

Project Documentation

Project Title

Porting Zephyr RTOS to BGM220P (EFR32BG22 SoC)

Objective

To enable Zephyr RTOS support for the **Silicon Labs BGM220P** module by creating a custom board configuration and successfully building and running a test application (`my_app`)

Target Platform

- **Board:** Silicon Labs BGM220P
- **SoC:** EFR32BG22C224F512IM40
- **Architecture:** ARM Cortex-M33
- **RTOS:** Zephyr 4.2.0-rc1
- **Toolchain:** Zephyr SDK 0.17.2

Key Accomplishments

Custom Board Support

- `bgm220p.dts`, `bgm220p.dtsi`, `pinctrl.dtsi` files
- `bgm220p_defconfig`
- `bgm220p_custom.c`, `bgm220p_custom.cmake`, `bgm220p_custom.yaml`
- `Kconfig.bgm220p_customtxt`

SoC Integration

- `Kconfig.soc`, `Kconfig.defconfig` for **EFR32BG22**
- Defined `SOC_PART_NUMBER` and `SOC_SERIES` cleanly
- Avoided recursive symbol dependencies

Toolchain Setup

- Verified: `ZEPHYR_SDK_INSTALL_DIR`
- Implemented: `toolchain.cmake` with correct GCC path
- Resolved: Compiler not found issue (`-gcc`)

Device Tree Configuration

- Included `<arm/silabs/efr32bg22.dtsi>`
- Defined LED GPIO for basic testing (`my_app`)

Build System Improvements

- Fixed CMake dependency loops
- Cleaned Kconfig warnings
- Ensured complete and error-free `west build` process

Example Application – `my_app`

A simple test to validate the GPIO and timing system:

- Blinks LED using Zephyr `gpio` and `k_sleep()` APIs
- Confirms custom board and SoC integration works correctly

Project Structure (Key Files)

```
zephyr/
├── boards/arm/bgm220p/
│   ├── bgm220p_custom.dts
│   ├── bgm220p.dtsi
│   ├── pinctrl.dtsi
│   ├── bgm220p_defconfig
│   ├── bgm220p_custom.c
│   ├── bgm220p_custom.cmake
│   ├── bgm220p_custom.yaml
│   └── Kconfig.bgm220p_custom
├── soc/silabs/silabs_s2/efr32bg22/
│   ├── Kconfig.soc
│   └── Kconfig.defconfig
```

Troubleshooting & Fixes

1. Kconfig Dependency Loop

- **Issue:** SOC_SERIES_EFR32BG22 selected itself recursively
- **Fix:** Removed select inside SoC configs; let it be selected via PART_NUMBER

2. Hidden Symbol Assigned Directly

- **Issue:** Assigned CONFIG_SOC_SERIES_EFR32BG22=y
- **Fix:** Used CONFIG_SOC_PART_NUMBER_EFR32BG22C224F512IM40=y instead

3. Undefined Symbol: SOC_FAMILY_SILABS_S2

- **Issue:** Symbol assigned but never defined
- **Fix:** Removed from defconfig; created proper Kconfig symbol if needed.

4. Kconfig Warnings as Errors

- **Issue:** Untyped symbols, undefined defaults
- **Fix:** Added int, hex, bool types and cleaned up undefined defaults.

5. Toolchain Compiler Not Found (-gcc)

- **Issue:** Misconfiguration caused empty CC, leading to -gcc\

6. Toolchain Detected but Build Failed

- **Issue:** Extra slashes or unset CC/CROSS_COMPILE
- **Fix:** Verified CROSS_COMPILE and CC, cleared build directory and rebuilt

Final Outcome

- Successfully ported Zephyr RTOS to BGM220P
- Achieved clean and reproducible build of blinky_app
- Fixed all Kconfig, DTS, and build system issues
- Ready for further application development on BGM220P under Zephyr

```
Activities Terminal Jul 9 14:41 sudersan@Victus: ~/zephyrproject/my_app
nucleo_wb55rg
stm32f0_disco
nucleo_f412zg
stm32u083c_dk
nucleo_h753zi
stm32f723e_disco
nucleo_l4r5zi
nucleo_c031c6
stm32h735g_disco
nucleo_wba65ri
nucleo_g031k8
nucleo_f446re
stm32mp157c_dk2
odroid_go
fk7b0m1_vbt6
fk750m1_vbt6
fk743m5_xlh6
mikroe_clicker_2
mikroe_quail
mikroe_clicker_ra4m1
mikroe_stm32_m4_clicker
mikroe_mini_m4_for_stm32
t1sr9518adk80d
osd32mp1_brk
pico_plus2
CMake Error at /home/sudersan/zephyrproject/zephyr/cmake/modules/boards.cmake:226 (message):
  Invalid BOARD; see above.
Call Stack (most recent call first):
  /home/sudersan/zephyrproject/zephyr/cmake/modules/zephyr_default.cmake:131 (include)
  /home/sudersan/zephyrproject/zephyr/share/zephyr-package/cmake/ZephyrConfig.cmake:66 (include)
  /home/sudersan/zephyrproject/zephyr/share/zephyr-package/cmake/ZephyrConfig.cmake:97 (include_boilerplate)
  CMakeLists.txt:2 (find_package)

-- Configuring incomplete, errors occurred!
FATAL ERROR: command exited with status 1: /usr/local/bin/cmake -DWEST_PYTHON=/home/sudersan/zephyr-env/bin/python3.12 -B/home/sudersan/zephyrproject/my_app/build -GNinja -DBOARD=bgm220p_custom -S/home/sudersan/zephyrproject/my_app
((zephyr-env) ) sudersan@Victus:~/zephyrproject/my_app$
```

Invalid BOARD: bgm220p_custom

Which means Zephyr still doesn't recognize your custom board.

Let's **definitely** fix this now.

1. Do you have this folder?

`~/zephyrproject/zephyr/boards/arm/bgm220p_custom/`

`ls ~/zephyrproject/zephyr/boards/arm/bgm220p_custom`

You must see these 3 files:

`bgm220p_custom.yaml`

`bgm_220p_custom.dts`

`CmakeLists.txt`

2. Run this to tell Zephyr where to look:

`export ZEPHYR_BOARD_ROOT=$HOME/zephyrproject/zephyr`

`echo $ZEPHYR_BOARD_ROOT`

3. Check if Zephyr sees your board now:

`west boards | grep bgm`

`bgm220p_custom`

Still Doing on This Page trying to Port the Zephyr for the Silicon Labs BGM220P