CEC Continuing Education Center

DesignNews

Constructing a Raspberry Pi RP2040 Low-Power Sensor Node

Day 1:

Assembling the RP2040 Linux Toolchain

Sponsored by

Digi-Key ELECTRONICS

informa markets

Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Attendee Chat’ by maximizing the chat widget in your dock.

Fred Eady

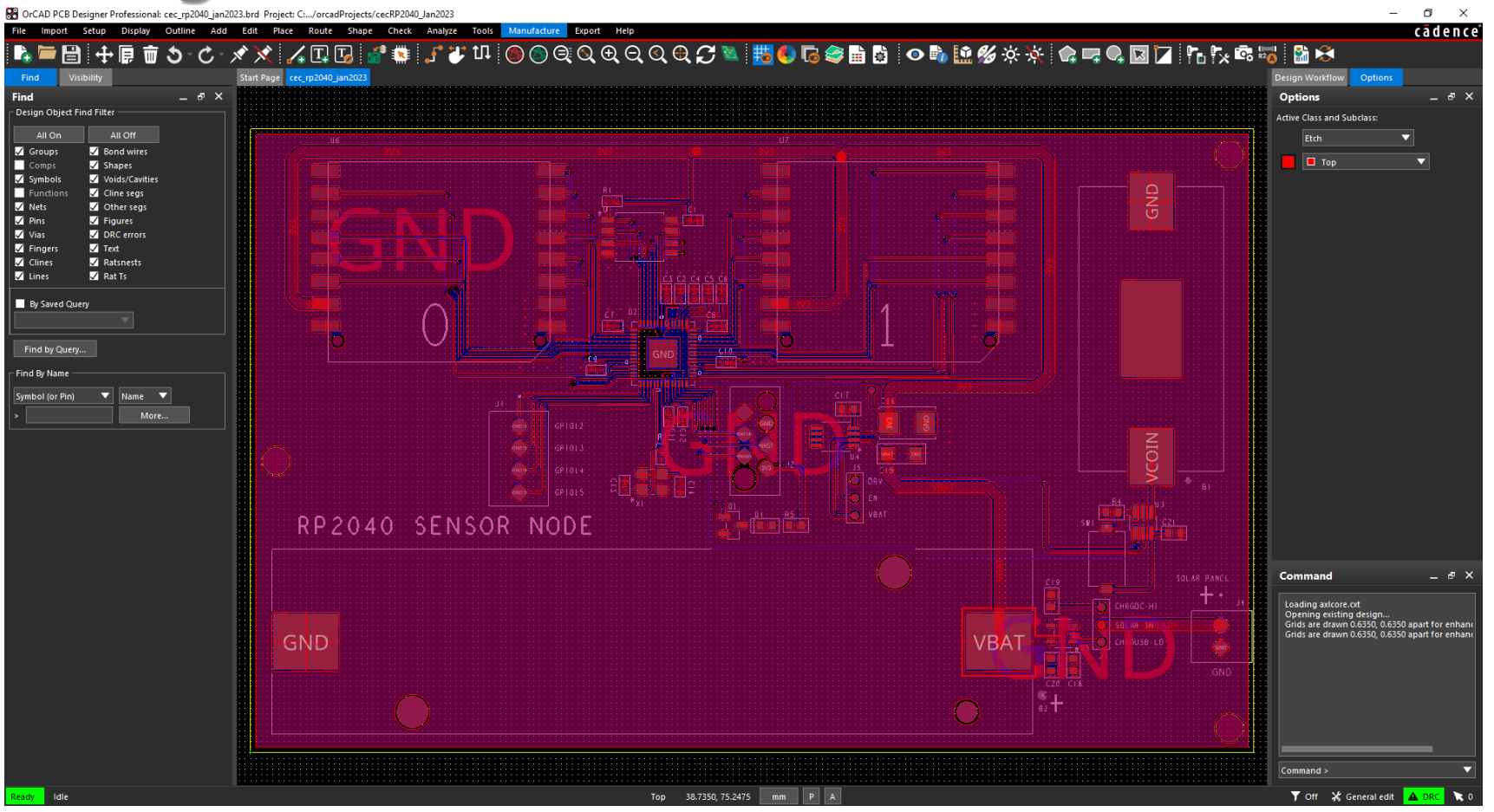
Visit ‘Lecturer Profile’ in your console for more details.

CEC Continuing Education Center

Sponsored By Digi-Key Corporation

AGENDA

- Install the Raspberry Pi Pico C/C++ SDK
- Install the *gcc-arm-none-eabi* Toolchain
- Install and Configure Visual Studio Code
- Test Flight



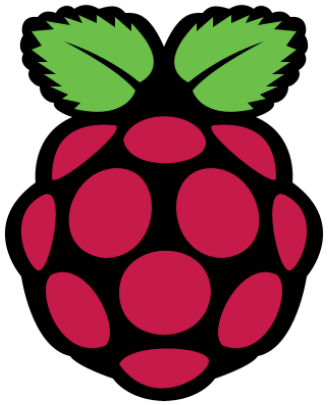
Create the *pico* Directory and Install git

```
fred@shoptc1660: ~/pico
File  Edit  View  Search  Terminal  Help

fred@shoptc1660:~$ mkdir pico
fred@shoptc1660:~$ cd pico
fred@shoptc1660:~/pico$
```

```
fred@shoptc1660: ~/pico
File  Edit  View  Search  Terminal  Help

fred@shoptc1660:~/pico$ sudo apt install git
[sudo] password for fred:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libflashrom1 libftdi1-2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 4,112 kB of archives.
After this operation, 20.9 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 liberror-perl all 0.17029-1 [26.5 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git-man all 1:2.34.1-1ubuntu1.5 [953 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git amd64 1:2.34.1-1ubuntu1.5 [3,132 kB]
Fetched 4,112 kB in 1s (4,085 kB/s)
Selecting previously unselected package liberror-perl.
(Reading database ... 198974 files and directories currently installed.)
Preparing to unpack .../liberror-perl_0.17029-1_all.deb ...
Unpacking liberror-perl (0.17029-1) ...
Selecting previously unselected package git-man.
Preparing to unpack .../git-man_1%3a2.34.1-1ubuntu1.5_all.deb ...
Unpacking git-man (1:2.34.1-1ubuntu1.5) ...
Selecting previously unselected package git.
Preparing to unpack .../git_1%3a2.34.1-1ubuntu1.5_amd64.deb ...
Unpacking git (1:2.34.1-1ubuntu1.5) ...
Setting up liberror-perl (0.17029-1) ...
Setting up git-man (1:2.34.1-1ubuntu1.5) ...
Setting up git (1:2.34.1-1ubuntu1.5) ...
Processing triggers for man-db (2.10.2-1) ...
fred@shoptc1660:~/pico$
```

Install the Raspberry Pi Pico C/C++ SDK

```
fred@shoptc1660: ~/pico
File  Edit  View  Search  Terminal  Help

fred@shoptc1660:~/pico$ git clone https://github.com/raspberrypi/pico-sdk.git --branch master
Cloning into 'pico-sdk'...
remote: Enumerating objects: 5748, done.
remote: Counting objects: 100% (1306/1306), done.
remote: Compressing objects: 100% (291/291), done.
remote: Total 5748 (delta 1096), reused 1051 (delta 1011), pack-reused 4442
Receiving objects: 100% (5748/5748), 2.36 MiB | 4.09 MiB/s, done.
Resolving deltas: 100% (2992/2992), done.
fred@shoptc1660:~/pico$
```

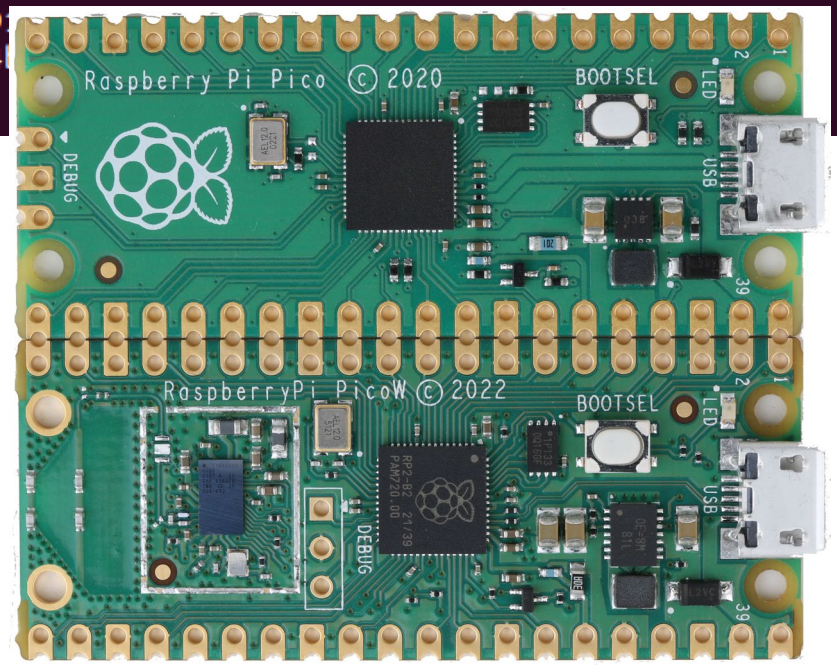


Install *tinyusb* and the Pico W Drivers

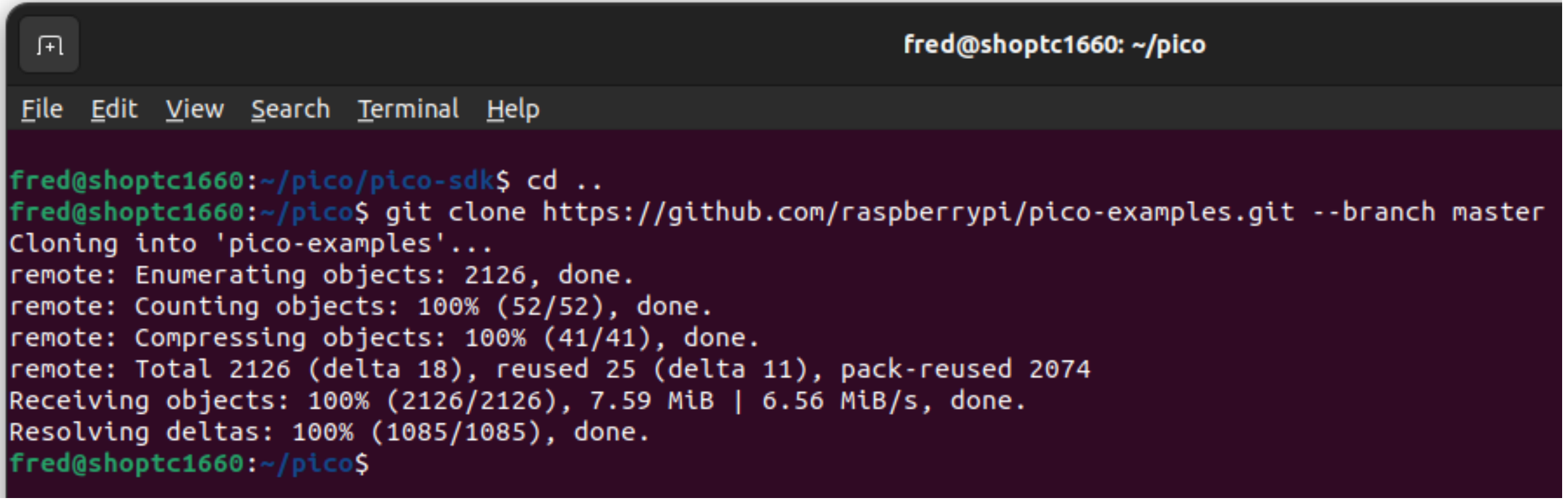
fred@shoptc1660: ~/pico/pico-sdk

File Edit View Search Terminal Help

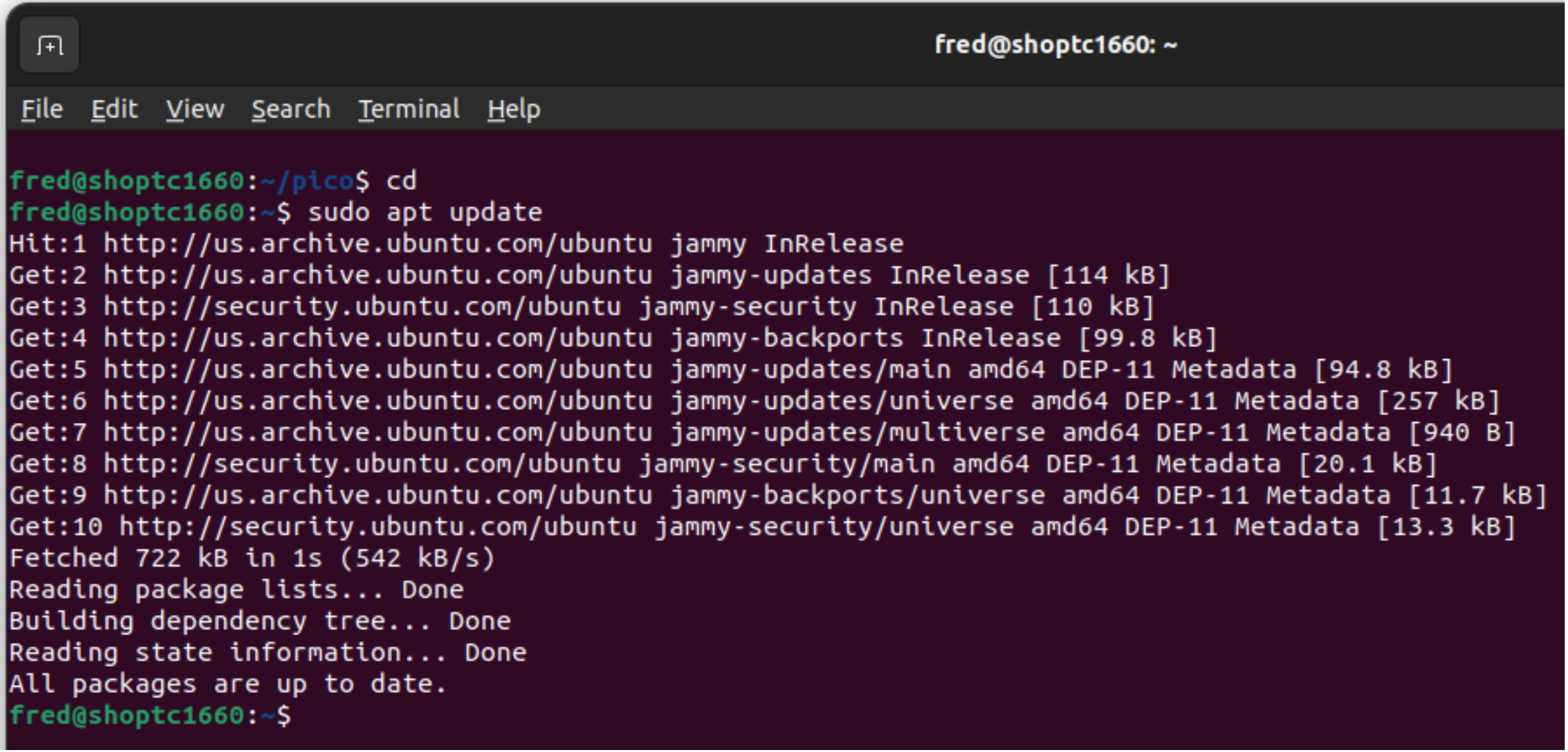
```
fred@shoptc1660:~/pico$ cd pico-sdk
fred@shoptc1660:~/pico/pico-sdk$ git submodule update --init
Submodule 'lib/cyw43-driver' (https://github.com/georgerobotics/cyw43-driver.git) registered for path 'lib/cyw43-driver'
Submodule 'lib/lwip' (https://github.com/lwip-tcpip/lwip.git) registered for path 'lib/lwip'
Submodule 'tinyusb' (https://github.com/hathach/tinyusb.git) registered for path 'lib/tinyusb'
Cloning into '/home/fred/pico/pico-sdk/lib/cyw43-driver'...
Cloning into '/home/fred/pico/pico-sdk/lib/lwip'...
Cloning into '/home/fred/pico/pico-sdk/lib/tinyusb'...
Submodule path 'lib/cyw43-driver': checked out '195dfcc10bb6f379e3dea45147590db2203d3c7b'
Submodule path 'lib/lwip': checked out '2399
Submodule path 'lib/tinyusb': checked out '4
fred@shoptc1660:~/pico/pico-sdk$
```



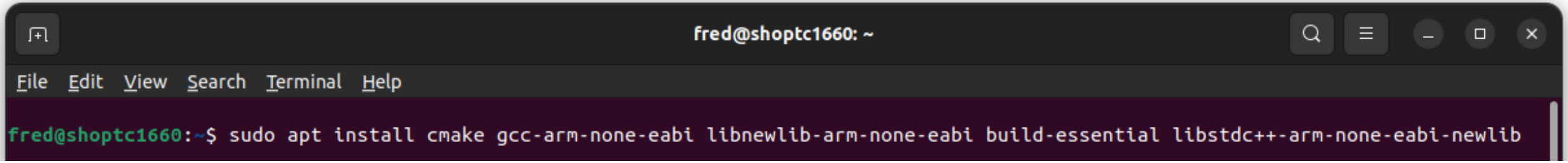
Install the Pico Examples



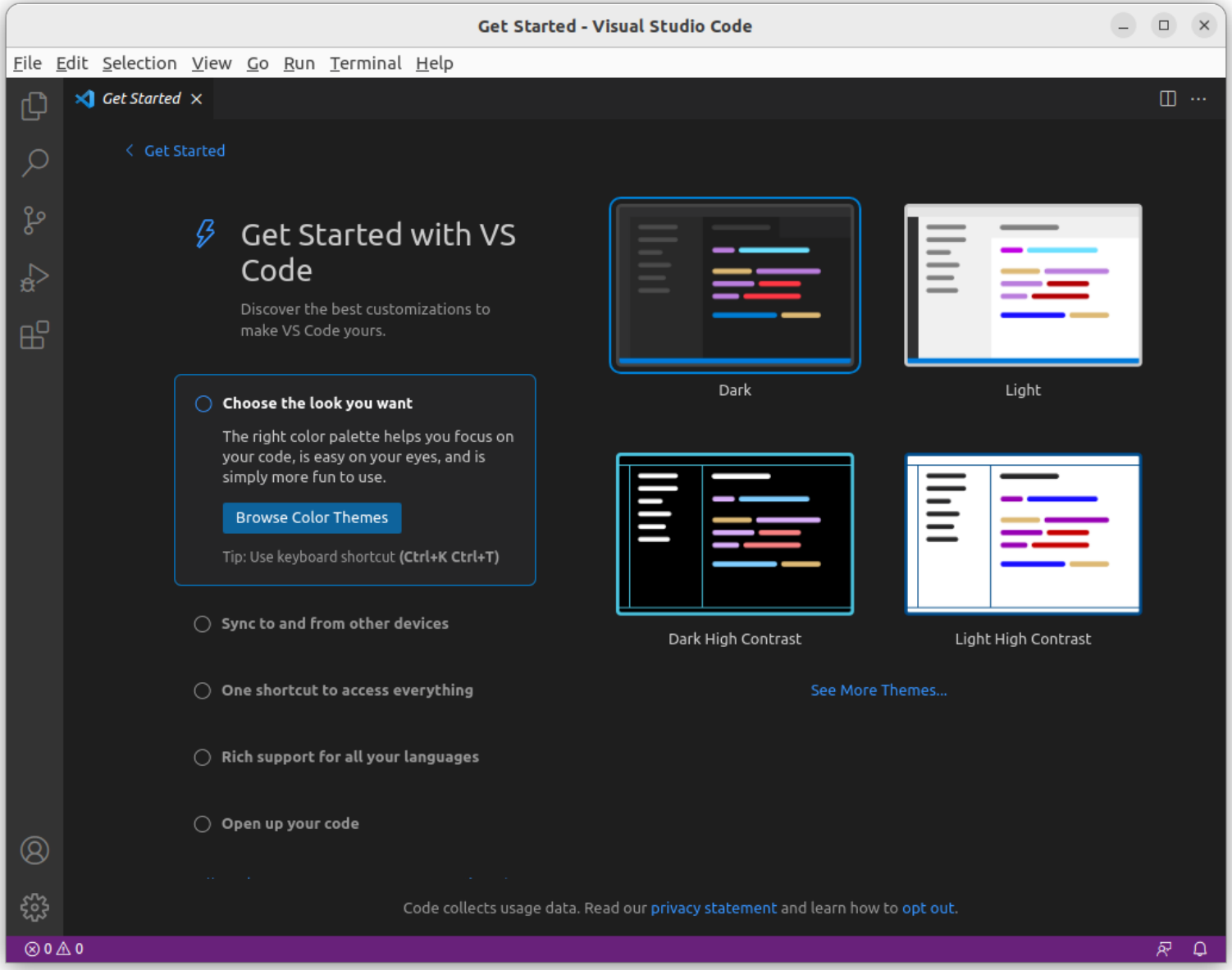
Execute an Ubuntu Update



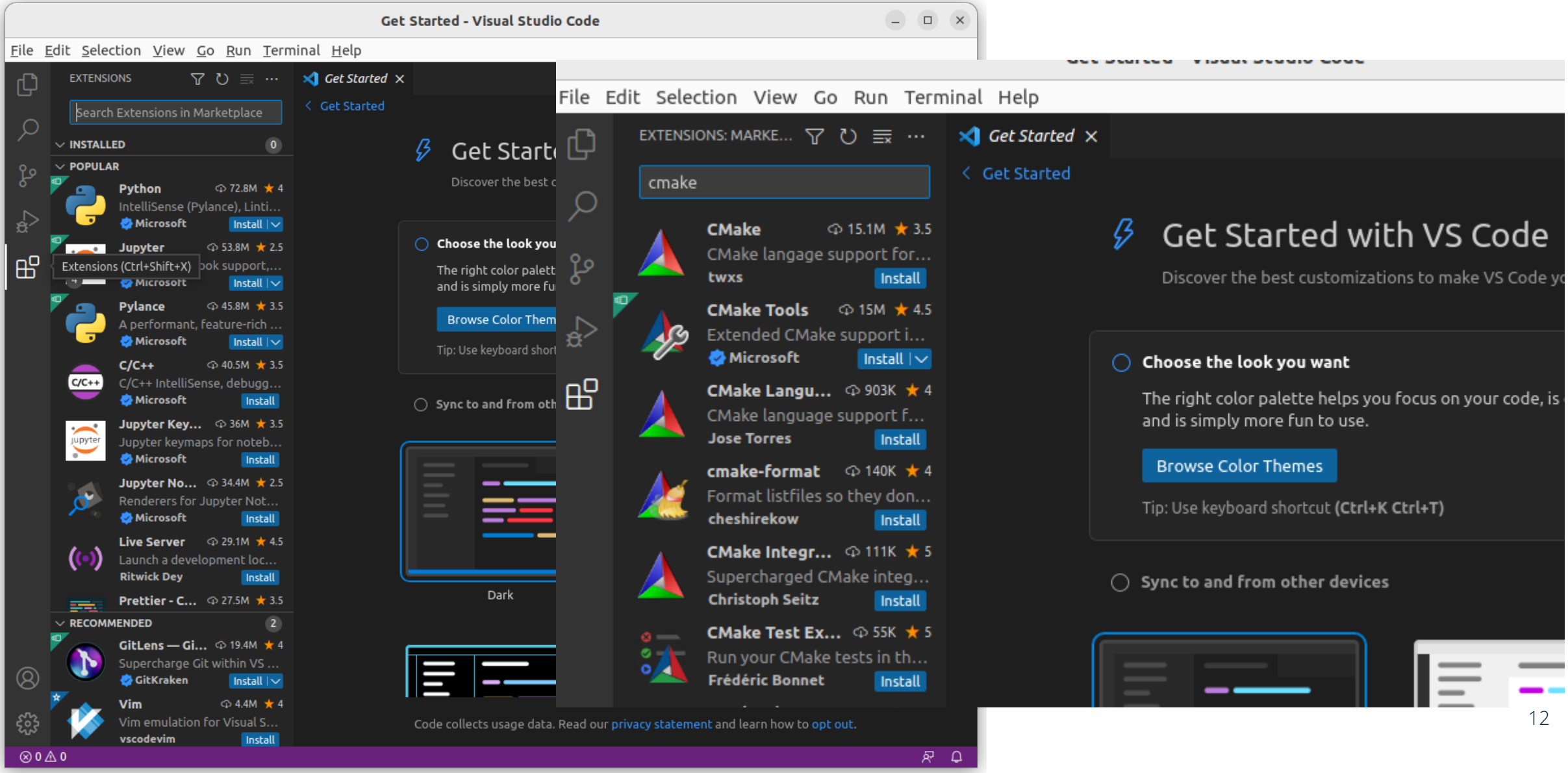
Install cmake – build-essential – libs and the GCC Toolchain



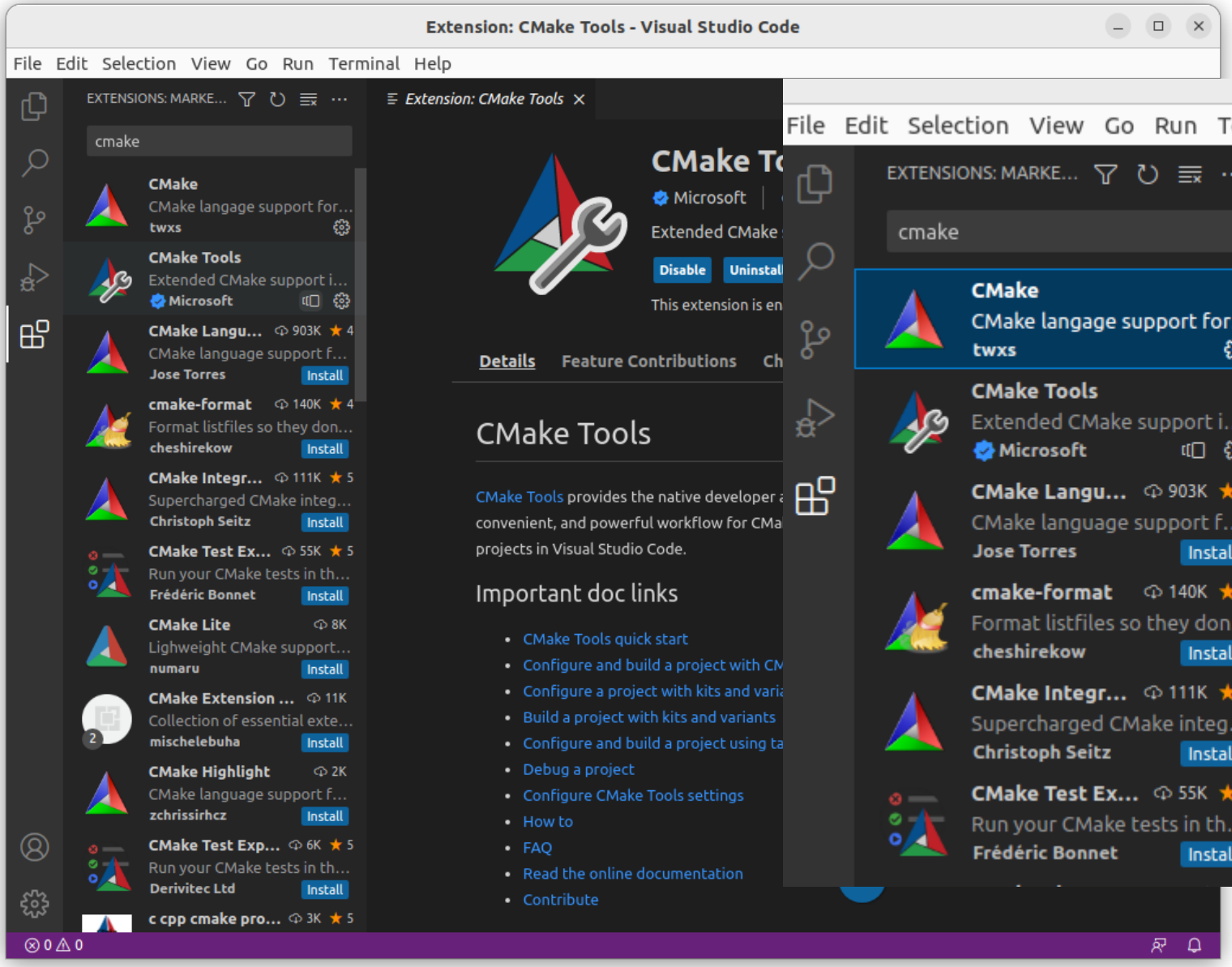
Download and Install Visual Studio Code

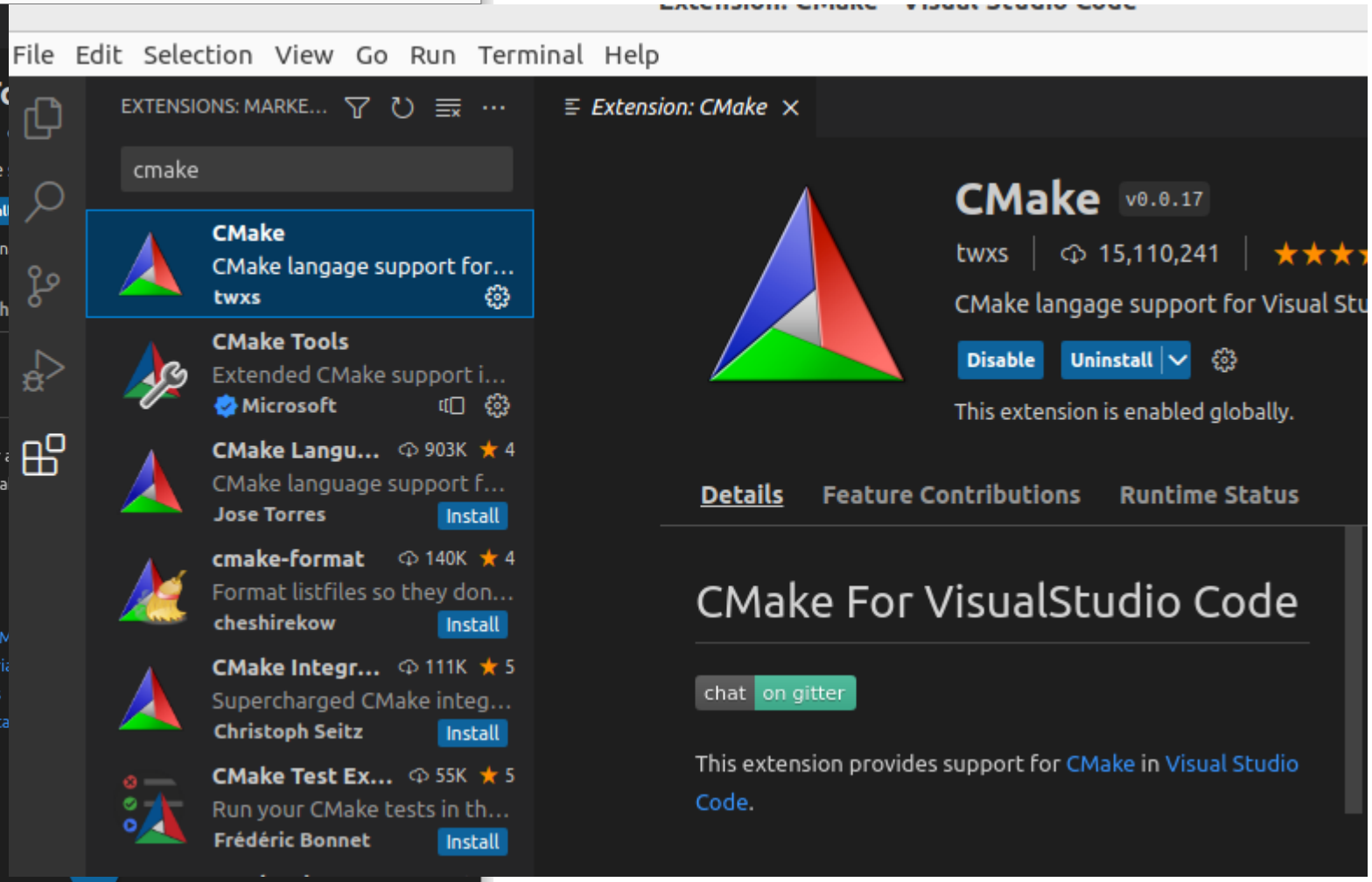


Install the CMake Tools Extension

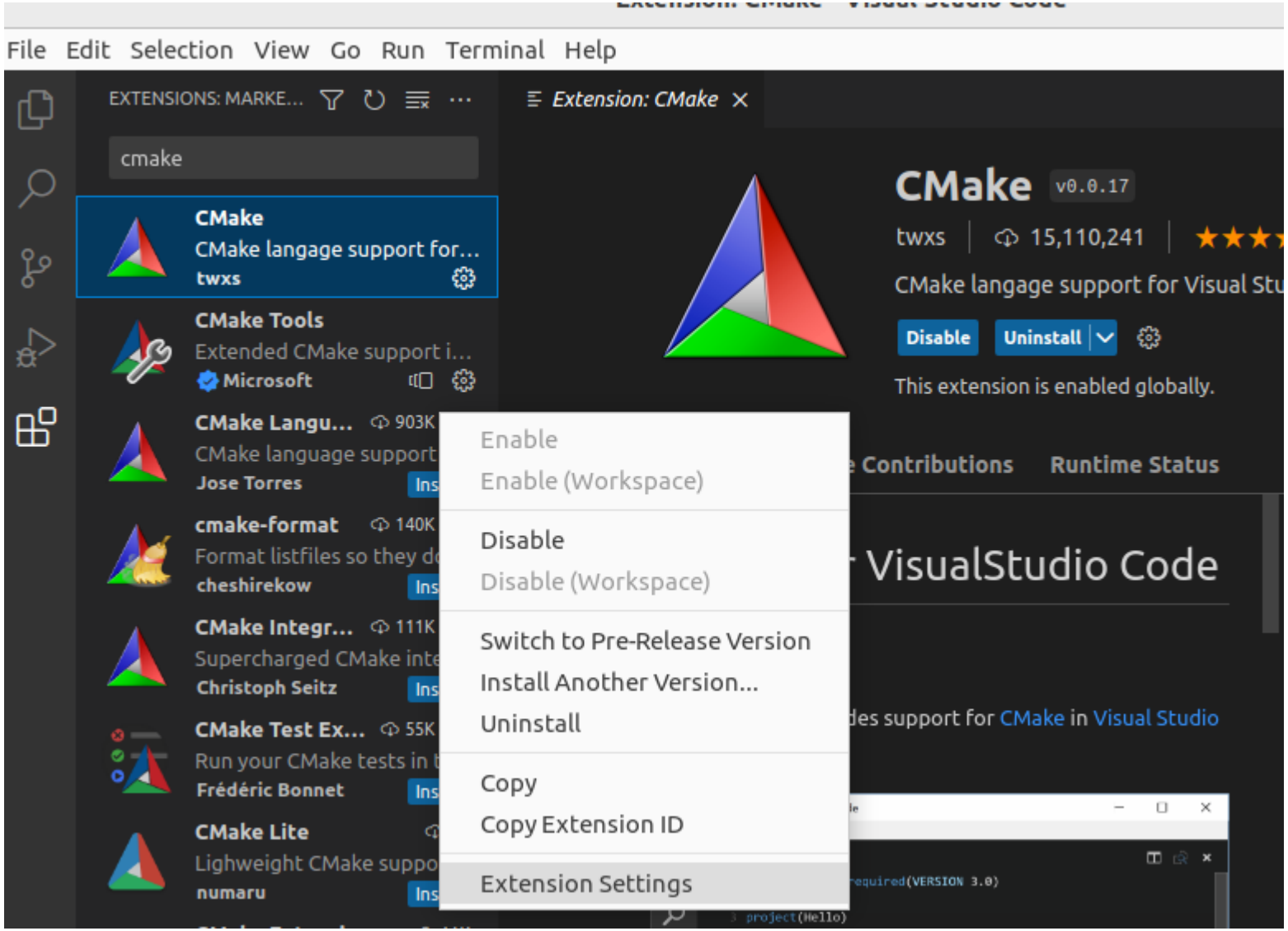


Install the CMake Tools Extension

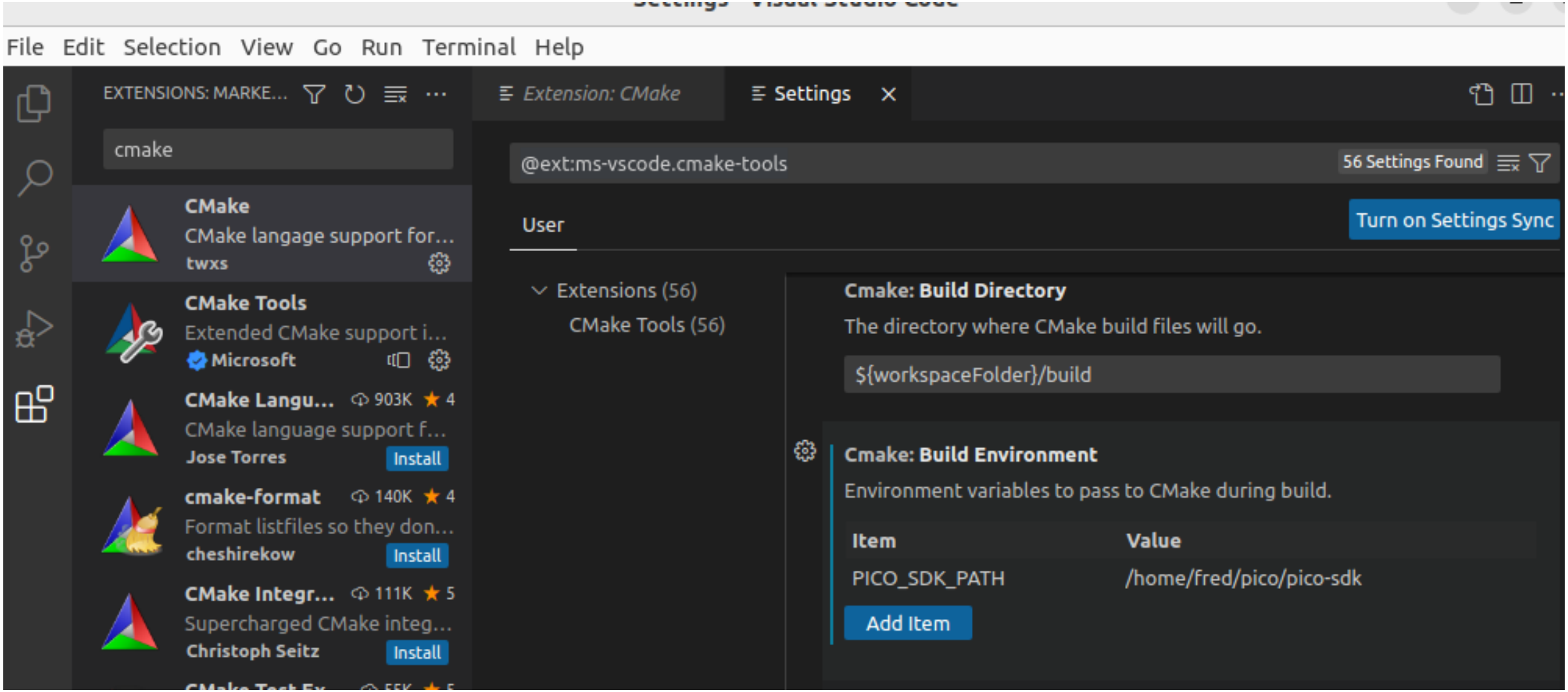




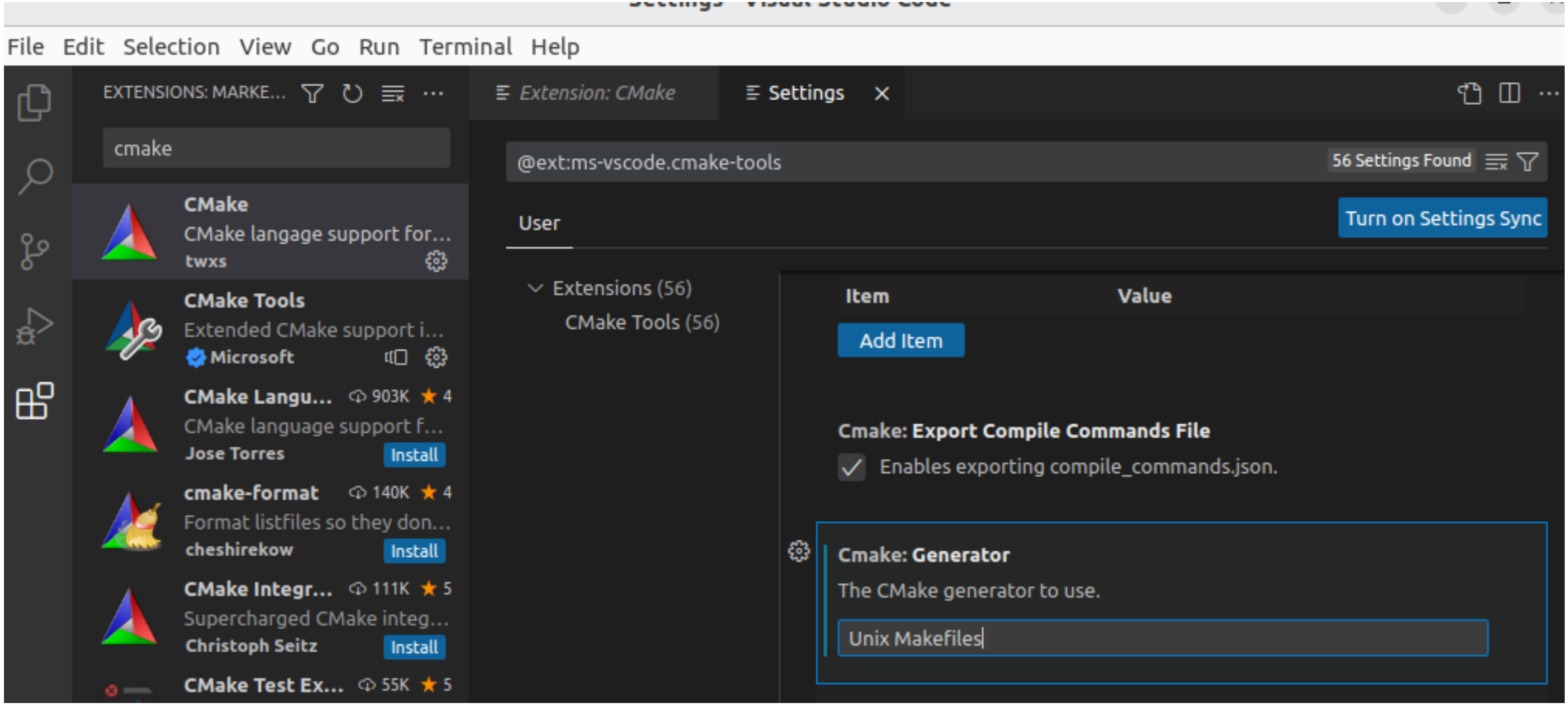
Configure the CMake Tools Extension



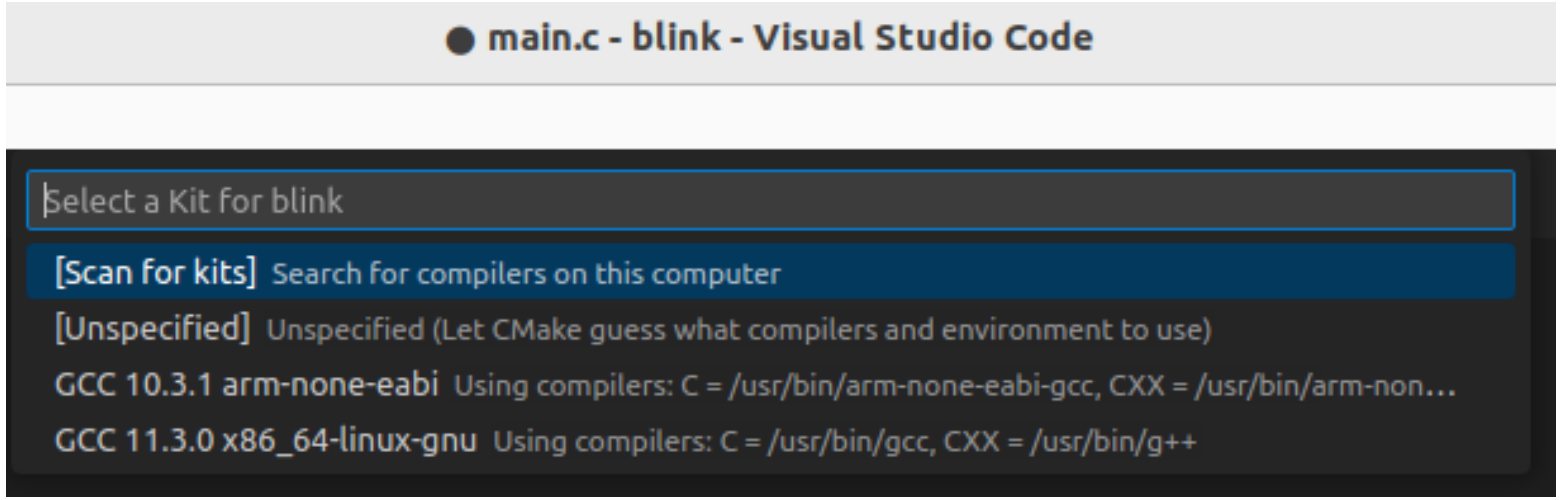
Configure the SDK Path

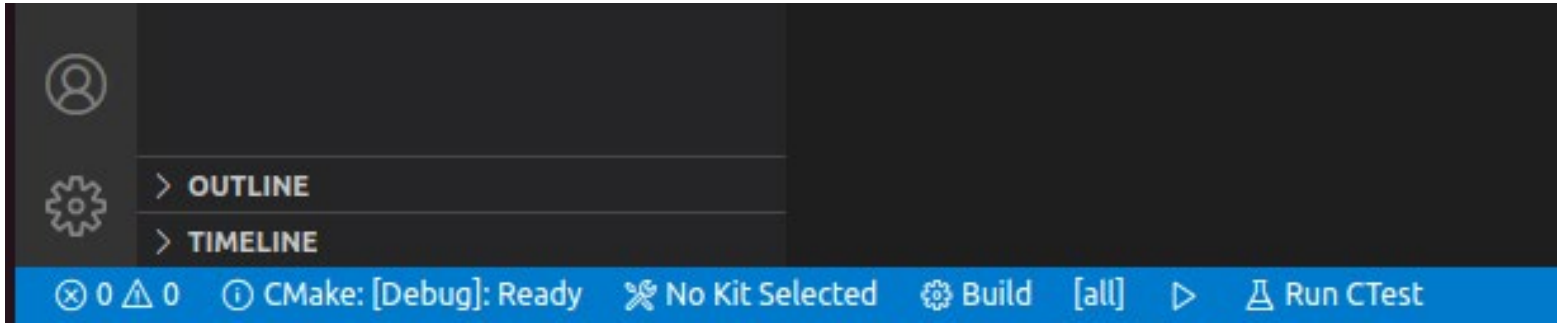


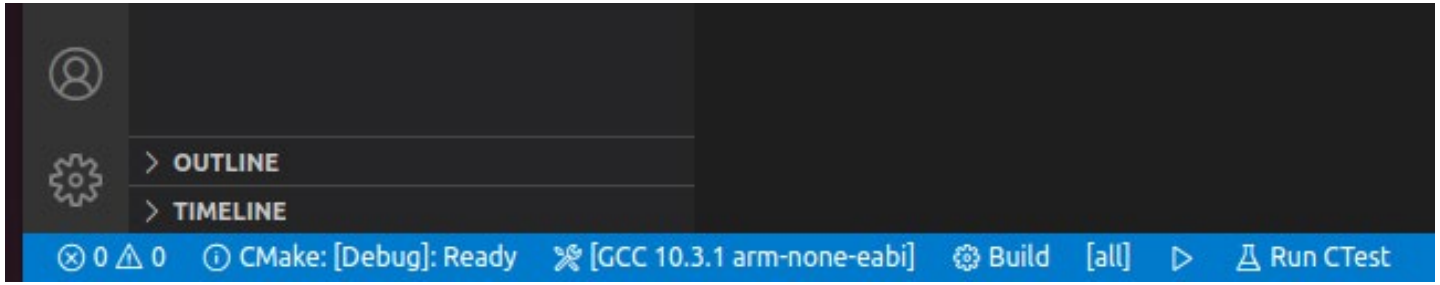
Specify Unix Makefiles



Discover and Select the Desired GCC Toolchain







Designate the SDK Paths

Open .bashrc ~/ Save

```
 86
 87 # colored GCC warnings and errors
 88 #export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01'
 89
 90 # some more ls aliases
 91 alias ll='ls -alF'
 92 alias la='ls -A'
 93 alias l='ls -CF'
 94
 95 # Add an "alert" alias for long running commands.  Use like so:
 96 #   sleep 10; alert
 97 alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || echo error)" "$
    (history|tail -n1|sed -e '\''s/^\s*[0-9]\+\s*//;s/[;&|]\s*alert$//'\'')"'
 98
 99 # Alias definitions.
100 # You may want to put all your additions into a separate file like
101 # ~/.bash_aliases, instead of adding them here directly.
102 # See /usr/share/doc/bash-doc/examples in the bash-doc package.
103
104 if [ -f ~/.bash_aliases ]; then
105     . ~/.bash_aliases
106 fi
107
108 # enable programmable completion features (you don't need to enable
109 # this, if it's already enabled in /etc/bash.bashrc and /etc/profile
110 # sources /etc/bash.bashrc).
111 if ! shopt -oq posix; then
112   if [ -f /usr/share/bash-completion/bash_completion ]; then
113     . /usr/share/bash-completion/bash_completion
114   elif [ -f /etc/bash_completion ]; then
115     . /etc/bash_completion
116   fi
117 fi
118
119 export PICO_SDK_PATH=/home/fred/pico/pico-sdk
120 export PICO_EXAMPLES_PATH=/home/fred/pico/pico-examples
121 export PICO_TOOLCHAIN_PATH=/home/fred/usr/bin
```

sh Tab Width: 8 Ln 1, Col 1 INS

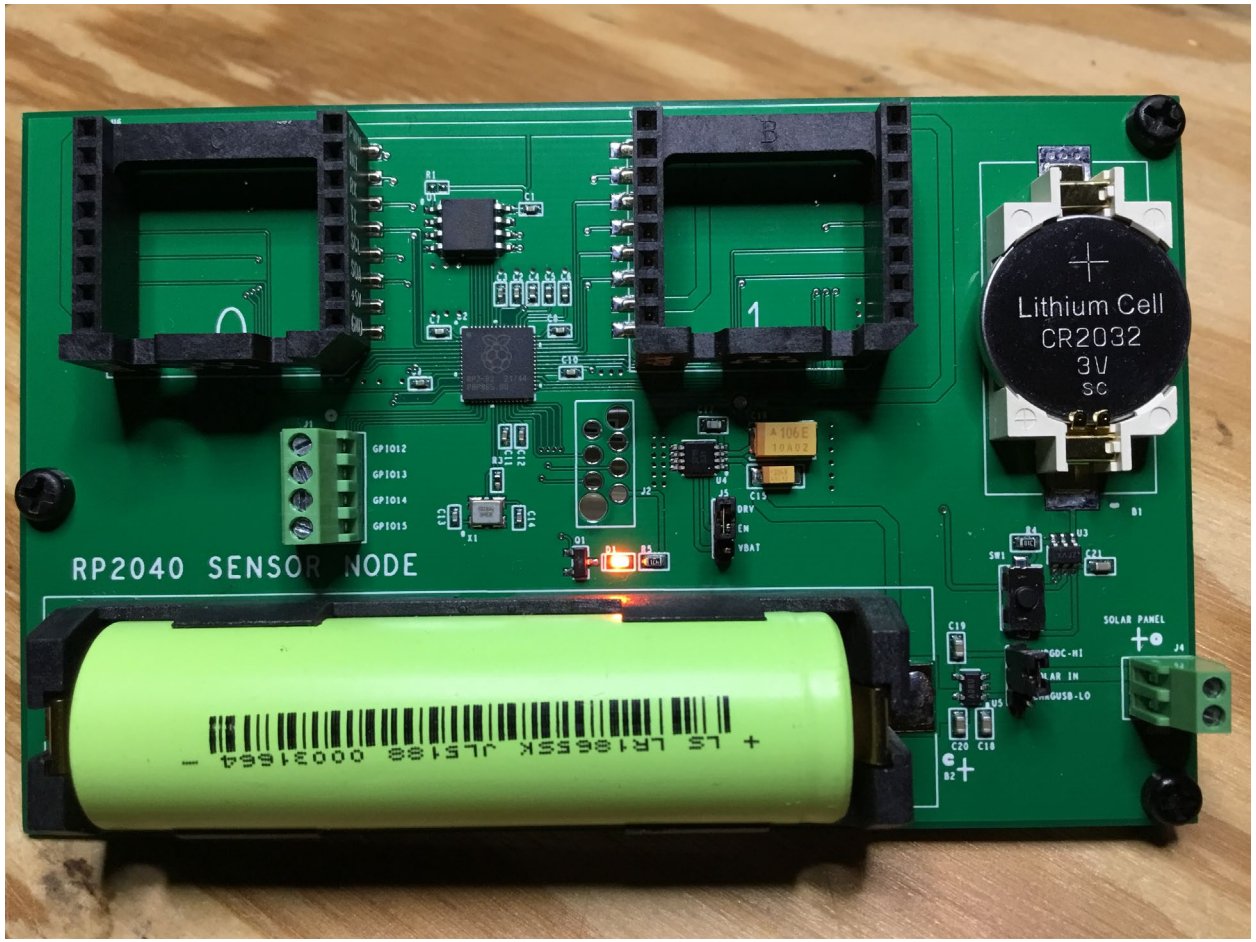
Known-Good Hardware

Test Fight Code – CMakeLists.txt

CMakeLists.txt | main.c

CMakeLists.txt

```
# Include build functions from Pico SDK
include($ENV{PICO_SDK_PATH}/external/pico_sdk_import.cmake)

# Set name of project (as PROJECT_NAME) and C/C++ standards
project(blink C CXX ASM)
set(CMAKE_C_STANDARD 11)
set(CMAKE_CXX_STANDARD 17)

# Creates a pico-sdk subdirectory in our project for the libraries
pico_sdk_init()

# Tell CMake where to find the executable source file
add_executable(${PROJECT_NAME}
    main.c
)

# Create map/bin/hex/uf2 files
pico_add_extra_outputs(${PROJECT_NAME})

# Link to pico_stdlib (gpio, time, etc. functions)
target_link_libraries(${PROJECT_NAME}
    pico_stdlib
)

# Enable usb output, enable uart output
pico_enable_stdio_usb(${PROJECT_NAME} 0)
pico_enable_stdio_uart(${PROJECT_NAME} 1)
```

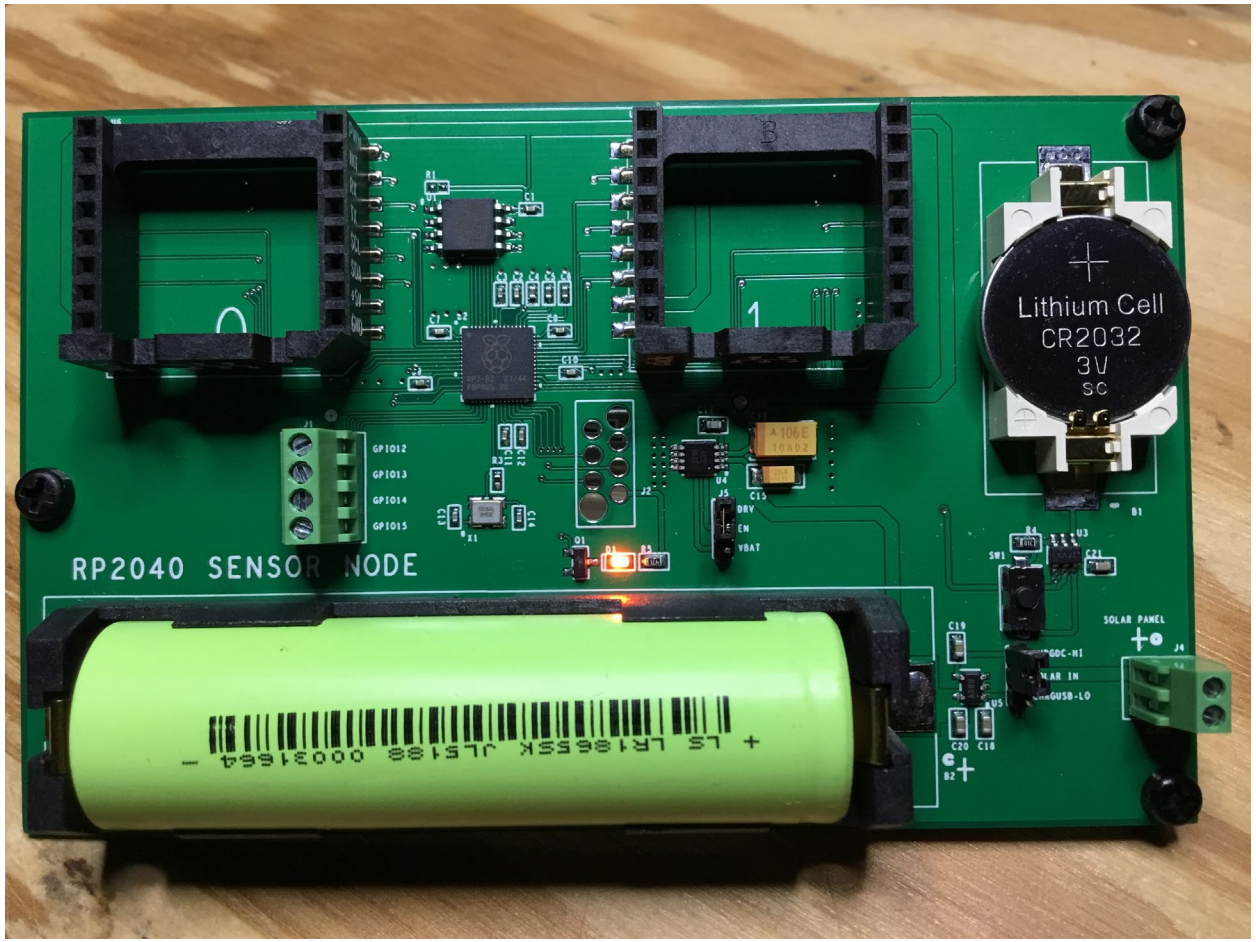


Test Fight Code - main.c

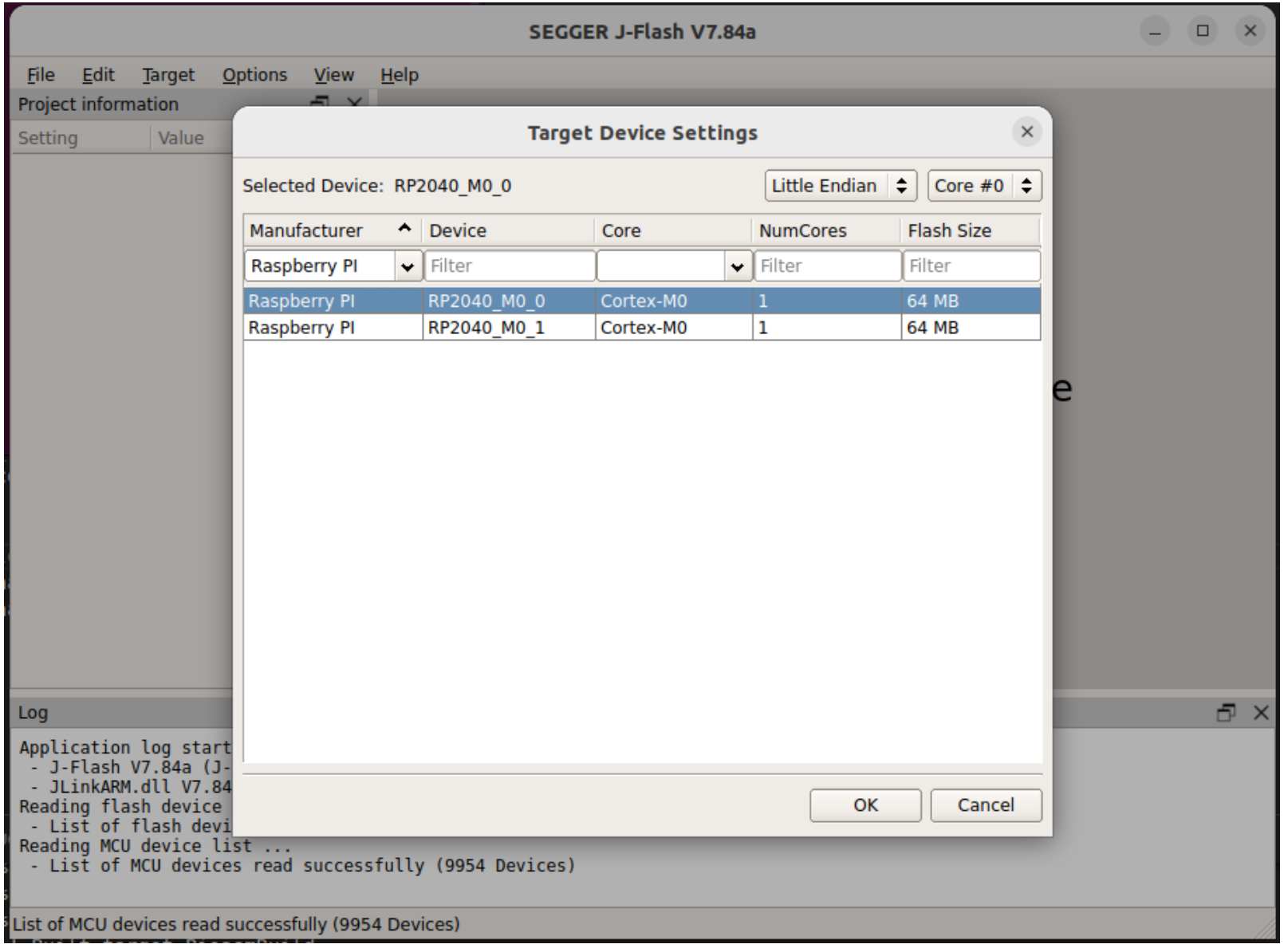
CMakeLists.txt | main.c

main.c

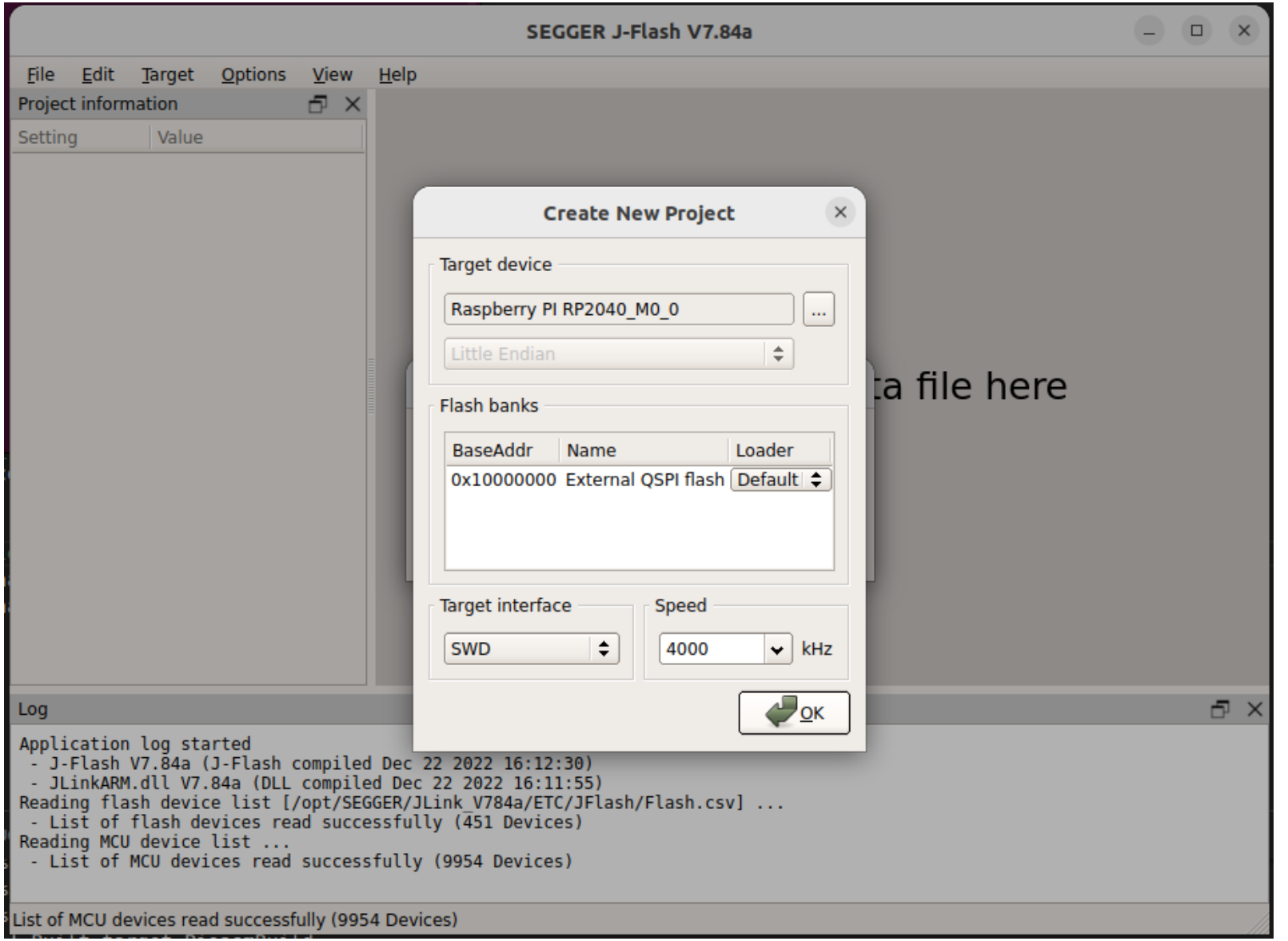
```c
#include <stdio.h>
#include "pico/stdlib.h"

int main() {

    const uint led_pin = 17;
    const uint tx0_pin = 0;
    const uint rx0_pin = 1;

    // Initialize LED pin
    gpio_init(led_pin);
    gpio_set_dir(led_pin, GPIO_OUT);
    // Initialize chosen serial port
    stdio_uart_init_full(uart0, 115200, tx0_pin, rx0_pin);

    // Loop forever
    while (true)
    {

        // Blink LED
        printf("RP2040 Low-Power Sensor Node Test Flight\r\n");
        gpio_put(led_pin, true);
        sleep_ms(500);
        gpio_put(led_pin, false);
        sleep_ms(500);
    }
}
```



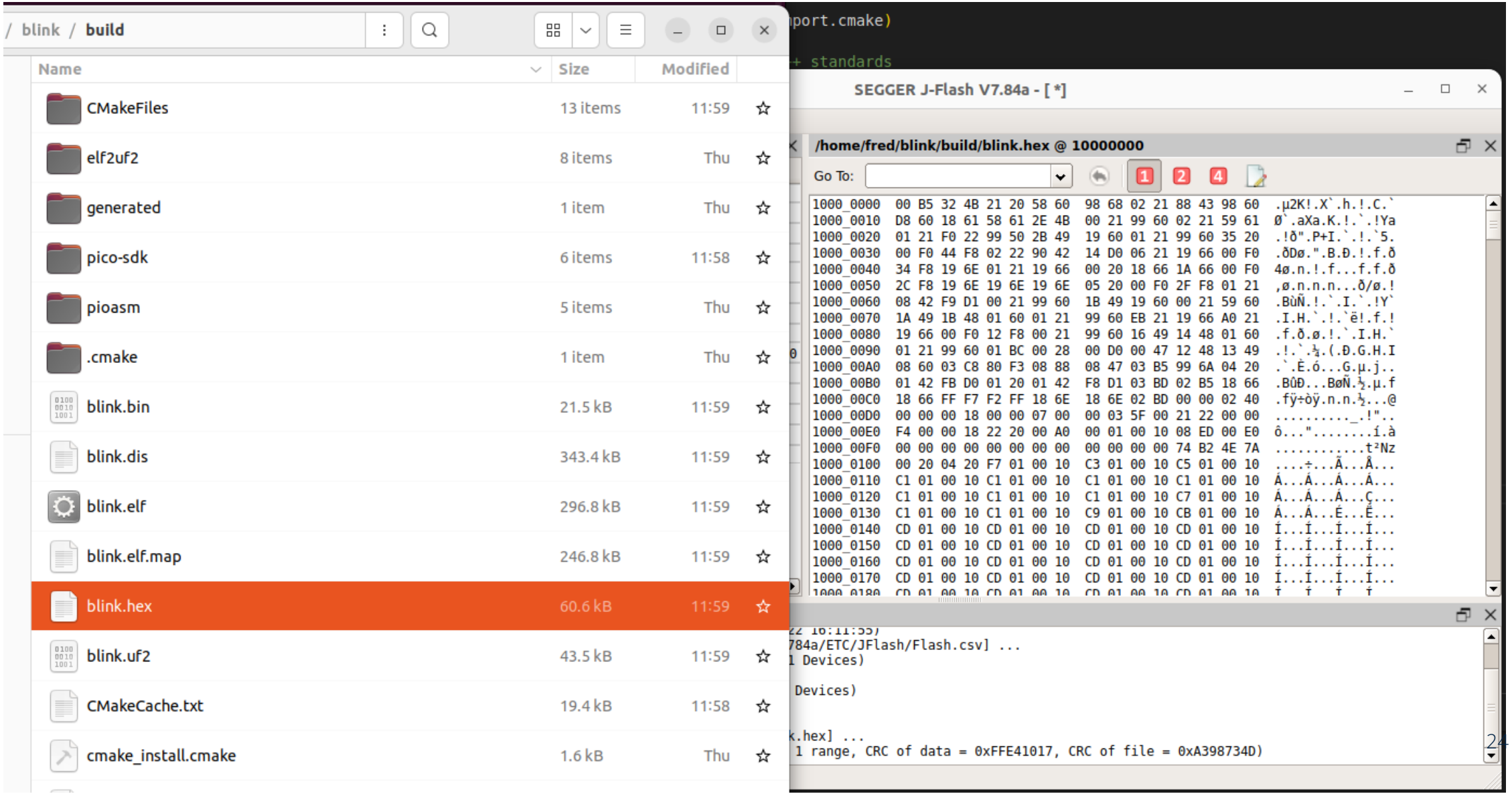
Fuel Up


SEGGER J-Flash V7.84a
File Edit Target Options View Help
Project information
Setting Value
Target Device Settings
Selected Device: RP2040_M0_0
Little Endian
Core #0
Manufacturer Device Core NumCores Flash Size
Raspberry PI Filter Filter Filter Filter
Raspberry PI RP2040_M0_0 Cortex-M0 1 64 MB
Raspberry PI RP2040_M0_1 Cortex-M0 1 64 MB
OK
Cancel
Log
Application log start
- J-Flash V7.84a (J-
- JLinkARM.dll V7.84
Reading flash device
- List of flash devi
Reading MCU device list ...
- List of MCU devices read successfully (9954 Devices)
List of MCU devices read successfully (9954 Devices)

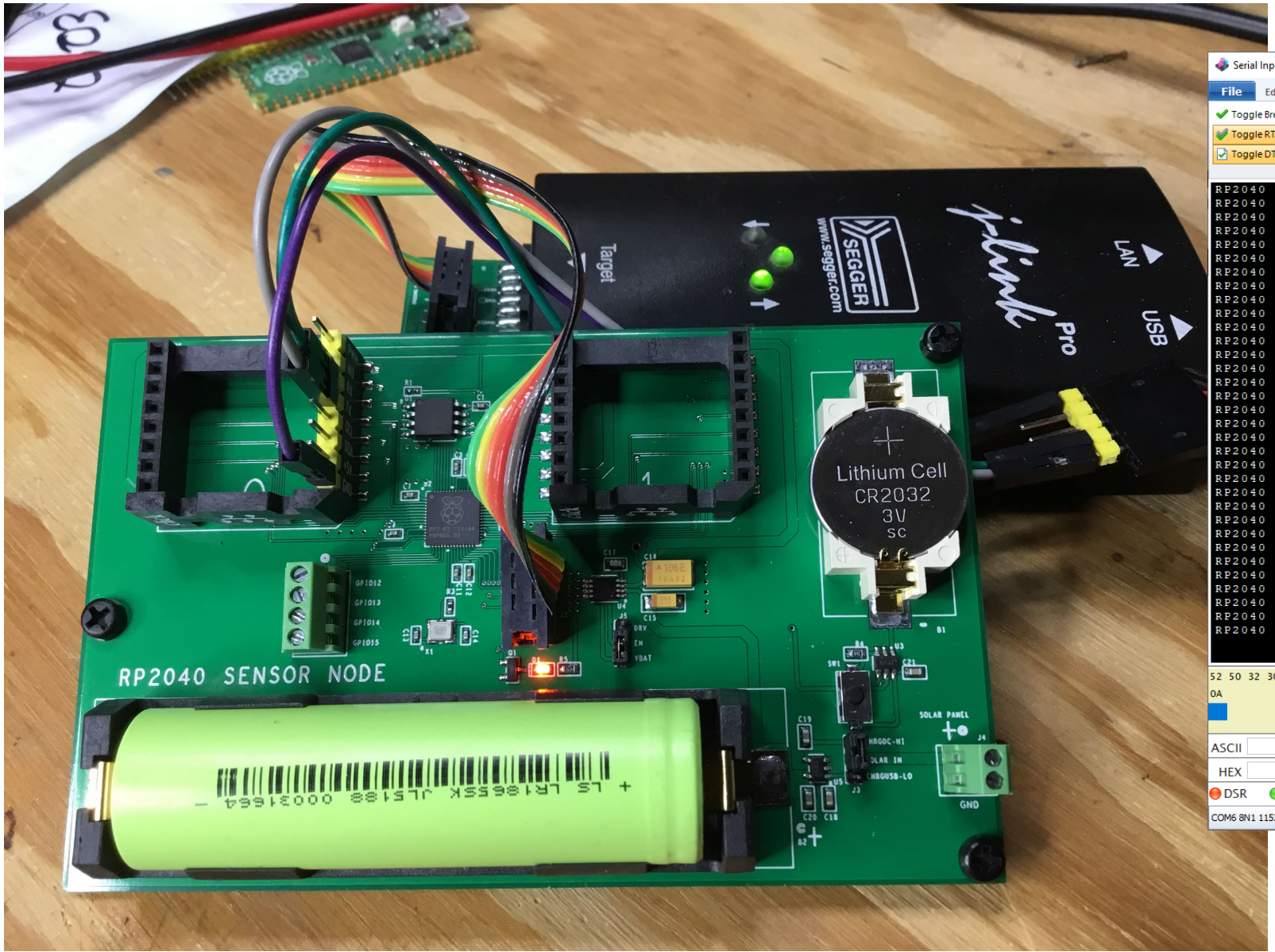
Engine Start

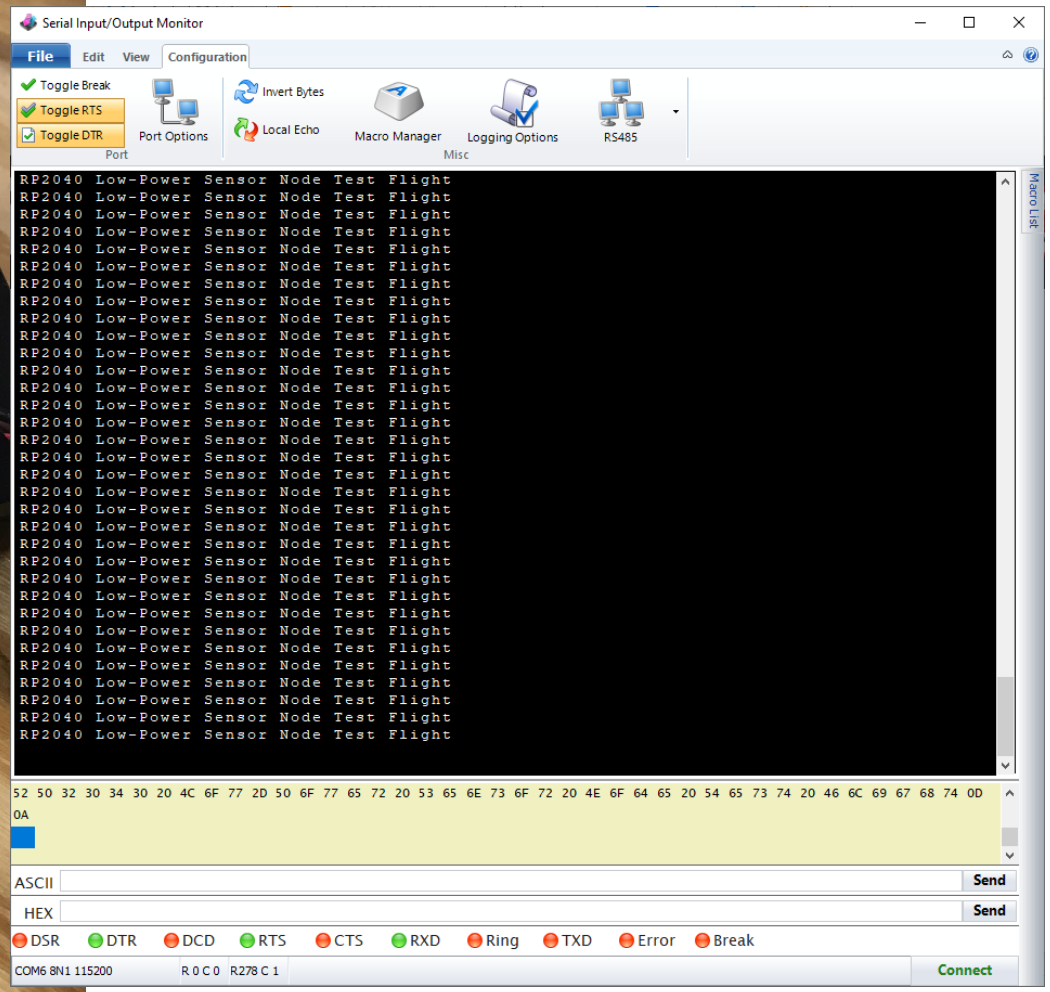


Takeoff Roll



Wheels UP!


Target
SEGGER
www.segger.com
J-Link
Pro
LAN
USB
Lithium Cell
CR2032
3V
sc
RP2040 SENSOR NODE
SOLAR PANEL
GND

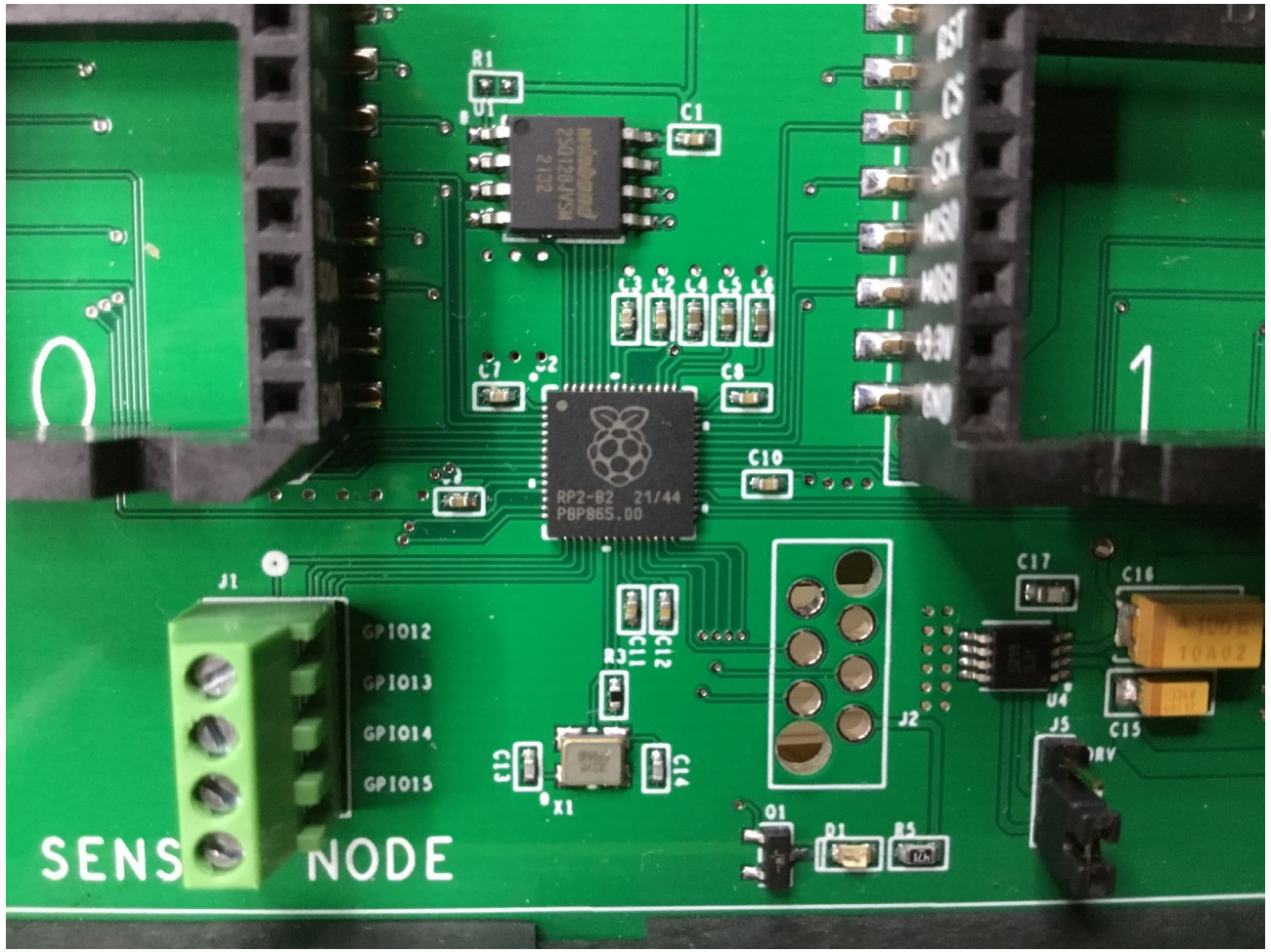

Serial Input/Output Monitor
File
Edit
View
Configuration
Toggle Break
Toggle RTS
Toggle DTR
Port
Port Options
Invert Bytes
Local Echo
Macro Manager
Logging Options
RS485
Misc
RP2040 Low-Power Sensor Node Test Flight
52 50 32 30 34 30 20 4C 6F 77 2D 50 6F 77 65 72 20 53 65 6E 73 6F 72 20 4E 6F 64 65 20 54 65 73 74 20 46 6C 69 67 68 74 0D 0A
ASCII
HEX
Send
DSR
DTR
DCD
RTS
CTS
RXD
Ring
TXD
Error
Break
COM6 8N1 115200
R 0 C 0
R278 C 1
Connect

Thank you for attending!!!

Please consider the resources below:

- raspberrypi.org
- RP2040 Data Sheet
- Raspberry Pi Pico C/C++ SDK

MORE TO COME..



CEC Continuing Education Center

DesignNews

Thank You

Sponsored by

Digi-Key ELECTRONICS

informa markets