Zephyr RTOS – Quick-Start Porting Guide for Silicon Labs BGM220P (EFR32BG22)

> **Audience** Embedded developers who already have an up-to-date Zephyr workspace and want to bring up a BGM220P module (or the Silabs SLTB010A eval board) quickly, building **LED Blink** and **Hello World-on-Button-Press** samples, and flashing them with a SEGGER J-Link probe.

---

0 · Prerequisites

Requirement	Notes
Host OS	Ubuntu 22.04-based (any Linux distribution works)
Toolchain	Zephyr SDK ≥ 0.17.2 already installed and in PATH
Zephyr repo	Workspace at ~/zephyrproject (created via west init/update)
J-Link hardware	USB-connected to the SLTB010A board (or to the BGM220P via SWD pins)
udev access	User belongs to the plugdev group **or** correct J-Link udev rules are installed

---

1 · Verify Your Workspace (optional)

```
# Build the stock LED blink for an BGM220P board to ensure toolchain works
west build -b sltb010a -s zephyr/samples/basic/blinky -d build_bg22 --pristine
```

---

2 · Build & Flash LED Blink on BGM220P

2.1 Compile

```
west build \
  -b sltb010a \              # SLTB010A eval-board alias for EFR32BG22/BGM220P
  -s zephyr/samples/basic/blinky \
  -d build_bg22 \
  --pristine
```

2.2 Flash (first attempt)

```
# Unsupported: OpenOCD cannot drive the Silabs J-Link MCU, so this fails
west flash -d build_bg22 --runner openocd   # ↯ expected to fail
```

2.3 Install SEGGER J-Link utilities

```
# 1· Grab the latest J-Link package (adjust version if needed)
wget https://www.segger.com/downloads/jlink/JLink_Linux_x86_64.deb

# 2· Update package cache & install dependencies
sudo apt update

# 3· Remove any conflicting old udev package (optional but recommended)
sudo apt remove segger-jlink-udev-rules   # only if previously installed

# 4· Install the new package
sudo dpkg -i Jlink_Linux_x86_64.deb

# Fix unmet deps if prompted
sudo apt-get install -f
```

2.4 Flash with J-Link

```
west flash -d build_bg22 --runner jlink         # default SWD, 4MHz

# --- OR, script JLinkExe directly ---
cd build_bg22/zephyr
JLinkExe -device EFR32BG22C224F512IM40 -if SWD -speed 4000 -autoconnect 1
```

flash.jlink ---on host

```
r
loadfile zephyr.hex
r
g
exit
```

> **Serial output:** Attach a terminal at **115 200 baud** on /dev/ttyACM0 (or similar) to see “LED Blink” startup logs.

---

3 · Create a Custom hello_app (button-triggered Hello World)

3.1 Skeleton layout

```
hello_app/
├── CMakeLists.txt   # minimal boilerplate
├── prj.conf         # enable console, button & LED drivers
└── src/
    └── main.c       # logic
```

CMakeLists.txt

```
cmake_minimum_required(VERSION 3.20.0)
find_package(Zephyr REQUIRED HINTS $ENV{ZEPHYR_BASE})
project(hello_app)
```

prj.conf

```
# 115200-baud console
CONFIG_CONSOLE=y
CONFIG_UART_CONSOLE=y

# GPIO drivers for button & LED aliases
CONFIG_GPIO=y
```

src/main.c

```
#include <zephyr/kernel.h>
#include <zephyr/drivers/gpio.h>
#include <zephyr/sys/printk.h>

#define LED   DT_ALIAS(led0)
#define BTN   DT_ALIAS(sw0)
static const struct gpio_dt_spec led  = GPIO_DT_SPEC_GET(LED, gpios);
static const struct gpio_dt_spec btn  = GPIO_DT_SPEC_GET(BTN, gpios);

void main(void)
{
   gpio_pin_configure_dt(&led, GPIO_OUTPUT_INACTIVE);
   gpio_pin_configure_dt(&btn, GPIO_INPUT | GPIO_PULL_UP);

   int last = 1;
   for (;;) {
      int state = gpio_pin_get_dt(&btn);
      if (!state && last) {
         printk("Hello World! %s\n", CONFIG_BOARD);
         gpio_pin_toggle_dt(&led);
      }
      last = state;
      k_msleep(10);
   }
}
```

3.2 Build & flash

```
west build -b sltb010a -s hello_app -d build_bg22 --pristine
west flash -d build_bg22 --runner jlink
```

Press the BTN0/PB0 user button → terminal prints **Hello World! sltb010a** and LED toggles.

---

4 · Common Pitfalls & Fixes

Problem	Fix
west flash cannot find J-Link	Ensure ~/.west/config has runner = jlink; confirm /opt/SEGGER/JLink in PATH.
udev permissions: *"no access to USB device"*	Add user to plugdev, then sudo udevadm control --reload-rules && sudo udevadm trigger.
Unknown device error in JLinkExe	Check exact part-number string (EFR32BG22C224F512IM40, etc.).
Console shows gibberish	Wrong baud rate (use **115200 8N1**) or wrong /dev/tty*.

---

5 · Next Steps

- Integrate **Bluetooth LE**: add CONFIG_BT=y and use the Zephyr hci-uart sample.