Some useful functions to use *La BlueFrog*'s hardware :

Set_System()
In : <project name>/src/hw_config.c
Role :
Enable the system configuration module clock.
Configure EXTI Line18 to be able to react to interrupts from USB IP
Initialize the MAL (Media Access Layer used by USB) i.e. the data Flash (= FLASH_Init())

PWR_Init()
In : BlueFrogV1-Lib/src/inframod_pwr.c
Role :
Set up GPIOs and I2C module to interface with Power Management IC
Configure Power Management IC (VCC and current levels)
Configure EXTI_Line0 interrupt to react to ONSTAT signal from Power Managemen IC (deglitched version of power ON/OFF switch) to which STM32 will react by adequate setting of POWERACK

LED_Config()
In : <project name>/src/hw_config.c
Role :
Configure the two GPIO that drive the green LED and the red LED

Slide_Config()
In : <project name>/src/hw_config.c
Role :
Configure the two GPIO that monitor the state of the slider switch

FLASH_Init()
In : BlueFrogV1-Lib/src/inframod_spi_flash.c
Role :
Set up GPIOs and SPI module to interface with Data Flash IC
/ Somewhat redundant with MAL_Init() in Set_System */*

OLED_Init()
In : BlueFrogV1-Lib/src/inframod_spi_oled.c
Role :
Set up GPIOs and SPI module to interface with Data Flash IC
Sets up GPIO to enable/disable VDDH=14V to OLED and disable it

OLED_Config(uint16_t *fconf)
In : BlueFrogV1-Lib/src/inframod_oled.c
Role :
Sets up OLED as per datasheets of OLED (Densitron DD-160128-FC-1A) and its display driver (SEPS525) using either default values (if fconf[0] == 0xFFFF) or custom values present in buffer pointed by fconf (typically, first obtained from some configuration file).

UART_Init()

In : BlueFrogV1-Lib/src/inframod_uart.c
Role :
Set up GPIOs and USART module to interface with external connector pins UART_Tx/UART_Rx

HORLOGE_Init();
In : BlueFrogV1-Lib/src/inframod_horloge.c
Role :
Set up the RTC (Real-Time Clock) Module

HORLOGE_EXTI_Init();
In : BlueFrogV1-Lib/src/inframod_horloge.c
Role :
Set up and enable interrupt EXTI_Line20 from RTC module

HORLOGE_Init();
In : BlueFrogV1-Lib/src/inframod_horloge.c
Role :

Set_USBClock();
In : <project name>/src/hw_config.c
Role :
Enable clock to the USB module

USB_Interrupts_Config();
In : <project name>/src/hw_config.c
Role :
Set up and enable low-priority USB interrupt (USB_LP_IRQn) and USB wake-up interrupt (USB_FS_WKUP_IRQn)

USB_Init();
In : STM32_USB-FS-Device_Driver/src/usb_init.c
Role :
Initialization of the USB module. Then USB operation is interrupt-driven.

See also example codes provided for other convenient macros and functions to perform basic functions around LEDs, GPIOs etc.

Data Flash hardware abstraction libraries :

In File: inframod_spi_flash.c

Function Name: void FLASH_Init(void)
Brief Description : Sets up physical interface (SPI etc.) to access Flash

Function Name: void FLASH_WriteBuffer(uint8_t* *pBuffer*, uint32_t *WriteAddr*, uint32_t *NumByteToWrite*)
Brief Description : Writes the buffer pointed by *pBuffer* of length *NumByteToWrite* bytes into Flash, starting at address *WriteAddr*
Note : *currently only uses one SRAM buffer of ADESTO Flash. Should be*

optimized by taking advantage of second SRAM buffer (ping-pong to improve throughput).

Function Name: void FLASH_ReadBuffer(uint8_t* *pBuffer*, uint32_t *ReadAddr*, uint32_t *NumByteToRead*)
Brief Description : Reads *NumByteToWrite* bytes from Flash starting at address *ReadAddr and* store them into a buffer pointed by *pBuffer*

Function Name: uint32_t FLASH_ReadID(void)
Brief Description : Returns the Manufacturer ID (Byte0) and Device ID (Byte 1 and Byte2) information stored in the Data Flash

Function Name: void FLASH_EraseBulk(void)
Brief Description : Perfoms a complete erase of the Flash