Zephyr RTOS Setup & Hello World Guide

📋 Overview

This guide walks you through setting up the Zephyr development environment on Ubuntu Linux and running your first 'Hello World' and 'Blinky' samples on an nRF52832 DK board. It includes a setup for debugging with VS Code and serial output via Minicom.

🛠️ Prerequisites

- Ubuntu 22.04

- nRF52832 Development Kit (DK)

- USB cable

- Internet access

✅ Step 1: Install Required Packages

Open a terminal and run:

```
sudo apt update
sudo apt upgrade
sudo apt install terminator net-tools minicom gdb-multiarch
```

📘 Step 2: Follow Zephyr Getting Started Guide

Go to: https://docs.zephyrproject.org/latest/develop/index.html

Follow all steps up to the SDK installer (**not included**).

📦 Step 3: Install Zephyr SDK and Toolchain

Run the following commands:

```
cd ~/zephyrproject/zephyr
west sdk install -t arm-zephyr-eabi
```

🖥️ Step 3.5: Install VS Code and Extensions

Download and install VS Code:

```
cd ~/Downloads
wget
```

```
https://update.code.visualstudio.com/latest/linux-deb-x64/stable -O vscode.deb
sudo dpkg -i vscode.deb
```

Launch VS Code and install these extensions:

- Cortex-Debug
- C/C++ (by Microsoft)

🔧 Step 3.6: Install SEGGER J-Link Debugger Tools

Download and install J-Link tools:

```
wget https://www.segger.com/downloads/jlink/JLink_Linux_V840_x86_64.deb
sudo dpkg -i ~/Downloads/JLink_Linux_V840_x86_64.deb
```

🧪 Step 4: Build and Flash Hello World

Create and enter a new Zephyr app folder:

```
cd ~/zephyrproject/zephyr
mkdir hello_world && cd hello_world
```

Create the following files:

src/main.c

```
#include <zephyr/kernel.h>
void main(void) {
 printk("Hello World from Zephyr!\n");
}
```

prj.conf

```
CONFIG_PRINTK=y
```

CMakeLists.txt

```
cmake_minimum_required(VERSION 3.20.0)
find_package(Zephyr REQUIRED HINTS $ENV{ZEPHYR_BASE})
project(hello_world)
target_sources(app PRIVATE src/main.c)
```

Build and flash:

```
west build -p always -b nrf52dk/nrf52832
west flash
```

🔍 Step 4.1: Verify Serial Output with Minicom

Open terminal and run:

```
sudo minicom -D /dev/ttyACM0
```

🔨 Step 5: Build and Flash Blinky Sample

```
cd ~/zephyrproject/zephyr
west build -b nrf52dk/nrf52832 samples/basic/blinky
west flash
```

💡 Step 6: Debug with VS Code

Create the following file in the relevant folder: `.vscode/launch.json`

```
{
 "configurations": [
 {
 "cwd": "${workspaceFolder}",
 "executable": "${workspaceFolder}/build/zephyr/zephyr.elf",
 "name": "Debug with JLink",
 "request": "launch",
 "type": "cortex-debug",
 "device": "NRF52832_xxAA",
 "servertype": "jlink",
 "runToEntryPoint": "main",
 "armToolchainPath": "${HOME}/zephyr-sdk-0.17.1/arm-zephyr-
eabi/bin/",
```

```
 "gdbPath": "/usr/bin/gdb-multiarch"
 }
 ]
}
```

Add the following in file /home/user/.config/Code/User/settings.json:

```
{

   "editor.minimap.enabled": false,

   "cortex-debug.armToolchainPrefix": "arm-zephyr-eabi",

   "cortex-debug.JLinkGDBServerPath": "/usr/bin/JLinkGDBServer"

}
```

in prj.conf add the following:

```
CONFIG_DEBUG_OPTIMIZATIONS=y
```

Try to debug (add breadpoints) the code (:

🔨 Step 7: Build and Flash Blinky Sample on different board

Now lets build and run on different board

```
cd ~/zephyrproject/zephyr
west build -b nrf52840dk/nrf52840 samples/basic/blinky
west flash
```