

Zephyr Project Setup Guide (Windows – PowerShell)

Follow these simple steps to set up your Zephyr environment on Windows using PowerShell.

Step 1: Open a Terminal

- Open **PowerShell** or **Command Prompt** (cmd.exe) as a **regular user** (not as Administrator).
-

Step 2: Navigate to Your Project Folder & Create a Virtual Environment

```
cd "C:\Users\Vantage\Documents\Zephyr_Project_0"  
python -m venv .venv
```

The **.venv** folder will contain your **Python virtual environment**.

Step 3: Activate the Virtual Environment

```
(.venv) PS C:\Users\Vantage\Documents\Zephyr_Project_0> .\.venv\Scripts\Activate.ps1
```

Important: You need to **activate** the **virtual environment** every time you open a new terminal and work on this project.

Step 4: Install West (Zephyr's project manager tool)

Once the virtual environment is activated, install West with:

```
(.venv) PS C:\Users\Vantage\Documents\Zephyr_Project_0> pip install west
```

Step 5: Initialize Zephyr in the Workspace

Make sure you're in the correct folder:

```
cd "C:\Users\Vantage\Documents\Zephyr_Project_0"
```

Then initialize Zephyr project named as **Zephyr-Src** and fetch Zephyr source code:

```
west init Zephyr-Src
```

Step 6: Organize the Folder Structure

Organize the Folder Structure

```
Zephyr_Project_0
|
├── Zephyr-Src
│   ├── zephyr
│   ├── modules
│   ├── .west
│   └── ...
└──
```

You need to **move the contents inside Zephyr-Src** one level up into the **Zephyr_Project_0** folder. After moving, it should look like:

```
Zephyr_Project_0
|
├── zephyr
├── modules
├── .west
├── .venv
└── ...
```

Then you can **delete the empty Zephyr-Src** folder.

Step 6: West Update

Make sure you're in the correct folder.
Then update Zephyr source code.

```
cd "C:\Users\Vantage\Documents\Zephyr_Project_0"

west update
```

Step 7: Export Zephyr CMake Package

Export Zephyr's CMake package. This enables CMake to locate and load the required boilerplate automatically when building applications

```
west zephyr-export
```

Step 8: Install Python Dependencies from Zephyr Packages

Use the **west packages** command to install all Python requirements defined by Zephyr modules.

```
west packages pip --install
```

Step 9: Install the Zephyr SDK

The **Zephyr Software Development Kit (SDK)** includes compilers, linkers, debuggers, and other essential tools for building and debugging applications across Zephyr's supported architectures.

To install the SDK on Windows using West:

```
cd "C:\Users\Vantage\Documents\Zephyr_Project_0\zephyr"  
west sdk install
```

Step 10: Build the Blinky Sample

To view the list of supported boards, run.

```
west boards
```

Once you've identified your board, you can build the Blinky example application. Replace **<your-board-name>** with your actual board identifier:

```
cd "C:\Users\Vantage\Documents\Zephyr_Project_0\zephyr"  
west build -p always -b stm32mp157c_dk2 samples\basic\blinky
```

```
west build -p always -b stm32mp157c_dk2 samples\basic\blinky
```

```

C:\
└─ Users\
    └─ <YourUsername>\
        └─ zephyrproject\
            ├── .venv\           # Python virtual environment
            │   ├── Scripts\
            │   ├── Lib\
            │   └─ ...
            ├── .west\          # West workspace configuration
            ├── zephyr\         # Zephyr RTOS source code
            │   ├── cmake\
            │   ├── boards\
            │   ├── drivers\
            │   ├── include\
            │   ├── kernel\
            │   └─ ...
            ├── modules\        # External modules (e.g., hal, lib)
            ├── tools\          # Additional tools (like west scripts)
            └─ west.yml          # Manifest file for west

```

