cancel

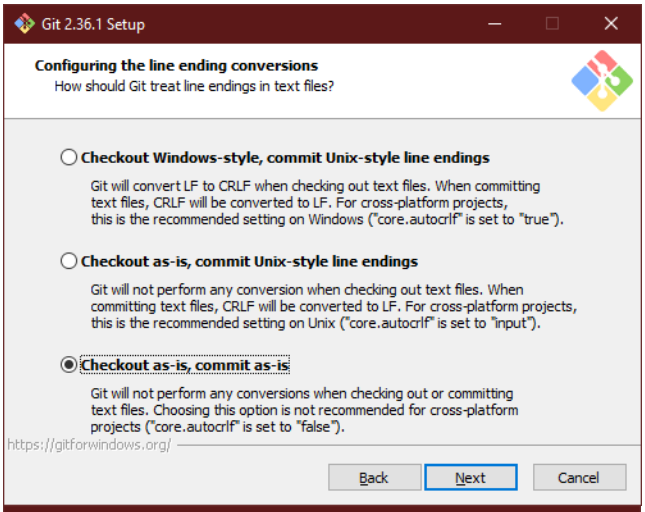

Git 2.36.1 Setup
Choosing the default editor used by Git
Which editor would you like Git to use?
Use the Nano editor by default
GNU nano is a small and friendly text editor running in the console window.
This is the recommended option for end users if no GUI editors are installed.
https://gitforwindows.org/
Back
Next
Cancel

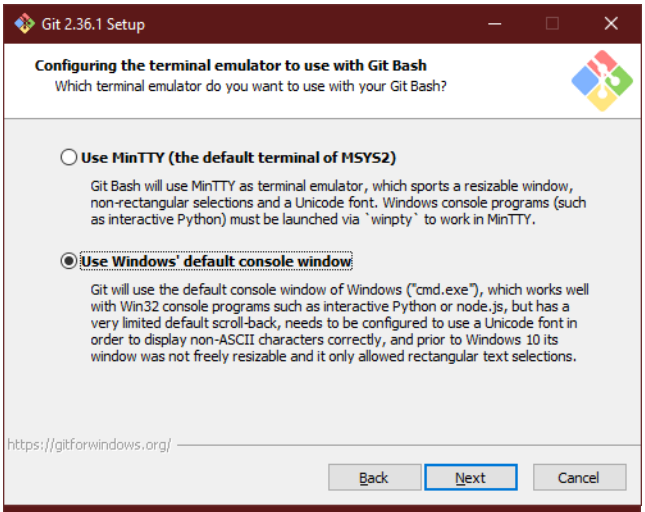

Git 2.36.1 Setup
Adjusting the name of the initial branch in new repositories
What would you like Git to name the initial branch after "git init"?
Let Git decide
Let Git use its default branch name (currently: "master") for the initial branch in newly created repositories. The Git project intends to change this default to a more inclusive name in the near future.
Override the default branch name for new repositories
NEW! Many teams already renamed their default branches; common choices are "main", "trunk" and "development". Specify the name "git init" should use for the initial branch:
main
This setting does not affect existing repositories.
https://gitforwindows.org/
Back
Next
Cancel

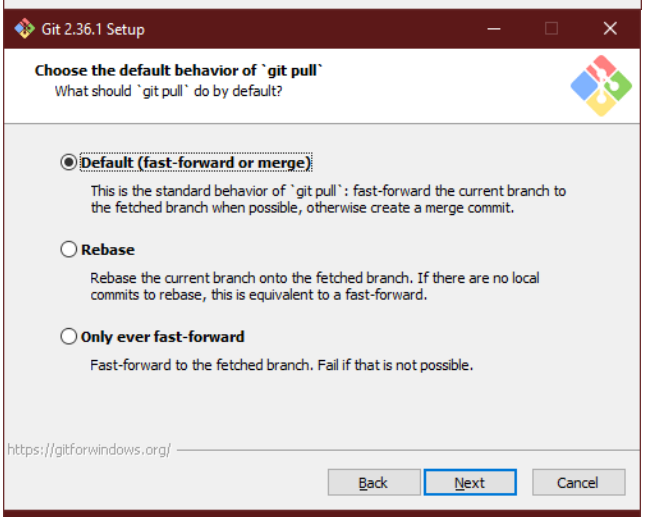

Git 2.36.1 Setup
Adjusting your PATH environment
How would you like to use Git from the command line?
Use Git from Git Bash only
This is the most cautious choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.
Git from the command line and also from 3rd-party software
(Recommended) This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools.
You will be able to use Git from Git Bash, the Command Prompt and the Windows PowerShell as well as any third-party software looking for Git in PATH.
Use Git and optional Unix tools from the Command Prompt
Both Git and the optional Unix tools will be added to your PATH.
Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.
https://gitforwindows.org/
Back
Next
Cancel

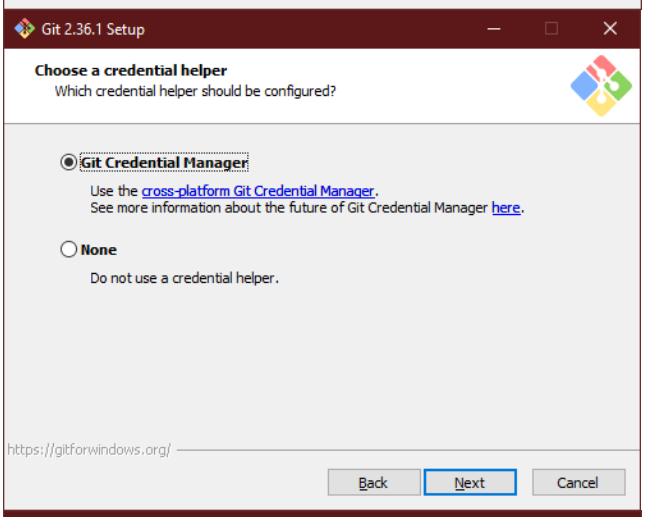

Git 2.36.1 Setup
Choosing the SSH executable
Which Secure Shell client program would you like Git to use?
Use bundled OpenSSH
This uses ssh.exe that comes with Git.
Use external OpenSSH
NEW! This uses an external ssh.exe. Git will not install its own OpenSSH (and related) binaries but use them as found on the PATH.
https://gitforwindows.org/
Back
Next
Cancel

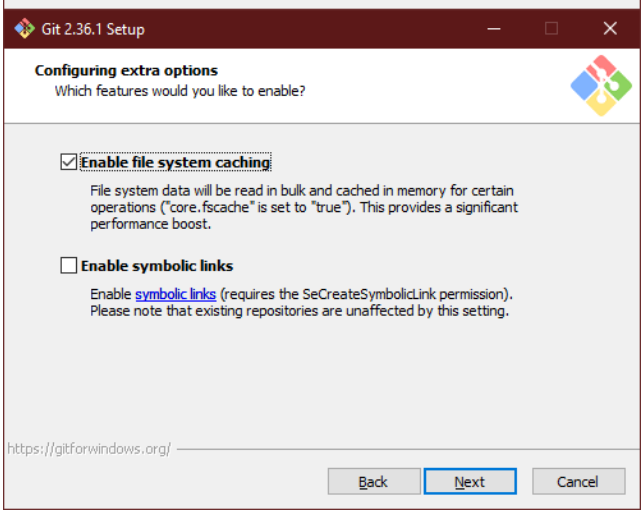

Git 2.36.1 Setup
Choosing HTTPS transport backend
Which SSL/TLS library would you like Git to use for HTTPS connections?
Use the OpenSSL library
Server certificates will be validated using the ca-bundle.crt file.
Use the native Windows Secure Channel library
Server certificates will be validated using 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

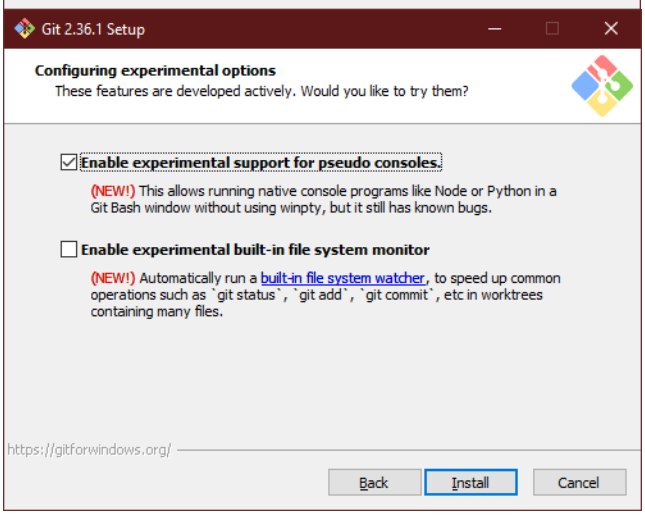












7. Use the windows admin cmd prompt to install the Pico SDK

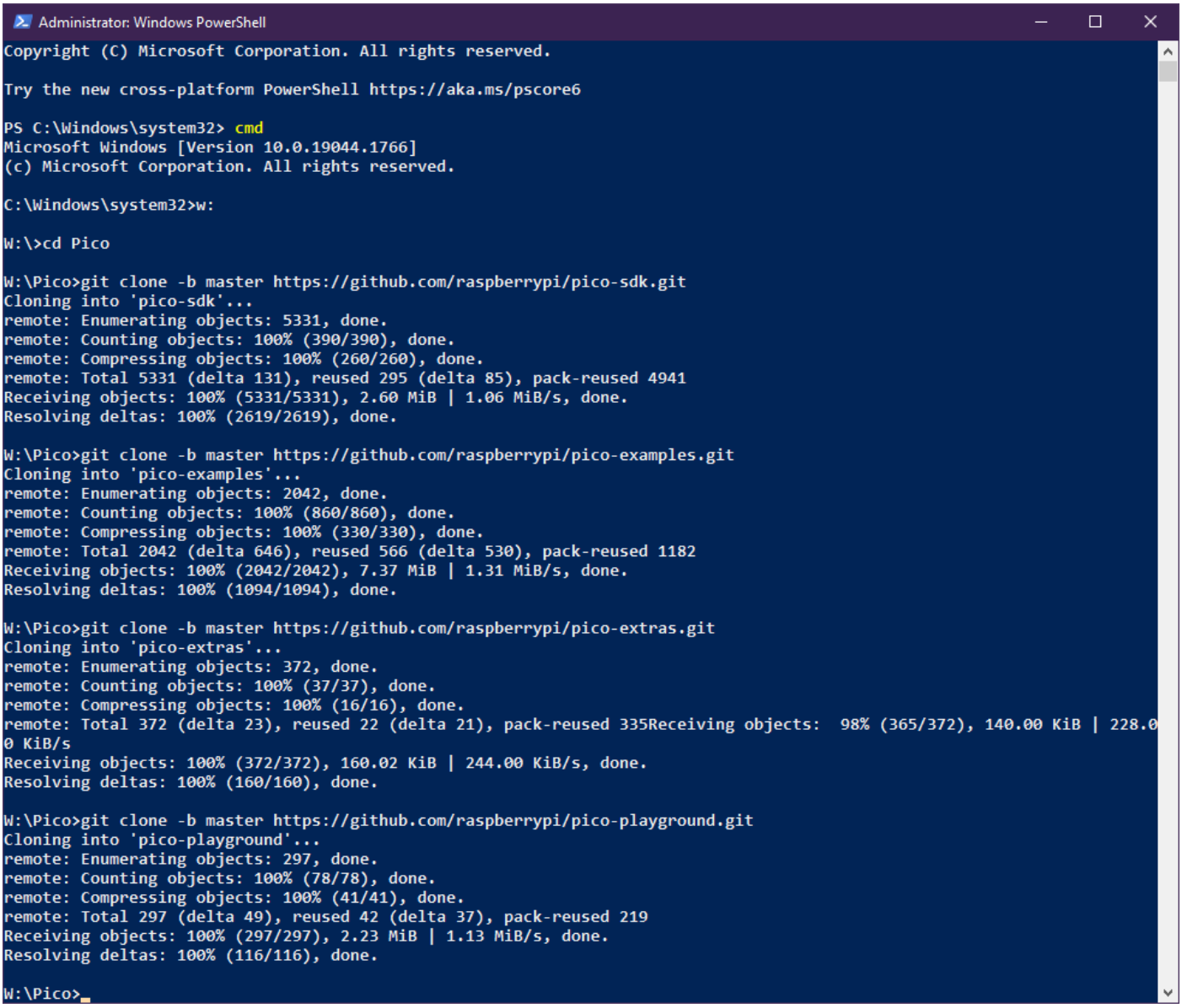
W:\Pico>git clone -b master https://github.com/raspberrypi/pico-sdk.git
Cloning into 'pico-sdk'...
remote: Enumerating objects: 5331, done.
remote: Counting objects: 100% (390/390), done.
remote: Compressing objects: 100% (260/260), done.
remote: Total 5331 (delta 131), reused 295 (delta 85), pack-reused 4941
Receiving objects: 100% (5331/5331), 2.60 MiB | 1.06 MiB/s, done.
Resolving deltas: 100% (2619/2619), done.

W:\Pico>git clone -b master https://github.com/raspberrypi/pico-examples.git
Cloning into 'pico-examples'...
remote: Enumerating objects: 2042, done.
remote: Counting objects: 100% (860/860), done.

remote: Compressing objects: 100% (330/330), done.
remote: Total 2042 (delta 646), reused 566 (delta 530), pack-reused 1182
Receiving objects: 100% (2042/2042), 7.37 MiB | 1.31 MiB/s, done.
Resolving deltas: 100% (1094/1094), done.

W:\Pico>git clone -b master https://github.com/raspberrypi/pico-extras.git
Cloning into 'pico-extras'...
remote: Enumerating objects: 372, done.
remote: Counting objects: 100% (37/37), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 372 (delta 23), reused 22 (delta 21), pack-reused 335Receiving objects: 98% (365/372), 140.00 KiB | 228.00 KiB/s
Receiving objects: 100% (372/372), 160.02 KiB | 244.00 KiB/s, done.
Resolving deltas: 100% (160/160), done.

W:\Pico>git clone -b master https://github.com/raspberrypi/pico-playground.git
Cloning into 'pico-playground'...
remote: Enumerating objects: 297, done.
remote: Counting objects: 100% (78/78), done.
remote: Compressing objects: 100% (41/41), done.
remote: Total 297 (delta 49), reused 42 (delta 37), pack-reused 219
Receiving objects: 100% (297/297), 2.23 MiB | 1.13 MiB/s, done.
Resolving deltas: 100% (116/116), done.

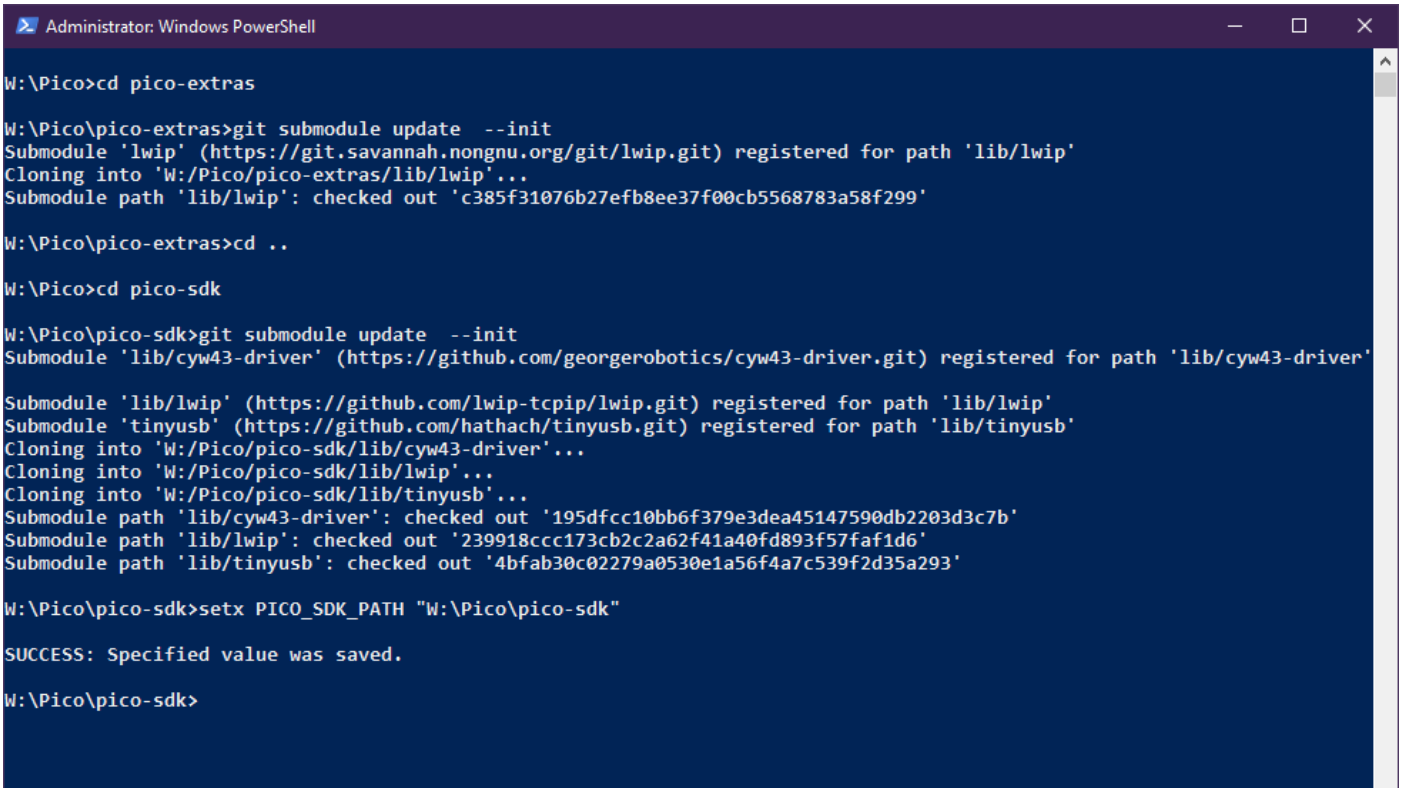


W:\Pico>cd pico-extras
W:\Pico\pico-extras>git submodule update --init
Submodule 'lwip' (https://git.savannah.nongnu.org/git/lwip.git) registered for path 'lib/lwip'
Cloning into 'W:/Pico/pico-extras/lib/lwip'...
Submodule path 'lib/lwip': checked out 'c385f31076b27efb8ee37f00cb5568783a58f299'

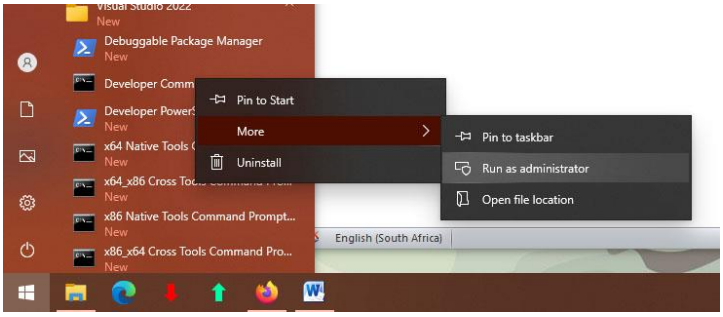
W:\Pico\pico-extras>cd ..
W:\Pico>cd pico-sdk

W:\Pico\pico-sdk>git submodule update --init
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 'tinyusb' (https://github.com/hathach/tinyusb.git) registered for path 'lib/tinyusb'
Cloning into 'W:/Pico/pico-sdk/lib/cyw43-driver'...
Cloning into 'W:/Pico/pico-sdk/lib/lwip'...
Cloning into 'W:/Pico/pico-sdk/lib/tinyusb'...
Submodule path 'lib/cyw43-driver': checked out '195dfcc10bb6f379e3dea45147590db2203d3c7b'
Submodule path 'lib/lwip': checked out '239918ccc173cb2c2a62f41a40fd893f57faf1d6'
Submodule path 'lib/tinyusb': checked out '4bfab30c02279a0530e1a56f4a7c539f2d35a293'

W:\Pico\pico-sdk>setx PICO_SDK_PATH "W:\Pico\pico-sdk"
SUCCESS: Specified value was saved.
W:\Pico\pico-sdk>



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



W:\Pico>cd pico-examples

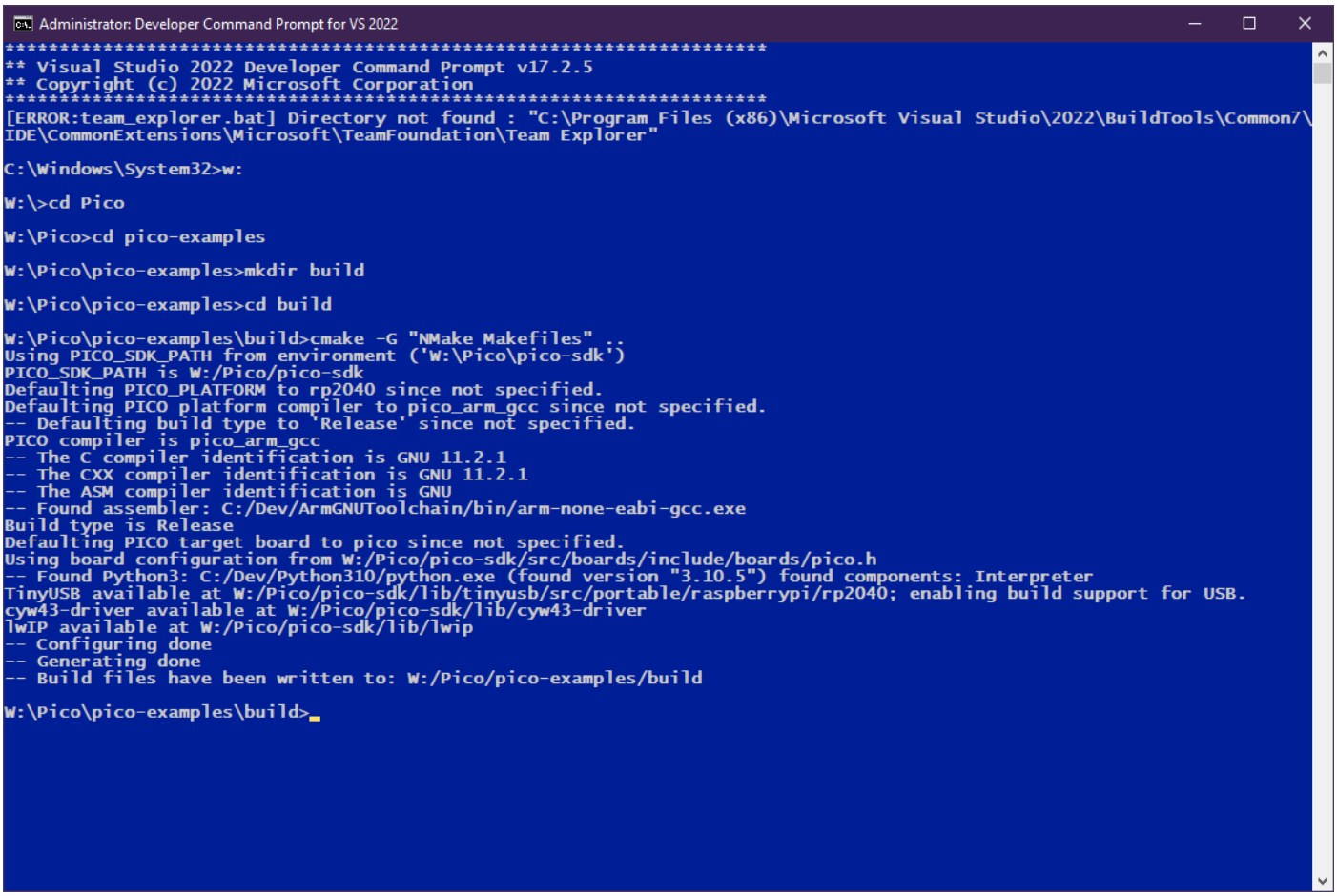
W:\Pico\pico-examples>mkdir build

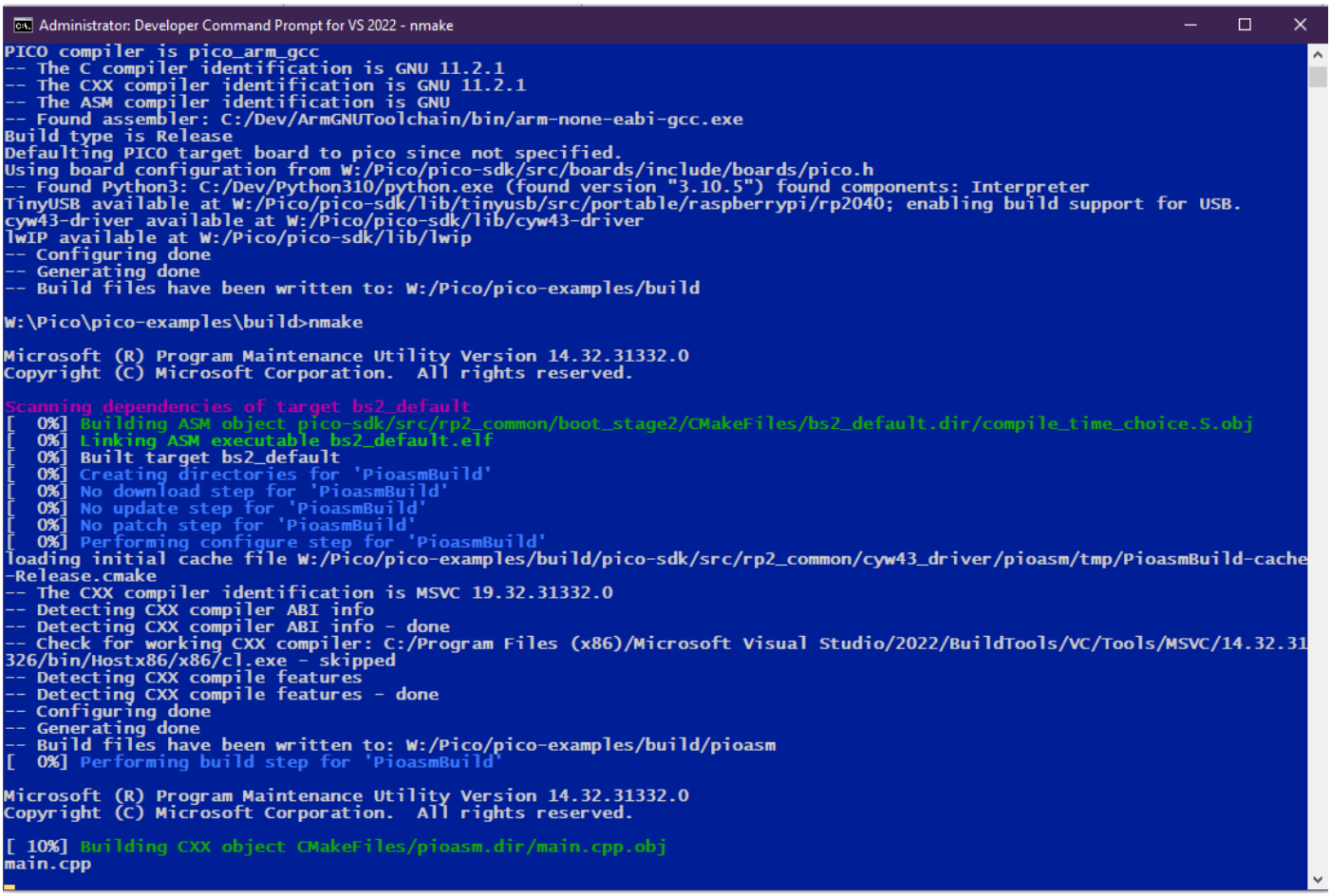
W:\Pico\pico-examples>cd build

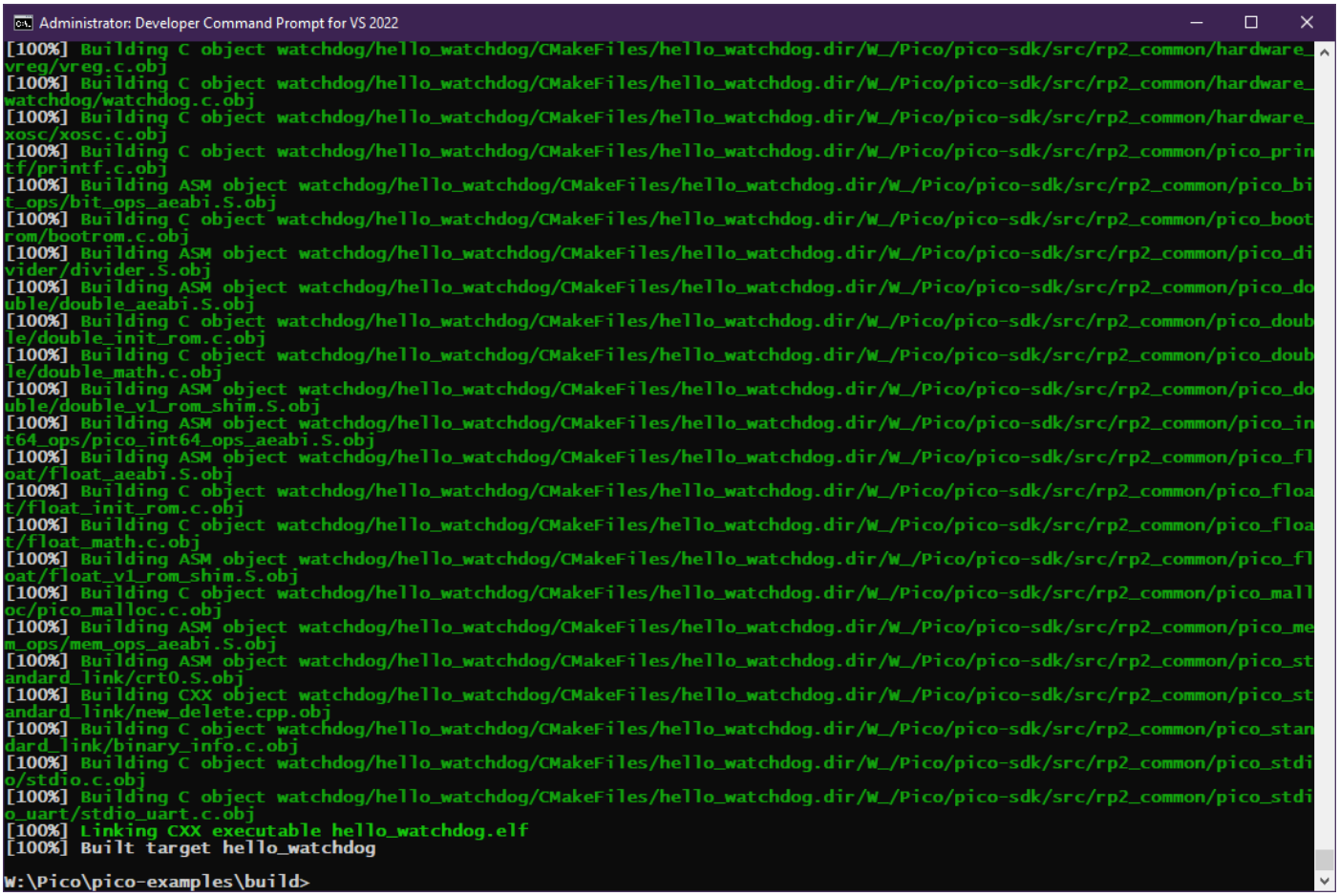
W:\Pico\pico-examples\build>cmake -G "NMake Makefiles" ..

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
-- The C compiler identification is GNU 11.2.1
-- The CXX compiler identification is GNU 11.2.1
-- The ASM compiler identification is GNU
-- Found assembler: C:/Dev/ArmGNUToolchain/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/Python310/python.exe (found version "3.10.5") found components: Interpreter
TinyUSB available at W:/Pico/pico-sdk/lib/tinyusb/src/portable/raspberrypi/rp2040; enabling build support for USB.
cyw43-driver available at W:/Pico/pico-sdk/lib/cyw43-driver
lwIP available at W:/Pico/pico-sdk/lib/lwip
-- Configuring done
-- Generating done
-- Build files have been written to: W:/Pico/pico-examples/build

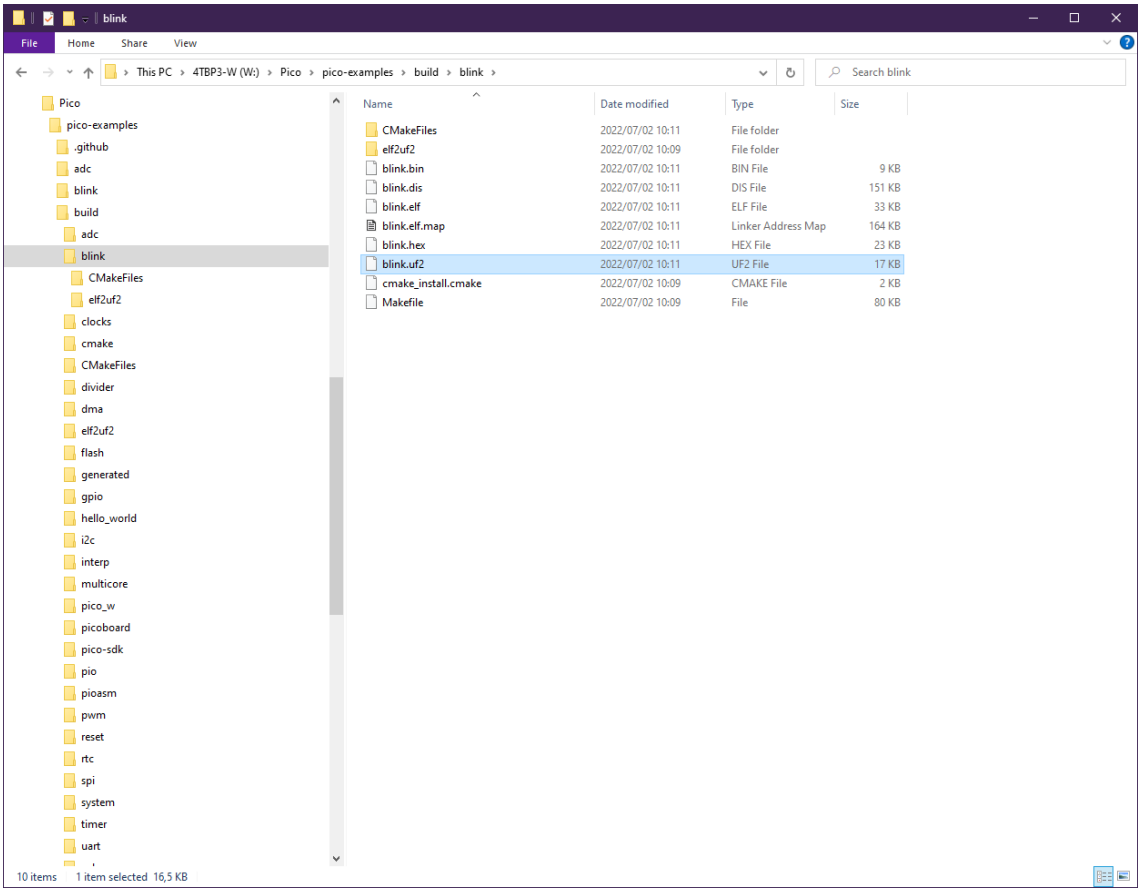
W:\Pico\pico-examples\build>nmake

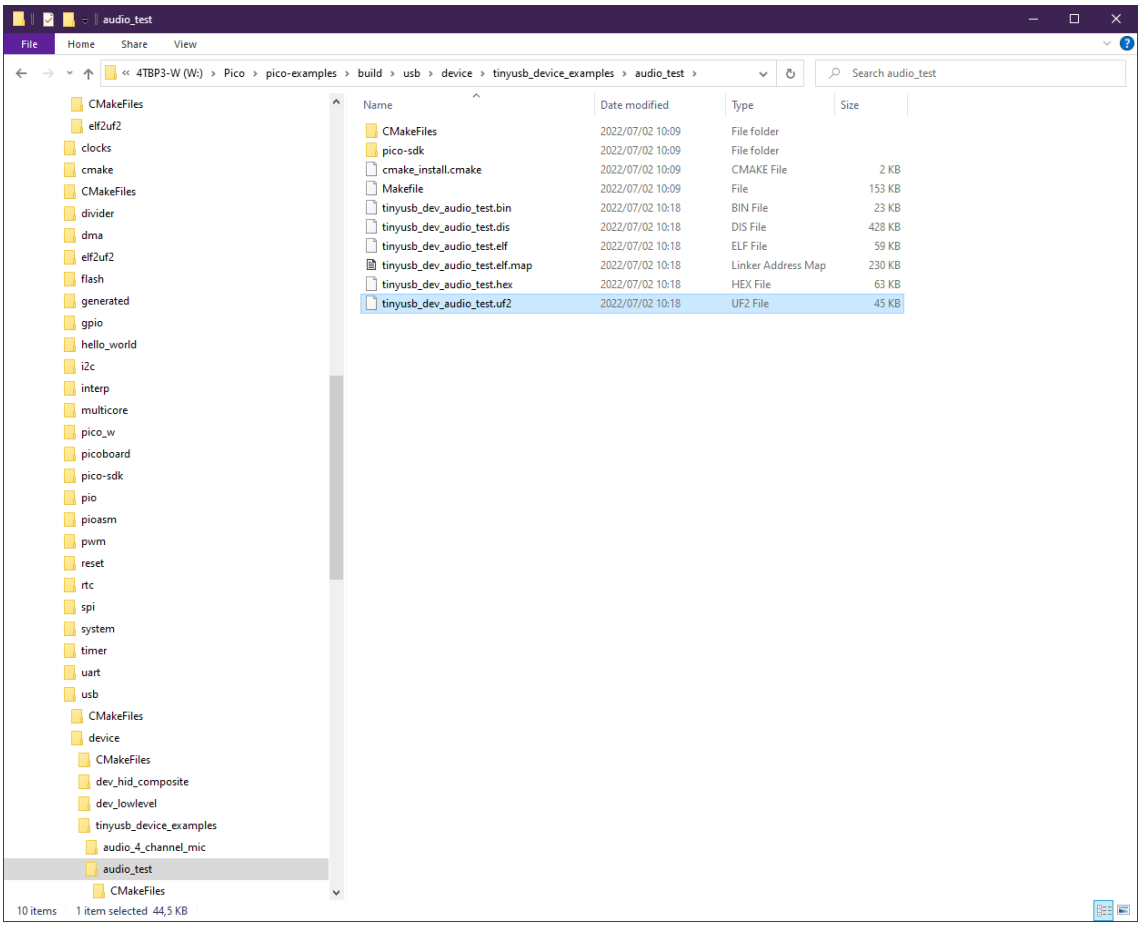






The uf2 files are in the pico-examples\build\sub-folders – including the tinyusb examples.

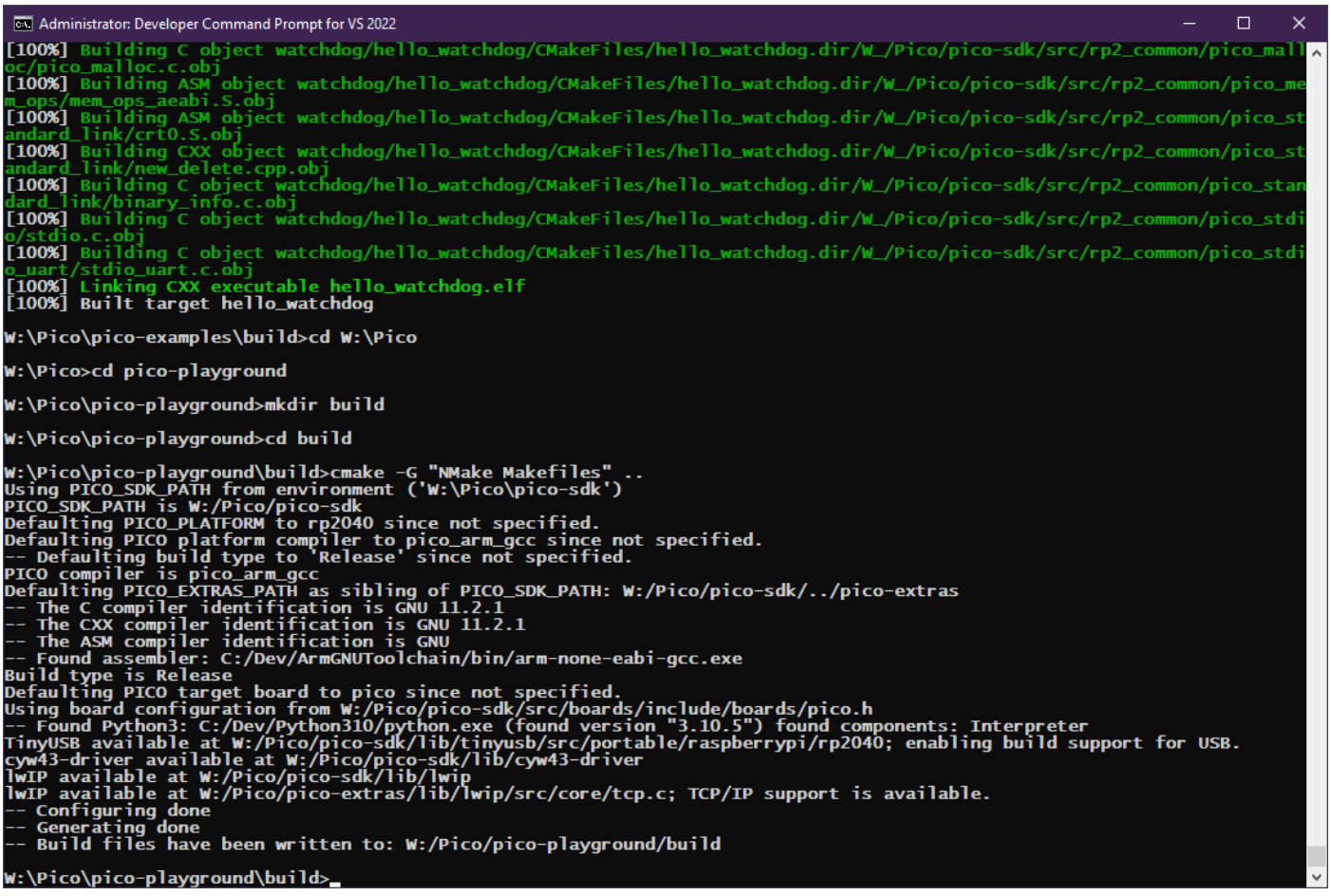

blink
File
Home
Share
View
This PC › 4TBP3-W (W:) › Pico › pico-examples › build › blink ›
Search blink
Pico
pico-examples
.github
adc
blink
build
adc
blink
CMakeFiles
elf2uf2
clocks
cmake
CMakeFiles
divider
dma
elf2uf2
flash
generated
gpio
hello_world
i2c
interp
multicore
pico_w
picoboard
pico-sdk
pio
pioasm
pwm
reset
rtc
spi
system
timer
uart
Name
Date modified
Type
Size
CMakeFiles 2022/07/02 10:11 File folder
elf2uf2 2022/07/02 10:09 File folder
blink.bin 2022/07/02 10:11 BIN File 9 KB
blink.dis 2022/07/02 10:11 DIS File 151 KB
blink.elf 2022/07/02 10:11 ELF File 33 KB
blink.elf.map 2022/07/02 10:11 Linker Address Map 164 KB
blink.hex 2022/07/02 10:11 HEX File 23 KB
blink.uf2 2022/07/02 10:11 UF2 File 17 KB
cmake_install.cmake 2022/07/02 10:09 CMAKE File 2 KB
Makefile 2022/07/02 10:09 File 80 KB
10 items
1 item selected 16,5 KB

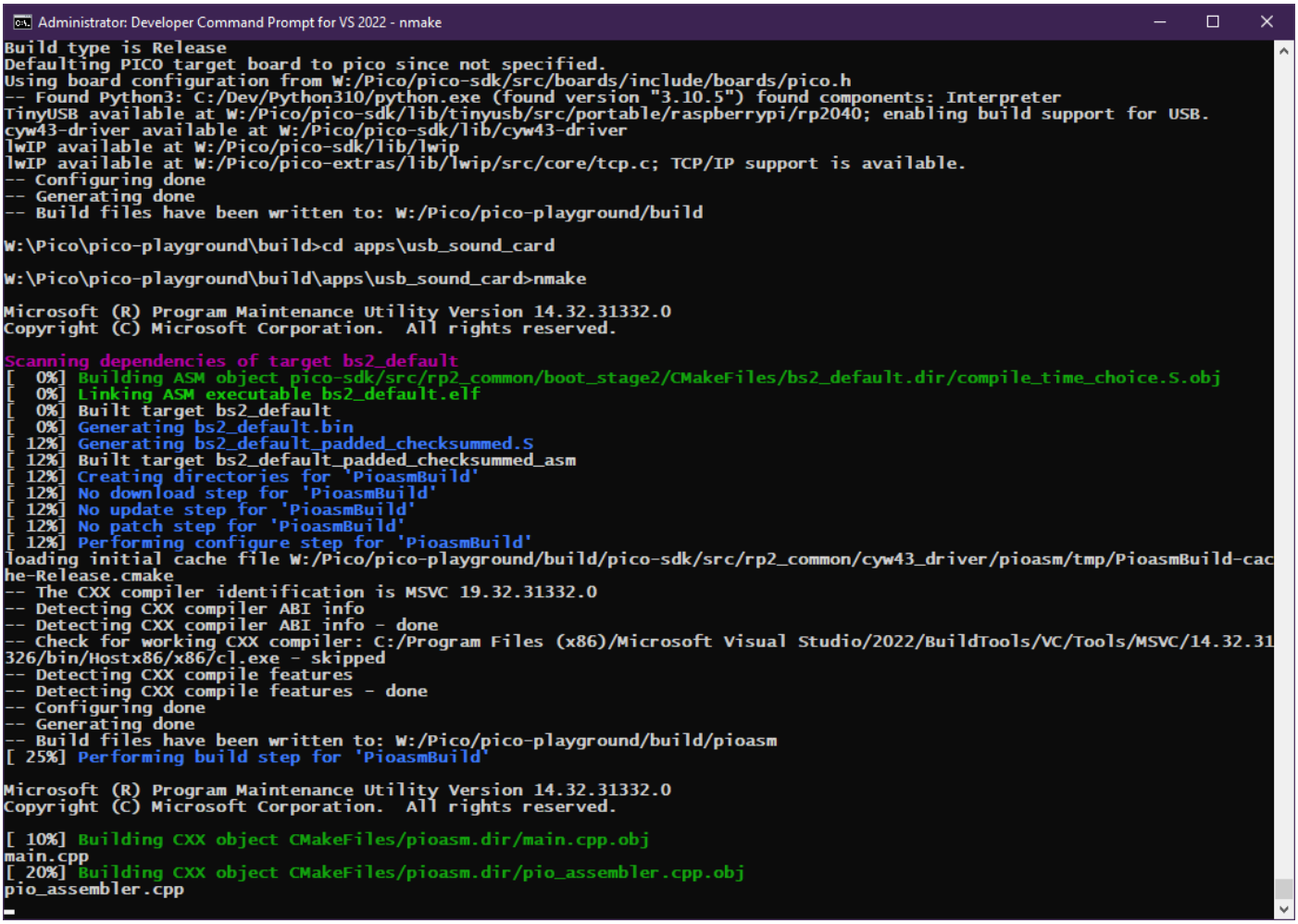

audio_test
File
Home
Share
View
« 4TBP3-W (W:) › Pico › pico-examples › build › usb › device › tinyusb_device_examples › audio_test ›
Search audio_test
CMakeFiles
elf2uf2
clocks
cmake
CMakeFiles
divider
dma
elf2uf2
flash
generated
gpio
hello_world
i2c
interp
multicore
pico_w
picoboard
pico-sdk
pio
pioasm
pwm
reset
rtc
spi
system
timer
uart
usb
CMakeFiles
device
CMakeFiles
dev_hid_composite
dev_lowlevel
tinyusb_device_examples
audio_4_channel_mic
audio_test
CMakeFiles
Name
Date modified
Type
Size
CMakeFiles 2022/07/02 10:09 File folder
pico-sdk 2022/07/02 10:09 File folder
cmake_install.cmake 2022/07/02 10:09 CMAKE File 2 KB
Makefile 2022/07/02 10:09 File 153 KB
tinyusb_dev_audio_test.bin 2022/07/02 10:18 BIN File 23 KB
tinyusb_dev_audio_test.dis 2022/07/02 10:18 DIS File 428 KB
tinyusb_dev_audio_test.elf 2022/07/02 10:18 ELF File 59 KB
tinyusb_dev_audio_test.elf.map 2022/07/02 10:18 Linker Address Map 230 KB
tinyusb_dev_audio_test.hex 2022/07/02 10:18 HEX File 63 KB
tinyusb_dev_audio_test.uf2 2022/07/02 10:18 UF2 File 45 KB
10 items
1 item selected 44,5 KB

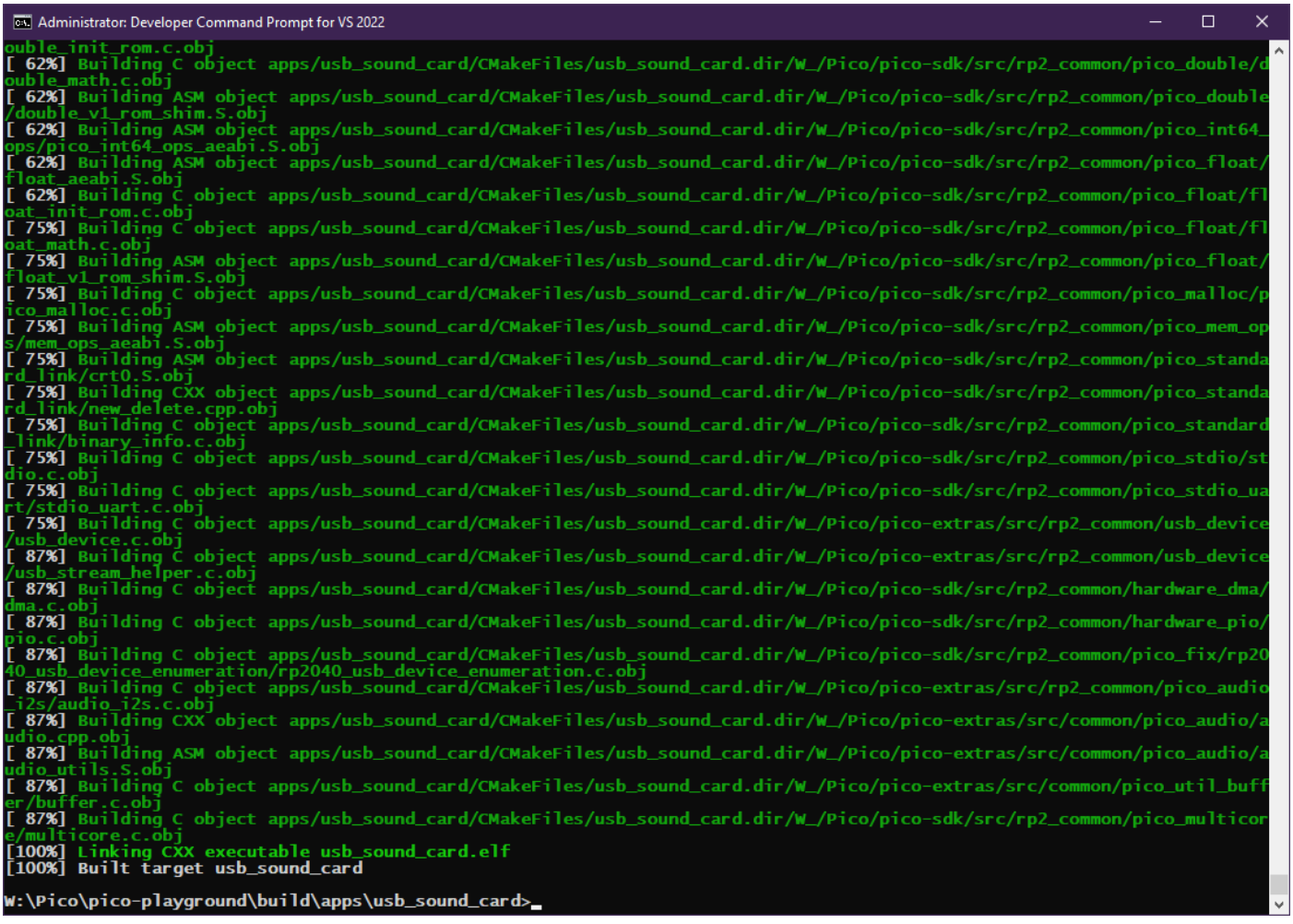
Building the usb_sound_card only:

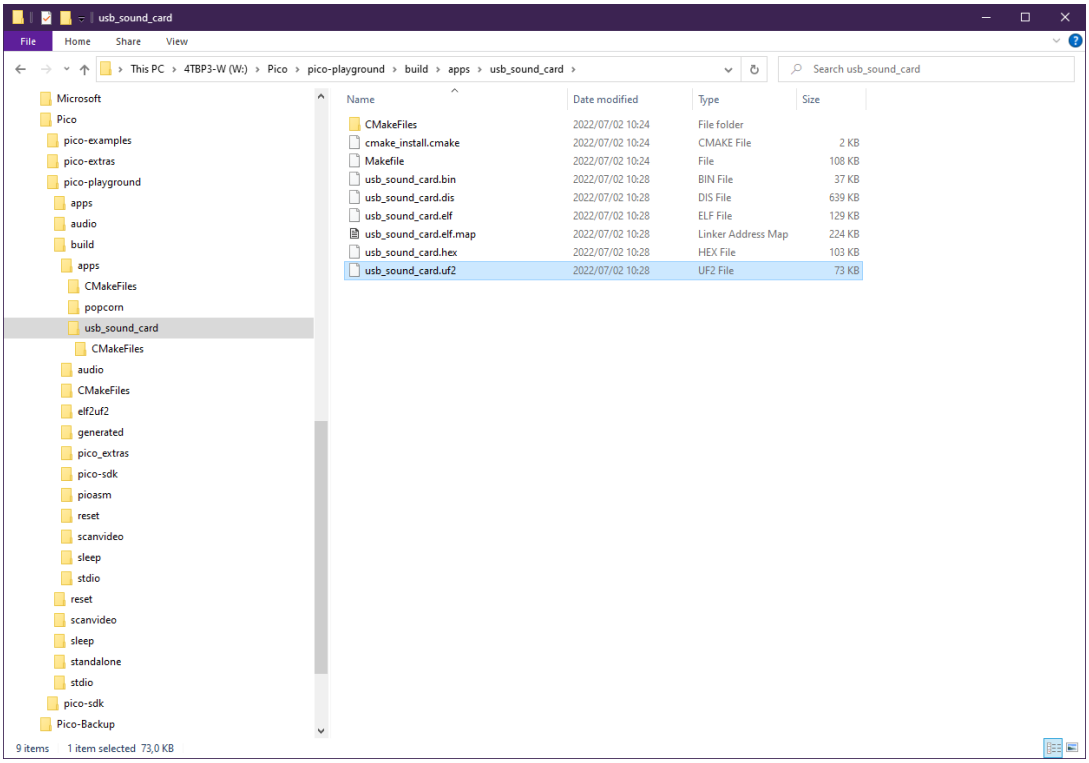
W:\Pico\pico-examples\build>cd W:\Pico
W:\Pico>cd pico-playground
W:\Pico\pico-playground>mkdir build
W:\Pico\pico-playground>cd build
W:\Pico\pico-playground\build>cmake -G "NMake Makefiles" ..
Using PICO_SDK_PATH from environment ('W:\Pico\pico-sdk')
PICO_SDK_PATH is W:/Pico/pico-sdk
Defaulting PICO_PLATFORM 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 11.2.1
-- The CXX compiler identification is GNU 11.2.1
-- The ASM compiler identification is GNU
-- Found assembler: C:/Dev/ArmGNUToolchain/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/Python310/python.exe (found version "3.10.5") found components: Interpreter
TinyUSB available at W:/Pico/pico-sdk/lib/tinyusb/src/portable/raspberrypi/rp2040; enabling build support for USB.
cyw43-driver available at W:/Pico/pico-sdk/lib/cyw43-driver
lwIP available at W:/Pico/pico-sdk/lib/lwip
lwIP available at W:/Pico/pico-extras/lib/lwip/src/core/tcp.c; TCP/IP support is available.
-- Configuring done
-- Generating done
-- Build files have been written to: 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




Administrator: Developer Command Prompt for VS 2022 - nmake
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/Python310/python.exe (found version "3.10.5") found components: Interpreter
TinyUSB available at W:/Pico/pico-sdk/lib/tinyusb/src/portable/raspberrypi/rp2040; enabling build support for USB.
cyw43-driver available at W:/Pico/pico-sdk/lib/cyw43-driver
lwIP available at W:/Pico/pico-sdk/lib/lwip
lwIP available at W:/Pico/pico-extras/lib/lwip/src/core/tcp.c; TCP/IP support is available.
-- Configuring done
-- Generating done
-- Build files have been written to: 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
Microsoft (R) Program Maintenance Utility Version 14.32.31332.0
Copyright (C) Microsoft Corporation. All rights reserved.
Scanning dependencies of target bs2_default
[0%] Building ASM object pico-sdk/src/rp2_common/boot_stage2/CMakeFiles/bs2_default.dir/compile_time_choice.S.obj
[0%] Linking ASM executable bs2_default.elf
[0%] Built target bs2_default
[0%] Generating bs2_default.bin
[12%] Generating bs2_default_padded_checksummed.S
[12%] Built target bs2_default_padded_checksummed_asm
[12%] Creating directories for 'PioasmBuild'
[12%] No download step for 'PioasmBuild'
[12%] No update step for 'PioasmBuild'
[12%] No patch step for 'PioasmBuild'
[12%] Performing configure step for 'PioasmBuild'
loading initial cache file W:/Pico/pico-playground/build/pico-sdk/src/rp2_common/cyw43_driver/pioasm/tmp/PioasmBuild-cache-Release.cmake
-- The CXX compiler identification is MSVC 19.32.31332.0
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: C:/Program Files (x86)/Microsoft Visual Studio/2022/BuildTools/VC/Tools/MSVC/14.32.31326/bin/Hostx86/x86/cl.exe - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: W:/Pico/pico-playground/build/pioasm
[25%] Performing build step for 'PioasmBuild'
Microsoft (R) Program Maintenance Utility Version 14.32.31332.0
Copyright (C) Microsoft Corporation. All rights reserved.
[10%] Building CXX object CMakeFiles/pioasm.dir/main.cpp.obj
main.cpp
[20%] Building CXX object CMakeFiles/pioasm.dir/pio_assembler.cpp.obj
pio_assembler.cpp


Administrator: Developer Command Prompt for VS 2022
ouble_init_rom.c.obj
[62%] Building C object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/W_/Pico/pico-sdk/src/rp2_common/pico_double/double_math.c.obj
[62%] Building ASM object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/W_/Pico/pico-sdk/src/rp2_common/pico_double/double_v1_rom_shim.S.obj
[62%] Building ASM object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/W_/Pico/pico-sdk/src/rp2_common/pico_int64_ops/pico_int64_ops_aeabi.S.obj
[62%] Building ASM object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/W_/Pico/pico-sdk/src/rp2_common/pico_float/float_aeabi.S.obj
[62%] Building C object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/W_/Pico/pico-sdk/src/rp2_common/pico_float/float_init_rom.c.obj
[75%] Building C object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/W_/Pico/pico-sdk/src/rp2_common/pico_float/float_math.c.obj
[75%] Building ASM object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/W_/Pico/pico-sdk/src/rp2_common/pico_float/float_v1_rom_shim.S.obj
[75%] Building C object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/W_/Pico/pico-sdk/src/rp2_common/pico_malloc/pico_malloc.c.obj
[75%] Building ASM object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/W_/Pico/pico-sdk/src/rp2_common/pico_mem_ops/mem_ops_aeabi.S.obj
[75%] Building ASM object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/W_/Pico/pico-sdk/src/rp2_common/pico_standard_link/crt0.S.obj
[75%] Building CXX object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/W_/Pico/pico-sdk/src/rp2_common/pico_standard_link/new_delete.cpp.obj
[75%] Building C object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/W_/Pico/pico-sdk/src/rp2_common/pico_standard_link/binary_info.c.obj
[75%] Building C object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/W_/Pico/pico-sdk/src/rp2_common/pico_stdio/stdio.c.obj
[75%] Building C object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/W_/Pico/pico-sdk/src/rp2_common/pico_stdio_uart/stdio_uart.c.obj
[75%] Building C object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/W_/Pico/pico-extras/src/rp2_common/usb_device/usb_device.c.obj
[87%] Building C object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/W_/Pico/pico-extras/src/rp2_common/usb_device/usb_stream_helper.c.obj
[87%] Building C object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/W_/Pico/pico-sdk/src/rp2_common/hardware_dma/dma.c.obj
[87%] Building C object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/W_/Pico/pico-sdk/src/rp2_common/hardware_pio/pio.c.obj
[87%] Building C object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/W_/Pico/pico-sdk/src/rp2_common/pico_fix/rp2040_usb_device_enumeration/rp2040_usb_device_enumeration.c.obj
[87%] Building C object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/W_/Pico/pico-extras/src/rp2_common/pico_audio_i2s/audio_i2s.c.obj
[87%] Building CXX object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/W_/Pico/pico-extras/src/common/pico_audio/audio.cpp.obj
[87%] Building ASM object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/W_/Pico/pico-extras/src/common/pico_audio/audio_utils.S.obj
[87%] Building C object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/W_/Pico/pico-extras/src/common/pico_util_buffer/buffer.c.obj
[87%] Building C object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/W_/Pico/pico-sdk/src/rp2_common/pico_multicore/multicore.c.obj
[100%] Linking CXX executable usb_sound_card.elf
[100%] Built target usb_sound_card
W:\Pico\pico-playground\build\apps\usb_sound_card>


usb_sound_card
File
Home
Share
View
This PC › 4TBP3-W (W:) › Pico › pico-playground › build › apps › usb_sound_card ›
Search usb_sound_card
Microsoft
Pico
pico-examples
pico-extras
pico-playground
apps
audio
build
apps
CMakeFiles
popcorn
usb_sound_card
CMakeFiles
audio
CMakeFiles
elf2uf2
generated
pico_extras
pico-sdk
pioasm
reset
scanvideo
sleep
stdio
reset
scanvideo
sleep
standalone
stdio
pico-sdk
Pico-Backup
Name
Date modified
Type
Size
CMakeFiles 2022/07/02 10:24 File folder
cmake_install.cmake 2022/07/02 10:24 CMAKE File 2 KB
Makefile 2022/07/02 10:24 File 108 KB
usb_sound_card.bin 2022/07/02 10:28 BIN File 37 KB
usb_sound_card.dis 2022/07/02 10:28 DIS File 639 KB
usb_sound_card.elf 2022/07/02 10:28 ELF File 129 KB
usb_sound_card.elf.map 2022/07/02 10:28 Linker Address Map 224 KB
usb_sound_card.hex 2022/07/02 10:28 HEX File 103 KB
usb_sound_card.uf2 2022/07/02 10:28 UF2 File 73 KB
9 items
1 item selected 73,0 KB