La BlueFrog-V2

Low-level API Contents

Release 0.a

Note : the .h files that expose the functions, types and macros defined below are located under …/libraries/BlueFrogV2-Lib/inc/LBF_API. The cores of the functions are under …/libraries/BlueFrogV2-Lib/src/LBF_API.

Custom Type Definitions :

(from file custom_types.h)

```
typedef enum {
  FALSE = 0,
  TRUE = 1
}
boolean_t;


typedef enum {
  NOK = 0,
  OK = 1
}
ReturnStatus_t;
```

On-board LEDs and Slider Switch Control:

(from file LBF_LED_Switches_lowlevAPI.h)

```
void Stm32_Led_ON(void);
void Stm32_Led_OFF(void);
void Stm32_Led_TOGGLE(void);

boolean_t State_Switch1_IsOn(void);
boolean_t State_Switch2_IsOn(void);
```

Data Flash Control :

(from file LBF_FLASH_lowlevAPI.h)

```
void FLASH_EraseBulk(void);
void FLASH_WritePage(uint8_t* pBuffer, uint32_t WriteAddr, uint16_t NumByteToWrite);
void FLASH_WriteBuffer(uint8_t* pBuffer, uint32_t WriteAddr, uint32_t NumByteToWrite);
void FLASH_ReadBuffer(uint8_t* pBuffer, uint32_t ReadAddr, uint32_t NumByteToRead);
uint32_t FLASH_ReadID(void);

#define FLASH_PAGE_LENGTH   0x0100  // 256 bytes per page mode
#define FLASH_NUMBER_OF_PAGES   32768
#define FLASH_CAPACITY      (FLASH_PAGE_LENGTH * FLASH_NUMBER_OF_PAGES)   // 32768 x
256B = 83888608 = 8MB = 64Mb
```

Direct STM32 GPIO Control :

(from file LBF_GPIO_lowlevAPI.h)

```
#define GPIO_HIGH(PORT,PIN)          HAL_GPIO_WritePin(PORT, PIN, GPIO_PIN_SET)
#define GPIO_LOW(PORT,PIN)           HAL_GPIO_WritePin(PORT, PIN, GPIO_PIN_RESET)
#define GPIO_TOGGLE(PORT,PIN)        HAL_GPIO_TogglePin(PORT, PIN)
```

OLED Control :

(from file LFB_OLED_lowlevAPI.h)

```
#define OLED_CS_LOW()          GPIO_LOW(OLED_CS_PORT, OLED_CS_PIN)
#define OLED_CS_HIGH()         GPIO_HIGH(OLED_CS_PORT, OLED_CS_PIN)
#define OLED_RS_LOW()          GPIO_LOW(OLED_RS_PORT, OLED_RS_PIN)
#define OLED_RS_HIGH()         GPIO_HIGH(OLED_RS_PORT, OLED_RS_PIN)
#define OLED_RESET_LOW()       GPIO_LOW(OLED_RESET_PORT, OLED_RESET_PIN)
#define OLED_RESET_HIGH()      GPIO_HIGH(OLED_RESET_PORT, OLED_RESET_PIN)

/* Colors for OLED in rgb565 format */
#define BLACK          0x0000
#define WHITE          0xFFFF
#define RED            0xF800
#define GREEN          0x07E0
#define BLUE           0x001F
#define YELLOW         0xFFE0
#define CYAN           0x07FF
#define LIGHT_BLUE     0x1C9F
#define ORANGE         0xFD20

/* Screen dimensions - 160x128 for DD-160128FC-1A */
#define X_FULL_SCREEN  160
#define Y_FULL_SCREEN  128

/*  OLED SPI */
uint8_t OLED_SPI_TransferByte (uint8_t octet);
void OLED_SendCmd (uint8_t Value);
void OLED_SendData (uint16_t Value);
void OLED_WriteReg (uint8_t RegName, uint8_t RegValue);
void OLED_DataStart (void);
void OLED_DataEnd (void);

/* OLED Gfx Generation */
void OLED_SetRegion(uint8_t x, uint8_t y, uint8_t width, uint8_t height);
void OLED_Fill(uint8_t x, uint8_t y, uint8_t width, uint8_t height, uint16_t color565);
void OLED_DisplayBuffer(uint8_t x, uint8_t y, uint8_t width, uint8_t height, uint16_t *buffer);
void OLED_Clear (void);

/* OLED High-voltage (14V) on/off control   */
void OLED_Switch_ON (void);
void OLED_Switch_OFF (void);

/* « printf » type utilities – CAUTION : rely on emWin middleware, which mus therefore be enabled */
void OLED_Overwrite_CurrentLine(void);
void OLED_PrintString(char* string);
void OLED_PrintDec(int32_t SignedInteger);
void OLED_PrintHex(uint16_t   Unsigned16);
```

/**** Note : for rich graphics and text generation, the emWin graphics library has a lot to offer   ********/

Power Management

(from file LBF_PWR_lowlevAPI.h)

```
void Turn_VDDH_On(void);
void Turn_VDDH_Off(void);
boolean_t Check_VDDH_On(void);
```

UART (#1 and #3) Control

(from file LBF_UART_lowlevAPI.h)

```
void UART_SendData (UartID_t Uart_ID, uint8_t data);
uint8_t UART_ReceiveData (UartID_t Uart_ID);
void UART_SendString(UartID_t Uart_ID, char* pString);
void UART_SendString_SwFlowControl(UartID_t Uart_ID, char* pString);

typedef enum {
 UART1 = 1,
 UART3 = 3
}
UartID_t;

// For UART software flow control
#define  XON   0x13
#define  XOFF  0x11
```

I2C #2 Control

(from file LBF_I2C2_lowlevAPI.h)

```
void      I2C2_WriteSingleReg (uint8_t ChipID, uint16_t RegAdd, uint8_t RegVal);
void      I2C2_WriteMultipleReg (uint8_t ChipID, uint16_t RegAdd, uint8_t* pVal, uint16_t NumByteToWrite );

uint8_t   I2C2_ReadSingleReg (uint8_t ChipID, uint16_t RegAdd);
void      I2C2_ReadMultipleReg (uint8_t ChipID, uint16_t RegAdd, uint8_t* pVal, uint16_t NumByteToRead );

void      I2C2_RmodWSingleReg (uint8_t ChipID, uint16_t RegAdd, uint8_t RegMask, uint8_t
RegUpdateVal);
```

Services

(from file Services.h)

```
void Delay_ms (volatile uint32_t nTime);
```

Pin aliases

(from file pin_aliases.h)

```
/*  ==== Power Management ============================================= */


/* --- LTC3533 PMIC ---*/

//PC2 - HPWR, STM32 output
#define HPWR_PIN                GPIO_PIN_2
#define HPWR_PORT               GPIOC

//PB6 - BUCK_ON, STM32 output
#define BUCK3V_ON_PIN                 GPIO_PIN_6
#define BUCK3V_ON_PORT                GPIOB

//PC13 - ONOFF_STAT (debounced On/Off push-button), STM32 input
#define ONOFF_STAT_PIN                GPIO_PIN_13
#define ONOFF_STAT_PORT               GPIOC


/* --- TPS22929 Power Switch (DC-DC Boost Converter On/Off) ---*/

// PC0 -  BOOSTCONV_EN,  STM32 output
#define VDDH_EN_PIN                   GPIO_PIN_0
#define VDDH_EN_PORT            GPIOC


/*  ==== LEDs  ========================================================= */

// PC3 -  STM32_LED,  STM32 output
#define STM32_LED_PIN           GPIO_PIN_3
#define STM32_LED_PORT          GPIOC


/*  ==== Selection Switches  ========================================== */

// PA15 -  SWITCH1,  STM32 input
#define SWITCH1_PIN             GPIO_PIN_15
#define SWITCH1_PORT            GPIOA

// PC8 -  SWITCH2,  STM32 input
#define SWITCH2_PIN             GPIO_PIN_8
#define SWITCH2_PORT            GPIOC

/*  ==== I2C1   ======================================================== */

// PB8 - SCL, STM32 output, Open-Drain
// PB9 - SDA, STM32 output/input, Open-Drain
#define I2C1_SCL_PIN            GPIO_PIN_8
#define I2C1_SDA_PIN            GPIO_PIN_9
#define I2C1_PORT               GPIOB

/*  ==== I2C2   ======================================================== */

// PB10 - SCL, STM32 output, Open-Drain
// PB11 - SDA, STM32 output/input, Open-Drain
#define I2C2_SCL_PIN            GPIO_PIN_10
#define I2C2_SDA_PIN            GPIO_PIN_11
#define I2C2_PORT               GPIOB

/*  ==== SPI1   ========================================================= */
```

```
// PA5 - CK, STM32 output, Std CMOS
// PA6 - MISO, STM32 input
// PA7 - MOSI, STM32 output, Std CMOS
#define SPI1_SCK_PIN        GPIO_PIN_5
#define SPI1_MISO_PIN       GPIO_PIN_6
#define SPI1_MOSI_PIN       GPIO_PIN_7
#define SPI1_PORT    GPIOA

/*  ==== SPI3   ============================================================ */

// PB3 - CK, STM32 output, Std CMOS
// PB4 - MISO, STM32 input
// PB5 - MOSI, STM32 output, Std CMOS
#define SPI3_SCK_PIN        GPIO_PIN_3
#define SPI3_MISO_PIN       GPIO_PIN_4
#define SPI3_MOSI_PIN       GPIO_PIN_5
#define SPI3_PORT           GPIOB

/*  ==== UART1  ============================================================ */
// PA9 - TX, STM32 output, Std CMOS
// PA10 - RX, STM32 input

#define UART1_TX_PIN        GPIO_PIN_9
#define UART1_RX_PIN        GPIO_PIN_10
#define UART1_PORT GPIOA

/*  ==== USART 2  ========================================================== */
// PA2 - TX, STM32 output, Std CMOS
// PA3 - RX, STM32 input
// PA4 - CK, STM32 output, Std CMOS

#define USART2_TX_PIN       GPIO_PIN_2
#define USART2_RX_PIN       GPIO_PIN_3
#define USART2_CK_PIN       GPIO_PIN_4
#define USART2_PORT         GPIOA

/*  ==== UART 3  =========================================================== */
// PC10 - TX, STM32 output, Std CMOS
// PC11 - RX, STM32 input

#define UART3_TX_PIN        GPIO_PIN_10
#define UART3_RX_PIN        GPIO_PIN_11
#define UART3_PORT          GPIOC

/*  ==== DATA FLASH (excl SPI)   =========================================== */

// PB7 = nCS
#define FLASH_CS_PIN        GPIO_PIN_7
#define FLASH_CS_PORT       GPIOB

/*  ==== BTLE (excl UART)       ========================================== */

// PC9 = BT_RST (active high)
#define BT_RST_PIN          GPIO_PIN_9
#define BT_RST_PORT         GPIOC

/*  ==== LSM6DS3 ACCEL/GYRO     ========================================== */

// PB15 = INT1_ACC_GYR, PC6 = INT2_ACC_GYR
#define INT1_ACC_GYR_PIN   GPIO_PIN_15
```

```
#define INT1_ACC_GYR_PORT          GPIOB
#define INT2_ACC_GYR_PIN   GPIO_PIN_6
#define INT2_ACC_GYR_PORT          GPIOC

/*  ==== BATTERY             ========================================= */

// PC1 = BATT_ADC_MEAS, PC7 = BATT_MEAS_EN
#define BATT_ADC_MEAS_PIN          GPIO_PIN_1
#define BATT_MEAS_EN_PIN GPIO_PIN_7
#define BATT_PORT              GPIOC

/*  ==== OLED (excl SPI)       ========================================= */

// PC4 - OLED_RS, STM32 output
#define OLED_RS_PIN            GPIO_PIN_4
#define OLED_RS_PORT                   GPIOC

// PC5, OLED_NCS, STM32 output
#define OLED_CS_PIN            GPIO_PIN_5
#define OLED_CS_PORT                   GPIOC

// PB1 - OLED_RESET, STM32 output
#define OLED_RESET_PIN                 GPIO_PIN_1
#define OLED_RESET_PORT                GPIOB
```

Aliases for on-board chips (ID, registers, etc.)

(from file OnBoard_chip_aliasas.h)

```
// Magnetometer :  ST LIS3MDL
#define LIS3MDL_CHIPID         0x1C
#define LIS3MDL_WHOAMI         0x0F
#define LIS3MDL_WHOAMI_CONTENTS        0x3D

// Accelerometer/Gyro :  ST LSM6DS3
#define LSM6DS3_CHIPID         0x6A
#define LSM6DS3_WHOAMI         0x0F
#define LSM6DS3_WHOAMI_CONTENTS        0x69

// ALS/Proximity : ST VL6180X
#define VL6180X_CHIPID         0x29
#define VL6180X_WHOAMI         0x00
#define VL6180X_WHOAMI_CONTENTS        0xB4

// Pressure/Temp Sensor : ST LPS25H
#define LPS25H_CHIPID          0x5C
#define LPS25H_WHOAMI          0x0F
#define LPS25H_WHOAMI_CONTENTS         0xBD
```

Global Variables

(from file global_variables.h)

```
 /*   ------- Handles on structures used by HAL API Functions   ------------- */

extern UART_HandleTypeDef huart1;             // initialized in LBF_UART1_Init.c
extern USART_HandleTypeDef husart2;           // initialized in LBF_USART2_Init.c
extern UART_HandleTypeDef huart3;             // initialized in LBF_USART3_Init.c
extern I2C_HandleTypeDef hi2c1;                 // initialized in LBF_I2C1_Init.c
extern I2C_HandleTypeDef hi2c2;                 // initialized in LBF_I2C2_Init.c
extern SPI_HandleTypeDef hspi1;                 // initialized in LBF_SPI1_Init.c
extern SPI_HandleTypeDef hspi3;                 // initialized in LBF_SPI3_Init.c
```