

🌐 <https://github.com/raspberrypi/pico-sdk/blob/master/.gitignore> を閲覧しました

🌐 <https://github.com/raspberrypi/pico-examples> を閲覧しました

🔍 "Pico SDK project ait ignore files .aitignore" を検索しました

10 結果

すべて表示

## Pico SDKプロジェクトのGitHubへのcommit対象ファイル

Raspberry Pi Pico 2 (RP2350ベース)のプロジェクトをVSCodeのRP2040拡張 (Pico SDKサポート付き) で作成した場合、GitHubにcommitする際は、ソースコードやビルド設定などの再現可能なファイルのみを含め、ビルド生成物や環境依存ファイルを除外するのがベストプラクティスです。これにより、チームメンバーが簡単にクローン・ビルドできるようになります。

Pico SDKの公式ドキュメントや例リポジトリ (例: raspberrypi/pico-examples) を基に、commitすべきファイルと避けるファイルをまとめます。VSCode拡張で生成される標準的なプロジェクト構造を想定しています。

### 1. commitすべきファイル (必須のプロジェクト本体)

これらはプロジェクトのコアで、SDKパスを環境変数で指定してビルド可能です。Gitに含めて共有しましょう。

### ファイル/ディレクトリ

CMakeLists.txt	プロジェクトのビルド設定。SDKのインポートやターゲット定義を含む。
pico_sdk_import.cmake	SDKをインポートするためのCMakeスクリプト (SDKからコピー)。
src/ ディレクトリ	ソースコード (例: main.c や main.cpp )。RP2350向けのコードをここに。
lib/ ディレクトリ (オプション)	カスタムライブラリや追加モジュール (存在する場合)。
README.md	プロジェクト説明、ビルド手順、依存関係 (SDKバージョンなど)。
.gitignore	無視ファイルリスト (後述)。これ自体をcommitして共有。
その他のヘッダー (.h)	ソースに関連するヘッダーファイル。

・ 例のプロジェクト構造 (commit後):

text

✕

折りたたむ

≡

広げる

📄

コピー

```

my_pico_project/
├─ CMakeLists.txt
├─ pico_sdk_import.cmake
├─ README.md
├─ .gitignore
└─ src/
    ├─ main.c (RP2350向けコード例: multicore, PIOなど)
    └─ CMakeLists.txt (サブディレクトリのビルド設定)

```

### 2. commitすべきでないファイル (無視するもの)

これらはビルド時生成やローカル環境依存のため、Gitに含めず `.gitignore` で除外。リポジトリをクリーンに保てます。

## ファイル/ディレクトリ

build/ ディレクトリ	CMakeによるビルド出力 (ELF, UF2, MAPファイルなど)。毎回再生成可能。
*.uf2 , *.elf , *.hex , *.bin	ファームウェアバイナリ。ビルド後に生成。
.vscode/ ディレクトリ	VSCode設定 (launch.json, tasks.jsonなど)。ユーザー固有で共有不要。
cmake-*.cache	CMakeキャッシュファイル。環境依存。
.DS_Store (macOS) や Thumbs.db (Windows)	OS生成のメタデータ。
PICO_SDK_PATH 関連のローカル設定	環境変数で管理。リポジトリにハードコードせず。

- ・推奨の .gitignore 内容 (Pico SDK公式のものを基にカスタマイズ):

text

✕ 折りたたむ

≡ 広げる

📄 コピー

```
# Build outputs
build/
*.uf2
*.elf
*.hex
*.bin
*.map
*.lst
```

```
# CMake
cmake-*/
CMakeCache.txt
CMakeFiles/
.cache
```

```
# VSCode
.vscode/
```

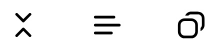
```
# OS
.DS_Store
Thumbs.db
```

```
# IDE
.idea/
*.swp
*.swo
```

### 3. commitの手順 (VSCode + Git使用時)

1. VSCodeでプロジェクトを開き、拡張でビルド確認 (Ctrl+Shift+P → "Pico: Build")。
2. ターミナルで:

text



```
git init # 初回のみ
git add . # .gitignoreがあるので自動除外
git commit -m "Initial Pico SDK project for RP2350"
git remote add origin https://github.com/yourusername/your-repo.git
git push -u origin main
```

3. README.mdにビルド手順を追加 (例: export PICO\_SDK\_PATH=/path/to/pico-sdk と mkdir build && cd build && cmake .. && make)。
4. Pico 2 (RP2350)特有の機能 (例: DCP, OTP)を使う場合、CMakeLists.txtで PICO\_PLATFORM=raspberrypi\_rp2350 を指定。

## 注意点

- ・ **SDK自体はcommitしない**: Pico SDKはGitHubからクローン `github.com` し、環境変数 `PICO_SDK_PATH` で参照。プロジェクトに含めるとリポジトリが巨大化。
- ・ **サブモジュール**: TinyUSBなどの依存はSDK内で管理。必要なら `git submodule update --init` をREADMEに記載。
- ・ **VSCode拡張特有**: 拡張が生成するテンプレートは上記に準拠。Pico 2サポートはSDK 2.0.0以降でOK (2025年現在、最新版使用推奨)。

これでクローンした人が簡単にビルドできます。追加質問があればどうぞ！

↳ [RP2350の新機能について詳しく](#)

↳ [Pico WのWiFiプロジェクト例](#)