Another simple device driver

1 Introduction

This lab is concerned with device driver development for the `LPC_2378_STK` development board.

2 In the lab

1. Download the file workspace.zip into a suitable directory either on a pen drive or in your University workspace. I suggest you call the directory `EN572/labs/lab03`. Unzip `workspace.zip`.

2. Start up EWARM and load the workspace `EN572/labs/lab03/workspace/workspace.eww`.

3. Connect a `LPC-2378-STK` board to a USB port on your computer.

4. Download and debug the project `lab02S`. This project gives solutions to the exercises set in `lab02`. Run it and observe its behaviour.

5. Carefully read the files `leds.h` and `leds.c` and make sure that you understand how the functionality for the USB CONNECT LED has been added.

6. Change to project `lab03` in your workspace. Now you can begin work on the driver for the buttons and joystick. Search `board.h` for the definitions related to the buttons and joystick.

7. Read `buttons.h`. This gives the prototypes for the functions that you must implement.

8. Add a call to `buttonsInit()` into `bspInit()` in the file `bsp.c`. This call should come directly after the call to `ledsInit()`. It is usual to initialise all your devices by calling the appropriate functions from `bspInit()`. In your applications, you can then initialise the board support package with a single call to `bspInit()` and this will initialise all the devices that the board support package supports. In this lab, you are using only the leds and the buttons; hence `bspInit()` contains calls to initialise only these devices. In later labs, you will be using a board support package that

supports more devices; their initialisation functions will be called from `bspInit()`.

9. Make sure that the project `lab03` now builds without warnings and/or errors. It will not yet behave as you want it to; but it should build successfully.

10. Modify `buttons.c` to complete the implementation of the functions for the buttons/joystick driver. As you develop the code for each function, check frequently that the project still builds successfully.

11. Modify your main function so that it repeatedly checks the state of the buttons and turns on the link led if `BUT_1` is pressed and turns off the link led if `BUT_1` is not pressed. Build and test your project.

12. Write a new main function that starts the link led flashing when `BUT_1` is pressed. The led should continue to flash until `BUT_2` is pressed. Build and test your project.

13. Modify your last main function so that the rate of flashing is increased by one hundredth of a second if `JS_UP` is pressed and is decreased by one hundredth of a second if `JS_DOWN` is pressed. The changes to the rate of flashing should occur only if the led is already flashing. Build and test your project.