
SpartanMC

Header file for peripheral access

Table of Contents

List of Figures

List of Tables

MANPAGE – PERIPHERALS.H(3)

NAME

peripherals.h – Project header file for access to peripheral components

SYNOPSIS

```
#include <system/peripherals.h>
```

DESCRIPTION

For any SpartanMC project, the system builder *jConfig* generates a header file providing the interface to all peripheral components present in your system. Variables pointing to the respective I/O and/or DMA memory base addresses will be automatically provided for each peripheral instance.

For a particular peripheral instance the name of the variable will be the respective identifier as shown in the system builder *jConfig* converted to UPPER CASE (e.g. *UART_LIGHT_0*). Each such variable will be a typed pointer tailored to the register I/O space of the particular peripheral. In case a component offers DMA space you will get another variable named `<PERIPHERAL_NAME>_DMA` pointing to the peripherals DMA base address.

The header files defining the respective variable types can be found at **spartanmc/include/peripherals/** (see section below for details). All files required for your systems peripherals will be automatically included via **peripherals.h**.

IMPLEMENT CUSTOM PERIPHERALS

For each type of peripheral component a header file is required at **spartanmc/include/peripherals/** declaring the particular data types (e.g. a struct) for register I/O and DMA access. The header files name must be the same as the peripherals hardware type.

If you add a custom peripheral component to the SpartanMC-SoC-Kit make sure you provide the corresponding header file. For a component named e.g. *my_peri* a file named **my_peri.h** is required. Within this file the following type declarations are expected to be found:

```
typedef ... my_peri_regs_t; /* register space access */
typedef ... my_peri_dma_t; /* DMA space access */
```

Note that in case your peripheral does not implement registers or DMA space the respective type declaration may be omitted. Basically, the interface to a peripheral component may be a pointer to an unsigned integer. In that case, the type definitions may look like the following:

```
typedef unsigned int my_peri_regs_t; /* register space access */
```

```
typedef unsigned int my_peri_dma_t; /* DMA space access */
```

Note that there is no explicit pointer- or array-like declaration. The point where the pointer comes in is at the variable instantiation in the generated header file.

To interface more complex peripherals it is wise declaring a structure with descriptive names for the particular registers. Additionally to the type declaration the header file may define bit constants to simplify bit wise access to the registers.

High level support functions operating on the peripherals registers and DMA space should be defined in arbitrary named header files located at **spartanmc/include**. The respective implementation of such functions should be part of *libperi*, but could virtually be implemented in any other library.

FILES

**<project_dir>/
system/
peripherals.h** Header file to include for access to generated peripheral variables

**<project_dir>/
system/
<subsystem_name>/
peripherals.h** Actual header file defining peripheral variables for the respective system. Included by **peripherals.h** depending on the actual subsystem the firmware is built for.

SEE ALSO

hardware.h(3), *spartanmc-libs(7)*

AUTHORS

Copyright (c) 2011, 2012 Dresden University of Technology, Institute for Computer Engineering, Chair for Embedded Systems.

Written by Markus Vogt