Zephyr RTOS Setup Guide

This repository is now configured with Zephyr RTOS and the minimal SDK for ARM development.

What Has Been Installed

1. System Dependencies

All required packages for Zephyr development have been installed, including:

- Build tools: `cmake`, `ninja-build`, `gperf`, `ccache`
- Device tools: `device-tree-compiler`, `dfu-util`
- Development libraries: `python3-dev`, `libsdl2-dev`

2. West Meta-Tool

- **Version**: v1.5.0
- **Location**: `~/.local/bin/west`
- West is Zephyr's meta-tool for managing repositories and building projects

3. Zephyr Repository

- **Location**: `/workspaces/LearnZephyrRTOS/zephyr/`
- The main Zephyr RTOS repository and all its modules have been cloned and initialized
- All dependencies have been pulled using `west update`

4. Python Dependencies

All Python packages required for Zephyr development have been installed, including:

- Build tools: `pyelftools`, `gcovr`, `pytest`
- Development tools: `mypy`, `pylint`, `ruff`
- Hardware interfaces: `pyocd`, `pyserial`, `pylink-square`

5. Zephyr SDK (ARM Only)

- **Version**: 0.17.4
- **Location**: `/workspaces/LearnZephyrRTOS/zephyr-sdk-0.17.4/`
- **Toolchain**: ARM (`arm-zephyr-eabi`) only - optimized for ARM Cortex-M development
- CMake package has been registered at: `~/.cmake/packages/Zephyr-sdk`

Directory Structure

```
/workspaces/LearnZephyrRTOS/
├── zephyr/                          # Main Zephyr RTOS source
├── zephyr-sdk-0.17.4/               # Zephyr SDK
│   └── arm-zephyr-eabi/             # ARM toolchain
├── modules/                         # Zephyr modules (HALs, libraries)
```

```
├── zephyrproject/                  # Your applications
│   └── apps/
│       └── 1_blinky/               # Example blinky application
├── setup_environment.sh            # Environment setup script
└── ZEPHYR_SETUP_GUIDE.md          # This file
```

Environment Setup

To set up your environment for each terminal session:

```
source /workspaces/LearnZephyrRTOS/setup_environment.sh
```

This sets:

- `ZEPHYR_BASE=/workspaces/LearnZephyrRTOS/zephyr`
- `ZEPHYR_SDK_INSTALL_DIR=/workspaces/LearnZephyrRTOS/zephyr-sdk-0.17.4`

Building Your First Application

Using the Existing Blinky Example

```
# Set up environment
source /workspaces/LearnZephyrRTOS/setup_environment.sh

# Navigate to your app directory
cd /workspaces/LearnZephyrRTOS/zephyrproject/apps/1_blinky

# Build for a specific board (example: STM32F429I Discovery)
west build -b stm32f429i_disc1

# Or for QEMU ARM
west build -b qemu_cortex_m3
```

Creating a New Application

```
# Create a new application directory
mkdir -p /workspaces/LearnZephyrRTOS/zephyrproject/apps/my_app
cd /workspaces/LearnZephyrRTOS/zephyrproject/apps/my_app

# Create minimal CMakeLists.txt
cat > CMakeLists.txt << 'EOF'
cmake_minimum_required(VERSION 3.20.0)
find_package(Zephyr REQUIRED HINTS $ENV{ZEPHYR_BASE})
project(my_app)

target_sources(app PRIVATE src/main.c)
```

```
EOF

# Create prj.conf
cat > prj.conf << 'EOF'
# Configuration options
CONFIG_PRINTK=y
CONFIG_SERIAL=y
EOF

# Create source directory and main.c
mkdir src
cat > src/main.c << 'EOF'
#include <zephyr/kernel.h>
#include <zephyr/sys/printk.h>

int main(void)
{
    printk("Hello Zephyr!\n");
    return 0;
}
EOF

# Build the application
west build -b qemu_cortex_m3
```

Common West Commands

```
# Build an application
west build -b <board_name>

# Clean build directory
west build -t clean

# Flash to hardware (when connected)
west flash

# Run in QEMU emulator
west build -t run

# List all supported boards
west boards

# Update Zephyr and modules
west update

# Get help
west --help
```

Supported ARM Boards (Examples)

Since you have the ARM toolchain installed, you can build for these boards:

STM32 Boards

- `stm32f429i_disc1` - STM32F429I Discovery
- `stm32f4_disco` - STM32F4 Discovery
- `nucleo_f429zi` - STM32 Nucleo-144
- `nucleo_f767zi` - STM32 Nucleo-144

Nordic Semiconductor

- `nrf52840dk_nrf52840` - nRF52840 DK
- `nrf5340dk_nrf5340_cpuapp` - nRF5340 DK

QEMU (For Testing)

- `qemu_cortex_m3` - QEMU ARM Cortex-M3
- `qemu_cortex_m0` - QEMU ARM Cortex-M0

To see all available boards:

```
west boards | grep arm
```

Testing the Installation

Build and run a simple test:

```
# Set up environment
source /workspaces/LearnZephyrRTOS/setup_environment.sh

# Build hello_world sample for QEMU
cd /workspaces/LearnZephyrRTOS/zephyr
west build -b qemu_cortex_m3 samples/hello_world

# Run in QEMU (Ctrl+A then X to exit)
west build -t run
```

Additional Resources

- **Zephyr Documentation**: https://docs.zephyrproject.org/
- **Getting Started Guide**: https://docs.zephyrproject.org/latest/getting_started/
- **Board Support**: https://docs.zephyrproject.org/latest/boards/index.html
- **West Tool**: https://docs.zephyrproject.org/latest/guides/west/

Troubleshooting

West not found

```
export PATH="$HOME/.local/bin:$PATH"
```

SDK not found

Ensure the environment variables are set:

```
source /workspaces/LearnZephyrRTOS/setup_environment.sh
```

Build fails

1. Make sure you've run `west update` at least once
2. Verify the board name with `west boards`
3. Check that CMakeLists.txt and prj.conf exist in your app directory

Next Steps

1. Read the existing guides in this repository:
   - `Zephyr_Devicetree_Overlay_Guide.md`
   - `Zephyr_Project_Configuration_Guide.md`
2. Explore sample applications:

   ```
   ls /workspaces/LearnZephyrRTOS/zephyr/samples/
   ```

3. Build the existing blinky application for your target board
4. Start developing your own embedded applications!

Setup completed on: $(date) **Zephyr SDK Version**: 0.17.4 **West Version**: 1.5.0 **Toolchains Installed**: ARM (arm-zephyr-eabi)