



Basic: fine tuning

BeRTOS allows you to fine tune each driver for your project and to specify hardware specific functions.

Configuring your program

If you have used the wizard to create your project, you have probably noticed that it created two directories, `cfg/` and `hw/` inside your project's directory.

The first one contains many files, with all the configuration options that you specified within the wizard. Using some include tricks, these files are looked up before the corresponding one inside the BeRTOS directory, effectively overriding the default settings.

How do you use the values specified in these files? Simple, include them using double quotes (") instead of brackets (<). For example:

```
#include "cfg/cfg_ser.h"
// this value will be the one specified inside the wizard
static uint8_t buf[CONFIG_UART0_TXBUFSIZE];

int main(void)
{
    // use buf
    //...
}
```

If you need to specify a custom value that overrides the defaults, just change the corresponding file inside the `cfg/` directory or use the wizard to edit the project.

Hardware specific definitions

In many real-world cases, the micro you are using is connected to external peripherals with some specific pins, which may differ depending on the project. BeRTOS provides `hw_*` files to account for these differences.

For example, the SD driver (`drv/sd.c`) needs a chip select pin. The corresponding hw file (`hw/hw_sd.h`) specifies 3 macros:

- ▶ `SD_CS_INIT()` : initializes the chip select circuitry (for SAM7 this means enabling PIO on a specific pin);
- ▶ `SD_CS_ON()` : asserts chip select line;
- ▶ `SD_CS_OFF()` : clears chip select line.

BeRTOS hw files usually come with empty definitions, so you must fill in each macro to ensure proper functionality.

Another example is the keyboard driver, which defines the `kdb_readkeys()` function; of course this is completely hardware dependent so you need to define it.

You can see an example of custom definition in `examples/demo/hw/hw_kbd.h`, where the readkey function is implemented to work on PCs.

DEVELOPER'S TRAC

On the developers website you will find all the information to contribute to BeRTOS development community, milestones, the possibility to open tickets to the community, the changelog and all the details on every feature.
dev.bertos.org

PREMIUM SUPPORT

By registering to Develer's advanced support services you will have online assistance, phone assistance and the possibility to report bugs directly to our BeRTOS engineering team. All services guarantee reply in 24 working hours.
bertos.org/supporto/premium/

CERTIFIED TRAINING

Customized training, onsite teaching, embedded application debug and deploy assistance, peer review on source code, training classes for single programmers or working teams.
bertos.org/supporto/formazione/



BeRTOS website by [Develer](http://Develer.com).