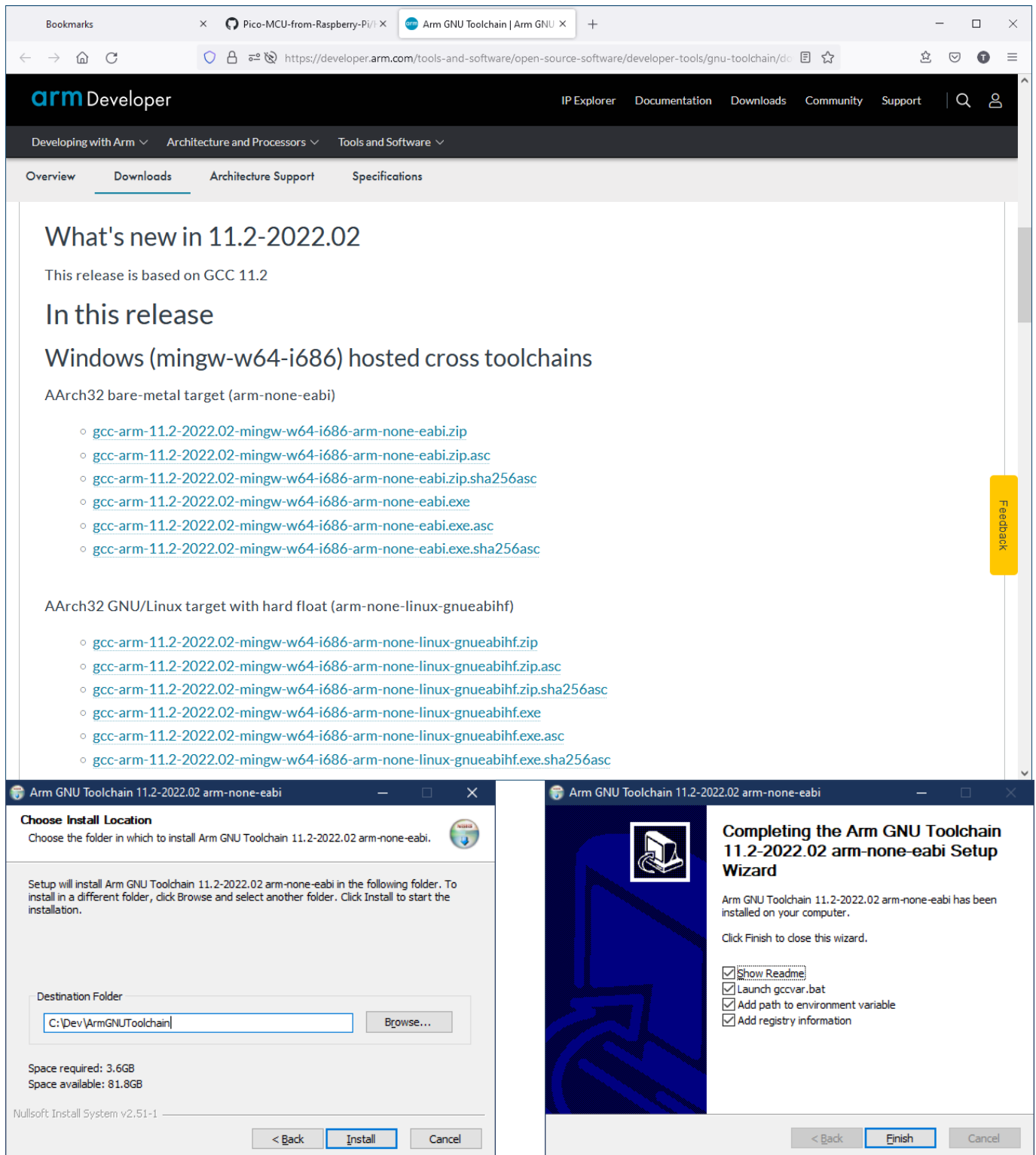


# Install Pico SDK in Windows 10x64 June 2022

Largely 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 C:\Pico).
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](https://github.com/Kitware/CMake/releases/download/v3.23.2/cmake-3.23.2-windows-x86_64.msi)

The screenshot shows the CMake website's download page in a web browser. The page has a navigation bar with links for About, Services, Resources, and Download. A table lists files for download, including a File Table v1, Cryptographic Hashes, and a PGP signature. Below this, there's a section for the Latest Release (3.23.2) with instructions on how to install it. Further down, there are sections for Source and Binary distributions, each with a table of available files. At the bottom, two Windows installer windows are shown: 'Install Options' and 'CMake Setup'. The 'Install Options' window has three radio buttons for adding CMake to the system PATH (all users, current user, or not at all) and a checked checkbox for 'Create CMake Desktop Icon'. The 'CMake Setup' window shows the 'Destination Folder' as 'C:\Dev\CMake\' with a 'Change...' button.

Role	Files
File Table v1	cmake-3.24.0-rc2-files-v1.json
Cryptographic Hashes	cmake-3.24.0-rc2-SHA-256.txt
PGP sig by 2D2CEF1034921684	cmake-3.24.0-rc2-SHA-256.txt.asc

Also see instructions on [Download Verification](#).

### Latest Release (3.23.2)

The release was packaged with CPack which is included as part of the release. The .sh files are self extracting gzipped tar files. To install a .sh file, run it with /bin/sh and follow the directions. The OS-machine.tar.gz files are gzipped tar files of the install tree. The OS-machine.tar.Z files are compressed tar files of the install tree. The tar file distributions can be untared in any directory. They are prefixed by the version of CMake. For example, the linux-x86\_64 tar file is all under the directory cmake-linux-x86\_64. This prefix can be removed as long as the share, bin, man and doc directories are moved relative to each other. To build the source distributions, unpack them with zip or tar and follow the instructions in README.rst at the top of the source tree. See also the [CMake 3.23 Release Notes](#).

Source distributions:

Platform	Files
Unix/Linux Source (has \n line feeds)	cmake-3.23.2.tar.gz
Windows Source (has \r\n line feeds)	cmake-3.23.2.zip

Binary distributions:

Platform	Files
Windows x64 Installer	cmake-3.23.2-windows-x86_64.msi
Windows x64 ZIP	cmake-3.23.2-windows-x86_64.zip
Windows i386 Installer	cmake-3.23.2-windows-i386.msi
Windows i386 ZIP	cmake-3.23.2-windows-i386.zip
macOS 10.13 or later	cmake-3.23.2-macos-universal.dmg

#### Install Options

Choose options for installing CMake 3.23.2

By default CMake does not add its directory to the system PATH.

☐ Do not add CMake to the system PATH

☒ Add CMake to the system PATH for all users

☐ Add CMake to the system PATH for the current user

☒ Create CMake Desktop Icon

Back Next Cancel

#### CMake Setup

##### Destination Folder

Click Next to install to the default folder or click Change to choose another.

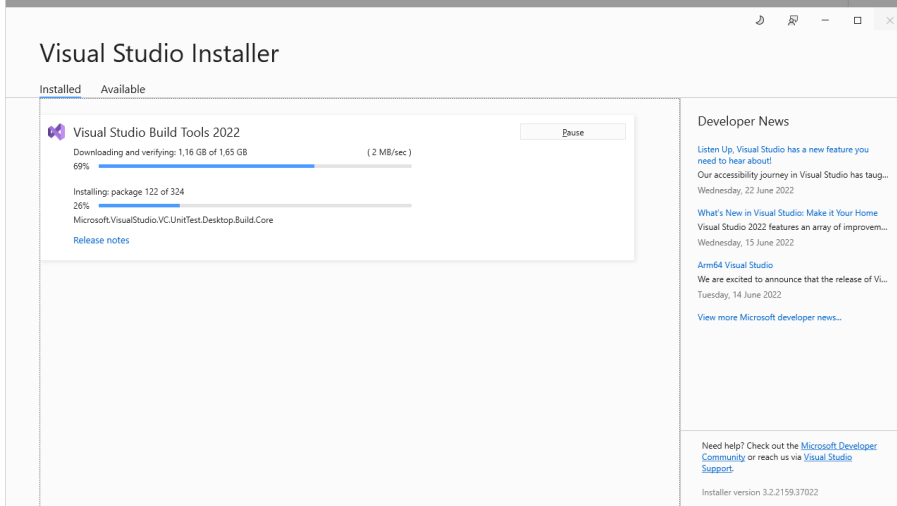
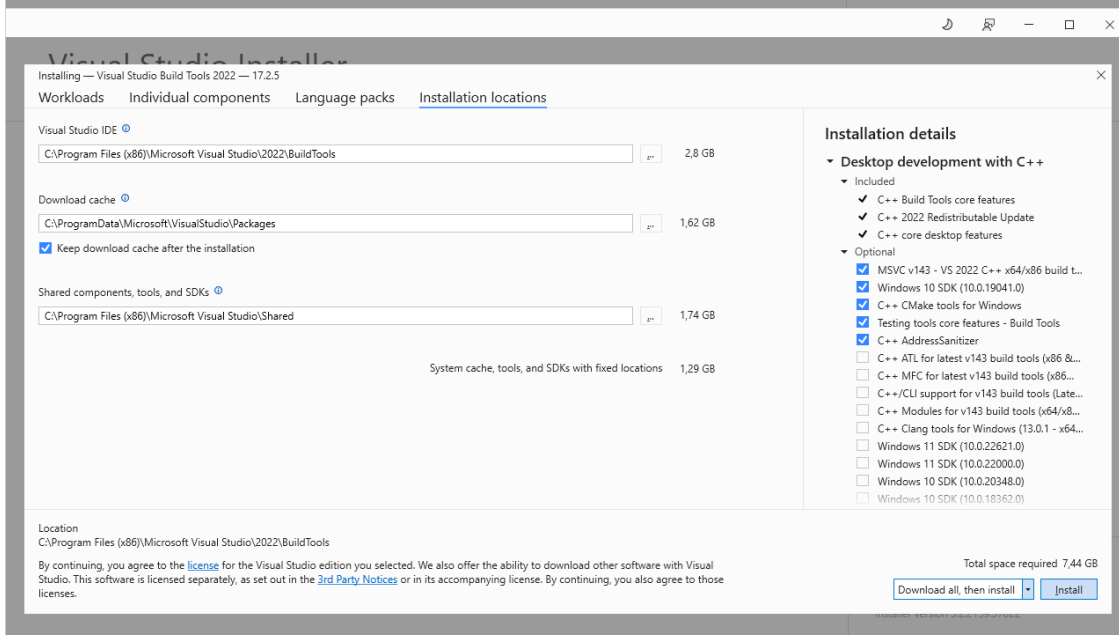
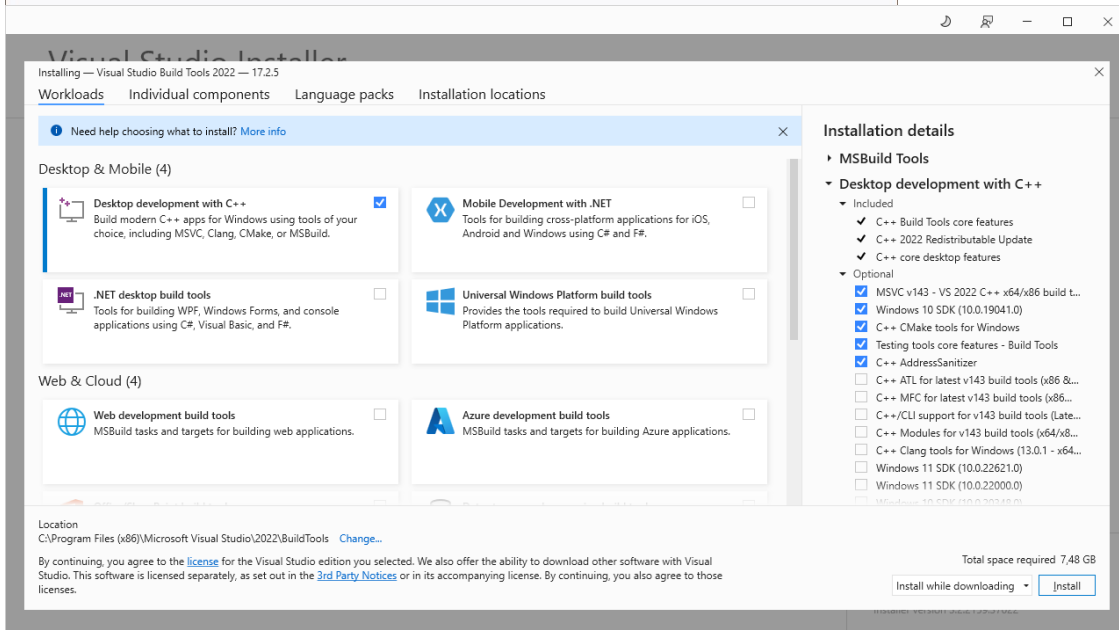
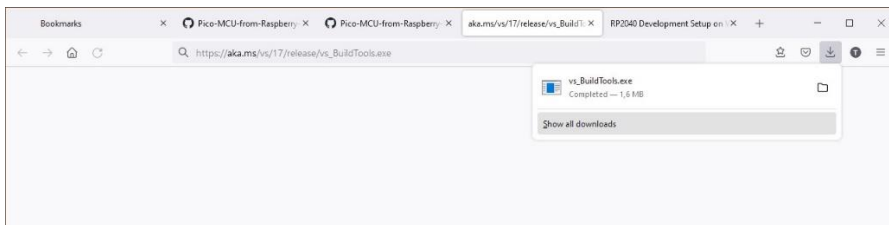
Install CMake to:

C:\Dev\CMake\

Change...

Back Next Cancel

4. Install vs\_BuildTools.exe from [https://aka.ms/vs/17/release/vs\\_BuildTools.exe](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>

The image shows a browser window displaying the 'Python Releases for Windows' page. The page lists stable releases (Python 3.10.5 and 3.9.13) and pre-releases (Python 3.11.0b3, 3.11.0b2, and 3.11.0b1). For each release, it provides download links for Windows embeddable packages (32-bit, 64-bit, ARM64) and Windows installers (32-bit, 64-bit, ARM64). Below the browser window, two screenshots of the Python 3.10.5 (64-bit) Setup window are shown. The first screenshot shows the 'Install Now' button, which includes IDLE, pip, and documentation, and creates shortcuts and file associations. It also shows the 'Customize installation' option. The second screenshot shows the 'Setup was successful' message, indicating that the installation is complete and the path length limit has been disabled.

Python Releases for Windows

- Latest Python 3 Release - Python 3.10.5
- Latest Python 2 Release - Python 2.7.18

Stable Releases

- Python 3.10.5 - June 6, 2022  
**Note that Python 3.10.5 cannot be used on Windows 7 or earlier.**
  - Download Windows embeddable package (32-bit)
  - Download Windows embeddable package (64-bit)
  - Download Windows help file
  - Download Windows installer (32-bit)
  - Download Windows installer (64-bit)
- Python 3.9.13 - May 17, 2022  
**Note that Python 3.9.13 cannot be used on Windows 7 or earlier.**
  - Download Windows embeddable package (32-bit)
  - Download Windows embeddable package (64-bit)
  - Download Windows help file
  - Download Windows installer (32-bit)
  - Download Windows installer (64-bit)

Pre-releases

- Python 3.11.0b3 - June 1, 2022
  - Download Windows embeddable package (32-bit)
  - Download Windows embeddable package (64-bit)
  - Download Windows embeddable package (ARM64)
  - Download Windows installer (32-bit)
  - Download Windows installer (64-bit)
  - Download Windows installer (ARM64)
- Python 3.11.0b2 - May 31, 2022
  - Download Windows embeddable package (32-bit)
  - Download Windows embeddable package (64-bit)
  - Download Windows embeddable package (ARM64)
  - Download Windows installer (32-bit)
  - Download Windows installer (64-bit)
  - Download Windows installer (ARM64)
- Python 3.11.0b1 - May 8, 2022
  - Download Windows embeddable package (32-bit)
  - Download Windows embeddable package (64-bit)
  - Download Windows embeddable package (ARM64)
  - Download Windows installer (32-bit)
  - Download Windows installer (64-bit)
  - Download Windows installer (ARM64)

Python 3.10.5 (64-bit) Setup

Install Python 3.10.5 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

**Install Now**  
C:\Dev\Python310

Includes IDLE, pip and documentation  
Creates shortcuts and file associations

→ **Customize installation**  
Choose location and features

☒ Install launcher for all users (recommended)  
☒ Add Python 3.10 to PATH

Cancel

Setup was successful

New to Python? Start with the [online tutorial](#) and [documentation](#). At your terminal, type "py" to launch Python, or search for Python in your Start menu.

See [what's new](#) in this release, or find more info about [using Python on Windows](#).

**Disable path length limit**  
Changes your machine configuration to allow programs, including Python, to bypass the 260 character "MAX\_PATH" limitation.

Close

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.

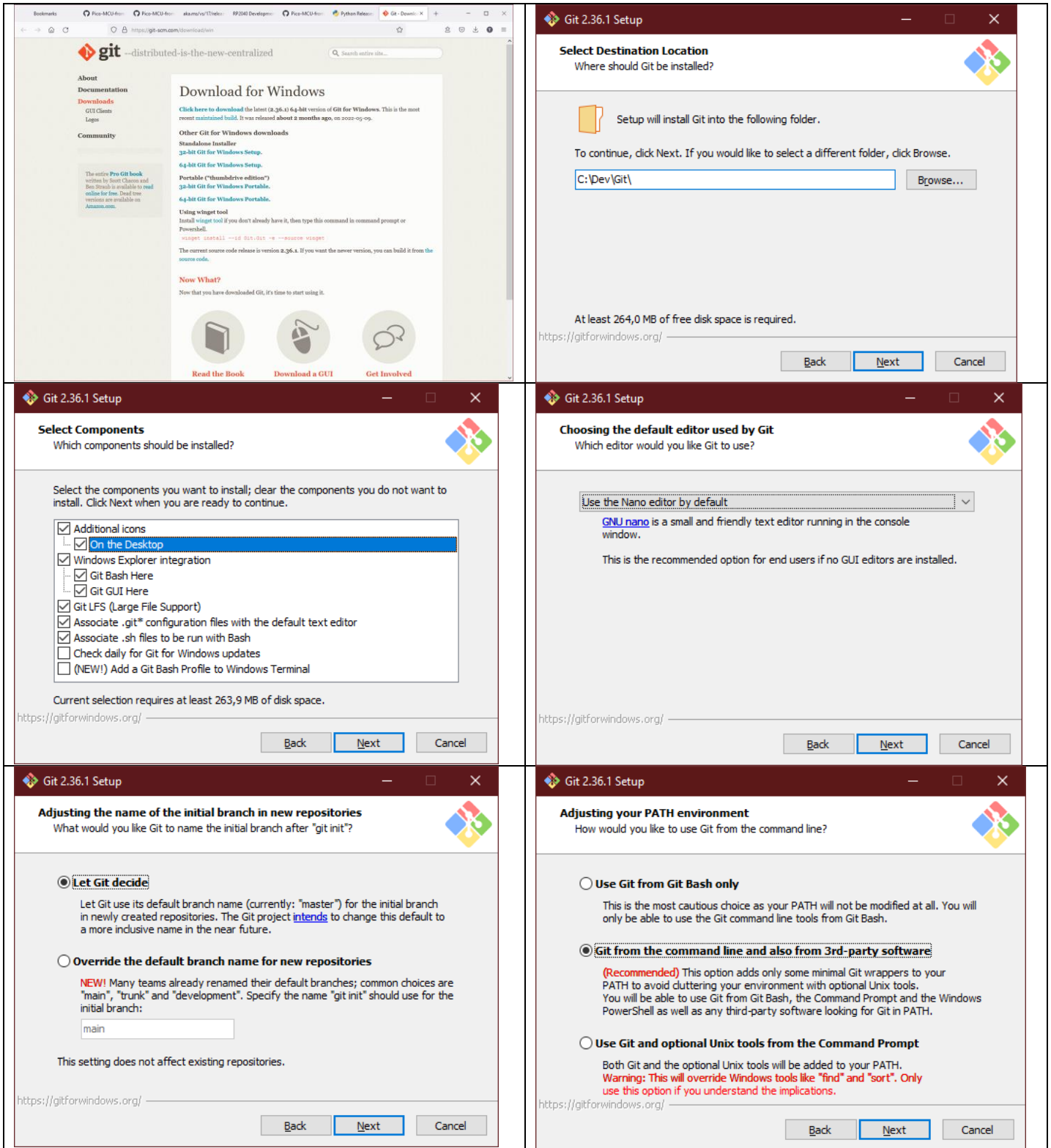
Name of the initial branch: Let Git decide

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"



<div><div>Git 2.36.1 Setup</div><div>Choosing the SSH executable</div><div>Which Secure Shell client program would you like Git to use?</div><div><div><input checked="" type="radio"/> Use bundled OpenSSH</div><div>This uses ssh.exe that comes with Git.</div><div><input type="radio"/> Use external OpenSSH</div><div><b>NEW!</b> This uses an external ssh.exe. Git will not install its own OpenSSH (and related) binaries but use them as found on the PATH.</div></div><div><div>https://gitforwindows.org/</div><div>BackNextCancel</div></div></div>	<div><div>Git 2.36.1 Setup</div><div>Choosing HTTPS transport backend</div><div>Which SSL/TLS library would you like Git to use for HTTPS connections?</div><div><div><input checked="" type="radio"/> Use the OpenSSL library</div><div>Server certificates will be validated using the ca-bundle.crt file.</div><div><input type="radio"/> Use the native Windows Secure Channel library</div><div>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.</div></div><div><div>https://gitforwindows.org/</div><div>BackNextCancel</div></div></div>
<div><div>Git 2.36.1 Setup</div><div>Configuring the line ending conversions</div><div>How should Git treat line endings in text files?</div><div><div><input type="radio"/> Checkout Windows-style, commit Unix-style line endings</div><div>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").</div><div><input type="radio"/> Checkout as-is, commit Unix-style line endings</div><div>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").</div><div><input checked="" type="radio"/> Checkout as-is, commit as-is</div><div>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").</div></div><div><div>https://gitforwindows.org/</div><div>BackNextCancel</div></div></div>	<div><div>Git 2.36.1 Setup</div><div>Configuring the terminal emulator to use with Git Bash</div><div>Which terminal emulator do you want to use with your Git Bash?</div><div><div><input type="radio"/> Use MinTTY (the default terminal of MSYS2)</div><div>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.</div><div><input checked="" type="radio"/> Use Windows' default console window</div><div>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.</div></div><div><div>https://gitforwindows.org/</div><div>BackNextCancel</div></div></div>
<div><div>Git 2.36.1 Setup</div><div>Choose the default behavior of 'git pull'</div><div>What should 'git pull' do by default?</div><div><div><input checked="" type="radio"/> Default (fast-forward or merge)</div><div>This is the standard behavior of 'git pull': fast-forward the current branch to the fetched branch when possible, otherwise create a merge commit.</div><div><input type="radio"/> Rebase</div><div>Rebase the current branch onto the fetched branch. If there are no local commits to rebase, this is equivalent to a fast-forward.</div><div><input type="radio"/> Only ever fast-forward</div><div>Fast-forward to the fetched branch. Fail if that is not possible.</div></div><div><div>https://gitforwindows.org/</div><div>BackNextCancel</div></div></div>	<div><div>Git 2.36.1 Setup</div><div>Choose a credential helper</div><div>Which credential helper should be configured?</div><div><div><input checked="" type="radio"/> Git Credential Manager</div><div>Use the <a href="#">cross-platform Git Credential Manager</a>. See more information about the future of Git Credential Manager <a href="#">here</a>.</div><div><input type="radio"/> None</div><div>Do not use a credential helper.</div></div><div><div>https://gitforwindows.org/</div><div>BackNextCancel</div></div></div>
<div><div>Git 2.36.1 Setup</div><div>Configuring extra options</div><div>Which features would you like to enable?</div><div><div><input checked="" type="checkbox"/> Enable file system caching</div><div>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.</div><div><input type="checkbox"/> Enable symbolic links</div><div>Enable <a href="#">symbolic links</a> (requires the SeCreateSymbolicLink permission). Please note that existing repositories are unaffected by this setting.</div></div><div><div>https://gitforwindows.org/</div><div>BackNextCancel</div></div></div>	<div><div>Git 2.36.1 Setup</div><div>Configuring experimental options</div><div>These features are developed actively. Would you like to try them?</div><div><div><input checked="" type="checkbox"/> Enable experimental support for pseudo consoles</div><div><b>(NEW!)</b> This allows running native console programs like Node or Python in a Git Bash window without using winpty, but it still has known bugs.</div><div><input type="checkbox"/> Enable experimental built-in file system monitor</div><div><b>(NEW!)</b> Automatically run a <a href="#">built-in file system watcher</a>, to speed up common operations such as 'git status', 'git add', 'git commit', etc in worktrees containing many files.</div></div><div><div>https://gitforwindows.org/</div><div>BackInstallCancel</div></div></div>



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

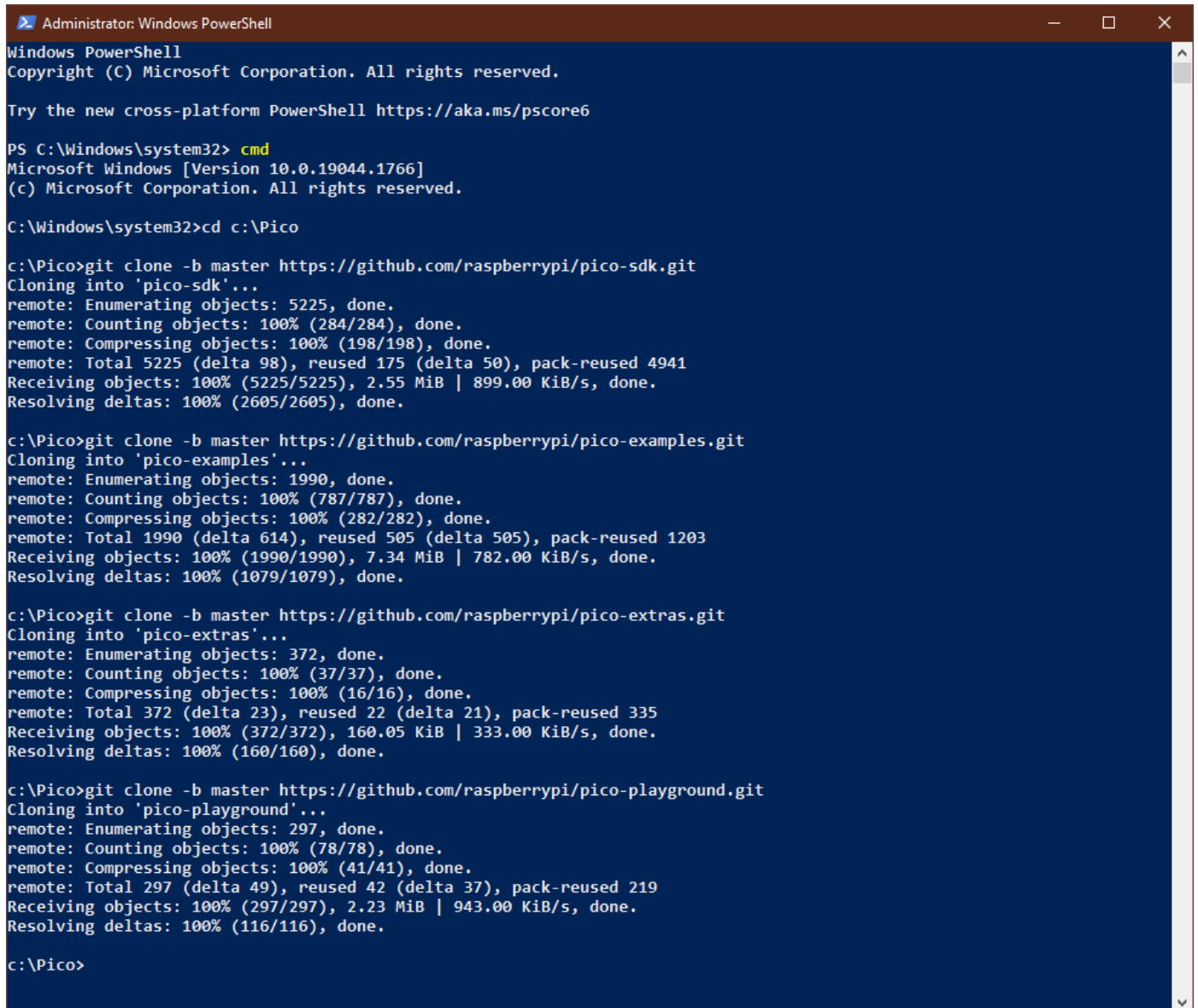
```
cd c:\Pico
```

```
c:\Pico>git clone -b master https://github.com/raspberrypi/pico-sdk.git
```

```
c:\Pico>git clone -b master https://github.com/raspberrypi/pico-examples.git
```

```
c:\Pico>git clone -b master https://github.com/raspberrypi/pico-extras.git
```

```
c:\Pico>git clone -b master https://github.com/raspberrypi/pico-playground.git
```

A screenshot of a Windows PowerShell terminal window titled "Administrator: Windows PowerShell". The terminal shows the execution of several git clone commands to install the Pico SDK and its submodules. The first command clones the pico-sdk repository. The second clones the pico-examples repository. The third clones the pico-extras repository. The fourth clones the pico-playground repository. Each command shows the progress of cloning, including enumerating, counting, and compressing objects, and receiving deltas. The terminal output is as follows:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> cmd
Microsoft Windows [Version 10.0.19044.1766]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd c:\Pico

c:\Pico>git clone -b master https://github.com/raspberrypi/pico-sdk.git
Cloning into 'pico-sdk'...
remote: Enumerating objects: 5225, done.
remote: Counting objects: 100% (284/284), done.
remote: Compressing objects: 100% (198/198), done.
remote: Total 5225 (delta 98), reused 175 (delta 50), pack-reused 4941
Receiving objects: 100% (5225/5225), 2.55 MiB | 899.00 KiB/s, done.
Resolving deltas: 100% (2605/2605), done.

c:\Pico>git clone -b master https://github.com/raspberrypi/pico-examples.git
Cloning into 'pico-examples'...
remote: Enumerating objects: 1990, done.
remote: Counting objects: 100% (787/787), done.
remote: Compressing objects: 100% (282/282), done.
remote: Total 1990 (delta 614), reused 505 (delta 505), pack-reused 1203
Receiving objects: 100% (1990/1990), 7.34 MiB | 782.00 KiB/s, done.
Resolving deltas: 100% (1079/1079), done.

c:\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 335
Receiving objects: 100% (372/372), 160.05 KiB | 333.00 KiB/s, done.
Resolving deltas: 100% (160/160), done.

c:\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 | 943.00 KiB/s, done.
Resolving deltas: 100% (116/116), done.

c:\Pico>
```

```
c:\Pico>cd pico-extras
```

```
c:\Pico\pico-extras>git submodule update - -init
```

```
c:\Pico\pico-extras>cd ..
```

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

```
c:\Pico\pico-sdk>git submodule update - -init
```

8. Then run a build of all the pico-examples:

```
c:\Pico\setx PICO_SDK_PATH "C:\Pico\pico-sdk"
```

```

c:\Pico>cd pico-extras

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

c:\Pico\pico-extras>cd ..

c:\Pico>cd pico-sdk

c:\Pico\pico-sdk>git submodule update --init
Submodule 'tinysub' (https://github.com/hathach/tinysub.git) registered for path 'lib/tinysub'
Cloning into 'C:/Pico/pico-sdk/lib/tinysub'...
Submodule path 'lib/tinysub': checked out '4bfab30c02279a0530e1a56f4a7c539f2d35a293'

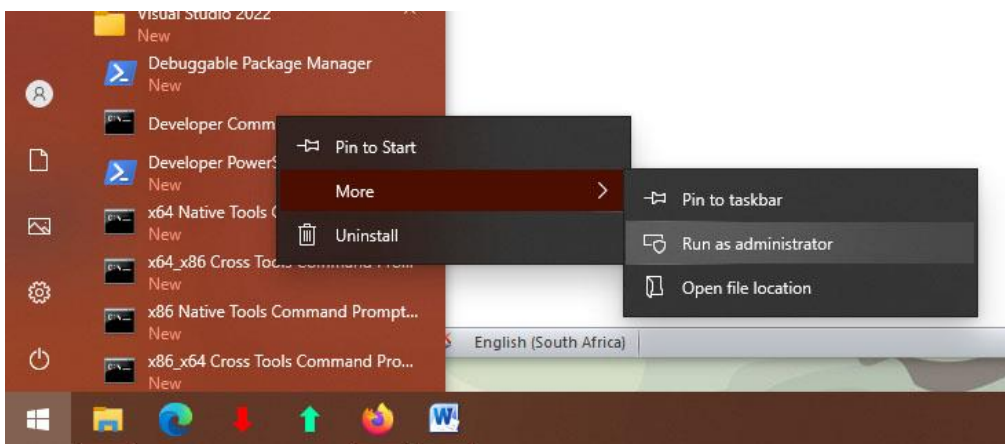
c:\Pico\pico-sdk>setx PICO_SDK_PATH "C:\Pico\pico-sdk"

SUCCESS: Specified value was saved.

c:\Pico\pico-sdk>_

```

9. Close the cmd window and run the VS Developer Command Prompt as admin.



```

c:\Windows\System32>cd c:\Pico
c:\Pico>cd pico-examples\
c:\Pico\pico-examples>mkdir build
c:\Pico\pico-examples>cd build
c:\Pico\pico-examples\build>cmake -G "NMake Makefiles" ..
c:\Pico\pico-examples\build>nmake

```



```
Administrator: Developer Command Prompt for VS 2022
*****
** Visual Studio 2022 Developer Command Prompt v17.2.5
** Copyright (c) 2022 Microsoft Corporation
*****
[ERROR:team_explorer.bat] Directory not found : "C:\Program Files (x86)\Microsoft Visual Studio\2022\BuildTools\Common7\IDE\CommonExtensions\Microsoft\TeamFoundation\Team Explorer"

C:\Windows\System32>cd c:\Pico\pico-sdk
c:\Pico\pico-sdk>cd ..
c:\Pico>cd pico-examples
c:\Pico\pico-examples>mkdir build
c:\Pico\pico-examples>cd build

c:\Pico\pico-examples\build>cmake -G "NMake Makefiles" .;
Using PICO_SDK_PATH from environment ('C:\Pico\pico-sdk')
PICO_SDK_PATH is C:/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 C:/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 C:/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:/Pico/pico-examples/build

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

```
Administrator: Developer Command Prompt for VS 2022 - nmake

c:\Pico\pico-examples\build>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
[ 0%] Generating bs2_default_padded_checksummed.S
[ 0%] Built target bs2_default_padded_checksummed_asm
[ 0%] Creating directories for 'ELF2UF2Build'
[ 0%] No download step for 'ELF2UF2Build'
[ 0%] No update step for 'ELF2UF2Build'
[ 0%] No patch step for 'ELF2UF2Build'
[ 0%] Performing configure step for 'ELF2UF2Build'
-- The C compiler identification is MSVC 19.32.31332.0
-- The CXX compiler identification is MSVC 19.32.31332.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: C:/Program Files (x86)/Microsoft Visual Studio/2022/BuildTools/VC/Tools/MSVC/14.32.31332/bin/Hostx86/x86/cl.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:/Program Files (x86)/Microsoft Visual Studio/2022/BuildTools/VC/Tools/MSVC/14.32.31332/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: C:/Pico/pico-examples/build/elf2uf2
[ 0%] Performing build step for 'ELF2UF2Build'

Microsoft (R) Program Maintenance Utility Version 14.32.31332.0
Copyright (C) Microsoft Corporation. All rights reserved.

[ 50%] Building CXX object CMakeFiles/elf2uf2.dir/main.cpp.obj
main.cpp
C:\Pico\pico-sdk\tools\elf2uf2\main.cpp(359): warning C4996: 'fopen': This function or variable may be unsafe. Consider using fopen_s instead. To disable deprecation, use _CRT_SECURE_NO_WARNINGS. See online help for details.
C:\Pico\pico-sdk\tools\elf2uf2\main.cpp(365): warning C4996: 'fopen': This function or variable may be unsafe. Consider using fopen_s instead. To disable deprecation, use _CRT_SECURE_NO_WARNINGS. See online help for details.
[100%] Linking CXX executable elf2uf2.exe
[100%] Built target elf2uf2
[ 0%] No install step for 'ELF2UF2Build'
```



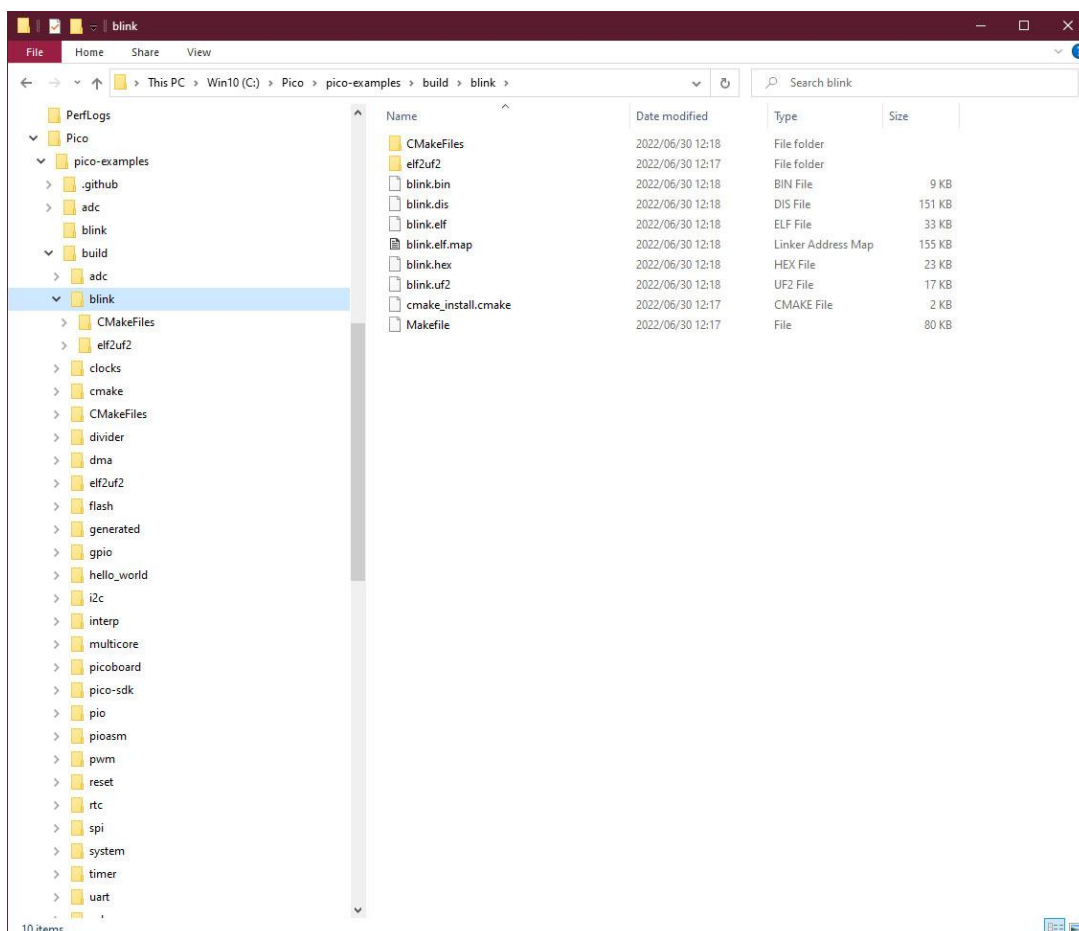
```

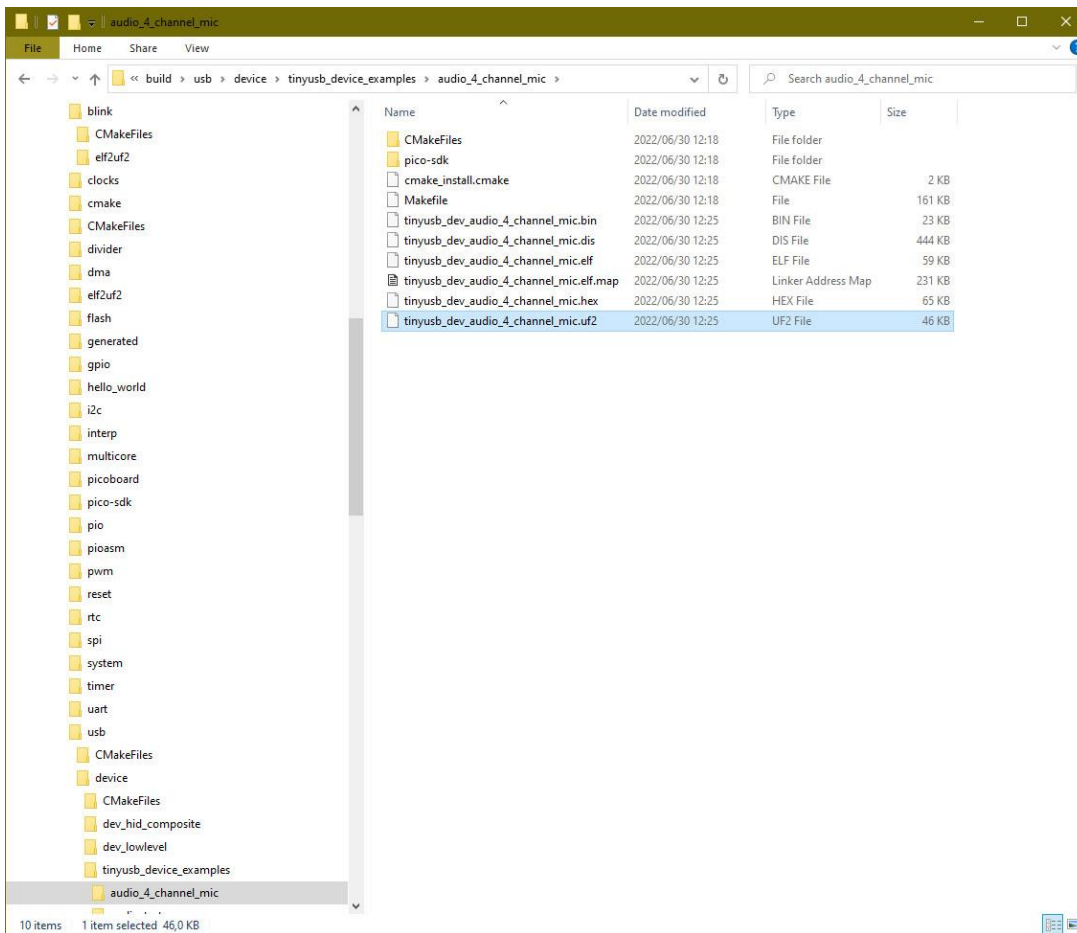
C:\Program Files\Microsoft Visual Studio\2022\Community\VC\Tools\MSVC\14.36.32532\bin\Hostx64\x64
[100%] Building C object watchdog/hello_watchdog/CMakeFiles/hello_watchdog.dir/C:/Pico/pico-sdk/src/rp2_common/hardware_vreg/vreg.c.obj
[100%] Building C object watchdog/hello_watchdog/CMakeFiles/hello_watchdog.dir/C:/Pico/pico-sdk/src/rp2_common/hardware_watchdog/watchdog.c.obj
[100%] Building C object watchdog/hello_watchdog/CMakeFiles/hello_watchdog.dir/C:/Pico/pico-sdk/src/rp2_common/hardware_xosc/xosc.c.obj
[100%] Building C object watchdog/hello_watchdog/CMakeFiles/hello_watchdog.dir/C:/Pico/pico-sdk/src/rp2_common/pico_printf/printf.c.obj
[100%] Building ASM object watchdog/hello_watchdog/CMakeFiles/hello_watchdog.dir/C:/Pico/pico-sdk/src/rp2_common/pico_bit_ops/bit_ops_aeabi.S.obj
[100%] Building C object watchdog/hello_watchdog/CMakeFiles/hello_watchdog.dir/C:/Pico/pico-sdk/src/rp2_common/pico_bootrom/bootrom.c.obj
[100%] Building ASM object watchdog/hello_watchdog/CMakeFiles/hello_watchdog.dir/C:/Pico/pico-sdk/src/rp2_common/pico_divider/divider.S.obj
[100%] Building ASM object watchdog/hello_watchdog/CMakeFiles/hello_watchdog.dir/C:/Pico/pico-sdk/src/rp2_common/pico_double/double_aeabi.S.obj
[100%] Building C object watchdog/hello_watchdog/CMakeFiles/hello_watchdog.dir/C:/Pico/pico-sdk/src/rp2_common/pico_double/double_init_rom.c.obj
[100%] Building C object watchdog/hello_watchdog/CMakeFiles/hello_watchdog.dir/C:/Pico/pico-sdk/src/rp2_common/pico_double/double_math.c.obj
[100%] Building ASM object watchdog/hello_watchdog/CMakeFiles/hello_watchdog.dir/C:/Pico/pico-sdk/src/rp2_common/pico_double/double_vlrom_shim.S.obj
[100%] Building ASM object watchdog/hello_watchdog/CMakeFiles/hello_watchdog.dir/C:/Pico/pico-sdk/src/rp2_common/pico_int64_ops/pico_int64_ops_aeabi.S.obj
[100%] Building ASM object watchdog/hello_watchdog/CMakeFiles/hello_watchdog.dir/C:/Pico/pico-sdk/src/rp2_common/pico_float/float_aeabi.S.obj
[100%] Building C object watchdog/hello_watchdog/CMakeFiles/hello_watchdog.dir/C:/Pico/pico-sdk/src/rp2_common/pico_float/float_init_rom.c.obj
[100%] Building C object watchdog/hello_watchdog/CMakeFiles/hello_watchdog.dir/C:/Pico/pico-sdk/src/rp2_common/pico_float/float_math.c.obj
[100%] Building ASM object watchdog/hello_watchdog/CMakeFiles/hello_watchdog.dir/C:/Pico/pico-sdk/src/rp2_common/pico_float/float_vlrom_shim.S.obj
[100%] Building C object watchdog/hello_watchdog/CMakeFiles/hello_watchdog.dir/C:/Pico/pico-sdk/src/rp2_common/pico_malloc/pico_malloc.c.obj
[100%] Building ASM object watchdog/hello_watchdog/CMakeFiles/hello_watchdog.dir/C:/Pico/pico-sdk/src/rp2_common/pico_memory_ops/mem_ops_aeabi.S.obj
[100%] Building ASM object watchdog/hello_watchdog/CMakeFiles/hello_watchdog.dir/C:/Pico/pico-sdk/src/rp2_common/pico_standard_link/crt0.S.obj
[100%] Building CXX object watchdog/hello_watchdog/CMakeFiles/hello_watchdog.dir/C:/Pico/pico-sdk/src/rp2_common/pico_standard_link/new_delete.cpp.obj
[100%] Building C object watchdog/hello_watchdog/CMakeFiles/hello_watchdog.dir/C:/Pico/pico-sdk/src/rp2_common/pico_standard_link/binary_info.c.obj
[100%] Building C object watchdog/hello_watchdog/CMakeFiles/hello_watchdog.dir/C:/Pico/pico-sdk/src/rp2_common/pico_stdio/stdio.c.obj
[100%] Building C object watchdog/hello_watchdog/CMakeFiles/hello_watchdog.dir/C:/Pico/pico-sdk/src/rp2_common/pico_stdio_uart/stdio_uart.c.obj
[100%] Linking CXX executable hello_watchdog.elf
[100%] Built target hello_watchdog

c:\Pico\pico-examples\build>

```

You can find the uf2 files in the `pico-examples\build\sub-folders`





Building the usb\_sound\_card only:

```
C:\Administrator: Developer Command Prompt for VS 2022
*****
** Visual Studio 2022 Developer Command Prompt v17.2.5
** Copyright (c) 2022 Microsoft Corporation
*****
[ERROR:team_explorer.bat] Directory not found : "C:\Program Files (x86)\Microsoft Visual Studio\2022\BuildTools\Common7\IDE\CommonExtensions\Microsoft\TeamFoundation\Team Explorer"

C:\Windows\System32>cd c:\Pico
c:\Pico>cd pico-playground
c:\Pico\pico-playground>mkdir build
c:\Pico\pico-playground>cd build
c:\Pico\pico-playground\build>cmake -G "NMake Makefiles" ..
Using PICO_SDK_PATH from environment ('C:\Pico\pico-sdk')
PICO_SDK_PATH is C:/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: C:/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 C:/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 C:/Pico/pico-sdk/lib/tinyusb/src/portable/raspberrypi/rp2040; enabling build support for USB.
lwIP available at C:/Pico/pico-extras/lib/lwip/src/core/tcp.c; TCP/IP support is available.
-- Configuring done
-- Generating done
-- Build files have been written to: C:/Pico/pico-playground/build
c:\Pico\pico-playground\build>
```



```

Administrator: Developer Command Prompt for VS 2022 - nmake
-- 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 C:/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 C:/Pico/pico-sdk/lib/tinyusb/src/portable/raspberrypi/rp2040; enabling build support for USB.
lwIP available at C:/Pico/pico-extras/lib/lwip/src/core/tcp.c; TCP/IP support is available.
-- Configuring done
-- Generating done
-- Build files have been written to: C:/Pico/pico-playground/build

c:\Pico\pico-playground\build>cd apps\usb_sound_card

c:\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 C:/Pico/pico-playground/build/pico_extras/src/common/pico_scanvideo/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: C:/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

```

```

Administrator: Developer Command Prompt for VS 2022
[ 62%] Building ASM object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/C:/Pico/pico-sdk/src/rp2_common/pico_double/float_vl_rom_shim.S.obj
[ 62%] Building ASM object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/C:/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/C:/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/C:/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/C:/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/C:/Pico/pico-sdk/src/rp2_common/pico_float/float_vl_rom_shim.S.obj
[ 75%] Building C object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/C:/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/C:/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/C:/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/C:/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/C:/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/C:/Pico/pico-sdk/src/rp2_common/pico_stdio/stdio.c.obj
[ 75%] Building C object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/C:/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/C:/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/C:/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/C:/Pico/pico-sdk/src/rp2_common/hardware_dma/dma.c.obj
[ 87%] Building C object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/C:/Pico/pico-sdk/src/rp2_common/hardware_pio/pio.c.obj
[ 87%] Building C object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/C:/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/C:/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/C:/Pico/pico-extras/src/common/pico_audio/audio.cpp.obj
[ 87%] Building ASM object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/C:/Pico/pico-extras/src/common/pico_audio/audio_utils.S.obj
[ 87%] Building C object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/C:/Pico/pico-extras/src/common/pico_util_buffer/buffer.c.obj
[ 87%] Building C object apps/usb_sound_card/CMakeFiles/usb_sound_card.dir/C:/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

c:\Pico\pico-playground\build\apps\usb_sound_card>

```

