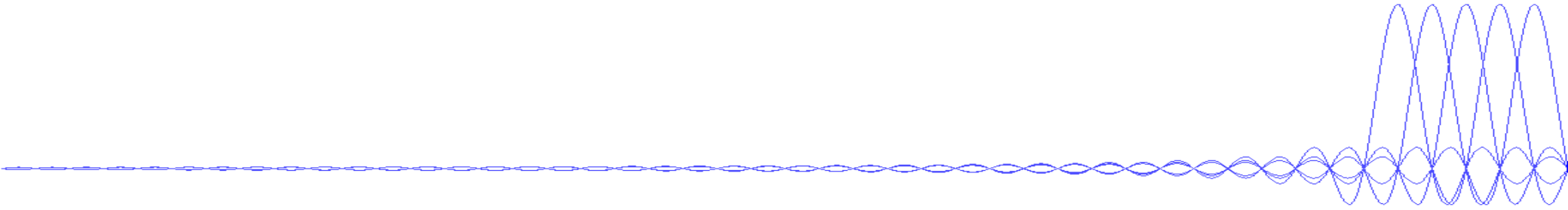
Raspberry PiとZERO 7

導入から設計，動作まで



Copyrights 2026 DSP Laboratory. All Rights Reserved.

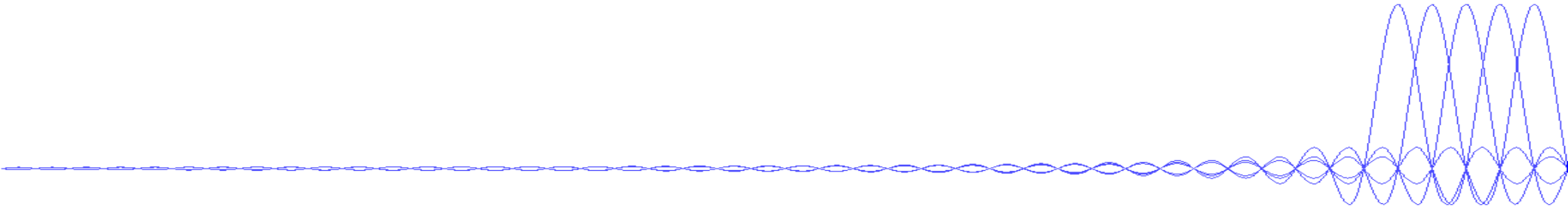
開発の流れ

- Raspberry Pi側

- ZERO7側

Copyrights 2026 DSP Laboratory. All Rights Reserved.

Raspberry Pi 側の開発

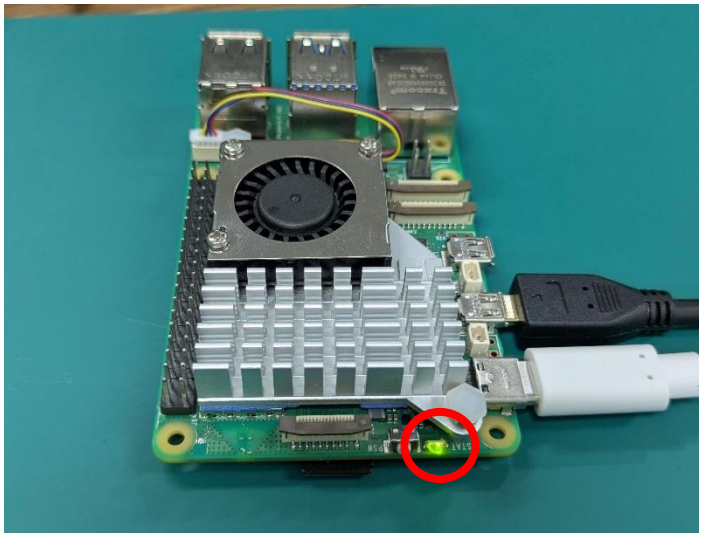


Copyrights 2026 DSP Laboratory. All Rights Reserved.

Raspberry Piの開発環境構築1

- Raspberry Pi用のOSインストール
  - 下記のリンクを参考にする
  - https://www.raspberrypi.com/documentation/computers/getting-started.html

- Raspberry Piをモニターに接続させる
- Raspberry Piを電源に接続させる
  - 電源に接続すると自動的に起動し，下記のようにLEDが緑に点灯する



Copyrights 2026 DSP Laboratory. All Rights Reserved.

Raspberry Piの開発環境構築2

- Raspberry Piを起動したら，モニターにデスクトップが表示される



Raspberry Piのデスクトップ画面

Copyrights 2026 DSP Laboratory. All Rights Reserved.

Raspberry Piの開発環境構築3

- 入出力インターフェースの接続
  - Raspberry Piの開発のために，USBやBluetoothでマウスやキーボードを接続
- インターネット接続
  - 無線LANもしくは，LANケーブルを用いてインターネット接続を行う
  - 詳細は，下記のリンクを参照
  - https://www.raspberrypi.com/documentation/computers/getting-started.html#wi-fi
- その他
  - Windows PCなどからSSH接続を用いて開発が可能なので，詳細はAppendixに記載する.

Copyrights 2026 DSP Laboratory. All Rights Reserved.

Raspberry Piのプログラム実行

- Pythonプログラムを用意する
  - 今回実行するプログラムが “raspi-zer07_button.py”
  - このマニュアルと同じフォルダに用意している
  - USBメモリなどを活用してRaspberry Piの任意のフォルダに展開する

Copyrights 2026 DSP Laboratory. All Rights Reserved.

Raspberry Piのプログラム実行2

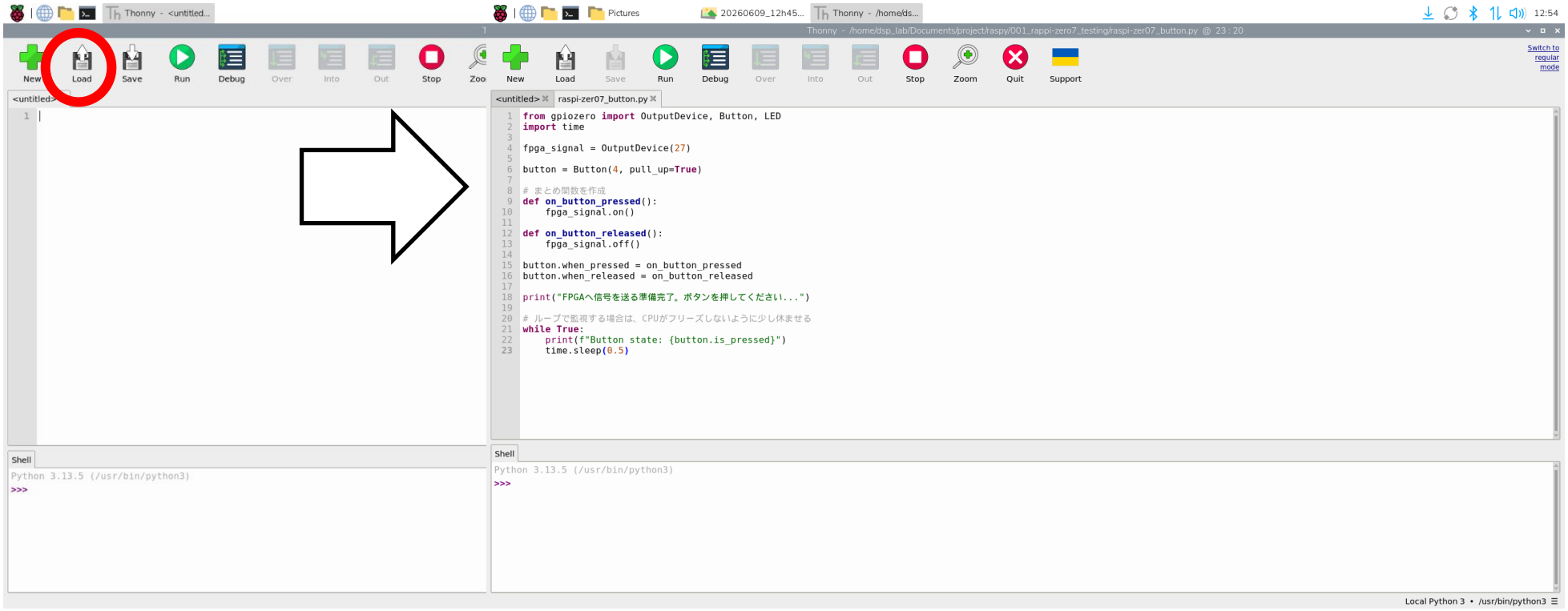
- Python実行用のソフトウェアの用意
  - Raspberry Piに初期状態からインストールされているThonnyを用いる
  - 左上のRaspberry Piマークをクリック→プログラミング→Thonnyの順でクリック



Copyrights 2026 DSP Laboratory. All Rights Reserved.

Raspberry Piのプログラム実行3

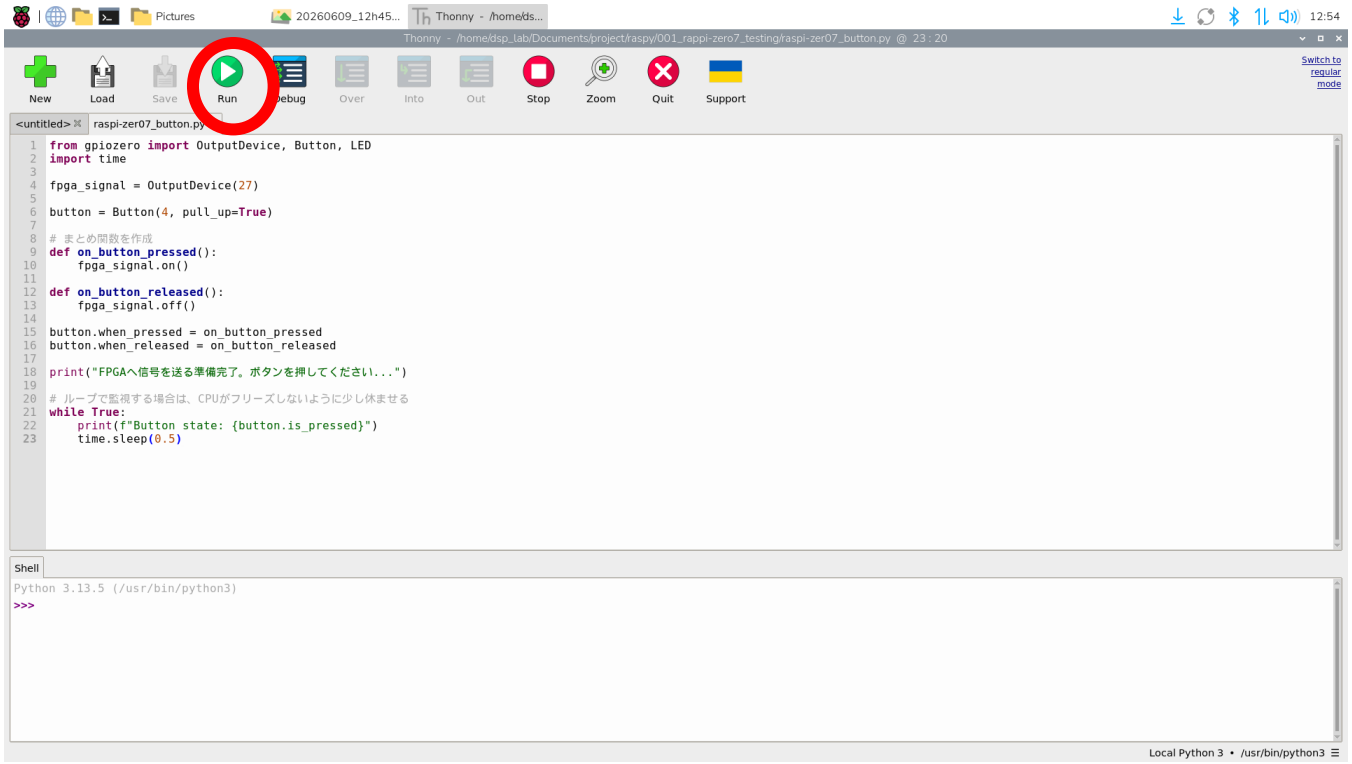
- Pythonプログラムの用意
  - Thonnyの実行画面から，“Load”を選択
  - ダウンロードした実行プログラムのファイルを選択する



Copyrights 2026 DSP Laboratory. All Rights Reserved.

Raspberry Piのプログラム実行4

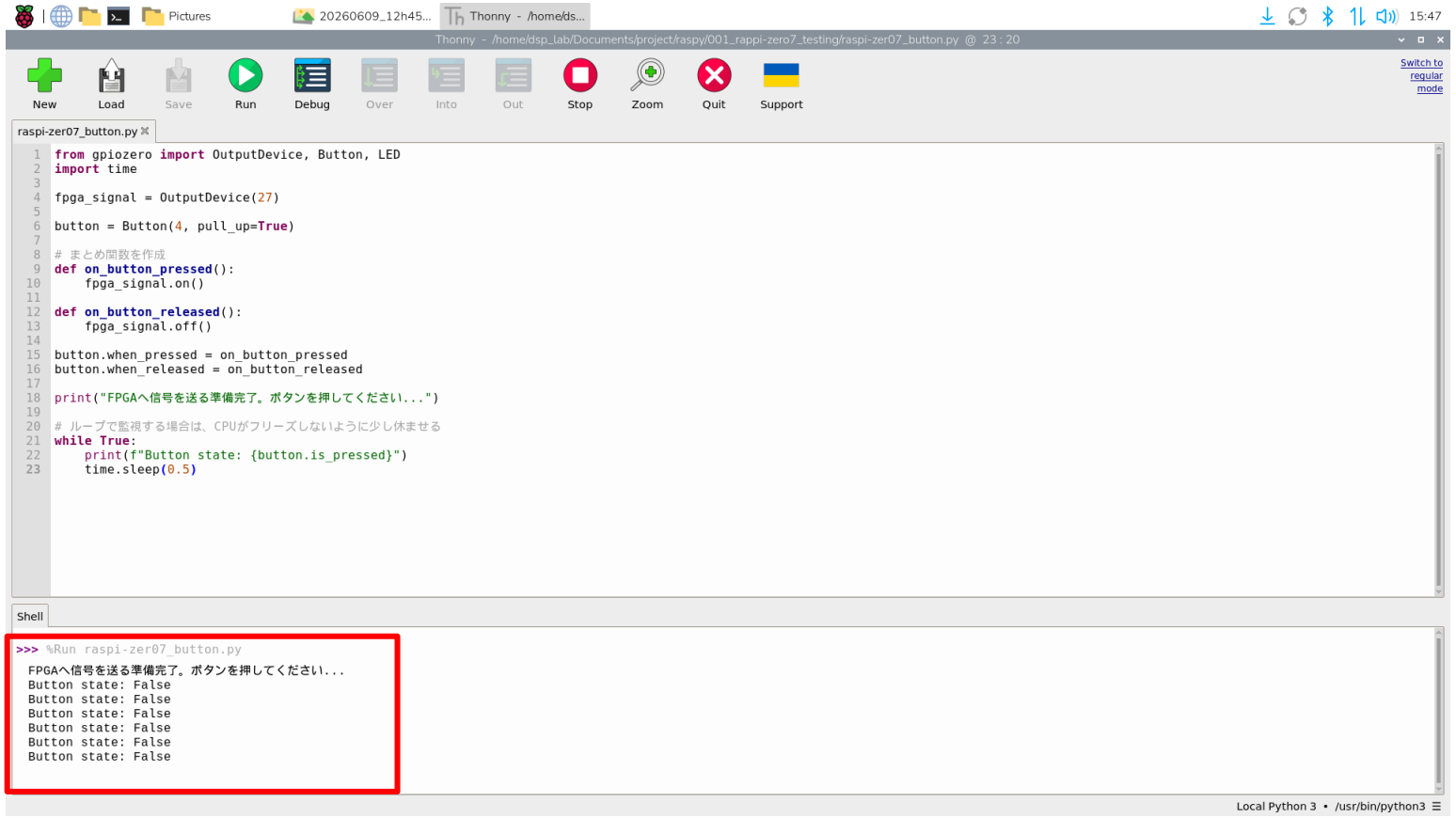
- Pythonプログラムの実行
  - 実行するPythonプログラムの用意
  - “Run”をクリックして，プログラムの実行を行う



Copyrights 2026 DSP Laboratory. All Rights Reserved.

実行の確認

- 本プログラムを実行されると，下記のようなShellの出力が得られる
  （まだ回路の設計していないので，Shellの表示は変わらない）



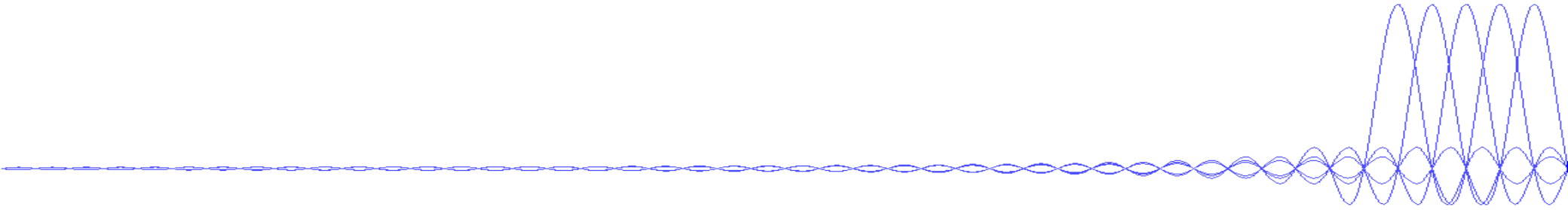
Copyrights 2026 DSP Laboratory. All Rights Reserved.

Raspberry Pi 側の開発

- 以上で，Raspberry Piの実行用意は完了した

Copyrights 2026 DSP Laboratory. All Rights Reserved.

ZERO7の開発



Copyrights 2026 DSP Laboratory. All Rights Reserved.

ZERO7仕様 / 開発環境

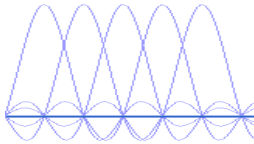
- ZERO7 取り扱い説明書を参照
- ZERO7 FPGA
  - シリーズ
    - Spartan 7
  - 型番
    - xc7s25csga225-1 ←―― Vivadoで開発する際に選択
- 環境
  - Windows 11
  - Vivado 2025.2

Copyrights 2026 DSP Laboratory. All Rights Reserved.

ZERO7開発の流れ

- Vivadoのインストール
- 使用するプログラム・ファイルの用意
- Vivavoプロジェクトの作成
  - プログラムの追加（Add Source）
  - テストベンチの実行←今回はパス
  - ILAの作成
  - Bitstreamの作成
  - ZERO7の用意
  - ZERO7とWindows PCの接続（JTAG）
  - ZERO7へbitファイルのアップロード

Copyrights 2026 DSP Laboratory. All Rights Reserved.



Vivadoのインストール

- Vivadoのインストールについては後ほど説明

Copyrights 2026 DSP Laboratory. All Rights Reserved.

Vivadoのインストール1

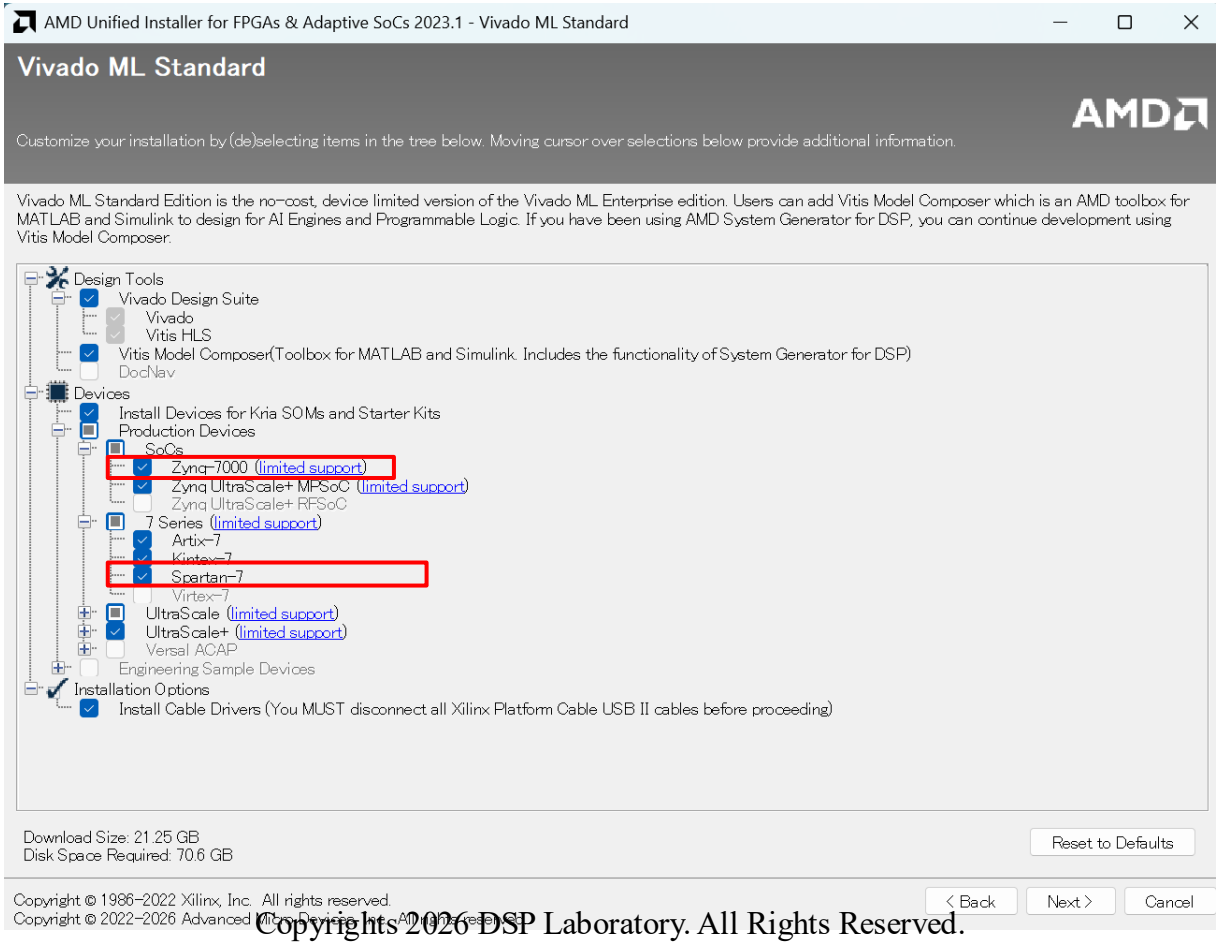
- インストール手順について下記を参照
  - Vivado インストール方法【Xilinx FPGA環境構築】Windows – kotoha sensorium

- 掲載されているURLの手順に従い，インストールを実施する.
  以降のスライドでは，URLの記載内容と異なる点のみを抜粋して説明する.

Copyrights 2026 DSP Laboratory. All Rights Reserved.

Vivadoのインストール2

- 開発対象デバイス（最低限）
  - 7Series → Spartan-7
  - SoCs → Zynq-7000



Copyrights 2026 DSP Laboratory. All Rights Reserved.

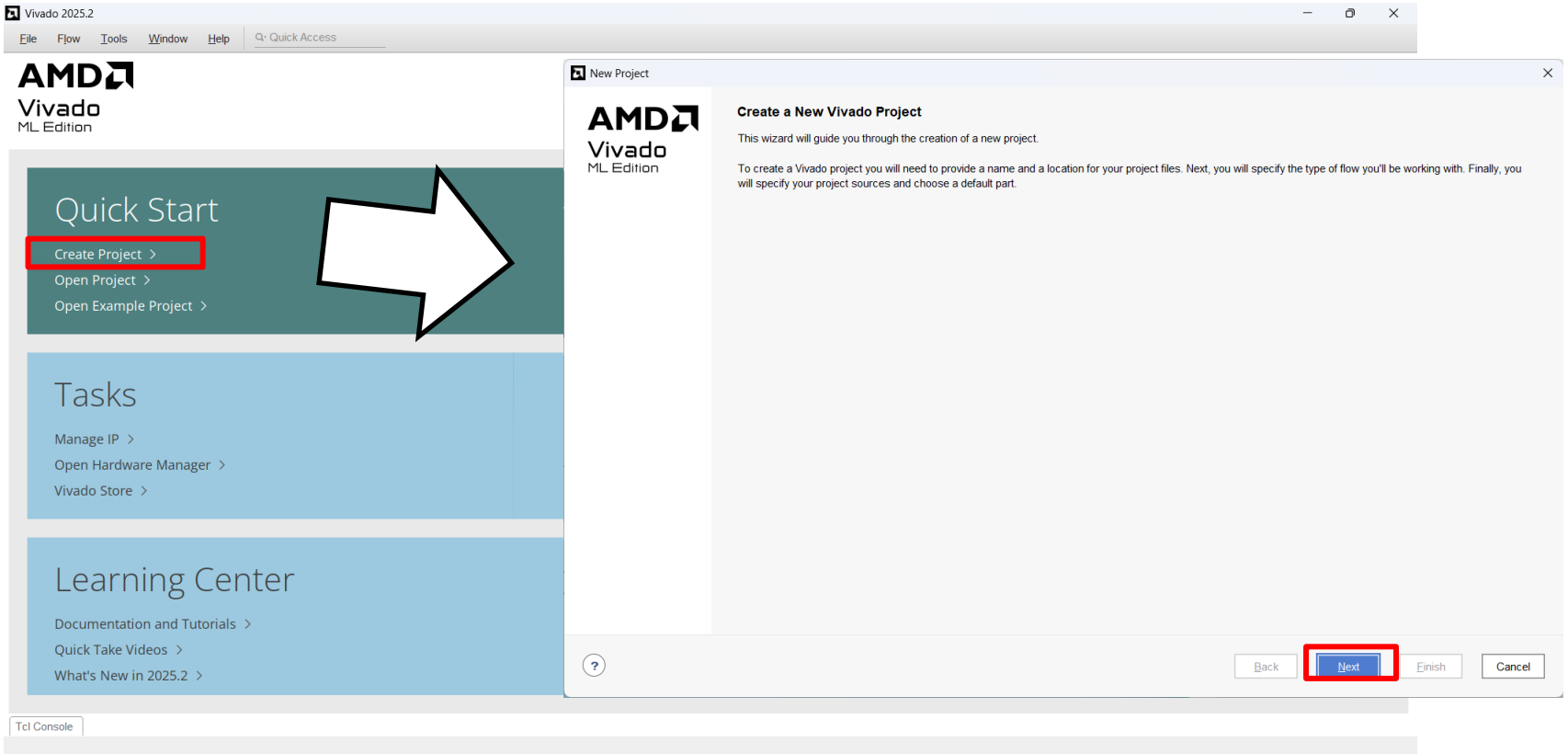
Verilogプログラムの用意

- 使用するプログラム・ファイルの用意
  - 使用するVerilogファイルは，“pi_button_receiver.v”
  - 使用するConstrainファイルは，“Zero7.xdc”

Copyrights 2026 DSP Laboratory. All Rights Reserved.

Vivavoプロジェクトの作成1

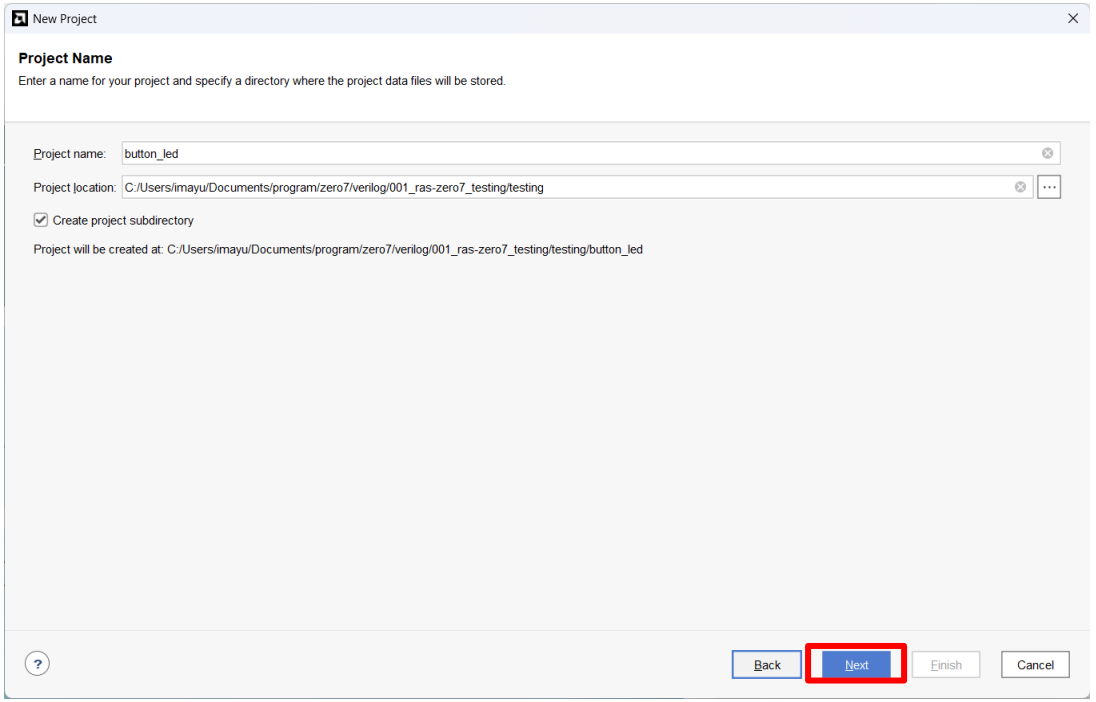
- Vivadoを起動し，新規プロジェクトを作成する
  - “Create Project”を選択
  - “Next”を選択



Copyrights 2026 DSP Laboratory. All Rights Reserved.

Vivavoプロジェクトの作成2

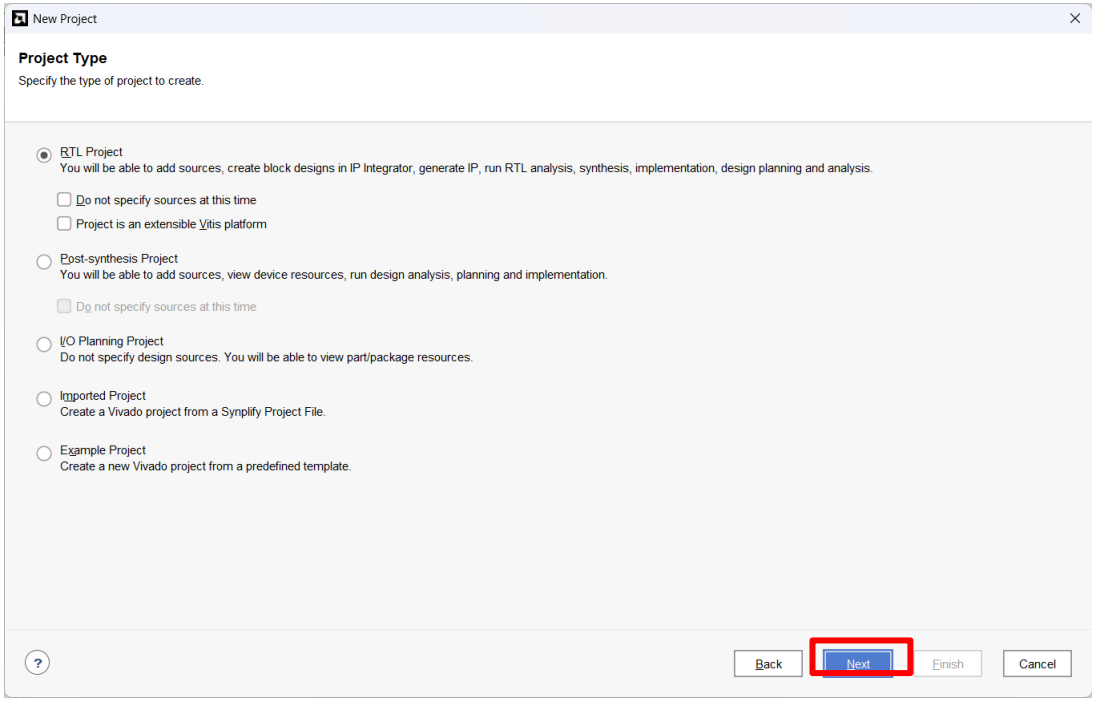
- プロジェクトのフォルダと名前を決める
  - 任意の値にする（長過ぎないように）
  - “Next”を選択



Copyrights 2026 DSP Laboratory. All Rights Reserved.

Vivavoプロジェクトの作成3

- プロジェクトの詳細設定
  - 基本的にすべて “Next”を選択
  - まずは，“Project Type”の選択
  - 初期状態の “RTL Project”の選択

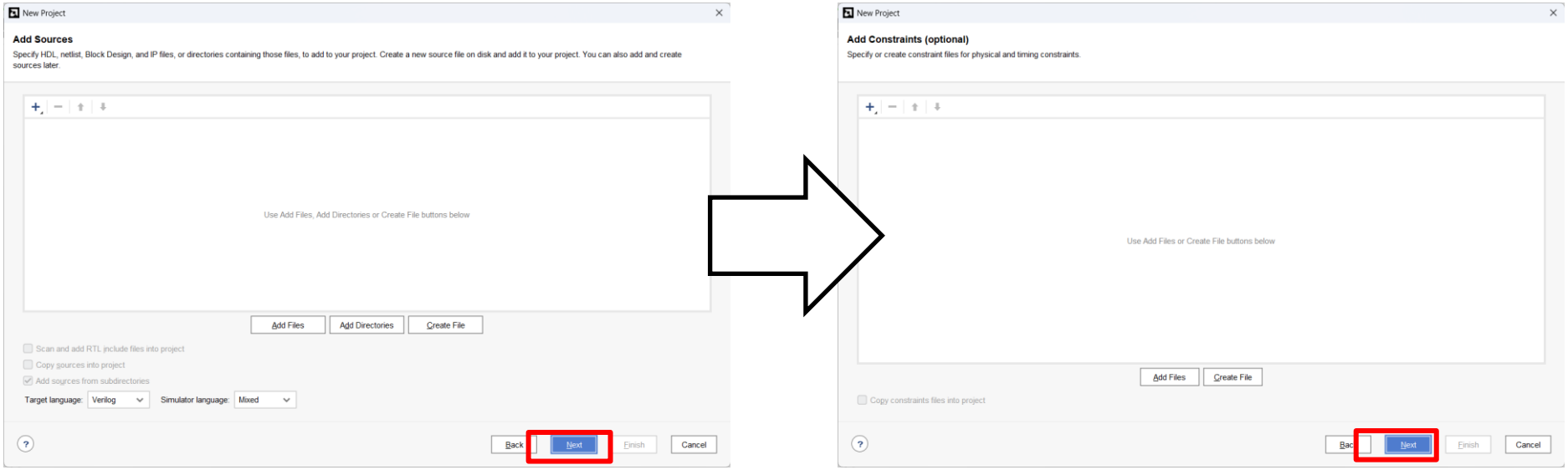


Copyrights 2026 DSP Laboratory. All Rights Reserved.

Vivavoプロジェクトの作成4

■ プロジェクトの詳細設定

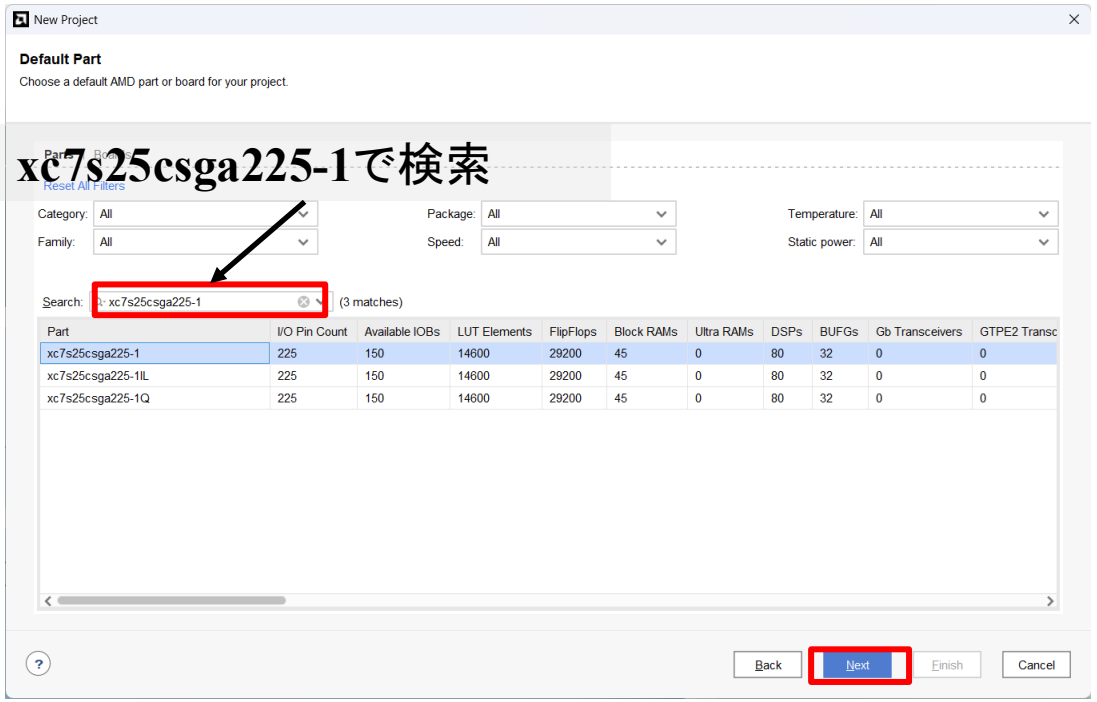
- □次に，“Add Sources”にて，“Next”をクリック
  - ■ここでVerilogファイルを選択しても良い
  - ■本資料では後ほど追加する
- □次に，“Add Constrains”にて，“Next”をクリック
  - ■ここで，XDCファイルを選択しても良い
  - ■本資料では後ほど追加する



Copyrights 2026 DSP Laboratory. All Rights Reserved.

Vivavoプロジェクトの作成5

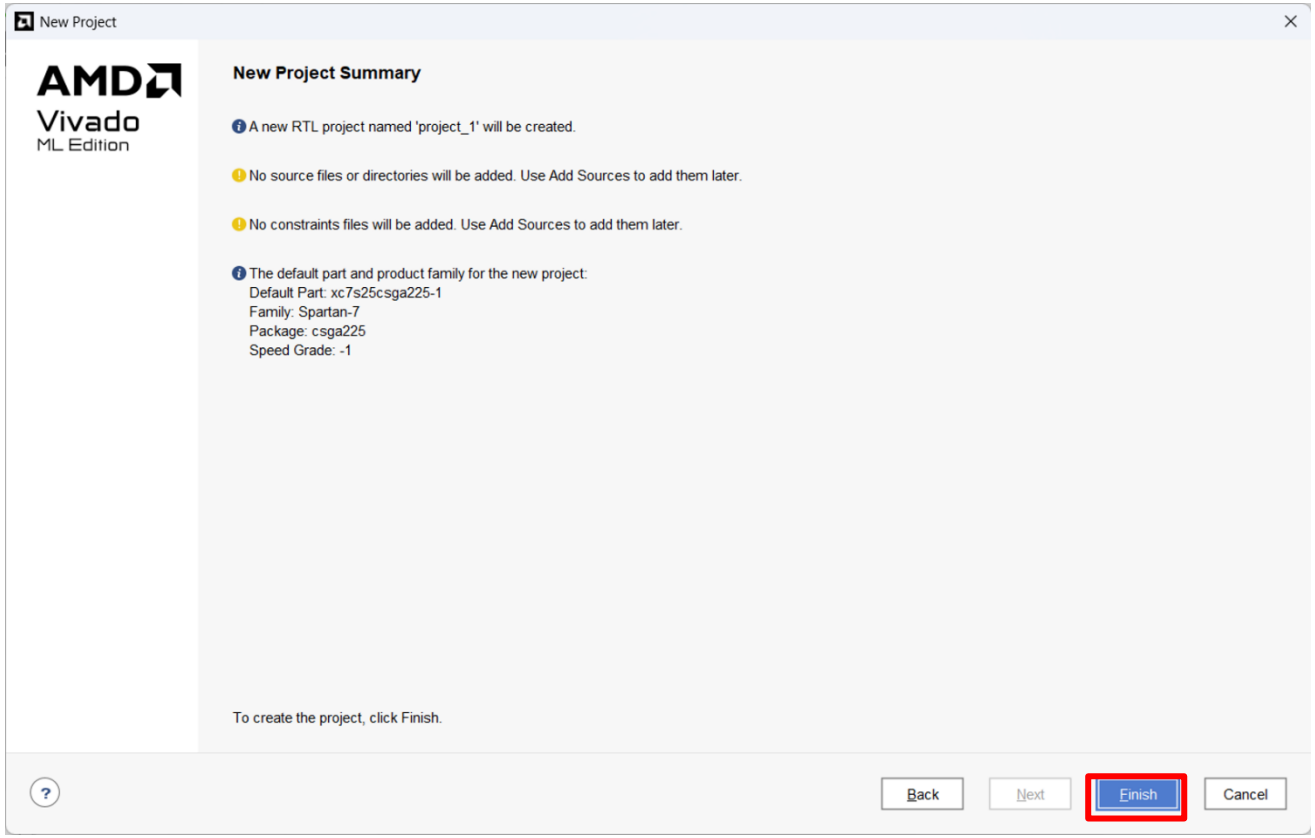
- プロジェクトの詳細設定
  - 使用するボードの選択
  - ZERO7は，“**xc7s25csga225-1**” を検索し，選択する
  - 選択して，青色になったら，“Next”の選択



Copyrights 2026 DSP Laboratory. All Rights Reserved.

Vivavoプロジェクトの作成6

- サマリーの確認
  - 下記のような表示であればOK
  - “Finish”を選択



Copyrights 2026 DSP Laboratory. All Rights Reserved.

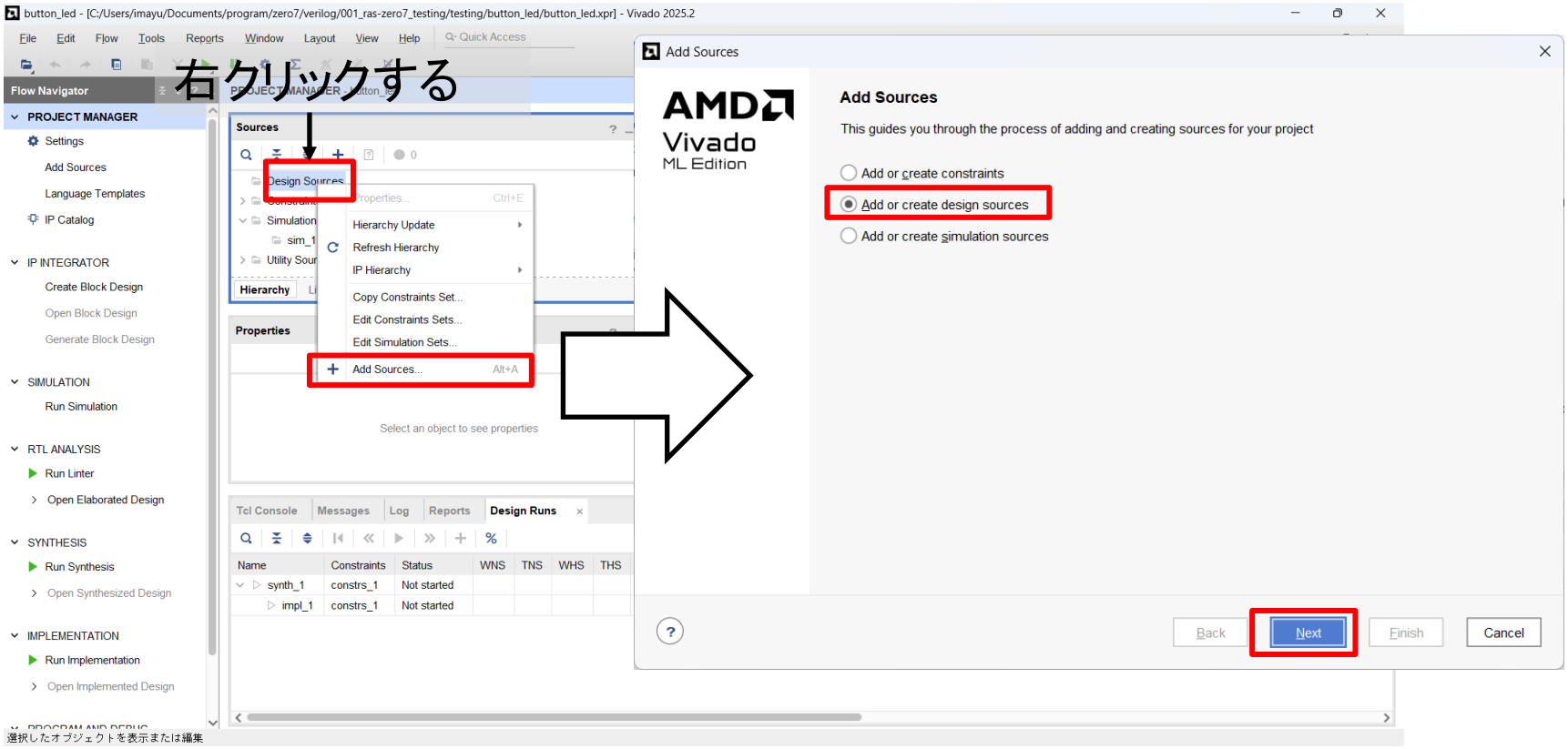
Vivavoプロジェクトの作成 | プログラムの追加1

- 2つのファイル・プログラムを追加する
  - Verilogファイル及び，Constrainファイル

Copyrights 2026 DSP Laboratory. All Rights Reserved.

Vivavoプロジェクトの作成 | プログラムの追加2

- Verilogファイルの追加
  - “Design Sources”を選択し，右クリック
  - “Add Sources”を選択
  - 新しく開いたWindowで，“Add or create design sources”を選択し，“Next”を選択

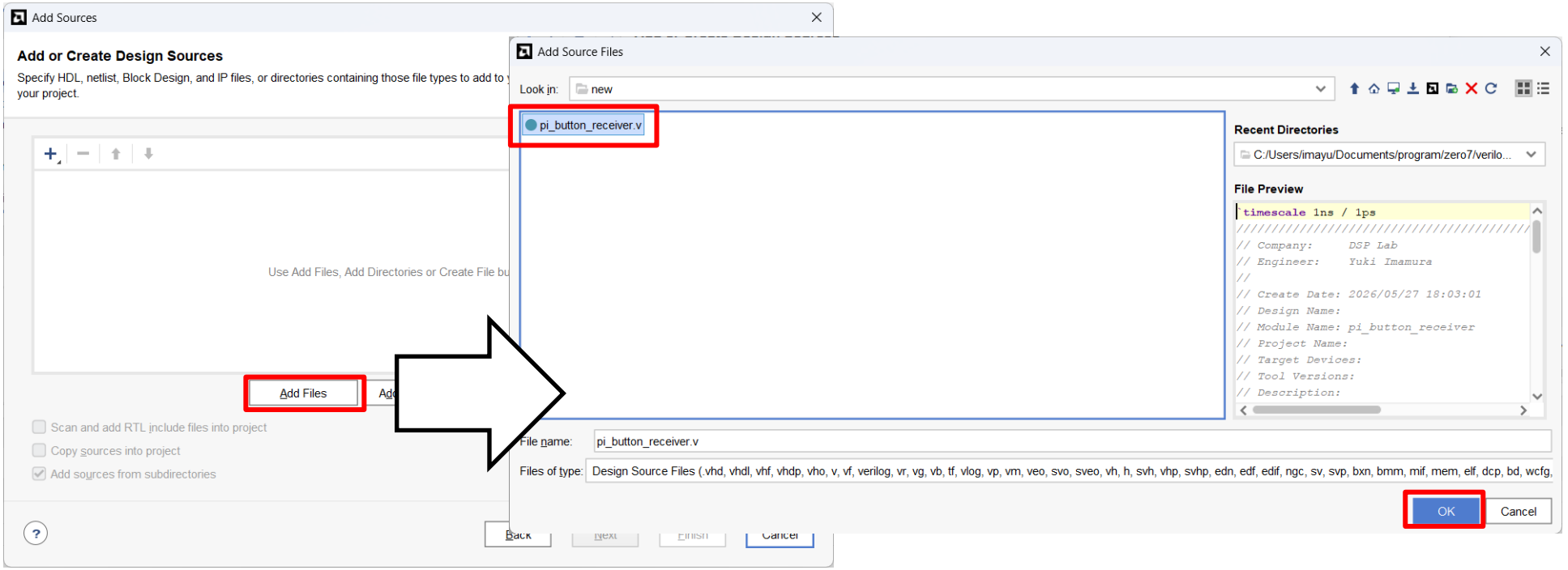


Copyrights 2026 DSP Laboratory. All Rights Reserved.

Vivavoプロジェクトの作成 | プログラムの追加3

Verilogファイルの追加

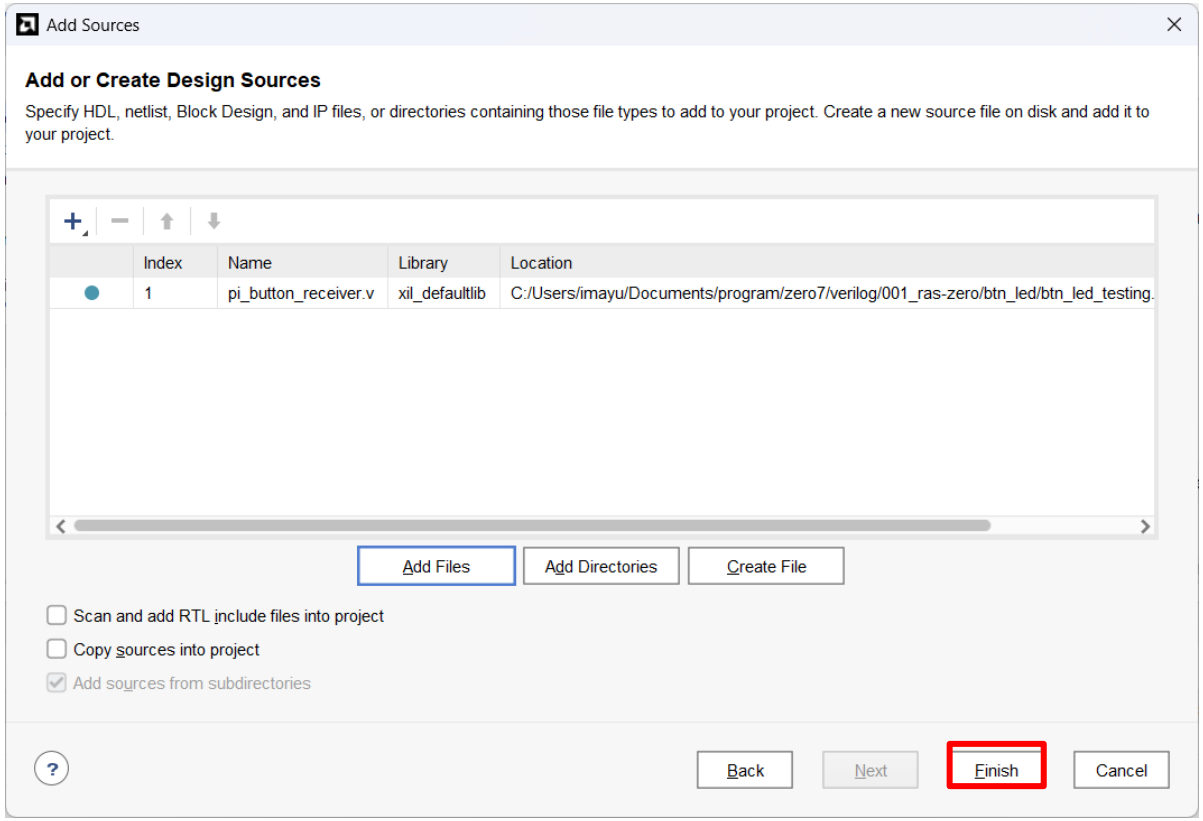
- “Add Files”の選択
- 新しい画面で，追加する “pi_button_receiver.v”を選択し，“OK”を選択



Copyrights 2026 DSP Laboratory. All Rights Reserved.

Vivavoプロジェクトの作成 | プログラムの追加4

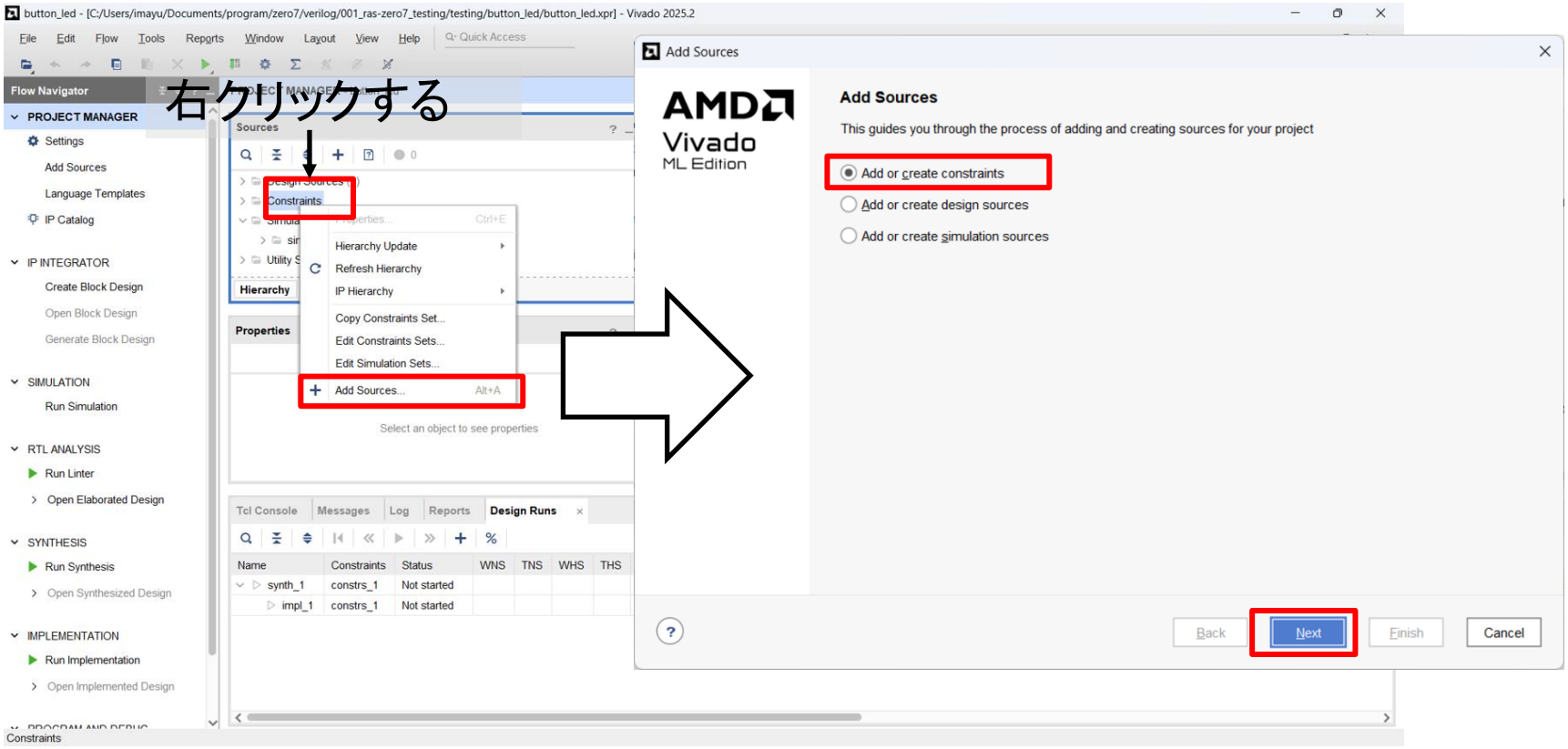
- Verilogファイルの追加
  - 追加したプログラムがあったら，“Finish”を選択



Copyrights 2026 DSP Laboratory. All Rights Reserved.

Vivavoプロジェクトの作成 | プログラムの追加5

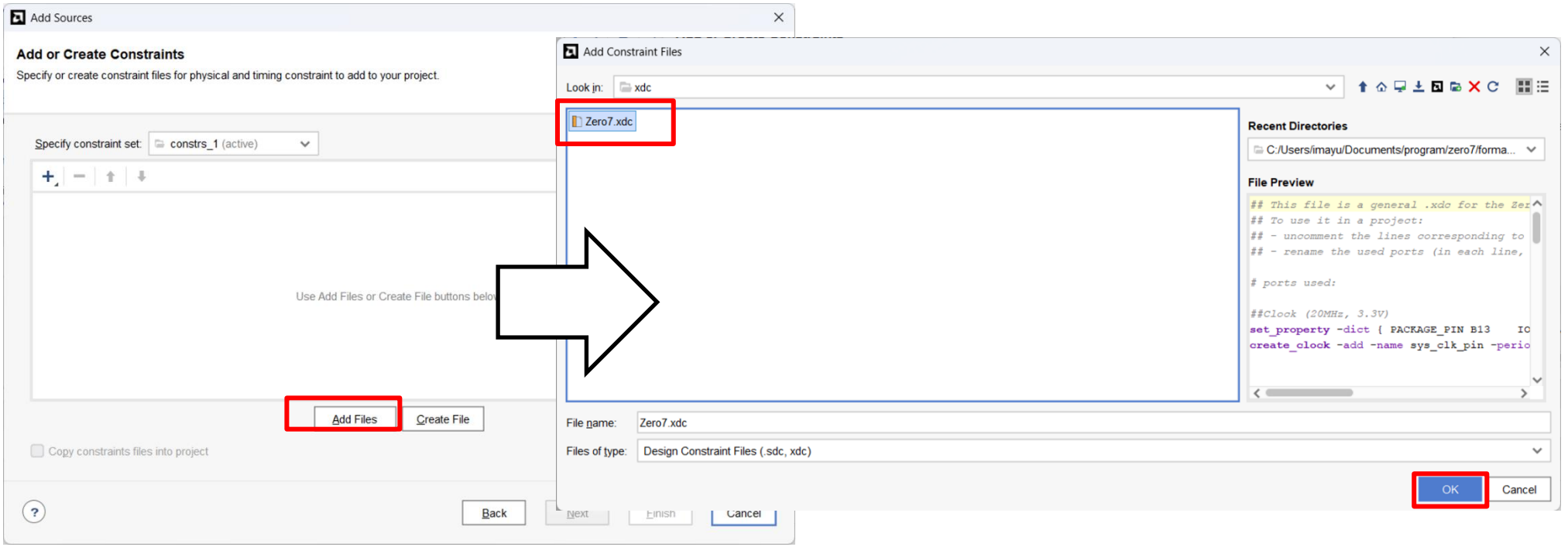
- Constraintファイルの追加
  - “Constraints”を選択し，右クリック
  - “Add Sources”を選択
  - 新しく開いたWindowで，“Add or create constraints”を選択し，“Next”を選択



Copyrights 2026 DSP Laboratory. All Rights Reserved.

Vivavoプロジェクトの作成 | プログラムの追加6

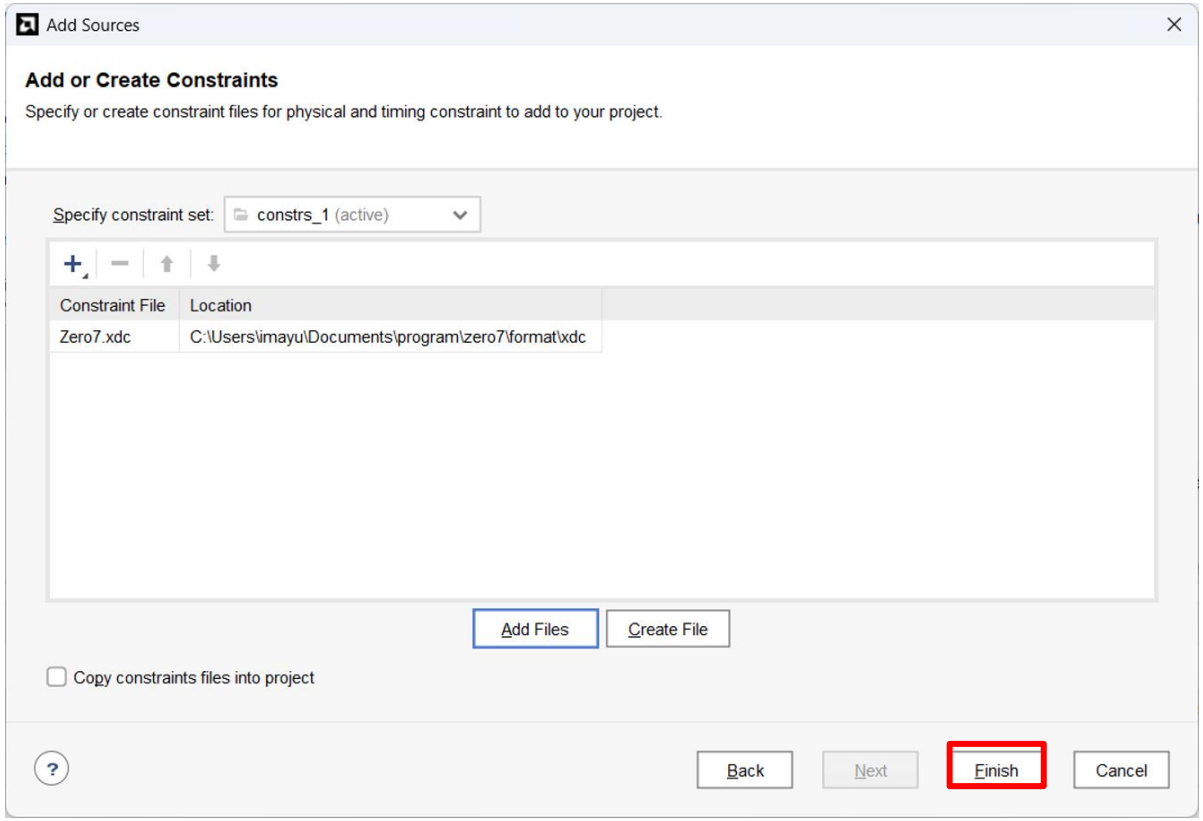
- Constrainファイルの追加
  - “Add Files”の選択
  - 新しい画面で，追加する “Zero7.xdc”を選択し，“OK”を選択



Copyrights 2026 DSP Laboratory. All Rights Reserved.

Vivavoプロジェクトの作成 | プログラムの追加7

- Constrainファイルの追加
  - 追加したプログラムがあったら，“Finish”を選択

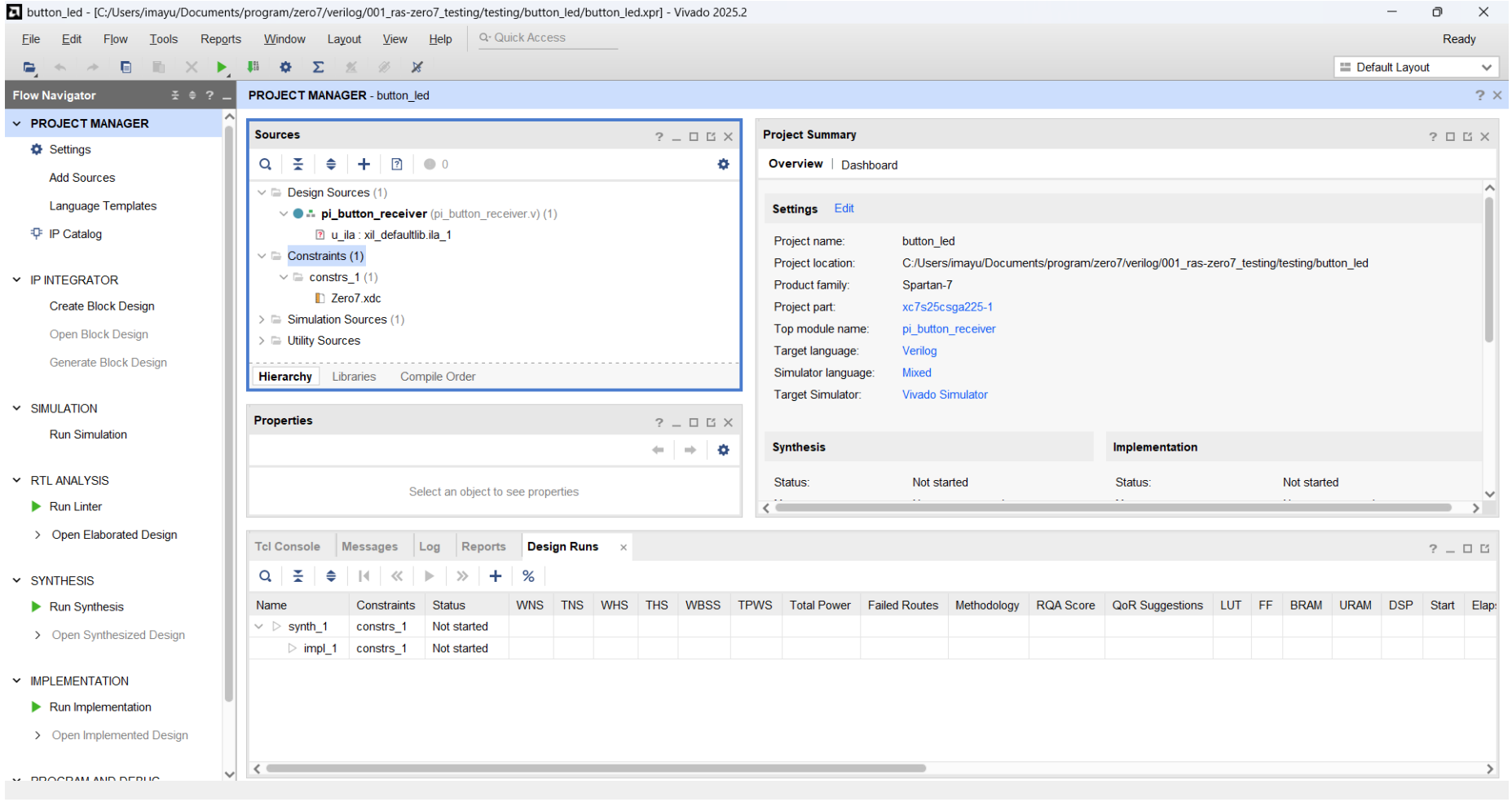


Copyrights 2026 DSP Laboratory. All Rights Reserved.

Vivavoプロジェクトの作成 | プログラムの追加8

■ 最終的な表示

□ 下記のように，ファイルが追加されておればOK



Copyrights 2026 DSP Laboratory. All Rights Reserved.

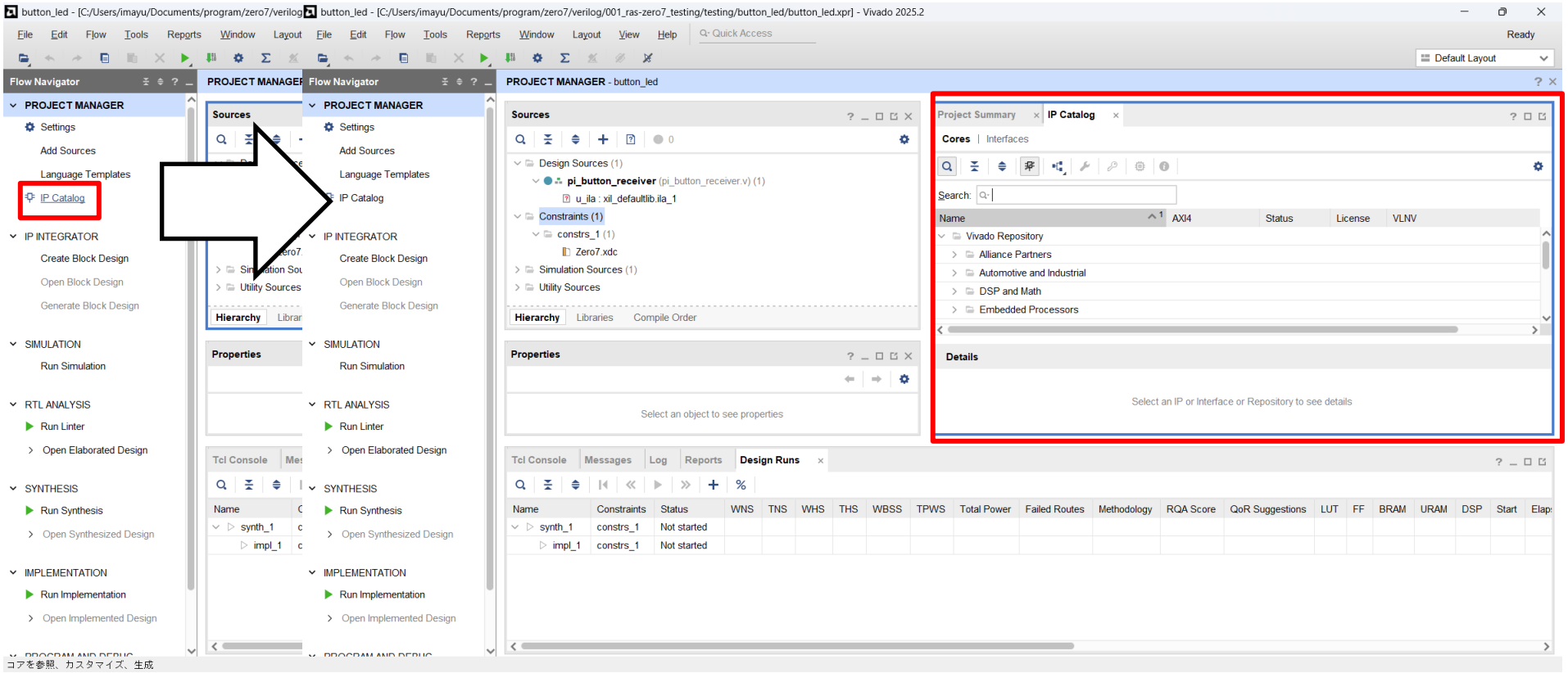
Vivavoプロジェクトの作成 | ILAの作成1

- ILA(Integrated Logic Analyzer)の追加
  - ZERO7内のFPGA Spartan7が正しく動作しているかを確認するためにのアナライザーの追加を行う

Copyrights 2026 DSP Laboratory. All Rights Reserved.

Vivavoプロジェクトの作成 | ILAの作成2

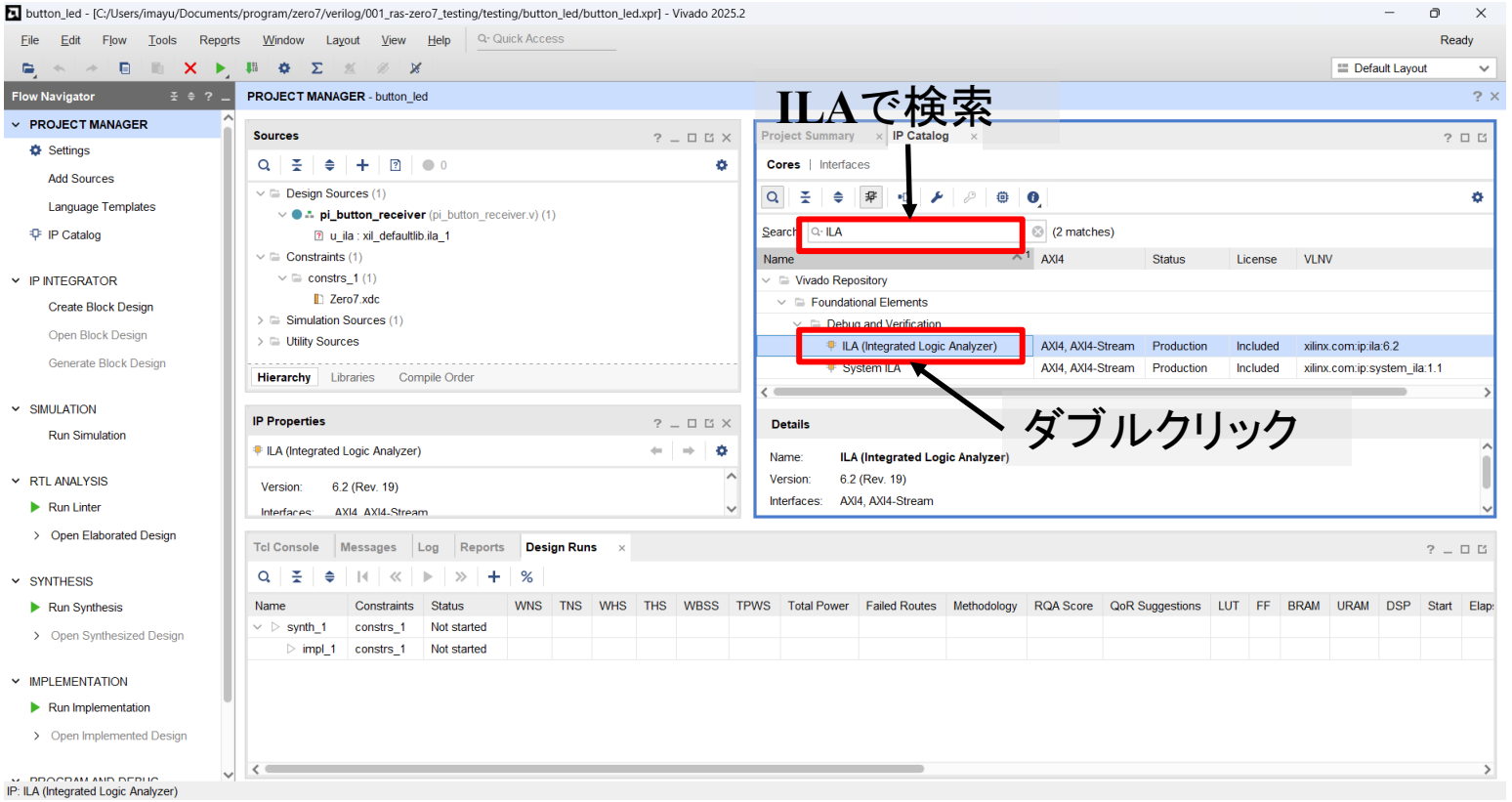
- ILA IP Blockの追加
  - “IP Catalog”の選択
  - “IP Catalog”のWindowが右側に表示される



Copyrights 2026 DSP Laboratory. All Rights Reserved.

Vivavoプロジェクトの作成 | ILAの作成3

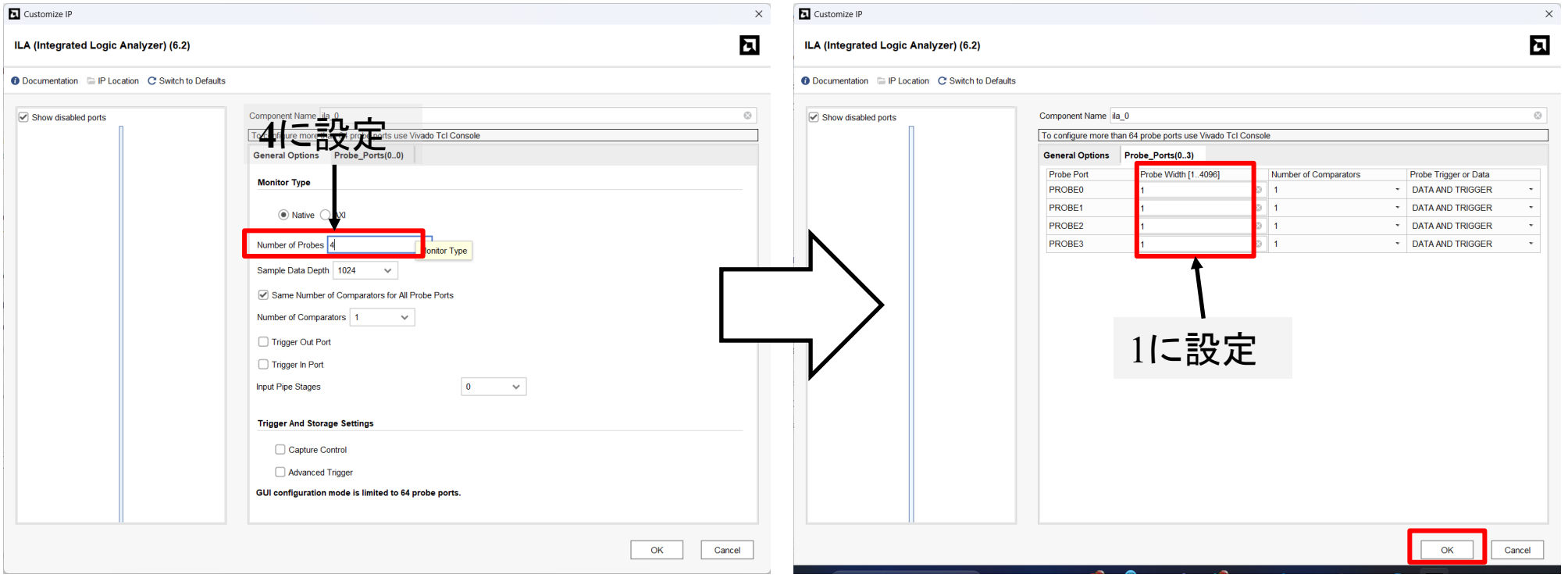
- ILA IP Blockの追加
  - 検索部分で “ILA”と入力
  - “ILA(Integrated Logic Analyzer)”をダブルクリック



Copyrights 2026 DSP Laboratory. All Rights Reserved.

Vivavoプロジェクトの作成 | ILAの作成4

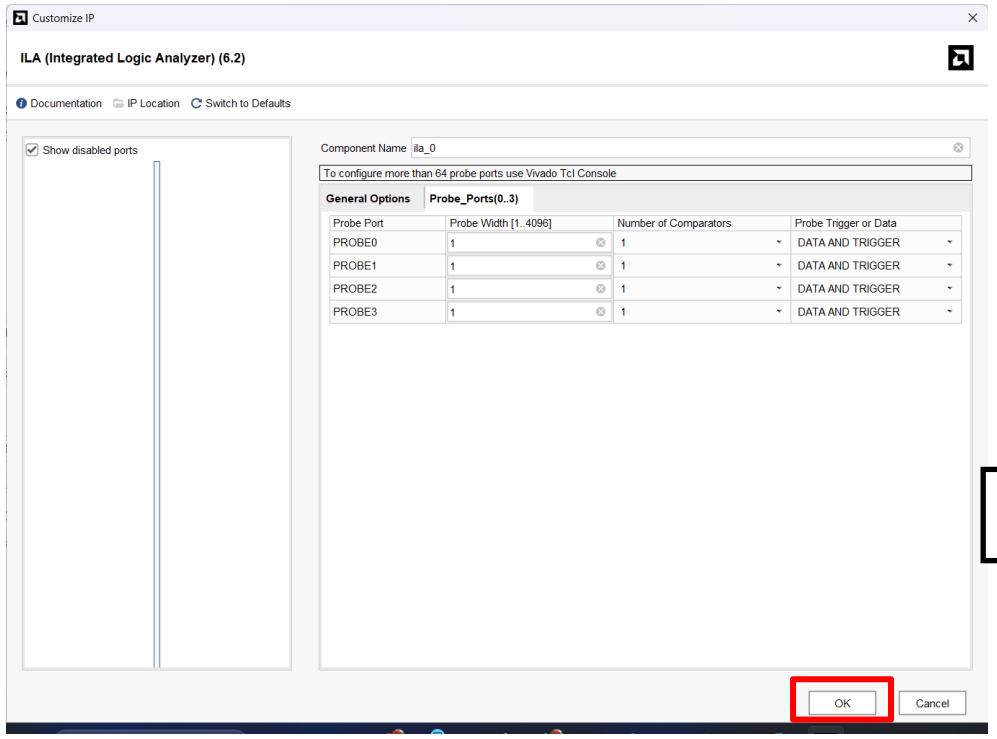
- ILA IP Blockの詳細設定
  - ILAの詳細設定画面が開く
  - “General Options”にて，“Number of Probes”を**4**に設定
  - Probeの “Probe Width”をすべて**1**に設定
  - 設定後，“OK”を選択

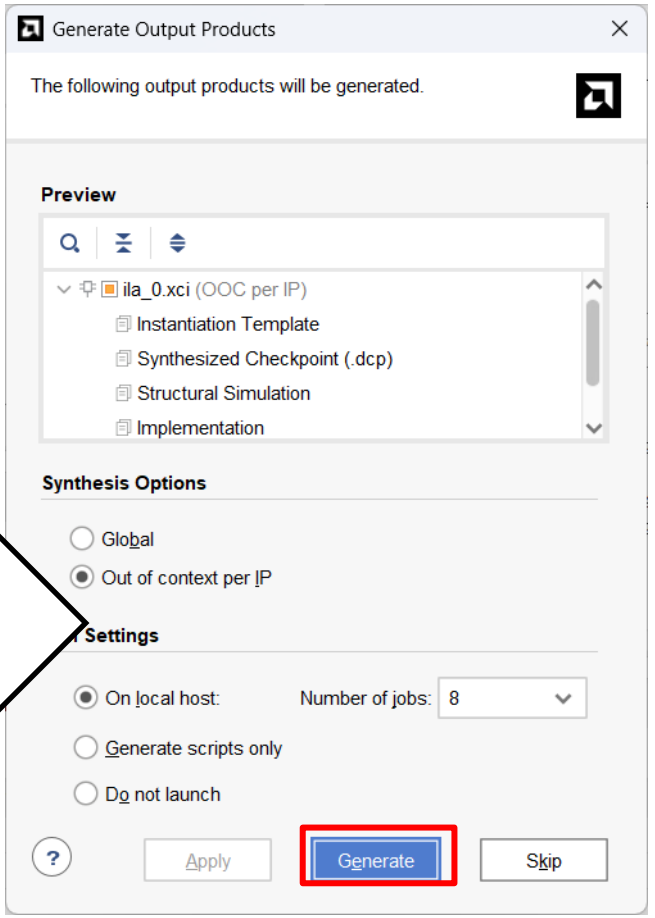


Copyrights 2026 DSP Laboratory. All Rights Reserved.

Vivavoプロジェクトの作成 | ILAの作成5

- ILA IP BlockのGenerate
  - “OK”選択後，下記のようなGenerateの画面が表示される

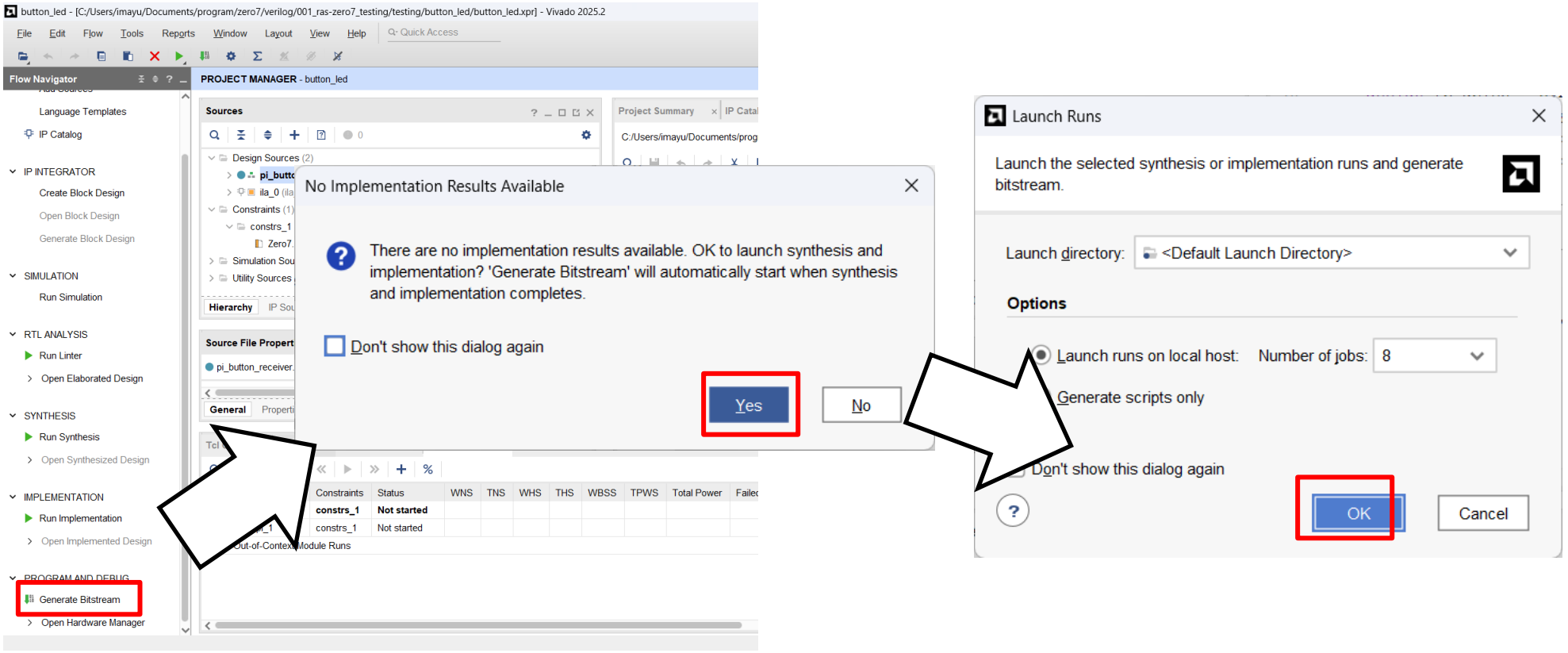




Copyrights 2026 DSP Laboratory. All Rights Reserved.

Vivavoプロジェクトの作成 |Bitstreamの作成1

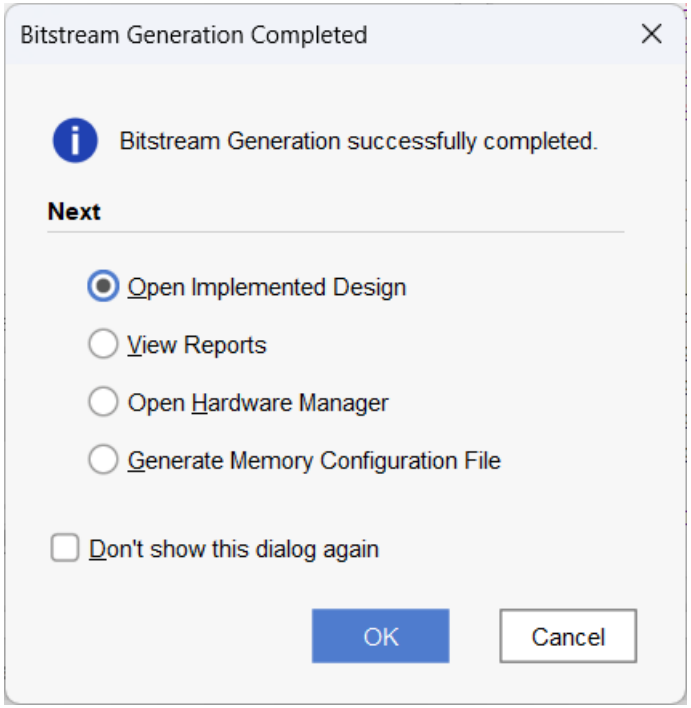
- 左側の一番した “Generate Bitstream”を選択
- Bitstreamの作成には時間がかかるので待つ



Copyrights 2026 DSP Laboratory. All Rights Reserved.

Vivavoプロジェクトの作成 |Bitstreamの作成2

- Bitstreamの作成できたら，下記の画面が表示される
- “OK”選択すると，今回のプロジェクトのレポートが表示される

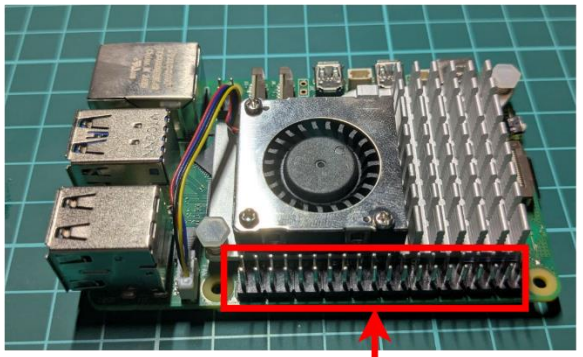


Copyrights 2026 DSP Laboratory. All Rights Reserved.

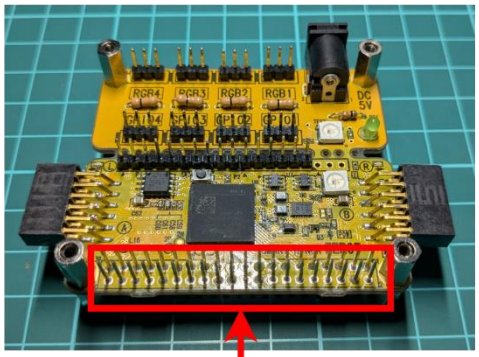
Vivavoプロジェクトの作成 | ZERO7の用意1

- Raspberry PiとZERO7を下記のように接続
  - それぞれのGPIO Header同士を接続

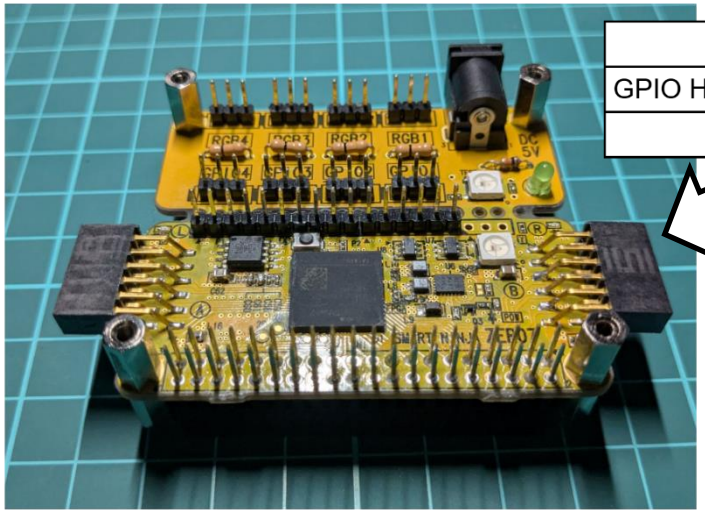
- Raspberry Piに電源が供給されると
  - ZERO7のLEDが点灯する



GPIO Header



GPIO Header

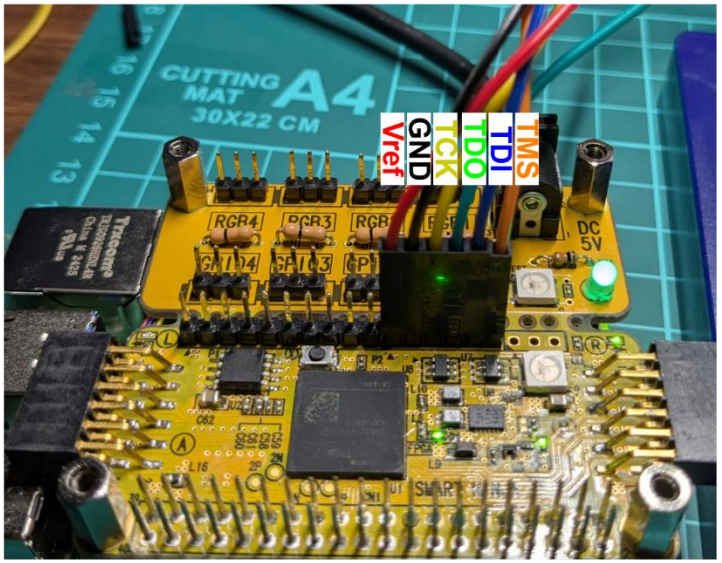


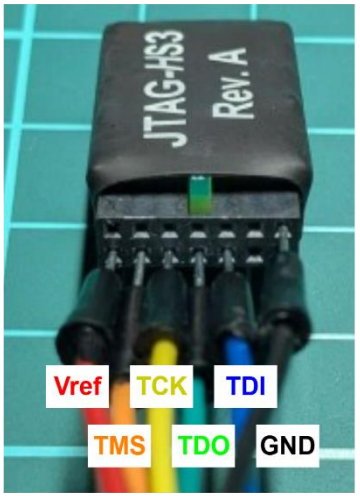
Copyrights 2026 DSP Laboratory. All Rights Reserved.

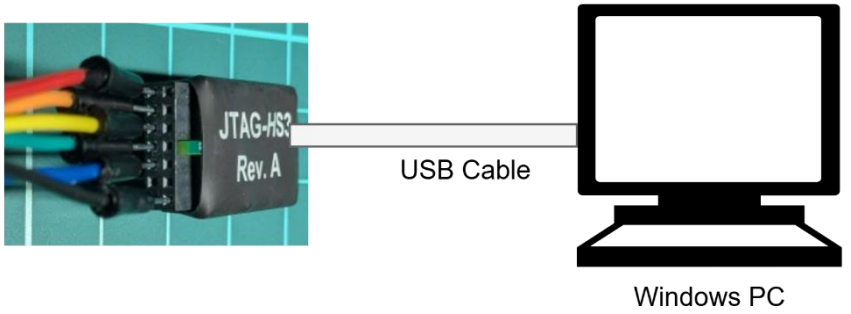
Vivavoプロジェクトの作成 | ZERO7とWindows PCの接続

- PCとZERO7をJTAGで接続
  - JTAG-H53とZERO7のJTAG部分をジャンパ線で接続させる
  - JTAG-H53とWindows PCをUSBケーブルで接続する

- Vref (red)
- TMS (orenge)
- TCK (yallow)
- TDO (green)
- TDI (blue)
- GND (black)



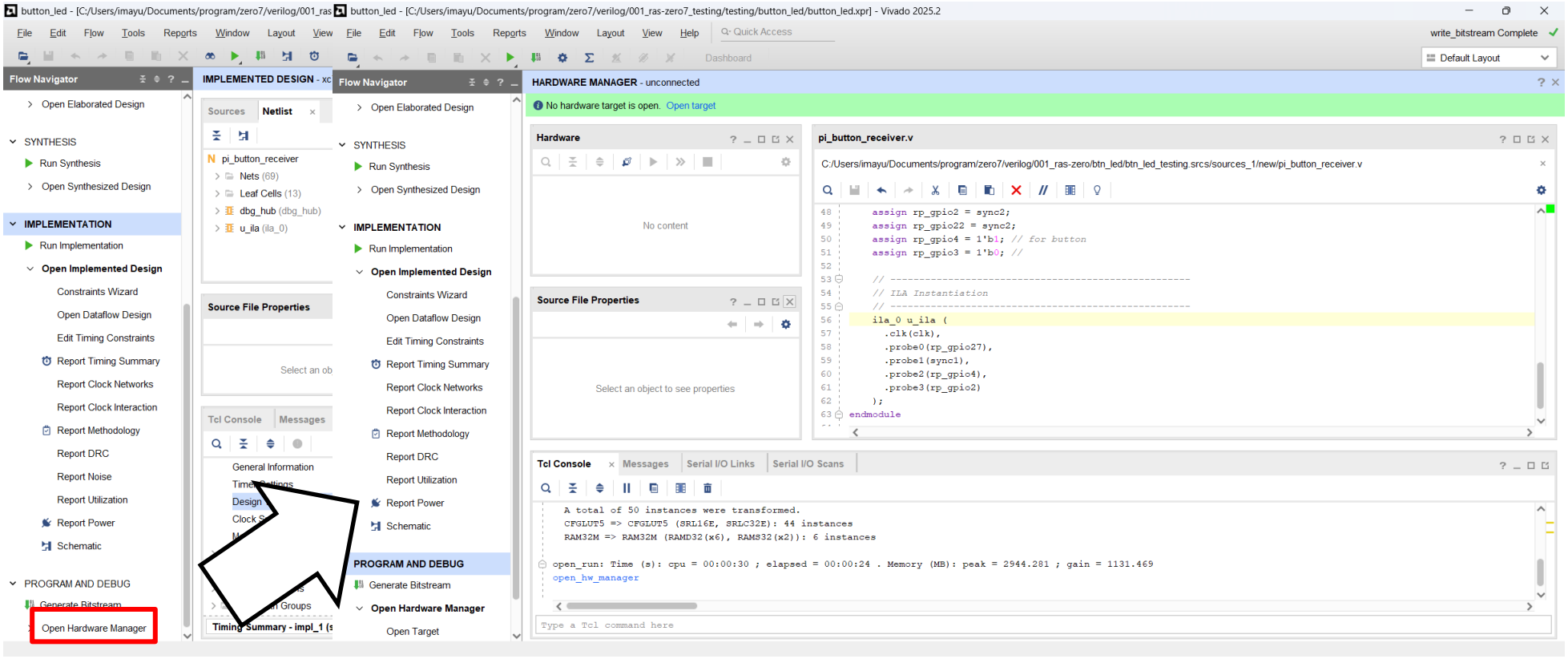




Copyrights 2026 DSP Laboratory. All Rights Reserved.

Vivavoプロジェクトの作成 | ZERO7へbitファイルのアップロード1

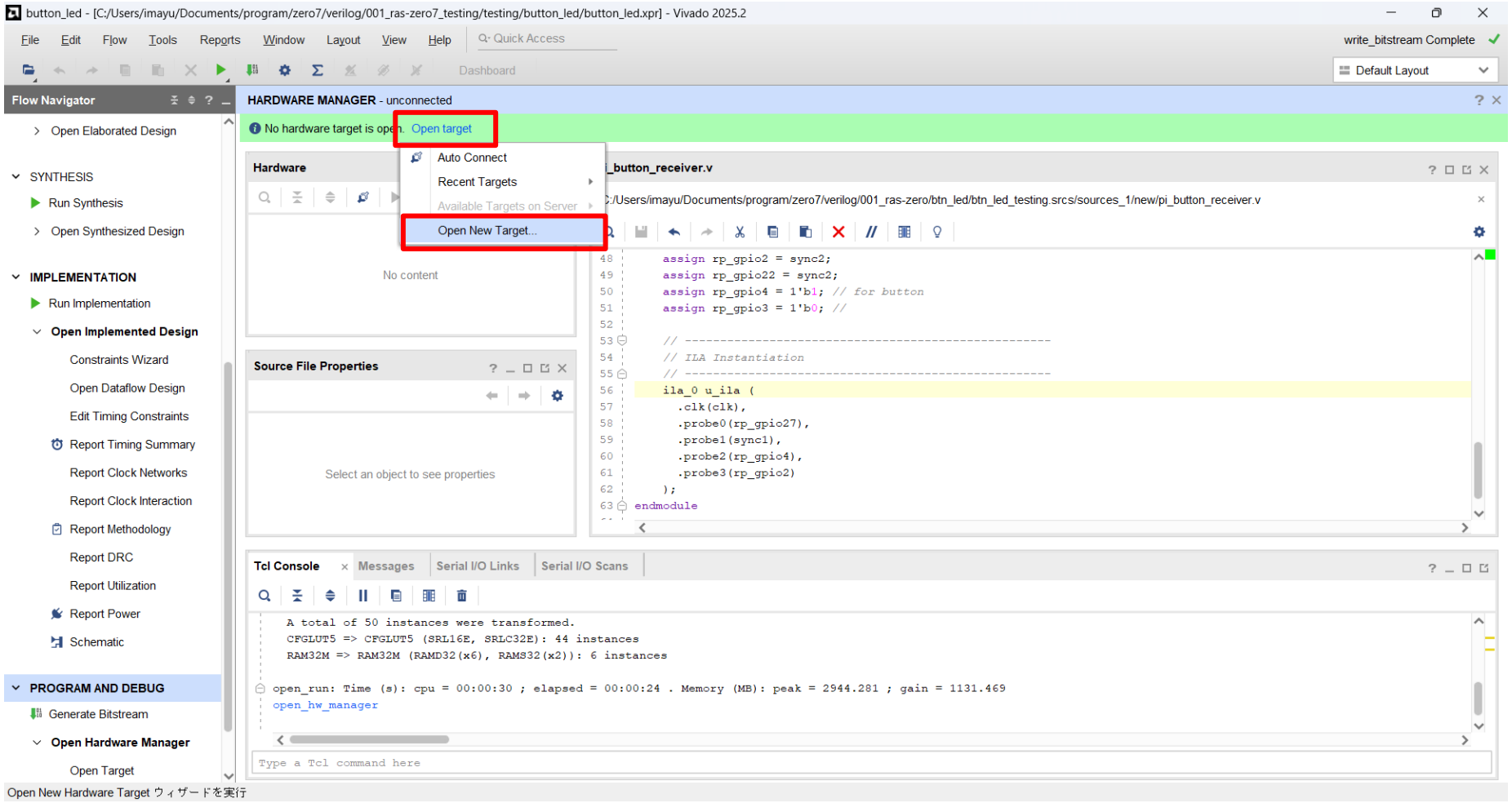
- “Open Hardware”でZERO7と接続
  - 再度Vivadoに戻る
  - “Generate Bitstream”の下の “Open Hardware”を選択する
  - “Hardware Manager”が開く



Copyrights 2026 DSP Laboratory. All Rights Reserved.

Vivavoプロジェクトの作成 | ZERO7へbitファイルのアップロード2

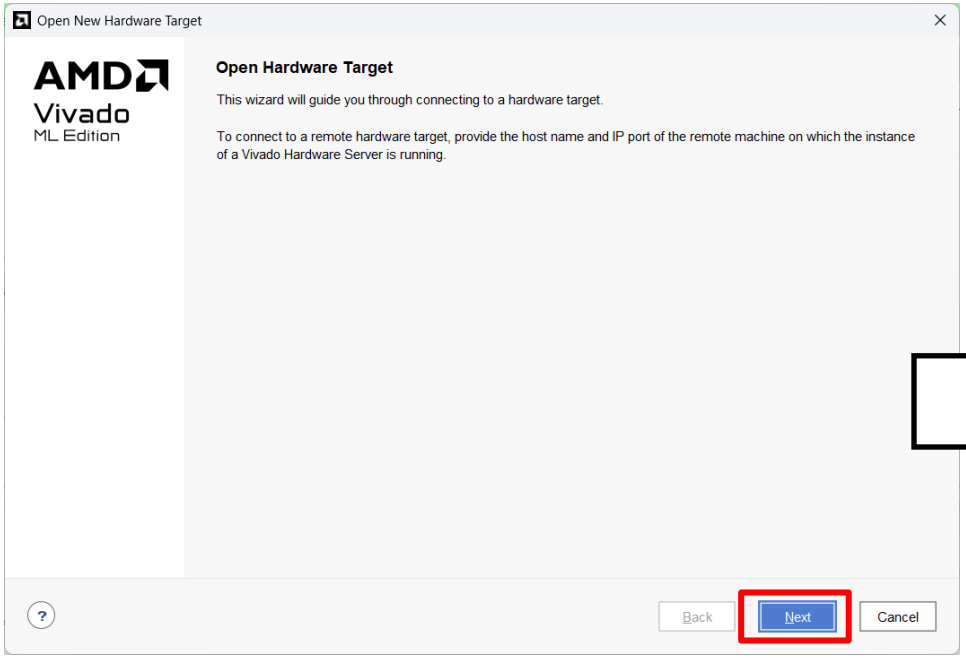
- “Open Hardware”でZERO7と接続
  - “Open target”を選択後，“Open New Target”を選択

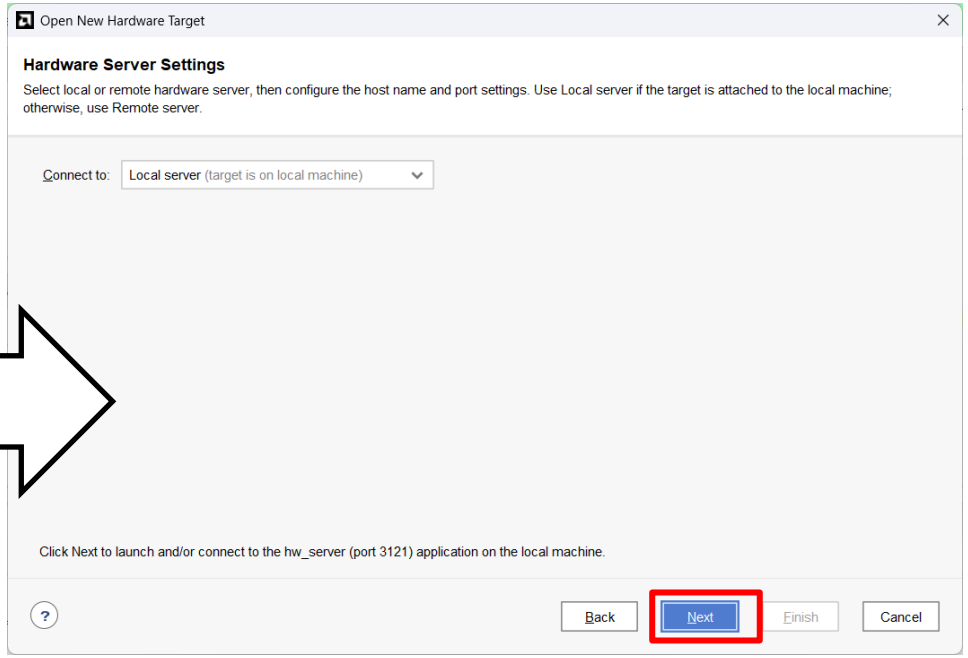


Copyrights 2026 DSP Laboratory. All Rights Reserved.

Vivavoプロジェクトの作成 | ZERO7へbitファイルのアップロード3

- “Open Hardware”でZERO7と接続
  - “Open New Target”を選択後，下記の画面が表示される
  - “Open Hardware Target”では，“Next”を押す
  - “Hardware Server Setting”では，“Local server”選択後，“Next”を選択

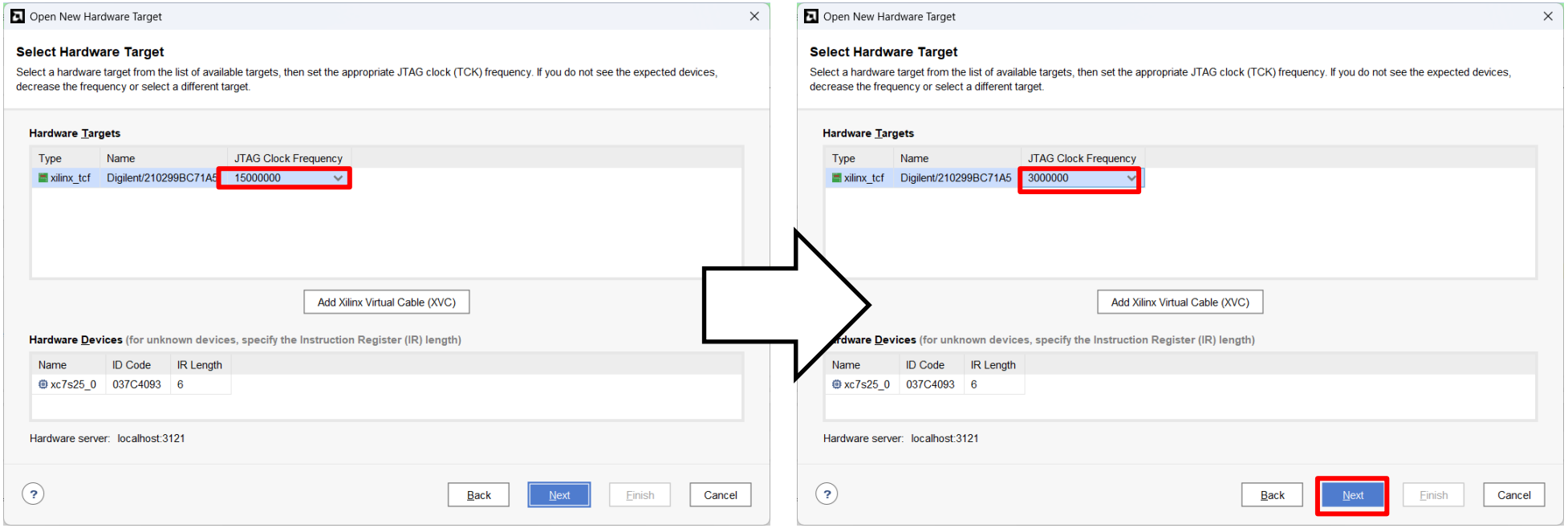




Copyrights 2026 DSP Laboratory. All Rights Reserved.

Vivavoプロジェクトの作成 | ZERO7へbitファイルのアップロード4

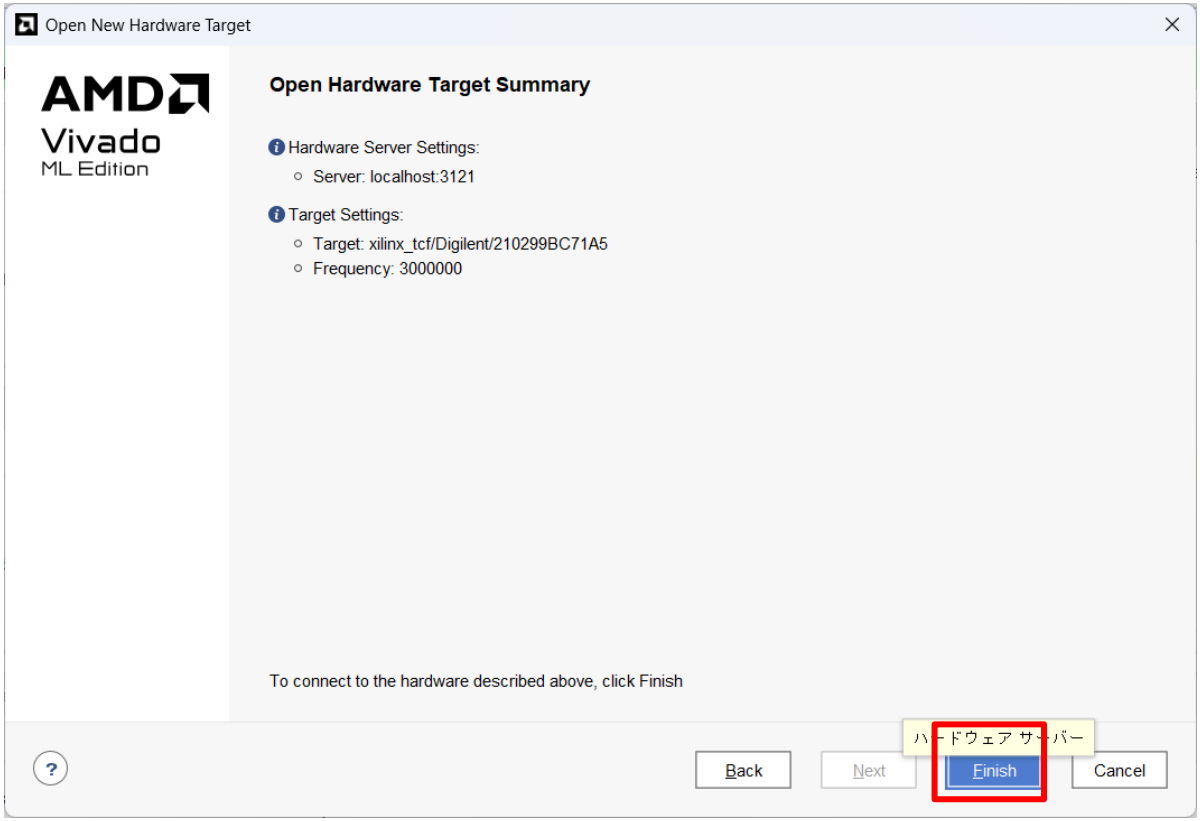
- “Open Hardware”でZERO7と接続
  - “Setting Hardware Target”において，JTAGクロック速度を設定する
  - 安定させるために，遅めの3,000,000に設定
  - 設定してから，“Next”を選択



Copyrights 2026 DSP Laboratory. All Rights Reserved.

Vivavoプロジェクトの作成 | ZERO7へbitファイルのアップロード5

- “Open Hardware”でZERO7と接続
  - サマリーが下記のようになっていれば適切に接続は可能

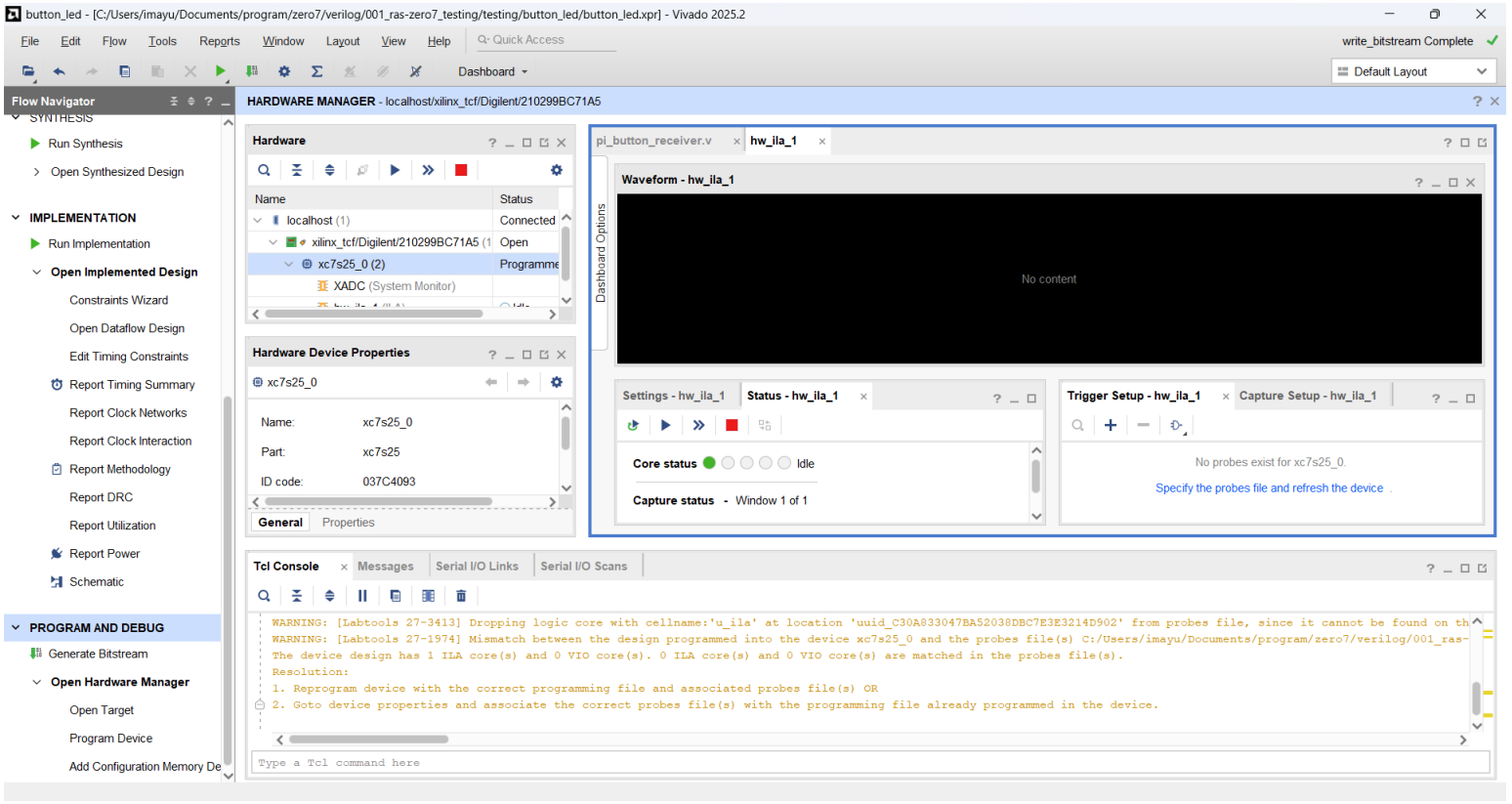


Copyrights 2026 DSP Laboratory. All Rights Reserved.

Vivavoプロジェクトの作成 | ZERO7へbitファイルのアップロード6

■ 接続の確認

□ 接続が正常にできると，下記のような画面に遷移する

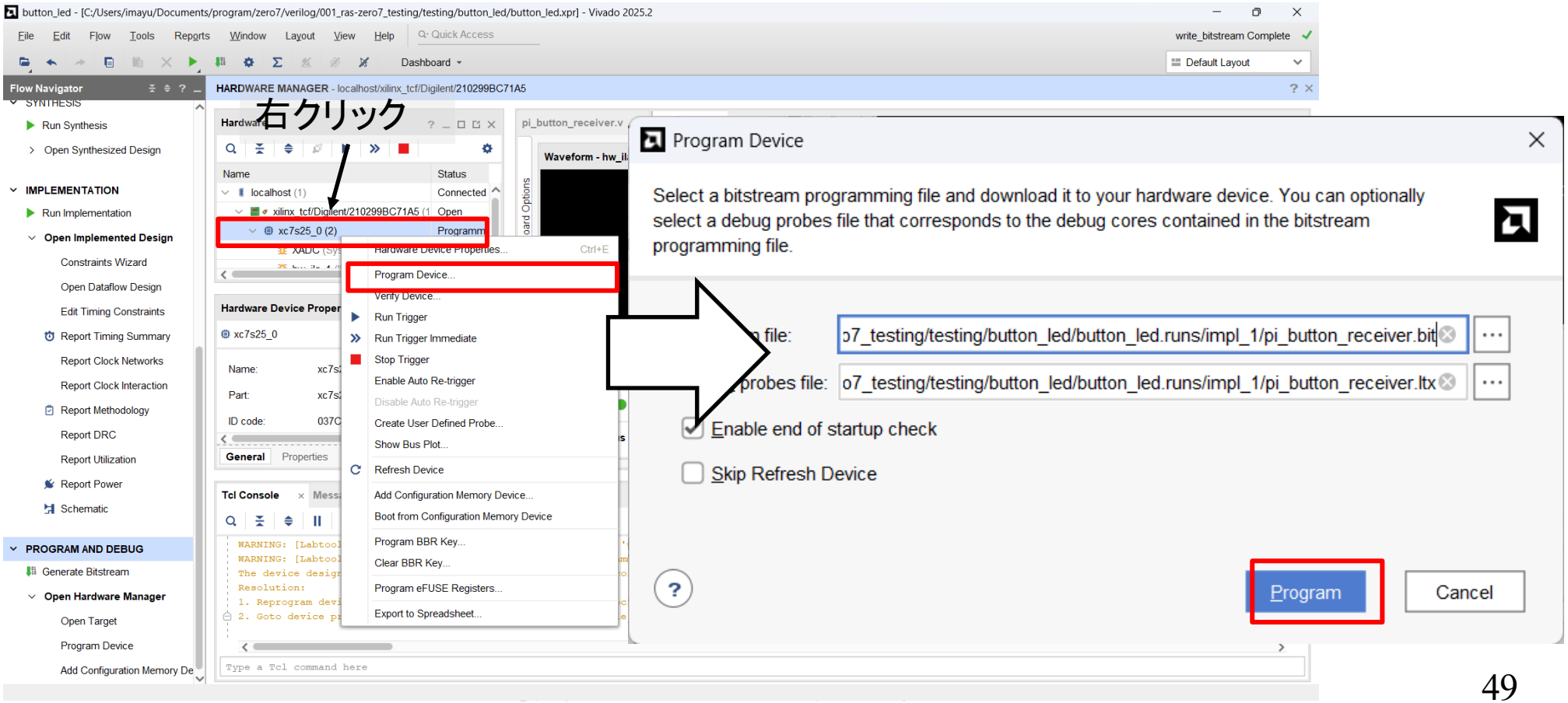


Copyrights 2026 DSP Laboratory. All Rights Reserved.

Vivavoプロジェクトの作成 | ZERO7へbitファイルのアップロード7

Bitファイルのアップロード

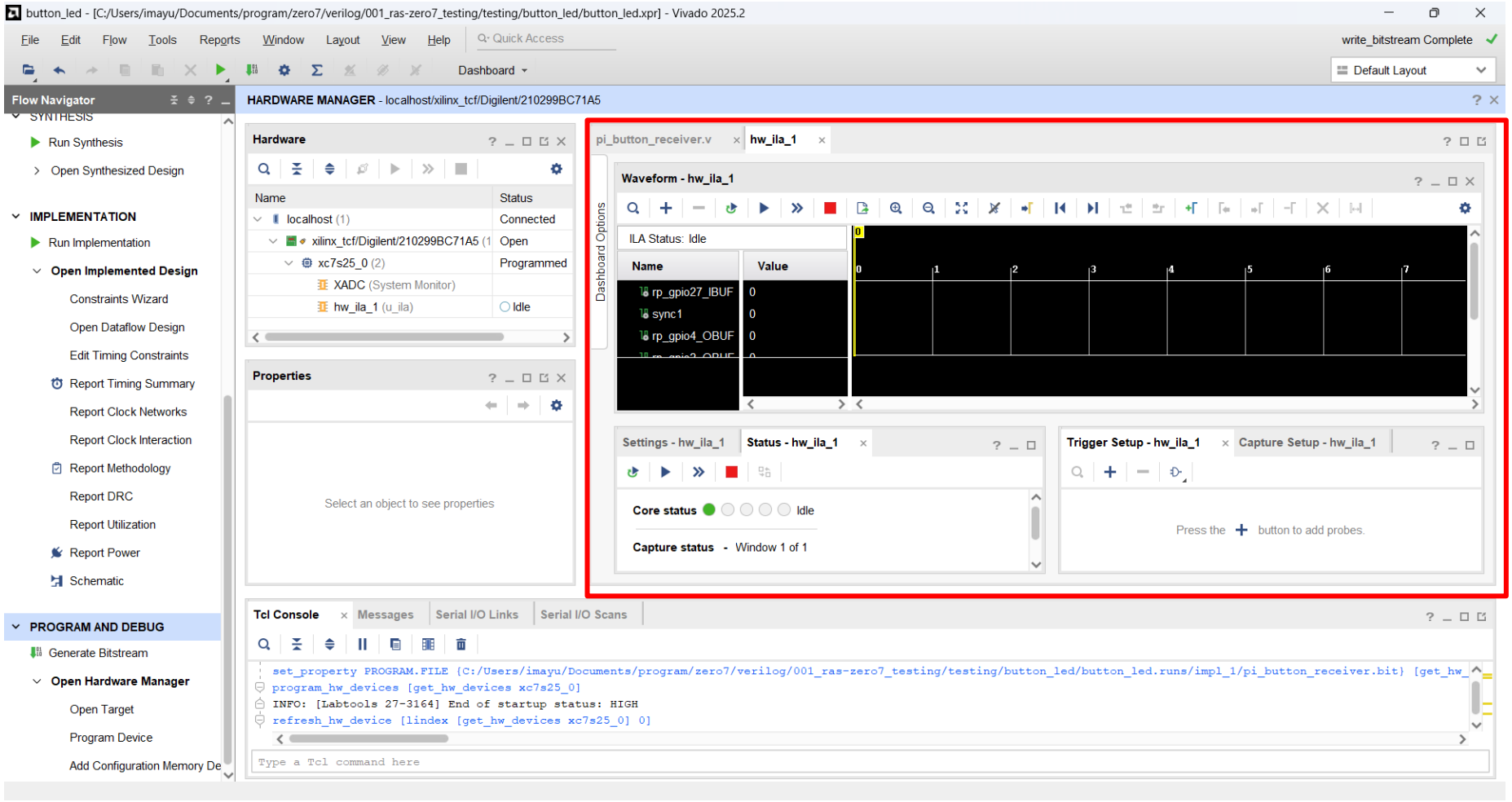
- “xs7s25”を右クリック→ “Program Device”を選択
- Bitファイル及びILA用のファイル（ltxファイル）
- “Program”を選択



Vivavoプロジェクトの作成 | ZERO7へbitファイルのアップロード8

■ Bitファイルのアップロードの確認

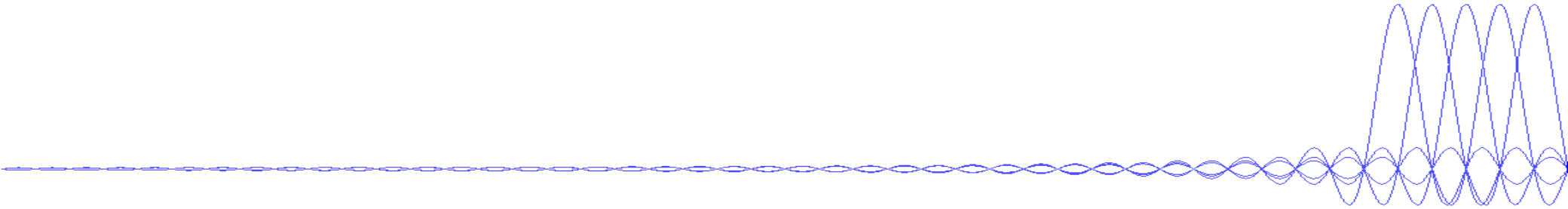
□ILAの画面が更新されるのが確認できる



Copyrights 2026 DSP Laboratory. All Rights Reserved.

Raspberry Pi & ZERO7

LED lighting



Copyrights 2026 DSP Laboratory. All Rights Reserved.

開発の流れ

- 回路の作成
- 実行を行う
- ILAの確認

Copyrights 2026 DSP Laboratory. All Rights Reserved.

今回導入したプログラムの仕様

今回導入したプログラムの概要

- Raspberry Pi側でボタンの入力を受け付け，その出力をZERO7に与える
- ZERO7はその入力を受け付けると，2段階同期をかけてLED点灯として出力する

詳細1

- Raspberry Pi
  - GPIO 4番でボタンの入出力を受け付ける
  - コンソール側出力
    - ボタンが押されていない → ‘False’
    - 場合ボタンが押された → ‘True’
  - ZERO7への出力は，GPIO 27番を使用する
- ZERO7
  - Raspberry Pi からの入力はGPIO 27番を使用する
  - 入力は2段階同期をかけてFPGA内部に取り込む
  - 出力として，GPIO 2番，GPIO 22番を立ち上げる
  - input

input/output	type	variable	Comments
input	wire	clk	
input	wire	rp_gpio27	From Rasp.
output	wire	rp_gpio2	For LED
output	wire	rp_gpio22	For LED
output	wire	rp_gpio3	
output	wire	rp_gpio4	For Button

Copyrights 2026 DSP Laboratory. All Rights Reserved.

仕様

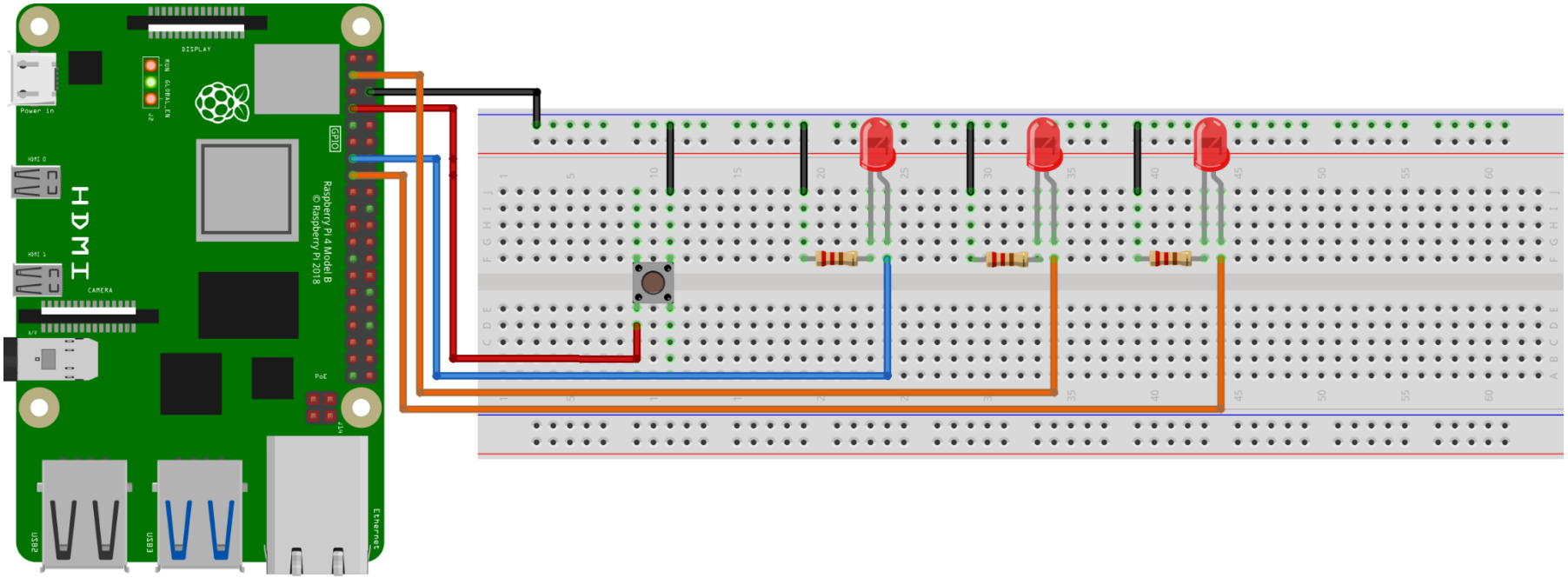
- 詳細2
  - DEBAG
    - ILAでの入力 GPIO 27番および出力 GPIO 22番の追跡を行う
    - GPIO 27番の出力をLEDに接続する

- 処理例
  - ボタンを押すと
    - ZERO 7からの出力としてLEDが2つ**同時**に点灯する
    - DEBAGとしてLEDが1つ点灯する

Copyrights 2026 DSP Laboratory. All Rights Reserved.

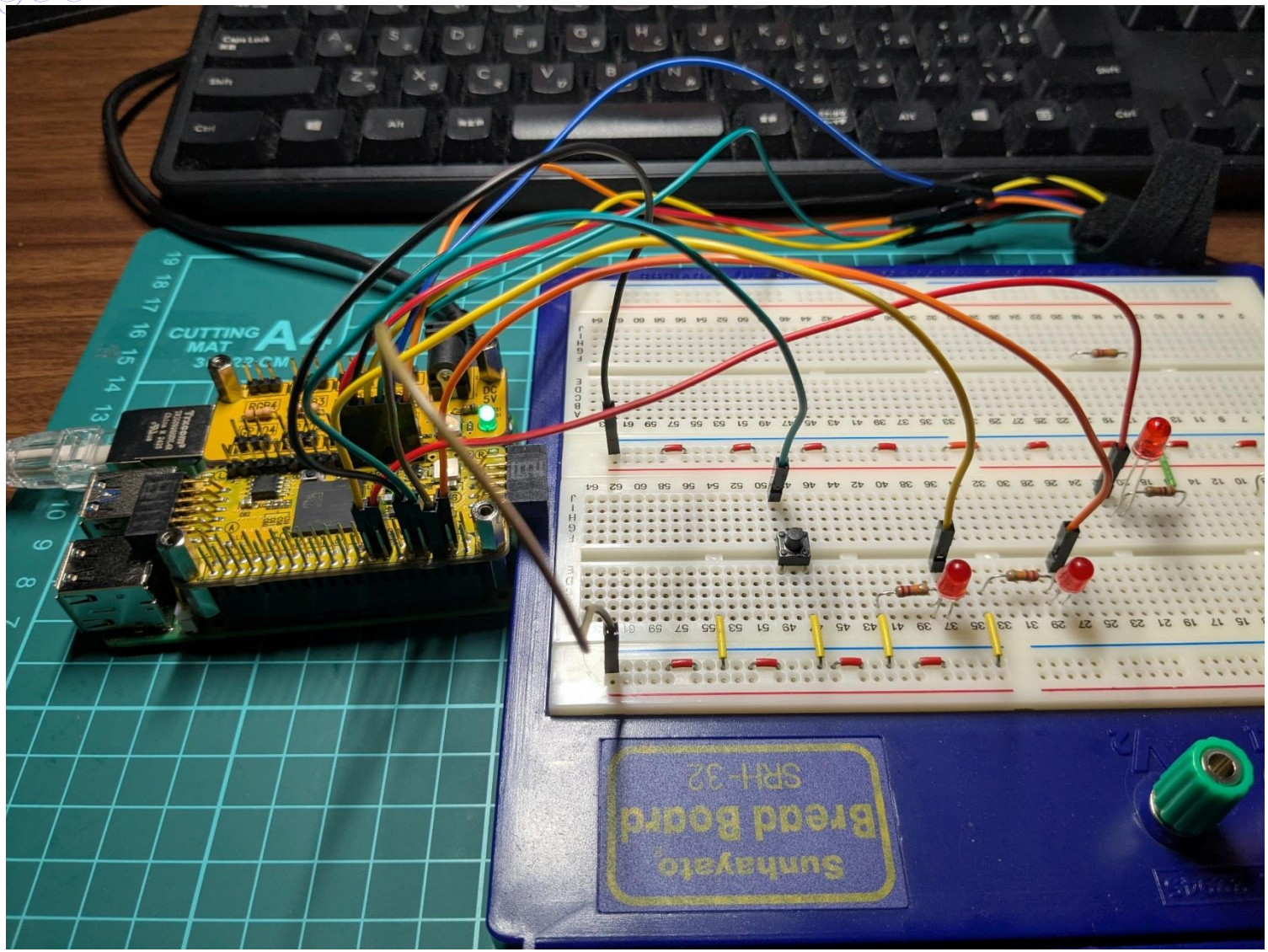
回路図

- 下記のようにブレッドボードとGPIOを接続する
  - 赤線：ボタン用
  - オレンジ線：ZERO7から出力用
  - 青線：Raspberry Pi → ZERO7への信号送信デバッグ用
  - 黒線：GND



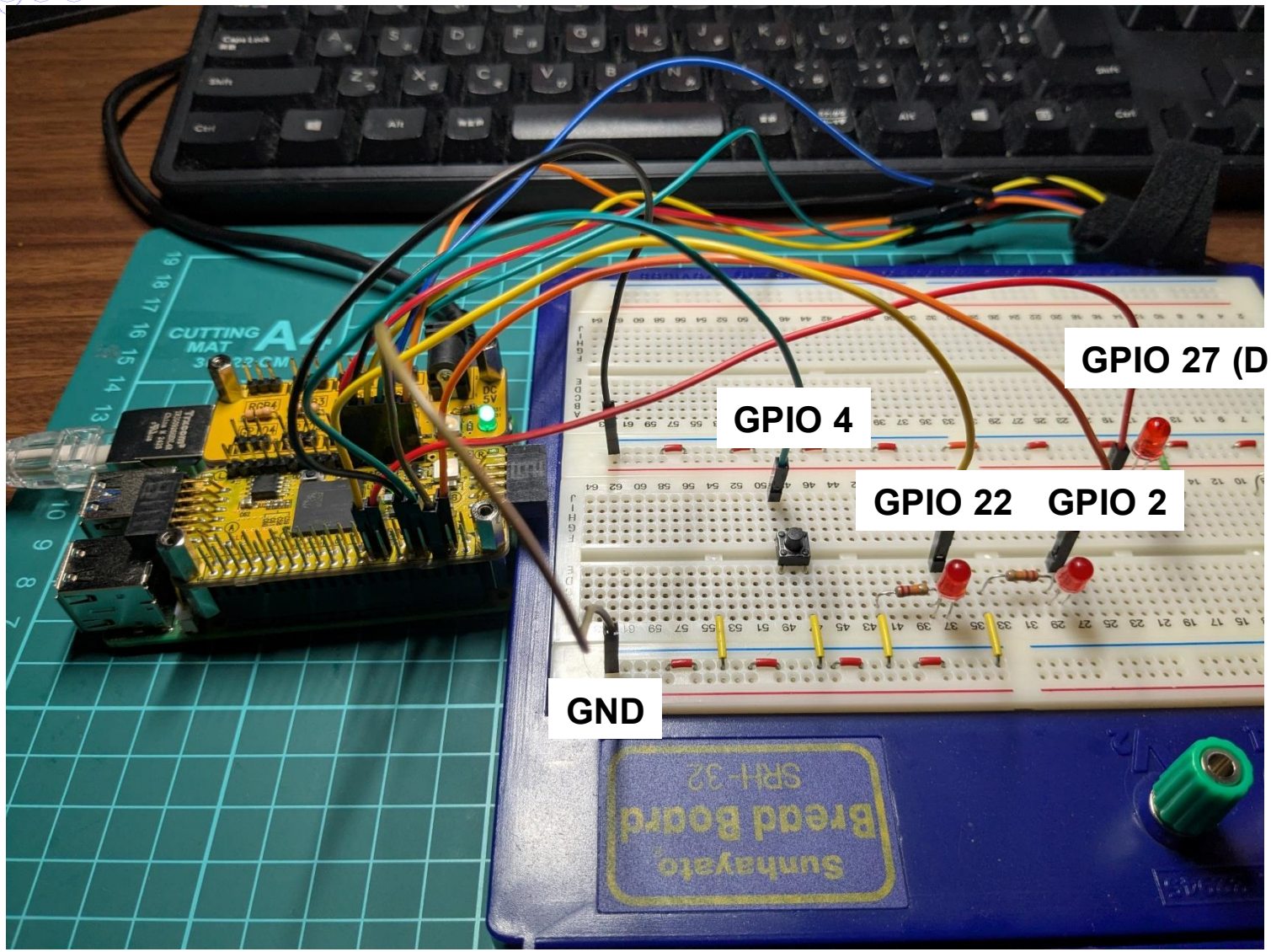
Copyrights 2026 DSP Laboratory. All Rights Reserved.

回路図



Copyrights 2026 DSP Laboratory. All Rights Reserved.

回路構成



Copyrights 2026 DSP Laboratory. All Rights Reserved.

結果

- ILAの出力
  - rp_gpio27_IBUF:　GPIO 27番を使ってZERO7へのRaspberry Piからの入力
  - rp_gpio2_IBUF:　GPIO 2番を使ってZERO7からの出力
  - 2段階同期回路が入っているので，入力から出力まで遅延が数クロック入っている

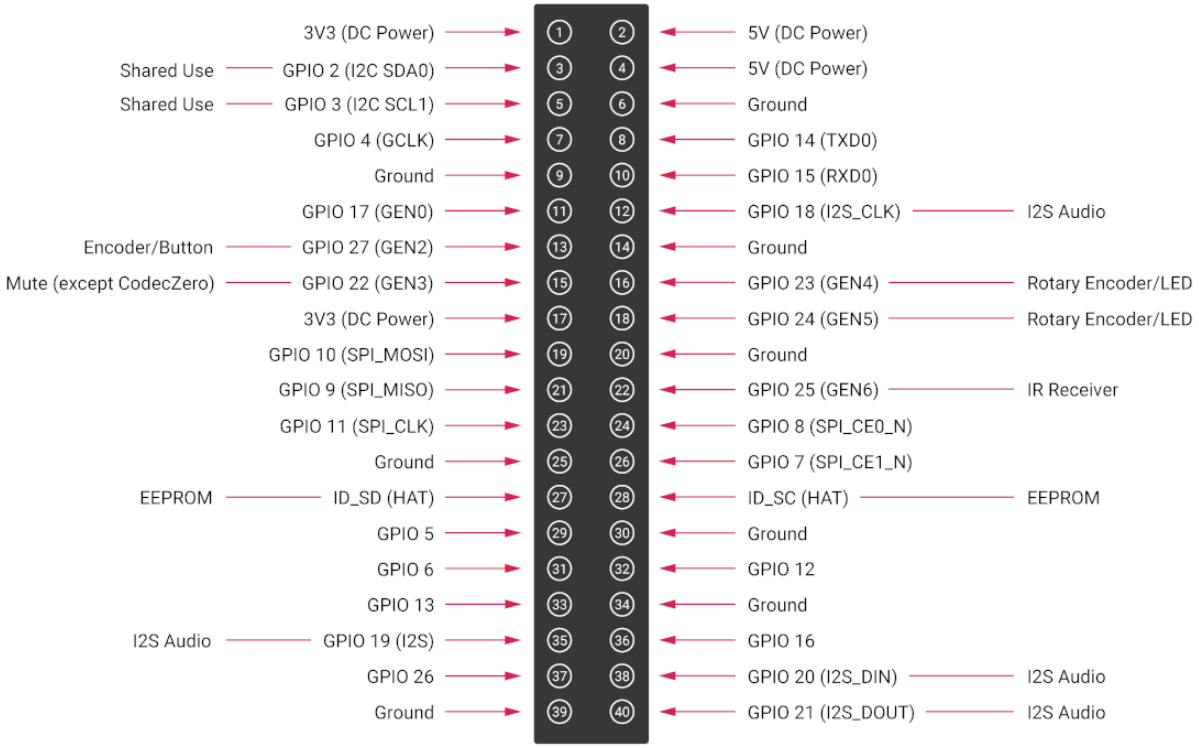


Copyrights 2026 DSP Laboratory. All Rights Reserved.

Appendix: GPIOとピンの配置

GPIOの関係

- Raspberry PiのGPIO番号とピン番号割り当てを下記に記載する.



参照: https://www.raspberrypi.com/documentation/accessories/audio.html#gpio-usage

Copyrights 2026 DSP Laboratory. All Rights Reserved.