
SpartanMC

SpartanMC software libraries

Table of Contents

List of Figures

List of Tables

MANPAGE – SPARTANMC-LIBS(7)

NAME

spartanmc-libs – SpartanMC software libraries and library build system

DESCRIPTION

The SpartanMC processor core comes with a number of supporting C libraries. The library functions are available for the firmware on the target system. Some library code is essential such as startup code or interrupt handling routines. The remainder of the library offers functions to the user to simplify access to peripheral components.

To use functions from a certain library, the corresponding header file must be included (see *spartanmc-headers*) in your source code and the linker must be told to include the library into the firmware binary. The latter is achieved by adding the library name to the list of link libraries in *config-build.mk* in the firmware directory. Note that pointer variables for access to peripheral registers are available by including *peripherals.h* (see **peripherals.h**).

To optimize the size of the resulting firmware binary, each library function is implemented in a separate source file. This allows the linker to remove all functions that are never called.

LIBRARIES

The following libraries are currently available for the SpartanMC processor core:

libc	System calls like printf, sleep, etc. Automatically linked with each project.
libgcc	Support function for the compiler. Automatically linked with each project.
libinterrupt	Required interrupt handler and support functions for default interrupt controller (intctrl). Could not be used together with libinterrupt_p .
libinterrupt_p	Required interrupt handler and interrupt support functions for interrupt controller with priority (intctrl_p). Could not be used together with libinterrupt .
libperi	Support functions for various peripheral components such as USB, UART and others (also see peripherals.h).
libstartup	Default startup code for the processor core. This library is included by default for each new project. Could be replaced by libstartup_loader .

- libstartup_loader** Startup code with support for firmware update via UART using a boot loader. For more details, see *spmc-loader* and *startup_loader*. Could not be used together with **libstartup**.
- librtos** Real Time Operating System. Can not be used together with **libstartup**.
- librtos_interrupt** Interrupt support for **librtos**. Can not be used together with **libstartup**.

IMPLEMENTING NEW FUNCTIONS

Extending an existing library

To add a function to an existing library, create a new file in the corresponding source folder in **spartanmc/lib_obj/src/<library_name>/**. To get a smaller binary through link time optimization, make sure to implement each function in a separate source file. The type of source file can be either C (*.c) or Assembler (*.s).

To make your new function known to the compiler, create or edit a header file in **spartanmc/include/**. Your implemented function can use other library functions. See section *Library build system* below on how to specify dependences between libraries.

Creating a new library

An entirely new library is create in a seperate folder named **spartanmc/lib_obj/src/<library_name>/**. As described above, create source files in that new folder to implement your library functions.

The new library must be included into the build system. Edit the file *spartanmc/lib_obj/Makefile* for that purpose. See below for details on the library build system.

The library build system

All source files for one certain SpartanMC library are placed in a dedicated directory. The location of the source files is **spartanmc/lib_obj/src/<library_name>**. Possible source file types are C (*.c) and assembler (*.s).

The library build process is controlled by the makefile **spartanmc/lib_obj/Makefile**, where the following variables are of interest:

- LIBS** List of libraries to build. Each name specified must correspond to a source directory **spartanmc/lib_obj/src/<library_name>**. All library names are specified without the prefix *lib*.
- OBJ_DIRS** List of directories with additional object files. These files are compiled but not explicitly bundled into a library archive file

(*a). Other library functions can use the resulting objects as dependences. This is useful for helper functions which could not explicitly be associated to a certain library.

DEPS_<library_name>

Specifies dependences for each library, if required. Each object file specified here is included in the library archive (*.a) additionally to the original library code. Valid objects are either from another library or from a directory specified via variable **OBJ_DIRS** (see above).

FILES AND DIRECTORIES

spartanmc/ lib_obj/src/ <library_name>	Source code for all functions of library <library_name>
spartanmc/ lib_obj/ Makefile	Makefile controlling the library build process

SEE ALSO

spartanmc-headers(3), spmc-loader(1), startup_loader(3)

AUTHORS

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

Written by Markus Vogt