Install OpenOCD and remote usage with CLion

Introduction

OpenOCD is meant to be run locally but with a few tricks we can execute it on a raspberry pi. The idea will be to forward openocd calls through SSH and forward the port 3333 from the localhost to the raspberry pi.

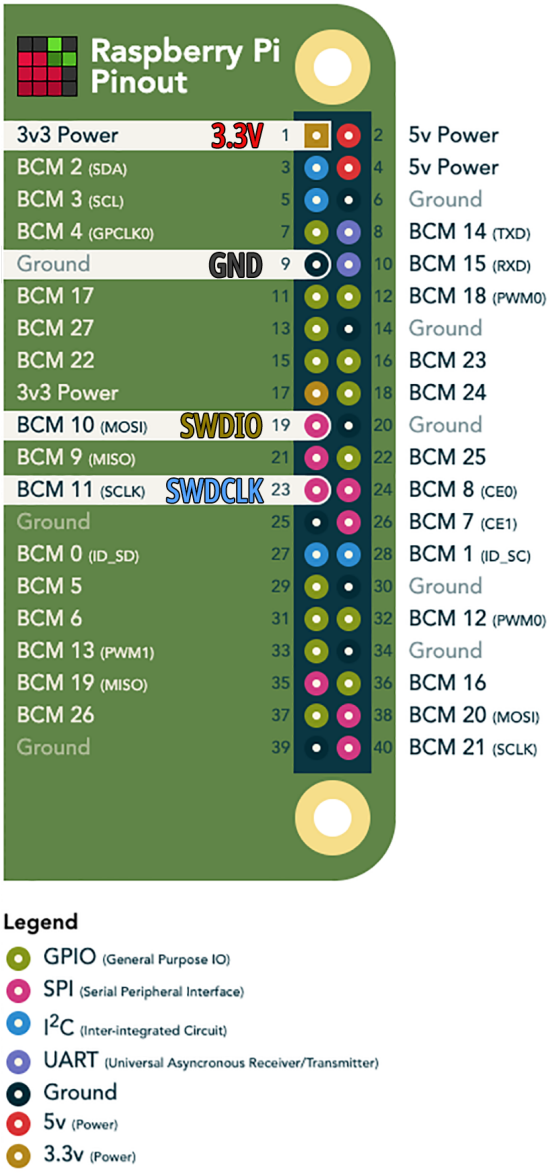
Setup

Wiring a Raspberry PI (zero, A+, B+,2,3,4)

The protocol to flash the Pinetime is SWD. In order to communicate with the Pinetime hardware, the flasher/debugger must be connected to the right pins.

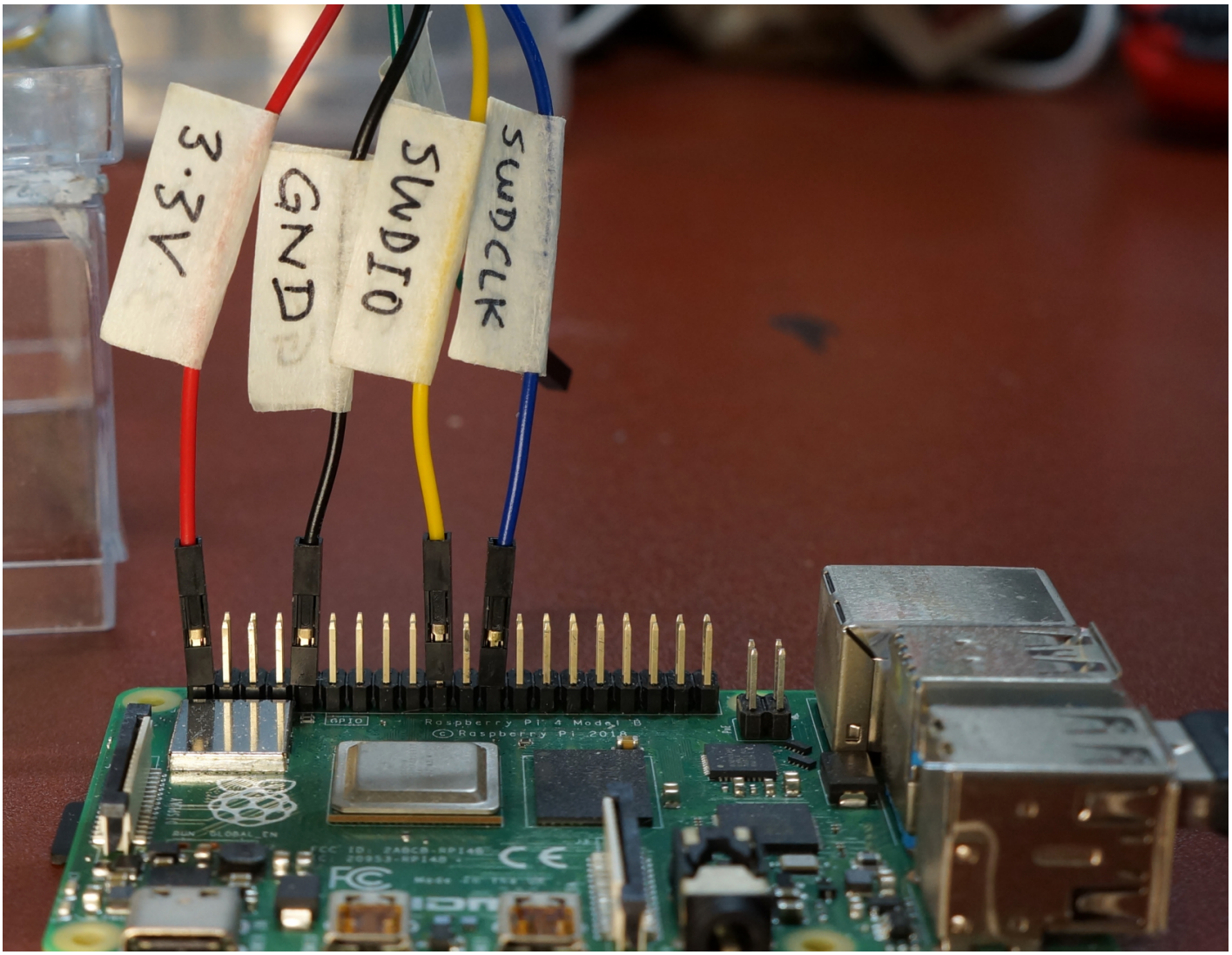


Pinetime SWD Pinout from
https://wiki.pine64.org/wiki/PineTime_Devkit_Wiring



Raspberry PI pinout from
https://github.com/lupyuen/visual-embedded-rust/blob/master/README.md#connect-pinetime-to-raspberry-pi
Valid for a Raspberry PI zero, A+, B+,2,3,4

Small trick: label your cables



Raspberry PI setup

Install raspbian

1. For simplicity, imager is a very good tool : package `rpi-imager` at least on Ubuntu and Archlinux repositories or download it from the official website

   https://www.raspberrypi.com/software/.
2. Flash your memory card with your favorite Raspberry PI OS (here I suppose raspbian lite)
3. Preflight configuration : mount the memory card on your dev system and
   1. Setup networking by editing /etc/wpa_supplicant/wpa_supplicant.conf and adding the following section

      ```
      network={
        ssid="YOUR_WIFI_SSID"
        psk="YOUR_WIFI_PASSWORD"
      ```

```
key_mgmt=WPA-PSK
  #scan_ssid=1  # Only needed if wifi ssid is hidden
}
```

2. Enable SSH by creating a `ssh` file in `/boot`. By example, run `touch /boot/ssh`.
3. Configure more by reading the documentation :

   https://www.raspberrypi.com/documentation/computers/configuration.html

4. **DO NOT CONNECT THE PINETIME IF IT IS NOT CHARGED**, it might drain to much current from the raspberry pins
5. Read the point above and then power up your raspberry PI
6. Monitor your router or DHCP server to find the PI ip address. Alternatively, you can scan your network with nmap or follow one of the other techniques documented here :

   https://www.raspberrypi.com/documentation/computers/remote-access.html
7. Connect to it through SSH with user `pi` and password `raspberry`. If mDNS works with your network environment, you can use `ssh pi@raspberrypi.local`.
   1. Good practice : Immediately set a new password using the `passwd` command.
   2. More security setup documented here :

      https://www.raspberrypi.com/documentation/computers/configuration.html#securing-your-raspberry-pi

8. Run `sudo raspi-config` to enable SPI
   1. Select Interfacing Options → SPI → Yes

Install OpenOCD-SPI

We need a patched version of OpenOCD to use SWD over SPI. Thus, the version from the official repository won't work. Instead, we'll install the `openocd-spi` fork by *lupyuen* :

https://github.com/lupyuen/openocd-spi. To simplify the Pinetime debugging operations, *lupyuen* created a tool called pinetime-updater which installs openocd-spi for us.

1. Install required tools : `apt install git`

Follow the instructions from the pinetime-updater repository or trust me with this :

```
# Clone the repo using
git clone https://github.com/lupyuen/openocd-spi.git  <-- do we need this ?


git clone https://github.com/lupyuen/pinetime-updater
# Run the bash script
cd pinetime-updater
```

```
chmod +x run.sh
./run.sh
```

Compilation troubleshooting

If you have compilation errors try reverting to a previous gcc version with the following steps. At the time of writing, the gcc version was 10, so the steps show how to revert to version 9 but you can easily transpose this (dear reader of the future).

1. Download previous gcc package `sudo apt install gcc-9`
2. Setup alternatives using the following two lines (10 and 20 refers to priorities ; bigger == higher)

   ```
   # Based on a stackoverflow response from Edd Inglis
   # https://askubuntu.com/questions/26498/how-to-choose-the-default-gcc-and-g-version

   # Remove previous setup
   sudo update-alternatives --remove-all gcc

   # Add the two options with higher priority for the latest version
   sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-9 10
   sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-10 20

   # Select the version to use
   sudo update-alternatives --config gcc
   ```

3. Execute the `run.sh` script again

Create OpenOCD configuration

Create the file `/home/pi/pinetime-updater/scripts/debug.ocd` with the content:

```
# This is an OpenOCD script that connects to the nRF52 for Cortex Debug.

# Debug Level must be 2 or greater or gdb will fail.
debug_level 2

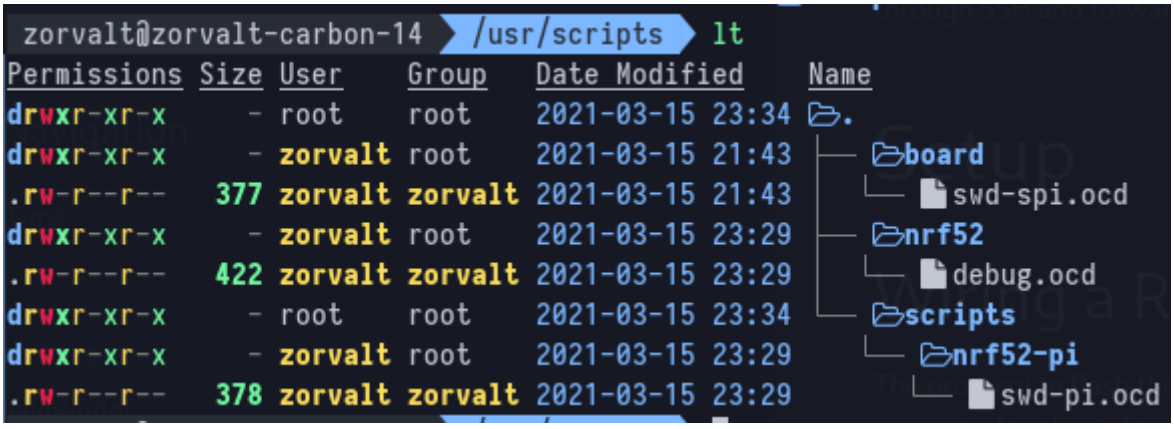
$_TARGETNAME configure -event reset-init {
    # Arm Semihosting is used to show debug console output and may only be enabled after init event.  We wait for the event and enable Arm Semihosting.
    echo "Enable ARM Semihosting to show debug output"
    arm semihosting enable
}
```

Host configuration for remote debugging

Local openocd configuration

Create the files under `/usr/scripts` with the same content as the files from `/home/pi/pinetime-updater/scripts` (same filename => same content).

You should endup with the following tree :



Forward openocd calls

Create the file /usr/bin/openocd with the following content :

```
#! /usr/bin/env bash
ssh -t -L 3333:localhost:3333 __REPLACE__THIS__ "cd pinetime-updater; openocd $@"
```

Then replace the `__REPLACE__THIS__` placeholder with your raspberry pi user and ip. Example: `pi@192.168.1.15` .

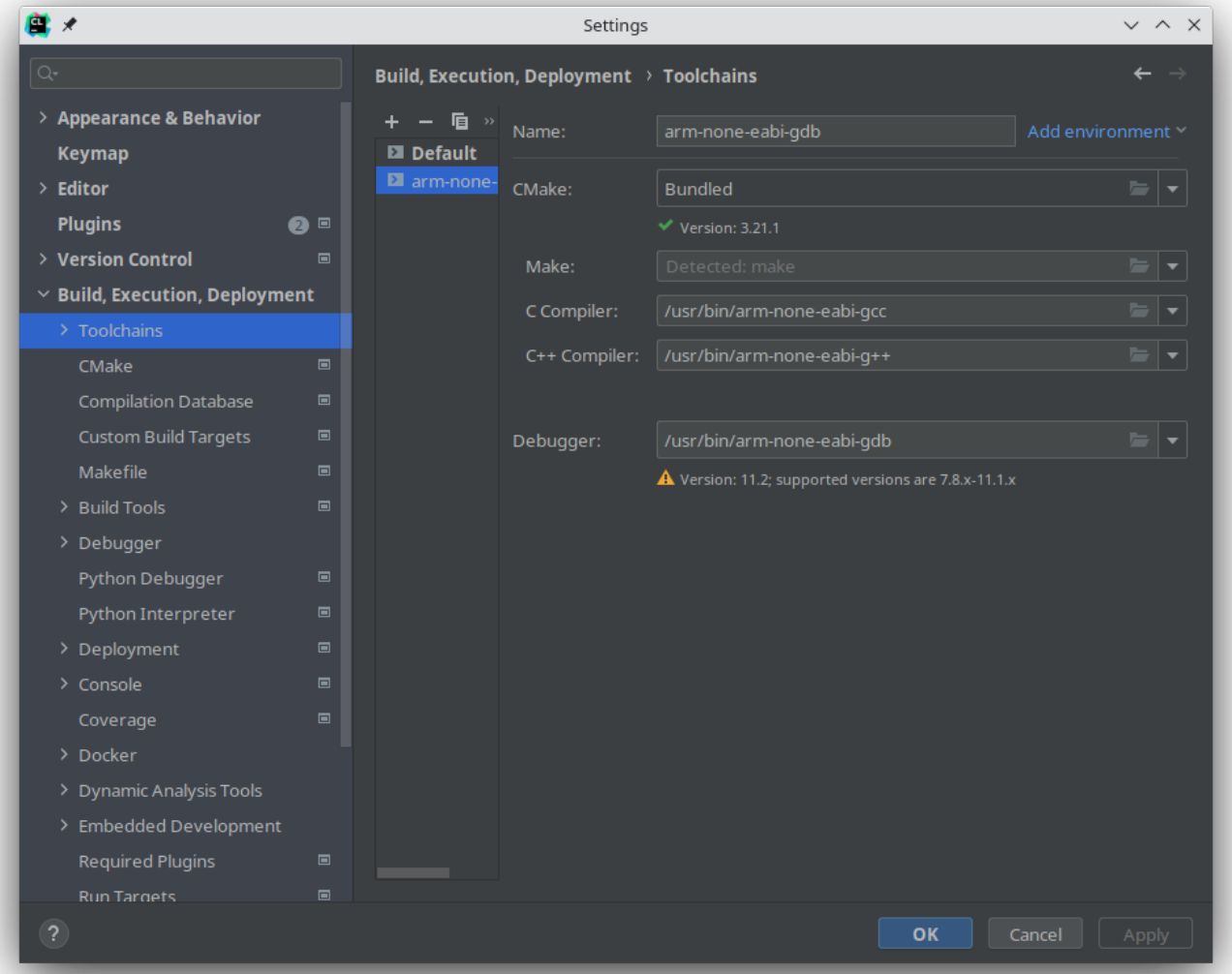
Do not forget to `chmod +x /usr/bin/openocd` when you are done.

Configure CLion

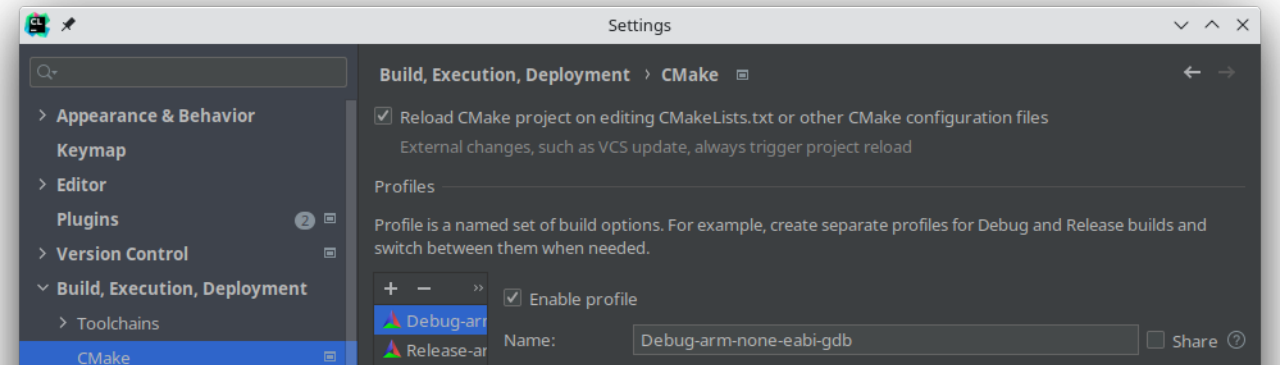
Toolchain

sudo apt-get install gcc-arm-none-eabi gdb-multiarch

sudo ln -s /usr/bin/gdb-multiarch /usr/bin/arm-none-eabi-gdb



CMake

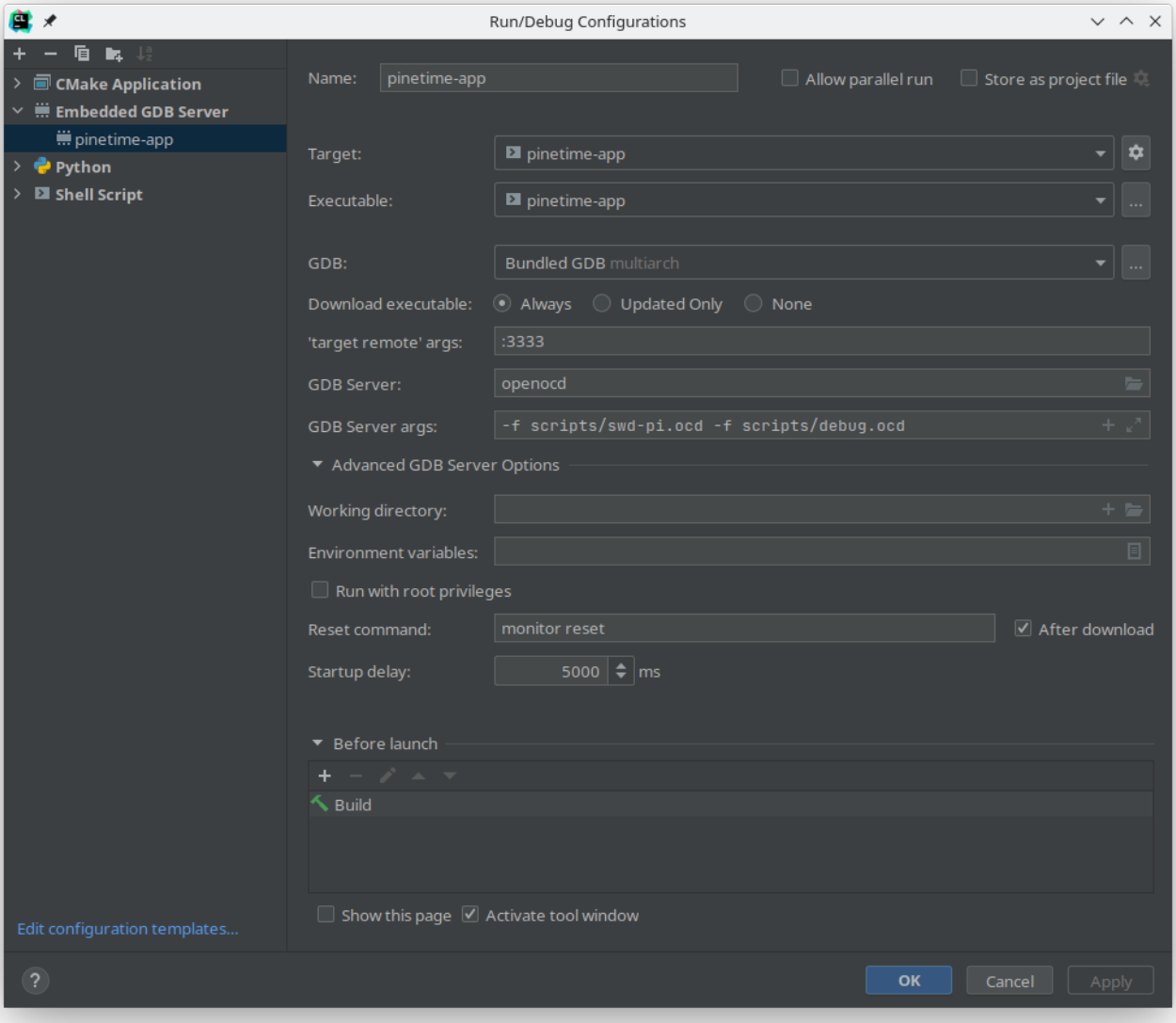


With `CMake options` field content :

```
-DCMAKE_C_COMPILER_FORCED=1
-DCMAKE_CXX_COMPILER_FORCED=1
-DARM_NONE_EABI_TOOLCHAIN_PATH="/usr"
-DNRF5_SDK_PATH="/opt/nrf5-sdk"
-DNRFJPROG="/usr/bin/nrfjprog"
-DUSE_OPENOCD=1
```

Run/Debug configuration

Pay attention to the `Startup delay` field. As the openocd command must first connect through SSH before starting a GDB server, you might experience some lag, especially on a Raspberry PI zero. Tune its value to your needs.



Revision #27

Created Mon, Mar 15, 2021 7:44 PM by Sylvain

Updated Sun, Mar 6, 2022 6:10 PM by Romain