

Custom Board Development for Genotyper in Zephyr

(Kindly read the whole section)

Introduction

We need to have a custom board for application testing and development which uses the *nRF52811* board. The names of the temporary custom boards defined are *genotyper_nrf52811* and *genotyper_som_nrf52811*. These two boards are defined by making necessary changes to *nrf52840dk_nrf52811* board definition (close variant).

Kindly clone the [ZephyrPlayground](#) and/or the [Genotyper-Firmware](#) repository to the local system.

The board definitions are provided in two different repositories.

- The [GitHub link](#) to the custom board files in Genotyper-Firmware repository is provided. The **boards** directory is present only in the *work-sp* branch of the repository.
- Also, [GitHub link](#) to both the custom board files in the Zephyr-Playground is provided. Kindly switch to the branch **work-sp** to find the same.

Essential changes

The following files of the custom boards were modified in order to build the applications.

1. *Kconfig*
2. *Kconfig.board*
3. *Kconfig.defconfig*
4. *board.cmake*
5. *genotyper_nrf52811.dts* (or) *genotyper_som_nrf52811.dts*
6. *genotyper_nrf52811.yaml* (or) *genotyper_som_nrf52811.yaml*
7. *genotyper_nrf52811_defconfig* (or) *genotyper_nrf52811_defconfig*

The modifications are mainly:

- Changing the name of the config in *Kconfig* and *Kconfig.board*.
- Changing the model name and compatible property in *.dts* file.
- Changing the identifier and name in YAML file.
- Enabling board config option in *board_defconfig* file.

To edit the files, kindly clone the repository and create a local branch with the name *work-sp*, by running the command

```
git branch work-sp
```

Also pull the *work-sp* branch from GitHub for the files by running

```
git pull origin work-sp
```

Important points to note

- It is a good practise to describe all the nodes in the devicetree source (.dts) files rather than having an overlay file for the board.
- The way to do this is by defining a new child node under the parent node. For example, if **izc** is the parent node, then **tmp100** can be defined as a new child node under **izc** along with all properties and labels.

Building sample applications for custom boards

Before trying to build and flash the applications for the two boards, kindly copy both the custom board directories to the *workspace zephyr directory*. For example, in the case of ZephyrPlayground (assuming **\$pwd** is ZephyrPlayground directory):

- Find the custom board files in the path **ZephyrPlayground/boards/arm/** and copy it to **../zephyr/boards/arm/** along with the definitions of all other boards that zephyr provides.
- Failing to do so, we get an error saying that the board definition is not found in the path where it looks for the boards which is **../zephyr/boards/arm/**

To build a zephyr application using the *west* tool for our custom board, run the command

```
west build -p auto -b genotyper_nrf52811 path-to-application
```

or

```
west build -p auto -b genotyper_som_nrf52811 path-to-application
```

Problems faced

Below are the issues faced while building various applications for the custom board *genotyper_nrf52811*.

- Oddly, the board successfully builds the applications such as *blinky* and *blinky_pwm* but does not build applications which include peripherals such as *mpu6050*, *tmp100*, *servo_motor*, etc. The error message is: *error: '__device_dts_ord_DT_N_ALIAS_pwm_servo_ORD' undeclared (first use in this function)*

Issues faced while building various applications for the custom board *genotyper_som_nrf52811*.

- Similar to the *genotyper_nrf52811*, the *genotyper_som_nrf52811* does not build any samples involving peripherals apart from GPIO (for example: *adc*, *i2c*, *pwm*, etc).
- The applications when built finds the devicetree source (.dts) file for this board successfully but does not seem to care about the contents. The direct meaning of the statement is that disabled nodes in the devicetree are accessed without any errors issued by the build system.

Attempted Solutions

Solutions attempted for the above mentioned issues:

- Clearing the contents of the cache folder present in *~.cache/zephyr/ToolchainCapabilityDatabase/*.
- Clearing the contents of cache folder present in *ZephyrPlay-ground/.cache/*

These solutions do not currently help solving the devicetree problem.

som_nrf52811 Board Definition

The *som_nrf52811* board definition is present in the [Genotyper-Firmware](#) and [ZephyrPlayground](#) repositories.

Important note: Highly recommended to checkout the devicetree source file.

A new board with the name *som_nrf52811* was defined in a similar manner i.e. by making necessary changes to the already existing *nrf52840dk_nrf52811* board.

- No change in *board.cmake*
- No change in *CMakeLists.txt*
- Change the *depends on* property to `BOARD_SOM_NRF52811`.
- Change the *config* to `BOARD_SOM_NRF52811` and *bool* property to "SOM NRF52811" in *Kconfig.board*.
- Change the *identifier* to *som_nrf52811* and *name* to *SOM – NRF52811* in *board.yaml* file.
- Change to `CONFIG_BOARD_SOM_NRF52811 = y` is *board_defconfig* file.
- Change to *if BOARD_SOM_NRF52811* and *default "som_nrf52811"* in the *Kconfig.defconfig* file.
- Change the *model* property to "Nordic SOM NRF52811" and *compatible* property to "nordic,som-nrf52811" in the *board.dts* file.

The definition of this board works as required and does not pose any issues. The devicetree of this new board can be edited as and when required. *Nodes* can be added and removed as per requirements.

The [demo video](#), the [board porting guide](#), and [custom board definition guide](#) can be used as references for multiple attempts.