RISC-V LED Shell on Arty Z7 – Quick Build & Run Guide

This guide explains how to build, program, and run the **RISC-V LED Shell** project on the **Digilent Arty Z7-20** board. The project demonstrates a PicoRV32-based SoC with UART and optional JTAG console for interactive LED control.

1. Folder Overview

Folder / File	Purpose
src/	All HDL source files (top.v, uart_core.v, jtag_uart.v, picorv32.v, rom_init.mem)
fw/	Firmware (uart_shell_led.c, Makefile)
constrs/	Constraint files for Arty Z7 pin mapping
create_project.tcl	Creates Vivado project automatically
deploy.tcl	Builds, implements, and programs FPGA
xsdb_listen.tcl	JTAG UART live console script
README.md	Documentation and usage notes

2. Build the Firmware

1. Open a terminal and navigate to the `fw/` folder.
2. Run:
make 3. Copy the generated ROM image to the source directory:
cp rom_init.mem ../src/

3. Build and Program the FPGA

1. From the project root, run:
vivado -mode batch -source deploy.tcl
This will automatically create the Vivado project, synthesize, implement, and program the FPGA.

4. Console Options

You can use either a physical UART or JTAG UART console:
- **UART mode**: Connect a USB-to-UART dongle to PMOD JD (pins JD1/JD2) and open a terminal at 115200 bps.
- **JTAG UART mode**: No dongle needed. Add this define in Vivado:
set_property verilog_define {USE_JTAG_UART} [current_fileset]
Then run:
xsdb xsdb_listen.tcl

5. Commands Available

Command	Description

help	Show available commands
on n	Turn on LED number n (0–7)
off n	Turn off LED number n
toggle n	Toggle LED number n
blink	Blink all LEDs once
status	Show current LED state

6. Example Session

After programming, open your console (UART or JTAG) and type:
=== RISC-V LED Shell === Type 'help' for commands. > help > on 3 > status LED state: 00001000

7. Notes

- This project targets **Arty Z7-20** (Zynq-7000) but can be adapted to other boards.
- Firmware updates only require rebuilding rom_init.mem and re-running the Tcl script.
- All scripts are Vivado 2023.2+ compatible.