Idiot's guide to compiling grbIHAL for RP2040 in Windows.

By Chuck Staton 1/5/2024

Download Pico SDK:

https://github.com/raspberrypi/pico-setup-windows?tab=readme-ov-file



Click "more info" and "run anyway"





Click next/ok/install/yes/ok/next/whatever/finish (just click through the install)

This will take a while because it's downloading all the stuff you need including VS Code, Python, etc.

| 8 | Raspberry Pi Pico SDK v1.5.1 | | | | | \times |
|----|--|---------------------|------------|-------------|---------|----------|
| 1 | installing | | | | | - |
| | Please wait while Raspberry Pi Pico SD | K v1.5.1 is being i | nstalled. | | | |
| | Downloading https://code.visualstudio | .com/sha/downlo | ad?build=s | table8os=wi | n32-x64 | |
| | | | | | | |
| | Show details | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| No | illsoft Install System v3.08 | | | | | _ |
| | | < Ba | sck. | Next > | Can | cel |
| | | | | | | |



If it sits here forever saying [0/51], it's waiting for you to press ENTER for some strange reason:



Just hit ENTER and it will finish:

Then hit ENTER again when it gets to the end:



You can close the Pico-setup command window.

Download and install Github Desktop

https://desktop.github.com/



| ← → × ↑ 🖊 | > Th | is PC > Downloads | | |
|----------------|------|------------------------------------|-------------------|-------------|
| 🔹 Quick access | | Name ^ | Date modified | Туре |
| Deckton | | GitHubDesktopSetup-x64 | 1/5/2024 12:24 AM | Application |
| | 7 | pice-setup-windows of L-scandalone | 1/4/2024 11:21 PM | Application |

Don't bother creating an account if you don't already have one, and don't plan on committing anything

Welcome to GitHub Desktop

GitHub Desktop is a seamless way to contribute to projects on GitHub and GitHub Enterprise. Sign in below to get started with your existing projects.





Click Clone:



Click URL, then enter https://github.com/grbIHAL/RP2040/, then click "Clone"

| Clone a repository | | |
|--------------------|-------------------|-----|
| GitHub.com | GitHub Enterprise | URL |
| | | |

Once it's done, **DO NOT** click "Open in VS Code." For some reason this causes errors later on.



Close Github and open VS Code from the start menu



If you already had VS Code installed, make sure you open the new one that says "Pico - VS Code"



If this comes up, tick all the boxes and click Allow



If this comes up, click YES:

| $\overline{\mathbf{v}}$ | Do you trust the authors o | f the files in this folder? | | | | | |
|-------------------------|---|----------------------------------|--|--|--|--|--|
| | Code provides features that may automatically execute files in this folder. | | | | | | |
| | If you don't trust the authors of these files, we recommend to continue in restricted mode as the files may be malicious. See our docs to learn more. | | | | | | |
| | C:\Users\Administrator\Documents\RP2040 | | | | | | |
| | Trust the authors of all files in the parent folder 'Documents' | | | | | | |
| | Yes, I trust the authors No, I don't trust the authors | | | | | | |
| | Trust folder and enable all features | Browse folder in restricted mode | | | | | |

Click "X" on all the notifications in the bottom right. Don't bother reading them, just close them.

Skip all the "welcome to VS Code" nonsense by clicking "Mark Done":



Note: Trying to clone the repository from inside VS Code does **<u>NOT</u>** work. Just use Github desktop as shown earlier..



After you open the cloned repo, there will be a message in the bottom right asking if you want to configure the project, click Yes. If you don't respond to the message in time, it will disappear, but you can get it back by clicking the bell icon in the bottom right.



Once you click YES, there will be a list of option in the top bar. Choose the bottom one (GCC 10.3.1 arm-...)



It will run some processes and spit out some text in the bottom of the screen. Check to make sure nothing in that box smells like an error (anything finishing with an exit code not equal to 0, is smelly).

Select CMakeLists.txt, right-click it, and click build all



It will do a bunch of thinking, generate a bunch of monologue, and create a new folder ("build") in the project directory, but there won't be any firmware in it because we didn't select a board. This was just a test.

If there are no errors, you can select a board by opening my_machine.h and uncommenting the board you're using (removing the "//" in front of it).

| EXPLORER ···· | C my_machine.h |
|--|--|
| RP2040 C ioexpand.h C ioports_analog.c C ioports.c C littlefs_hal.c C littlefs_hal.h C lwipopts.h C main.c C MCP3221.c | <pre>C my_machine.h > BOARD_BTT_SKR_PICO_10 19</pre> |
| C MCP3221.h | <pre>91 #define BOARD_BTT_SKR_PICO_10 // incomplete and untested!</pre> |
| c my_machine.h | 30 //#define BUARD_UNC_BUDSTERPACK |
| C Picesonice C pice_cnc_map.h C pice_cnc.c ≅ pice_sdk_import.cmake C picebob_dlx_g540_map.h C picebob_dlx_map.h | <pre>31 //#define BOARD_CITOH_CX6000 // C.ITOH CX-6000 HPGL plotter 32 //#define BOARD_GENERIC_4AXIS 33 //#define BOARD_MY_MACHINE // Add my_machine_map.h before enabling this! 34 35 // Configuration 36 // Uncomment to enable.</pre> |

Now, again select CMakeLists.txt, right-click it, and click build all. Now you should have some usable firmware in the Build folder.

Go into the new folder (build) and the grblHAL.uf2 file is the firmware you just created.

| C citon_cx6000_map.n | | - | Land and mail and | NE STATE OF STATE | nacio anticolari d | |
|-----------------------|-------------------|-----------|------------------------|--------------------|--------------------|----------|
| C citoh_cx6000.c | | 10 op | Lion (ADD MOTT "Add MO | WS Service OFF) | | |
| CMakeLists.tpl | | Id | TOTTADO HOTT AGO HO | THE OTTENT APT OFF | | |
| M CMakeLists.txt | | 1000 CO | 20.441 | | | |
| C cnc_boosterpack_ma | File Home | Share V | iew | | | |
| R COPYING | ← → • ↑ 📙 | > This PC | > Documents > GitHub > | RP2040 > build | | |
| C dhcpserver.c | | | Name | Date modified | Туре | Size |
| C dhcpserver.h | 🖈 Quick access | | | | | 10000 |
| C driver.c | Desktop 🛃 Desktop | * | .cmake | 1/5/2024 12:47 AM | Filefolder | |
| C driver.h | | * | CMakeFiles | 1/5/2024 1:08 AM | File folder | |
| () driver.ison | | | generated | 1/5/2024 12:47 AM | File folder | |
| E driverPIO nio | | Я | pico-sdk | 1/5/2024 12:47 AM | File folder | |
| C flash a | Pictures | 1 | i.ninja_deps | 1/5/2024 1:08 AM | NINJA_DEPS File | 96 KB |
| C flash.c | PanelPC | | ninja_log | 1/5/2024 1:08 AM | NINJA_LOG File | 22 KB |
| C flash.h | | _ | bs2_default.elf.map | 1/5/2024 1:08 AM | MAP File | 4 KB |
| C generic_map_4axis.h | This PC | | build.ninja | 1/5/2024 12:47 AM | NINJA File | 1,395 KB |
| C generic_map.h | Network | | cmake_install | 1/5/2024 12:47 AM | CMake Source File | 2 KB |
| C i2c.c | | | CMakeCache | 1/5/2024 12:47 AM | Text Document | 22 KB |
| C i2c.h | | | compile_commands | 1/5/2024 12:47 AM | JSON Source File | 1,155 KB |
| C ioexpand.h | | | c driverPIO.pio | 1/5/2024 1:08 AM | C Header Source F | 12 KB |
| C ioports_analog.c | | | grbIHAL.bin | 1/5/2024 1:08 AM | BIN File | 196 KB |
| C ioports.c | | | grbIHAL.dis | 1/5/2024 1:08 AM | DIS File | 3,061 KB |
| - Faul - E - I - | | | grbIHAL.elf | 1/5/2024 1:08 AM | ELF File | 2,200 KB |
| > OUTLINE | | | grbHAL.elf.map | 1/5/2024 1:08 AM | MAP File | 797 KB |
| > TIMELINE | | N | grbIHAL.hex | 1/5/2024 1:08 AM | HEX File | 551 KB |
| P master* ↔ ⊗ 0 △ 0 | | | grbIHAL.uf2 | 1/5/2024 1:08 AM | UF2 File | 392 KB |

If you're happy with it, drag and drop the grbIHAL.uf2 file into the board. When you first plug in a RP2040 board, it comes up as a USB mass storage device.

You just drag grbIHAL.uf2 out of the GitHub\RP2040\build folder and drop it into the RPI-RP2 device.

| | S ONTION AND WIFT | AND WIFT DETWOPETRE UFF | | | | - | - 0 | × | 🥪 🗹 📕 🖛 | Manage RPI-RP2 (D:) |
|------------------------------------|------------------------------|--|--|---|---|---|-----|-----|--|--|
| View | | | | | | | | ~ 😮 | File Home Share View D | rive Tools |
| Cut Copy path Paste shortcut | Move Copy to to Organize | New item • Desy access • Proper New | → Open - + + + + + + + + + + + + + + + + + + | Select all Select none Invert selection Select | | | | | Image: Weight of the second secon | Move to V X Delete V Copy to V Rename New fold Organize |
| C → Documen | ts > GitHub > RP2040 > build | | | | ~ | õ | | Q | $\leftarrow \rightarrow \checkmark \uparrow \checkmark \Rightarrow RPI-RP2 (D:)$ | |
| ^ Nat | me | Date modified | Type | Size | | | | | A Quick access | lame ^ |
| - 11 1 | .cmake CMakeFiles | 1/4/2024 11:15 PW 1/5/2024 7:48 AM | File folder | | | | | | This PC | |
| - I i | generated | 1/4/2024 11:13 PM | File folder | | | | | | 3D Objects | |
| | pico-sdk | 1/4/2024 11:13 PM | File folder | | | | | | Deskton | |
| | .ninja_deps | 1/5/2024 7:48 AM | NINJA_DEPS File | 96 KB | | | | | Personal Per | |
| | .ninja_log | 1/5/2024 7:48 AM | NINJA_LOG File | 22 KB | | | | | Documents | |
| ₩ 1 | bs2_default.elf.map | 1/5/2024 7:48 AM | Linker Address Map | 4 KB | | | | | Downloads | |
| | build.ninja | 1/4/2024 11:13 PM | NINJA File | 1,377 KB | | | | | Music | |
| E | cmake_install.cmake | 1/4/2024 11:13 PM | CMake Source File | 2 KB | | | | | E Pictures | |
| | CMakeCache.txt | 1/4/2024 11:13 PM | Text Document | 22 KB | | | | | 🖶 Videos 🔭 | |
| រ | compile_commands.json | 1/4/2024 11:13 PM | JSON File | 1,144 KB | | | | | 🐛 Local Disk (C:) | |
| ₩ | driverPIO.pio.h | 1/5/2024 7:48 AM | C/C++ Header | 12 KB | | | | | RPI-RDP (02) | |
| | grbIHAL.bin | 1/5/2024 7:48 AM | BIN File | 196 KB | | | | | Operating VMr. (E) | + Copy to RPI-RP2 (D:) |
| | grbIHAL.dis | 1/5/2024 7:48 AM | DIS File | 3,061 KB | | | | | Coperating vivis (c.) | |
| | grbIHAL.elf | 1/5/2024 7:48 AM | ELF File | 2,203 KB | | | | | VM Storage (F:) | |
| × | grbIHAL.elf.map | 1/5/2024 7:48 AM | Linker Address Map | 811 KB | | - | | | 🥪 File Storage (G:) | |
| - Goi 📄 | grbIHAL.hex | 1/5/2024 7:48 AM | HEX File | 551 KB | | | | | 🛆 cstatonftlg@gmail.com - Go | |
| il.con 📋 | grbIHAL.uf2 | 1/5/2024 7:48 AM | UF2 File | 392 KB • | | | | | 🛆 maintenance.ftlg@gmail.con | |
| nubli Y 1 KB | | | | | | | [| | 2 items | |
| tio 8 mos | 39 include(embroider) | y/CMakeLists.txt) | | | | | | | | |

The board will reboot and disappear. If it disappears, that means it worked, because it now has firmware and has booted into its normal mode.

On some boards (like mine, the BTT SKR Pico board) there is a jumper that you must install in order for this mass storage device mode to be turned on. And you must remove the jumper once firmware has been installed in it, or it will keep booting into mass storage mode.



A normal Raspberry Pi Pico board does not have this jumper, so it comes up in Mass Storage mode only the first time. If you want to reinstall/update firmware onto it, you must hold down the "bootsel" button while you plug it in.



If you want to make changes to the firmware, make them in...

• The pin map for your board (in my case GitHub\RP2040\btt_skr_pico_10_map.h):

| File Edit Selection View Go | o | $\leftarrow \rightarrow$ | P RP2040 (Admin | istrator] |
|---|---|---|--|--|
| EXPLORER | | C my_machine.h M | C btt_skr_pico_10_map.h | × |
| V RP2040 V webui oitattributes | | C btt_skr_pico_10_0 35 #define 1 36 #define b | map.h > TRINAMIC_STREAM 1 MAS_ROARD_INIT | |
| .gitatinbutes .gitignore .gitmodules | | 37 38 // Define | e step pulse output pins. | |
| C bluetooth.c C bluetooth.h | | 39 #define 9 40 #define) 41 #define) | STEP_PORT <_STEP_PIN /_STEP_PIN | GPIO_PIO_1 // Single pin PIO SM 11 6 |
| C btt_skr_pico_10_map.h | | 42 #define 2 43 | _STEP_PIN | 19 |
| C btt_skr_pico_10.c C citoh_cx6000_map.h C citoh_cx6000.c | | 44 // Define 45 #define 46 #define 47 #define) | ≥ step direction output p: DIRECTION_PORT DIRECTION_OUTMODE <_DIRECTION_PIN | ins. GPIO_OUTPUT GPIO_MAP 10 |
| M CMakeLists.txt C cnc_boosterpack_map.h | | 48 #define 49 #define 50 | /_DIRECTION_PIN Z_DIRECTION_PIN | 5 28 |
| COPYING C dhcpserver.c | | 51 // Define 52 #define | e stepper driver enable/d ENABLE_PORT | isable output pin. GPIO_OUTPUT |

This is where you change which pins are assigned to which functions. In my case, I want to utilize some of the unmapped pins as AUX relay outputs repurpose some of the mapped pins for other uses

- GitHub\RP2040\my_machine.h
 - This is where you select the board you are using and enable optional functions.
- GitHub\RP2040\grbl\config.h
 - More advanced settings here, you probably don't need to change anything.

...but leave everything else alone unless you know what you're doing. If you needed this guide, you should probably stick to those 3 files. Make your changes, and then select CMakeLists.txt, right-click it, and click build all, then **rename the new grbIHAL.uf2 file** and drag/drop the new file onto the board. The firmware file should always have a different name each time you send it to the board.