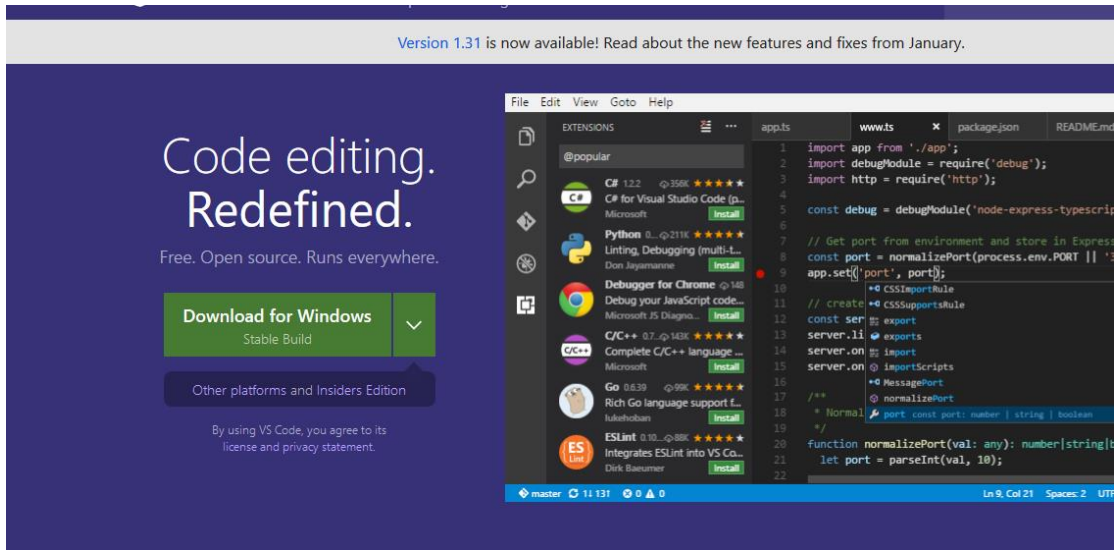


1、下载并安装 VSCode

下载地址: <https://code.visualstudio.com/>



2、下载 cmake

下载地址: <https://cmake.org/download/>

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.txt at the top of the source tree. See also the CMake 3.14 Release Notes. Source distributions:

Platform	Files
Unix/Linux Source (has \n line feeds)	cmake-3.14.0-rc2.tar.gz cmake-3.14.0-rc2.tar.Z
Windows Source (has \r\n line feeds)	cmake-3.14.0-rc2.zip

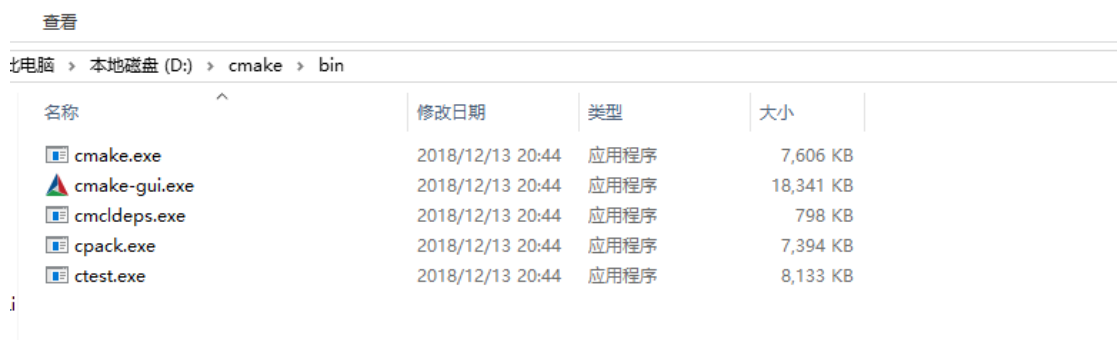
Binary distributions:

Platform	Files
Windows win64-x64 Installer: Installer tool has changed. Uninstall CMake 3.4 or lower first!	cmake-3.14.0-rc2-win64-x64.msi
Windows win64-x64 ZIP	cmake-3.14.0-rc2-win64-x64.zip
Windows win32-x86 Installer: Installer tool has changed. Uninstall CMake 3.4 or lower first!	cmake-3.14.0-rc2-win32-x86.msi
Windows win32-x86 ZIP	cmake-3.14.0-rc2-win32-x86.zip
Mac OS X 10.7 or later	cmake-3.14.0-rc2-Darwin-x86_64.dmg cmake-3.14.0-rc2-Darwin-x86_64.tar.gz
Linux x86_64	cmake-3.14.0-rc2-Linux-x86_64.sh cmake-3.14.0-rc2-Linux-x86_64.tar.gz

Download verification:

Role	Files
------	-------

解压重命名为 cmake,放到 D 盘根目录下



3、下载 K210 的 toolchain 和 openocd

下载地址: <https://github.com/kendryte/kendryte-gnu-toolchain/releases>

The screenshot shows the GitHub repository page for `kendryte / kendryte-gnu-toolchain`. The release `v8.2.0-20190409` is highlighted with a red box. Below the release information, the "Assets" section lists 10 files. The file `kendryte-toolchain-win-amd64-8.2.0-20190409.zip` is highlighted with a red box.

Release Information:

- Release: `v8.2.0-20190409`
- Author: `vowstar`
- Released on: 10 Apr
- Commits since release: 4

Version:

- Binutils 2.31.51
- GDB 8.2
- GCC 8.2.0

Build:

- Linux
- OSX
- Windows

Assets (10):

Asset Name	Size
<code>kendryte-toolchain-osx-mojave-8.2.0-20190409.tar.bz2</code>	46.3 MB
<code>kendryte-toolchain-osx-mojave-8.2.0-20190409.tar.xz</code>	21.4 MB
<code>kendryte-toolchain-ubuntu-amd64-8.2.0-20190409.tar.bz2</code>	45.6 MB
<code>kendryte-toolchain-ubuntu-amd64-8.2.0-20190409.tar.xz</code>	19.2 MB
<code>kendryte-toolchain-win-amd64-8.2.0-20190409.tar.xz</code>	18.2 MB
<code>kendryte-toolchain-win-amd64-8.2.0-20190409.zip</code>	51.4 MB
<code>kendryte-toolchain-win-i386-8.2.0-20190409.tar.xz</code>	16.8 MB
<code>kendryte-toolchain-win-i386-8.2.0-20190409.zip</code>	48.6 MB
Source code (zip)	
Source code (tar.gz)	

下载完成后，解压放到 D 盘根目录下。

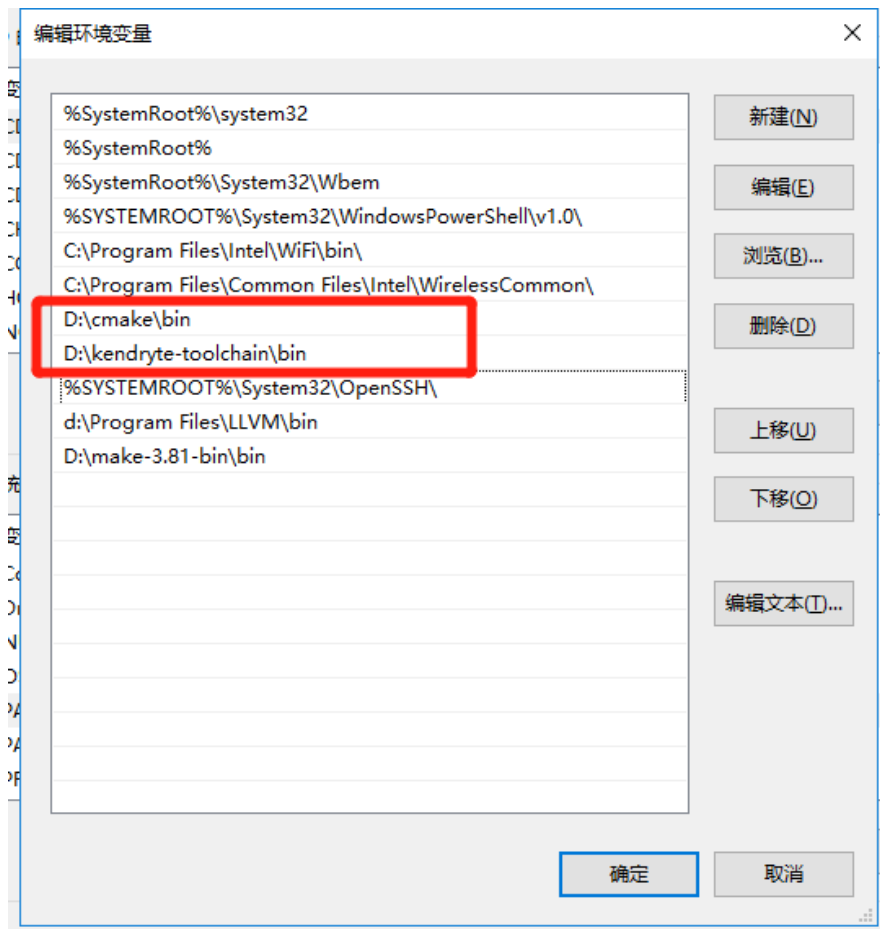
共享 查看

D:\kendryte-toolchain\bin

搜索"bin"

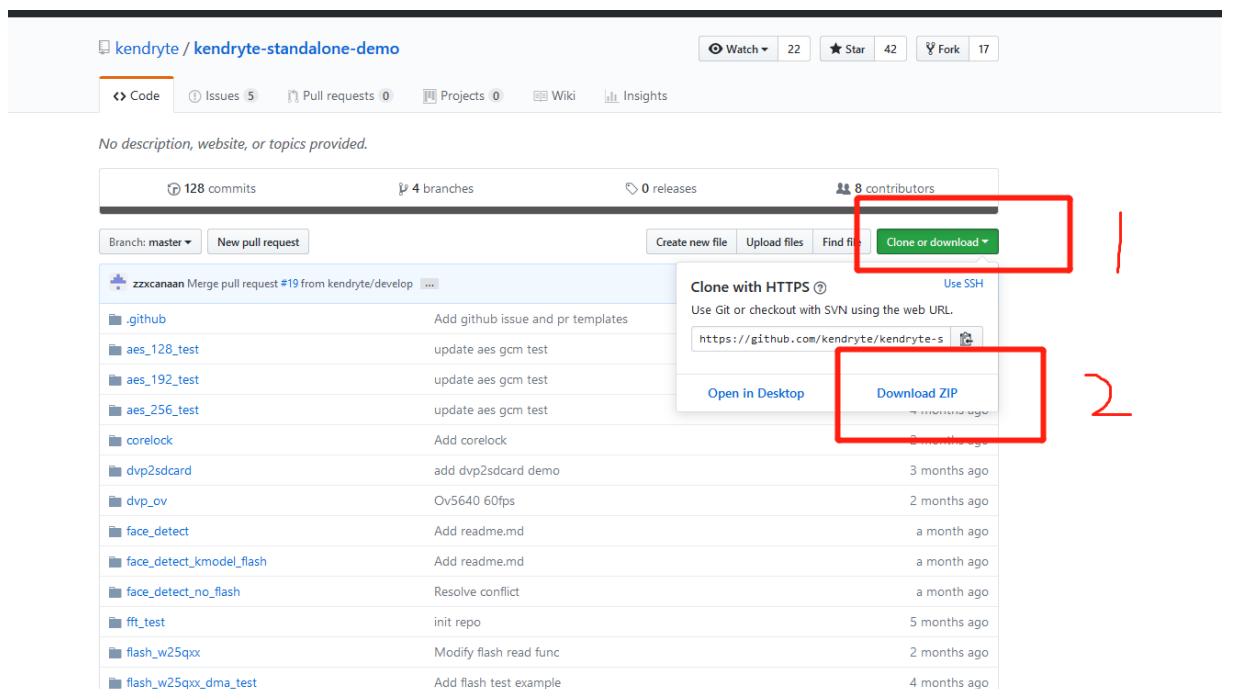
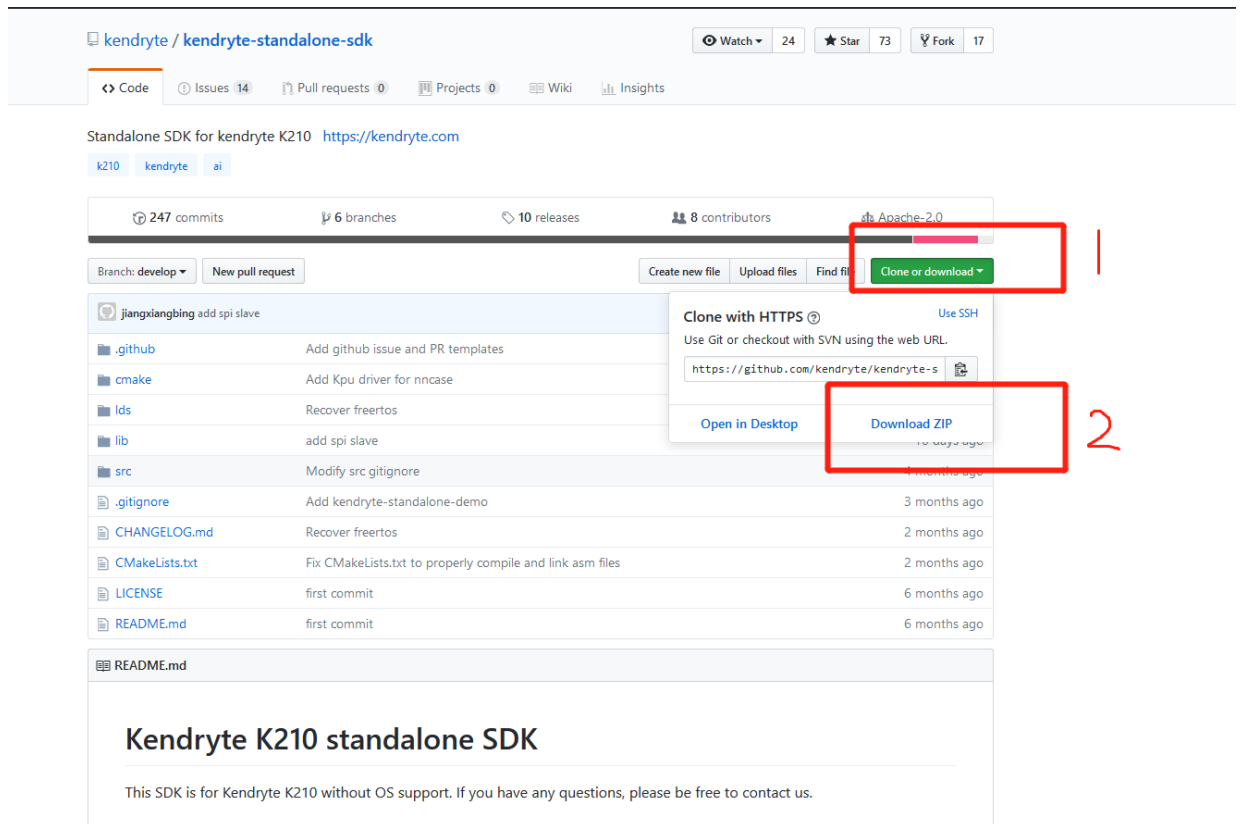
名称	修改日期	类型	大小
make.exe	2019/4/11 17:10	应用程序	214 KB
riscv64-unknown-elf-addr2line.exe	2019/4/11 17:10	应用程序	775 KB
riscv64-unknown-elf-ar.exe	2019/4/11 17:10	应用程序	797 KB
riscv64-unknown-elf-as.exe	2019/4/11 17:10	应用程序	1,048 KB
riscv64-unknown-elf-c++.exe	2019/4/11 17:10	应用程序	1,000 KB
riscv64-unknown-elf-c++filt.exe	2019/4/11 17:10	应用程序	772 KB
riscv64-unknown-elf-cpp.exe	2019/4/11 17:10	应用程序	998 KB
riscv64-unknown-elf-elfedit.exe	2019/4/11 17:10	应用程序	62 KB
riscv64-unknown-elf-g++.exe	2019/4/11 17:10	应用程序	1,000 KB
riscv64-unknown-elf-gcc.exe	2019/4/11 17:10	应用程序	997 KB
riscv64-unknown-elf-gcc-8.2.0.exe	2019/4/11 17:10	应用程序	997 KB
riscv64-unknown-elf-gcc-ar.exe	2019/4/11 17:10	应用程序	58 KB
riscv64-unknown-elf-gcc-nm.exe	2019/4/11 17:10	应用程序	58 KB
riscv64-unknown-elf-gcc-ranlib.exe	2019/4/11 17:10	应用程序	58 KB
riscv64-unknown-elf-gcov.exe	2019/4/11 17:10	应用程序	683 KB
riscv64-unknown-elf-gcov-dump.exe	2019/4/11 17:10	应用程序	561 KB
riscv64-unknown-elf-gcov-tool.exe	2019/4/11 17:10	应用程序	613 KB
riscv64-unknown-elf-gdb.exe	2019/4/11 17:10	应用程序	5,322 KB
riscv64-unknown-elf-gdb-add-index...	2019/4/11 15:34	应用程序	4 KB
riscv64-unknown-elf-gprof.exe	2019/4/11 17:10	应用程序	837 KB
riscv64-unknown-elf-ld.bfd.exe	2019/4/11 17:10	应用程序	1,168 KB
riscv64-unknown-elf-ld.exe	2019/4/11 17:10	应用程序	1,168 KB
riscv64-unknown-elf-nm.exe	2019/4/11 17:10	应用程序	786 KB

4、设置环境变量

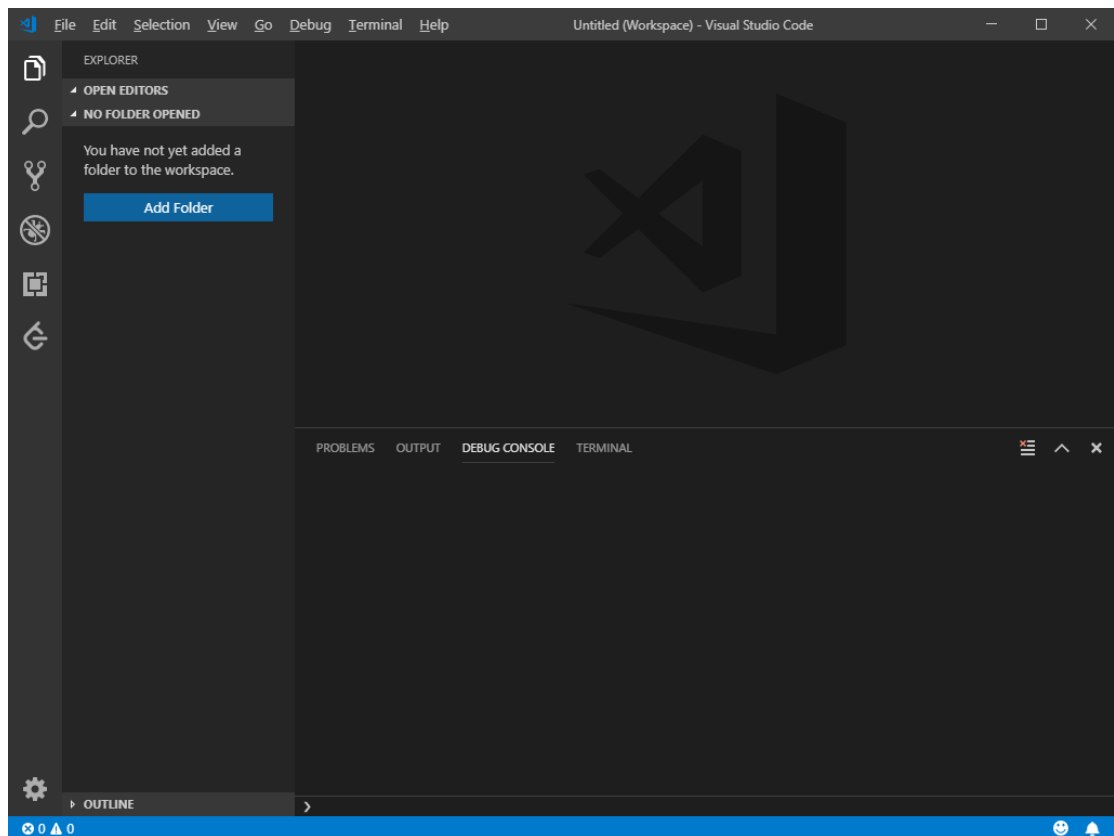


5、下载 K210 的最新 SDK 和 demo package

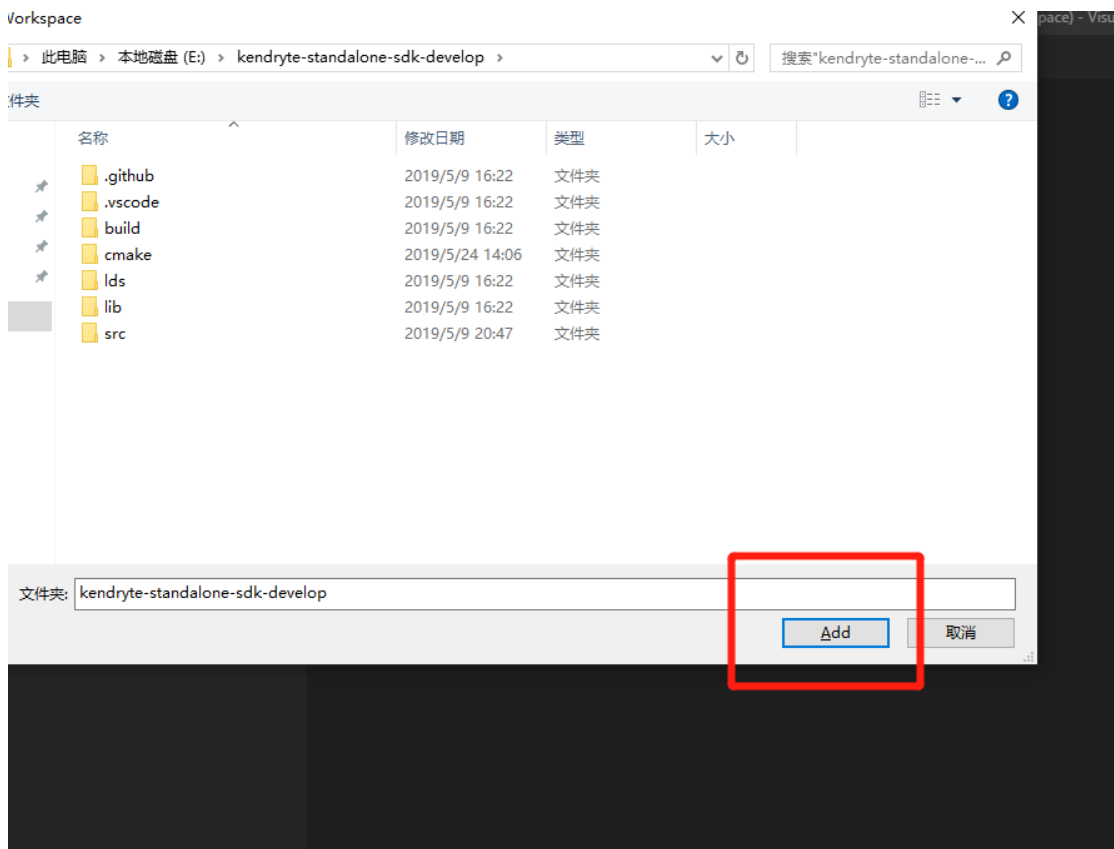
下载地址：<https://github.com/kendryte/kendryte-standalone-sdk>
<https://github.com/kendryte/kendryte-standalone-demo>



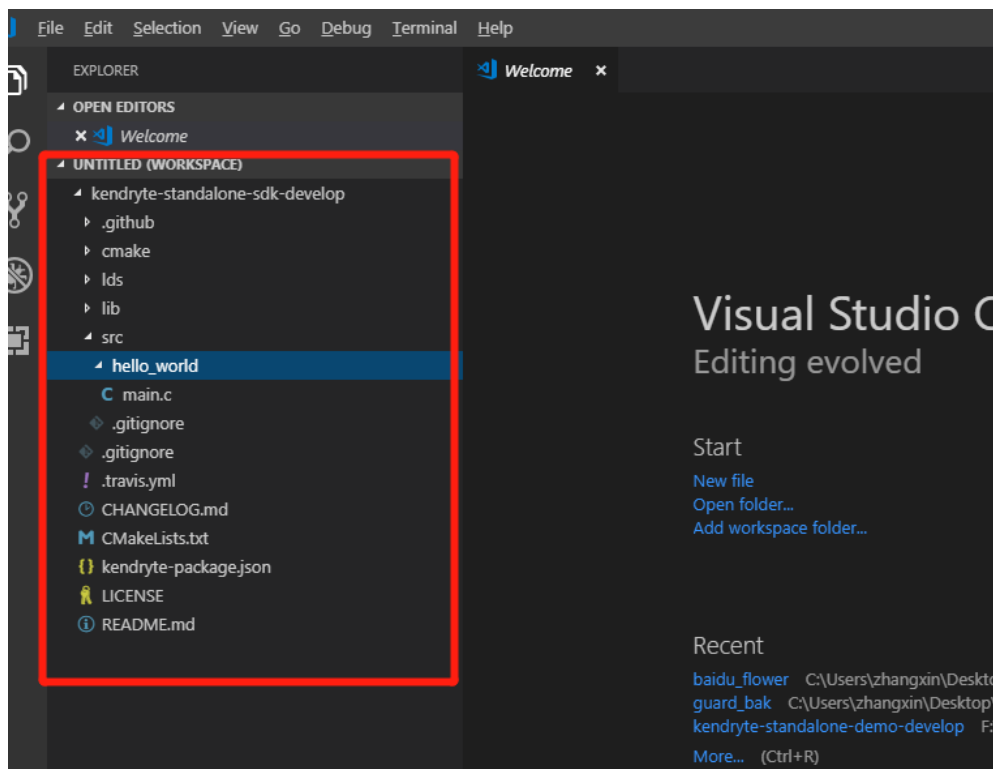
6、打开 VSCode



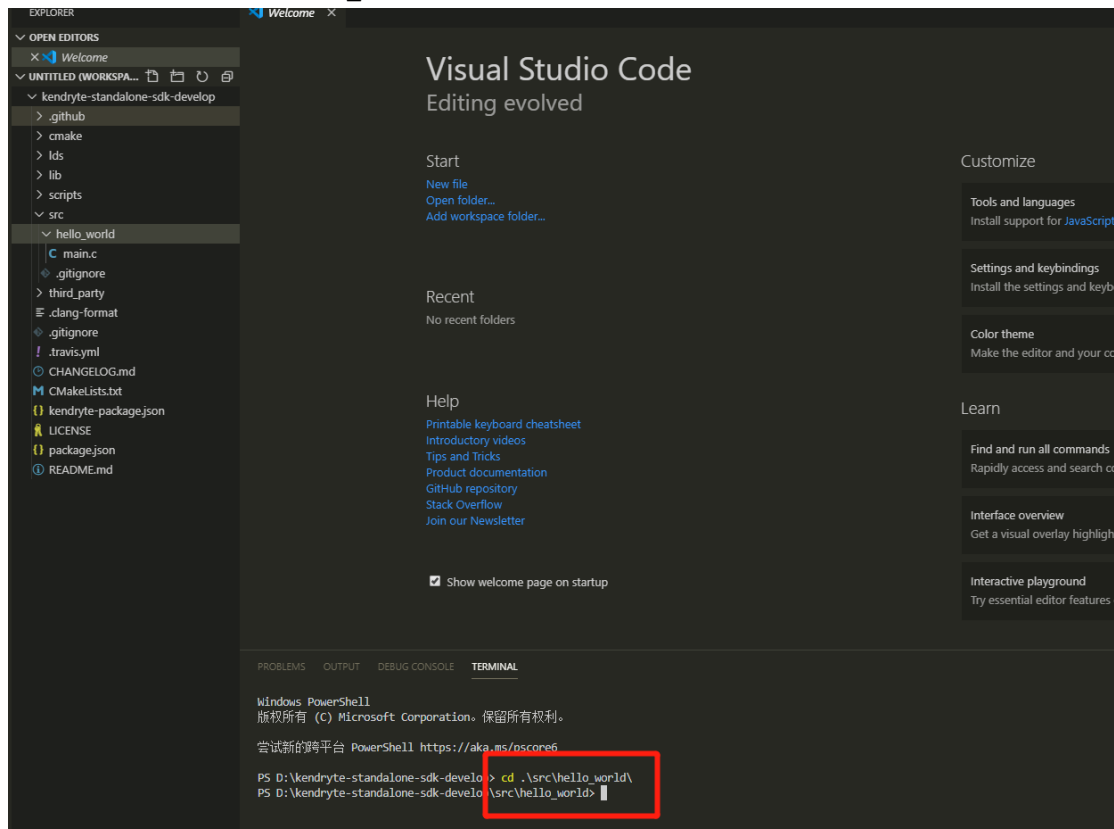
添加 kendryte-standalone-sdk-develop 文件夹到 VSCode 中,点击 “Add”



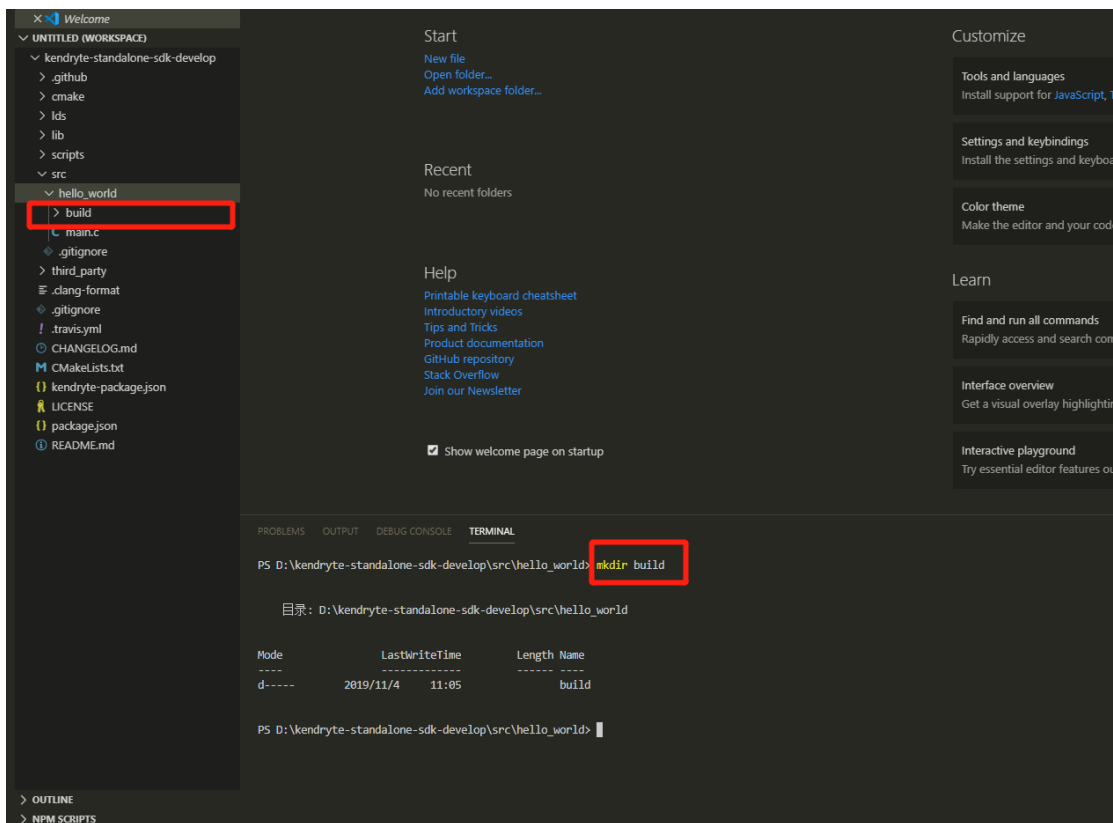
成功添加工程后的界面



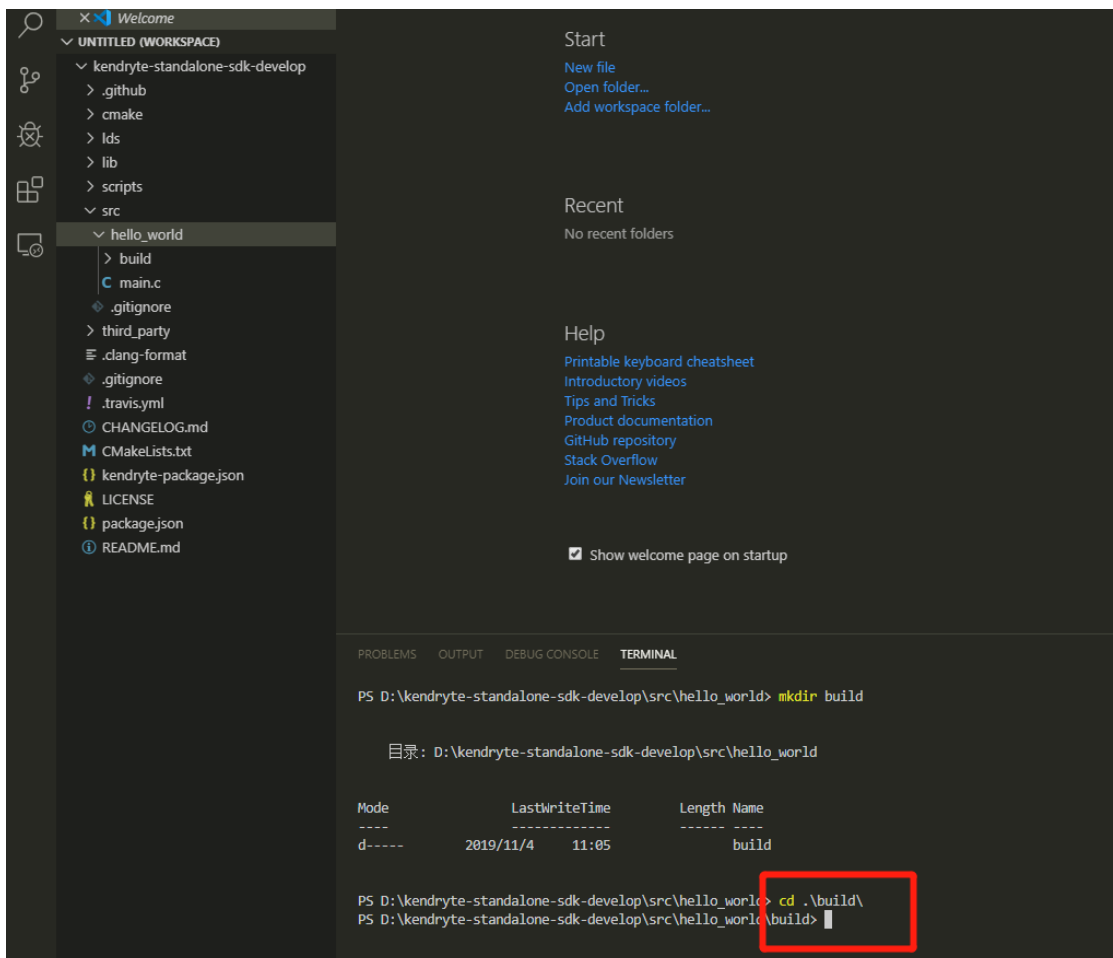
按键盘 `ctrl+shift+~`, 下方出现命令终端,
使用 `cd` 命令进入 `/src/hello_world` 文件夹内



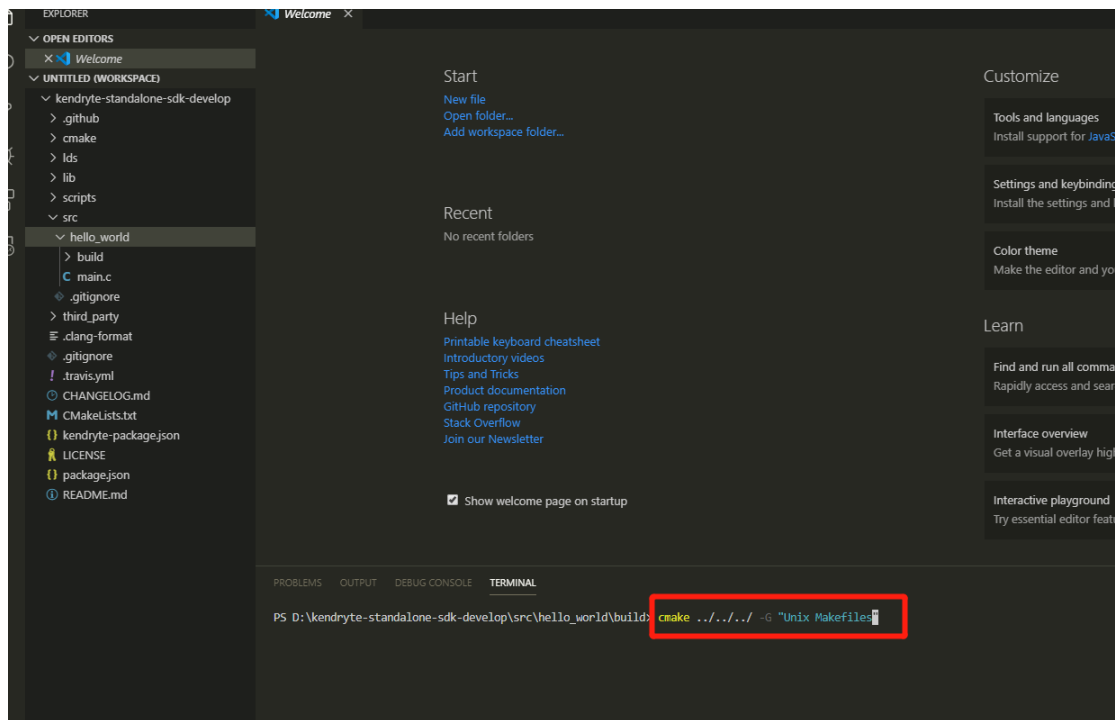
输入命令 `mkdir build` , 回车, 创建 `build` 文件夹



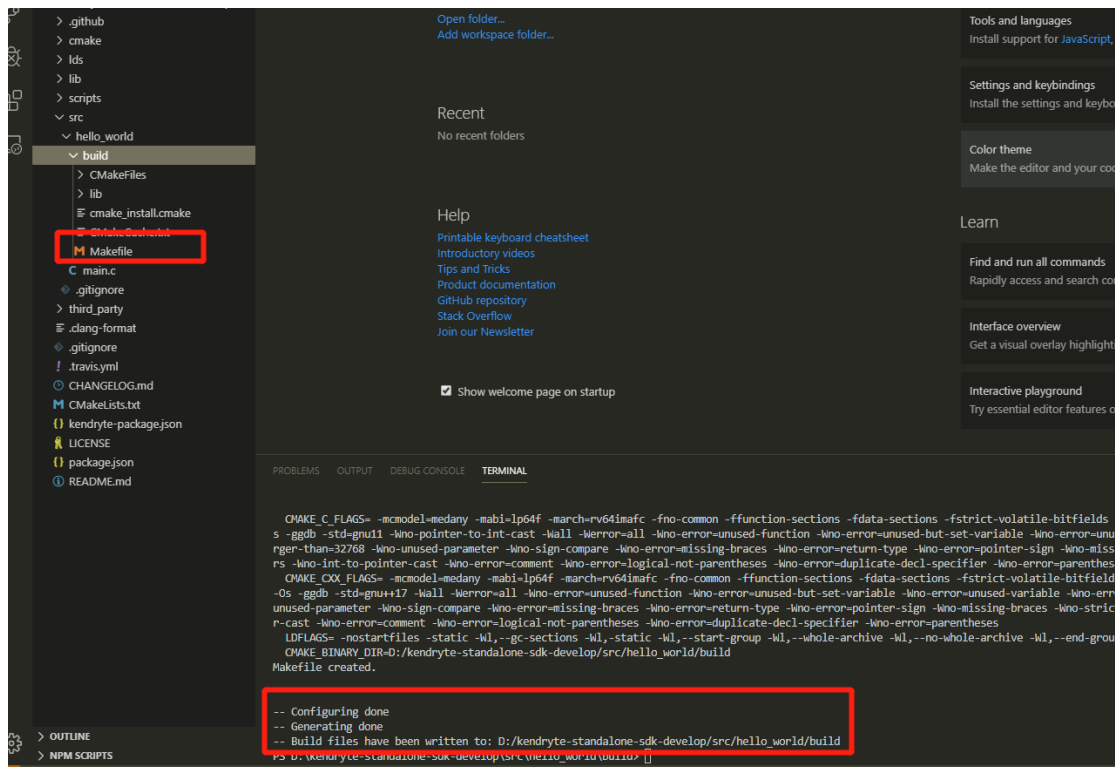
cd build 回车，进入 build 文件夹内。



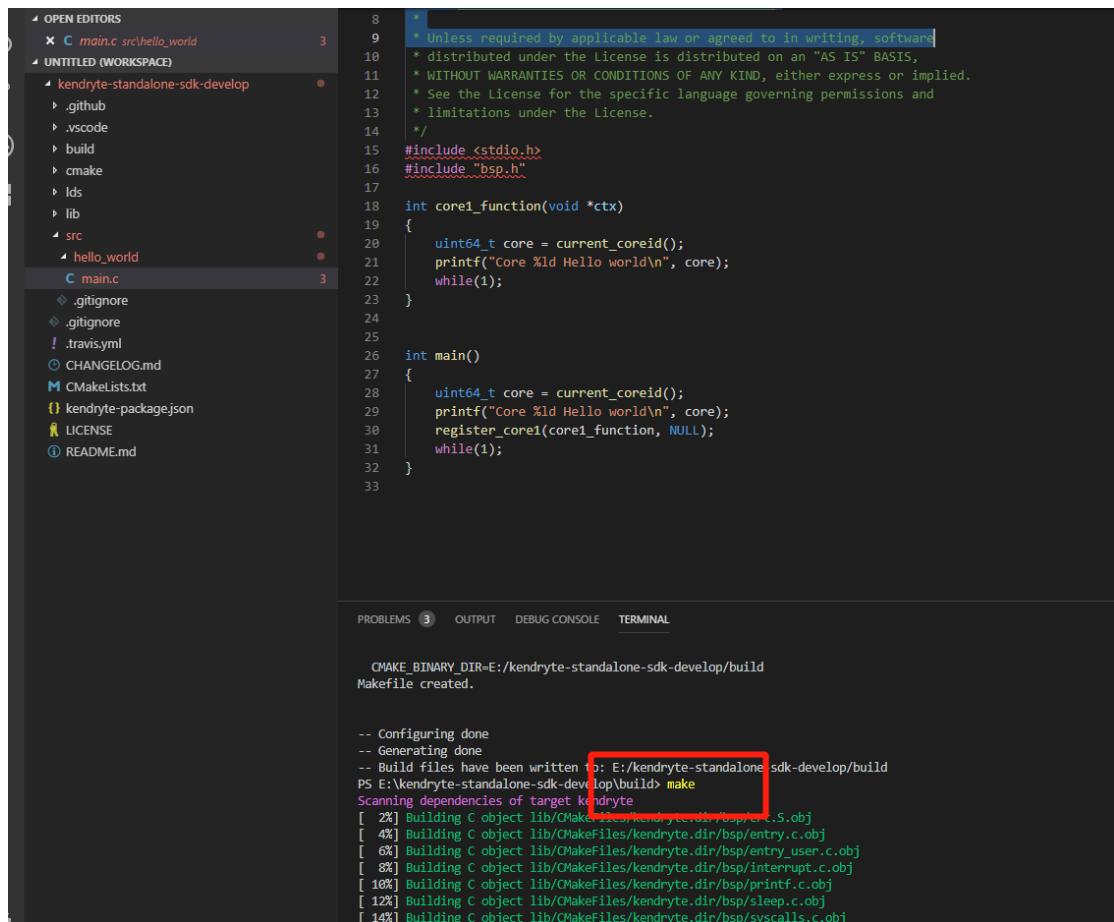
输入 `cmake ../../.. -G "Unix Makefiles"`
回车



此时，在 `build` 目录，`makefile` 文件已创建。



继续输入命令 make,回车



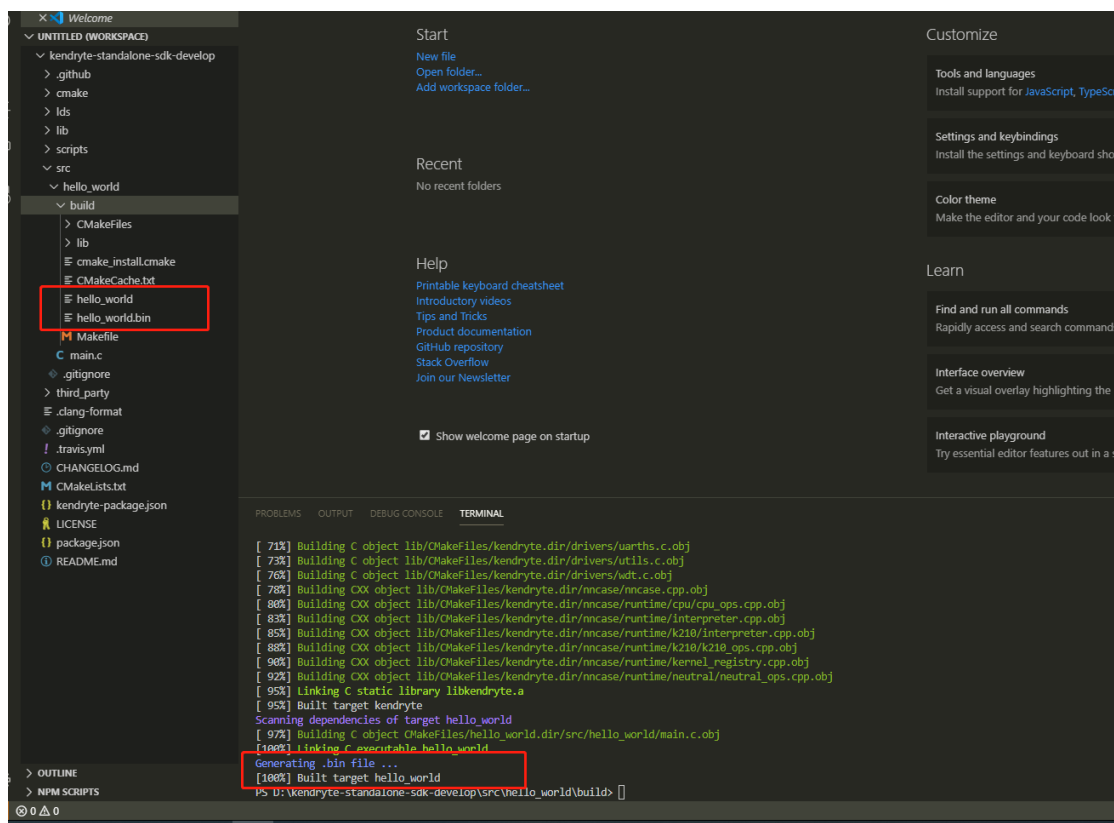
The screenshot shows the VS Code editor with the file explorer on the left displaying the project structure. The main.c file is open in the editor, showing a C program with a core1_function and a main function. The terminal at the bottom shows the output of the make command, including the generation of the Makefile and the compilation of various object files.

```
8 *
9 * Unless required by applicable law or agreed to in writing, software
10 * distributed under the License is distributed on an "AS IS" BASIS,
11 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
12 * See the License for the specific language governing permissions and
13 * limitations under the License.
14 */
15 #include <stdio.h>
16 #include "bsp.h"
17
18 int core1_function(void *ctx)
19 {
20     uint64_t core = current_coreid();
21     printf("Core %ld Hello world\n", core);
22     while(1);
23 }
24
25
26 int main()
27 {
28     uint64_t core = current_coreid();
29     printf("Core %ld Hello world\n", core);
30     register_core1(core1_function, NULL);
31     while(1);
32 }
33
```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL

CMAKE_BINARY_DIR=E:/kendryte-standalone-sdk-develop/build
Makefile created.

-- Configuring done
-- Generating done
-- Build files have been written to: E:/kendryte-standalone-sdk-develop/build
PS E:/kendryte-standalone-sdk-develop/build> make
Scanning dependencies of target kendryte
[2%] Building C object lib/CMakeFiles/kendryte.dir/bsp/uart.c.obj
[4%] Building C object lib/CMakeFiles/kendryte.dir/bsp/entry.c.obj
[6%] Building C object lib/CMakeFiles/kendryte.dir/bsp/entry_user.c.obj
[8%] Building C object lib/CMakeFiles/kendryte.dir/bsp/interrupt.c.obj
[10%] Building C object lib/CMakeFiles/kendryte.dir/bsp/printf.c.obj
[12%] Building C object lib/CMakeFiles/kendryte.dir/bsp/sleep.c.obj
[14%] Building C object lib/CMakeFiles/kendryte.dir/bsp/syscalls.c.obj



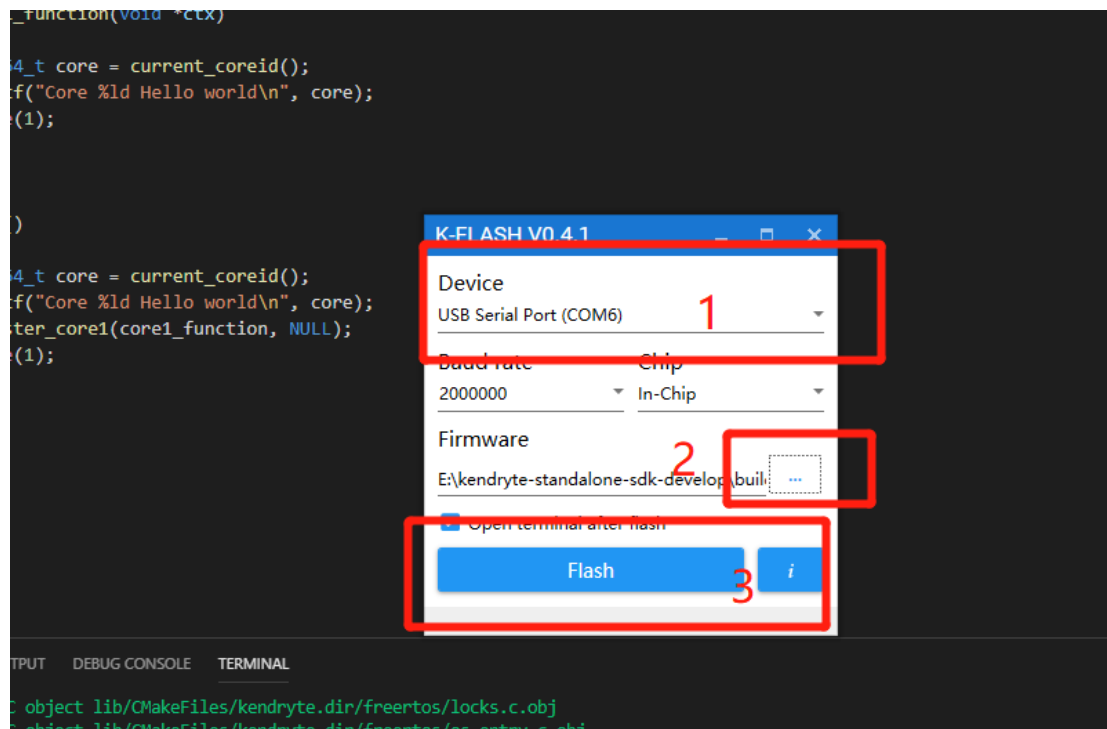
The screenshot shows the VS Code editor with the file explorer on the left displaying the project structure. The main.c file is open in the editor, showing a C program with a core1_function and a main function. The terminal at the bottom shows the output of the make command, including the generation of the Makefile and the compilation of various object files.

```
71%] Building C object lib/CMakeFiles/kendryte.dir/drivers/uarts.c.obj
73%] Building C object lib/CMakeFiles/kendryte.dir/drivers/utlis.c.obj
76%] Building C object lib/CMakeFiles/kendryte.dir/drivers/wdt.c.obj
78%] Building CXX object lib/CMakeFiles/kendryte.dir/nncase/nncase.cpp.obj
80%] Building CXX object lib/CMakeFiles/kendryte.dir/nncase/runtime/cpu/cpu_ops.cpp.obj
83%] Building CXX object lib/CMakeFiles/kendryte.dir/nncase/runtime/interpreter.cpp.obj
85%] Building CXX object lib/CMakeFiles/kendryte.dir/nncase/runtime/k210/interpreter.cpp.obj
88%] Building CXX object lib/CMakeFiles/kendryte.dir/nncase/runtime/k210/k210_ops.cpp.obj
90%] Building CXX object lib/CMakeFiles/kendryte.dir/nncase/runtime/kernel_registry.cpp.obj
92%] Building CXX object lib/CMakeFiles/kendryte.dir/nncase/runtime/neutral/neutral_ops.cpp.obj
95%] Linking C static library libkendryte.a
95%] Built target kendryte
Scanning dependencies of target hello_world
97%] Building C object CMakeFiles/hello_world.dir/src/hello_world/main.c.obj
100%] Linking C executable hello_world
Generating .bin file ...
100%] Built target hello_world
PS U:/kendryte-standalone-sdk-develop/src/hello_world/build: 
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

完成编译。其中 `hello_world.bin` 为可直接烧写的文件。

打开 K-Flash 工具, 加载刚才生成的 `bin` 文件, 点击 “Flash” 等待完成即可。



下载完成后, 自动弹出的命令行窗口会直接打印 “hello_wolrd”。

