通用驱动

详见 device 文件夹

```
/*****************************************************************************

* 功　能 : 设备文件系统功能实现  dev_file.c

*****************************************************************************/

void dev_init(void);

static int dev_get_fd_by_name(char* name);

static int dev_fd_chk(int fd);

void dev_register(file_op* op);

void dev_list(void);

int dev_open(char* name, int option);

int dev_write(int fd, char* buff, int size);

int    dev_probe(int fd, int msec);

int dev_read(int fd, char* buff, int size);

int dev_seek(int fd, int offset, int whence);

int dev_ioctl(int fd, int option, void* param);

void dev_close(int fd);

/*****************************************************************************

* 功　能 : 设备文件系统功能实现  end

*****************************************************************************/
```

```
/*******************************************************************************

* 功   能 : spi 设备功能实现  dev_spi.c

*******************************************************************************/

SPI_PARAM *BSP_SPI_Get_Param(unsigned char spi_num);

void BSP_SPI_SetAttr(SPI_PARAM* SPIParam);

void BSP_SPI_Init(unsigned char spi_num);

int BSP_SPI_SendRecv_Data(unsigned char spi_num, char send_data, char *recv_data);

int BSP_SPI_SendData(unsigned char spi_num, char *buff, int size);

int BSP_SPI_ReadData(unsigned char spi_num, char *buff, int size);

int BSP_SPI_ioctl(unsigned char spi_num, int op, void* param);

int BSP_SPI_open(unsigned char spi_num, int option);

void BSP_SPI_close(unsigned char subdev_num);

void Dev_SPI_Init(void);

/*******************************************************************************

* 功   能 : spi 设备功能实现  dev_spi.c   end

*******************************************************************************/




/*******************************************************************************

* 功   能 : can 设备功能实现  dev_can.c

*******************************************************************************/

CAN_PARAM* BSP_Can_GetParam(unsigned char can_num);
```

```
void    BSP_Can_TxIRQHandler(unsigned char num);

void    BSP_Can_Rx0IRQHandler(unsigned char num);

void    BSP_Can_GpioCfg(CAN_PARAM* Param);

void BSP_Can_ModeCfg(CAN_PARAM* Param, int baudrate);

void BSP_Can_FiltCfg(CAN_PARAM* Param);

void    BSP_Can_Init(unsigned char num);

int    BSP_Can_write(unsigned char subdev_num, char* buff, int size);

void BSP_Can_Timer(void *ptmr, void *p_arg);

int BSP_Can_Probe(unsigned char subdev_num, int msec);

int    BSP_Can_read(unsigned char subdev_num, char* buff, int size);

int    BSP_Can_open(unsigned char subdev_num, int option);

void BSP_Can_close(unsigned char subdev_num);

void Dev_Can_Init(void);

/*******************************************************************************

* 功    能 : can 设备功能实现  dev_can.c end

*******************************************************************************/



/*******************************************************************************

* 功    能 : 232、485 等设备功能实现  dev_com.c

*******************************************************************************/

COM_PARAM* BSP_Com_Get_Param(unsigned char com_num);

COM_PARAM* BSP_Com_Get_Param_By_DMA(DMA_Stream_TypeDef *DMA_Streamx);
```

```
void BSP_DMA_ClrItFlg(DMA_Stream_TypeDef *DMA_StreamX);

void BSP_Com_DMA_Rx_Init(USART_TypeDef           *USARTx,
                         DMA_Stream_TypeDef *DMA_StreamRx,
                         INT32U    DMA_ChRx,
                         void       *Rx_Buf,
                         unsigned int Rx_Buf_Size);
void BSP_Com_DMA_Tx_Init(USART_TypeDef           *USARTx,
                         DMA_Stream_TypeDef *DMA_StreamTx,
                         INT32U    DMA_ChTx,
                         void       *Tx_Buf,
                         unsigned int Tx_Buf_Size);
int BSP_Com_DMA_Tx(COM_PARAM* ComParam);

int BSP_Com_RS485_Ctrl(const UART_PARAM *UartParam, int opt);

void BSP_Com_IRQHandler(unsigned char com_num);

void BSP_Com_DMA_IRQHandler(DMA_Stream_TypeDef *DMA_Streamx);

void BSP_Com_SetAttr(COM_PARAM* UsartParam, struct termio* io);

void BSP_Com_Init(unsigned char com_num);

int    BSP_Com_ioctl(unsigned char subdev_num, int op, void* param);

void BSP_Com_Timer(void *ptmr, void *p_arg);

int BSP_Com_Probe(unsigned char com_num, int msec);

int BSP_Com_ReadData(unsigned char com_num, char* buff, int size);

int BSP_Com_SendData(unsigned char com_num, char* buff, int size);
```

```
int    BSP_Com_open(unsigned char subdev_num, int option);

void BSP_Com_close(unsigned char subdev_num);

void Dev_Com_Init(void);

/*****************************************************************************

* 功   能 : 232、485 等设备功能实现  dev_com.c end

*****************************************************************************/



/*****************************************************************************

* 功   能 : 标准输入、输出设备功能实现  dev_stdio.c

*****************************************************************************/

void Dev_Stdio_init(void)

int fputc(int ch, FILE *f)

int fgetc(FILE *f)

void print(const char *fmt, ...)

/*****************************************************************************

* 功   能 : 标准输入、输出设备功能实现  dev_stdio.c    end

*****************************************************************************/









/*****************************************************************************
```

```
* 功　能 : GPIO 输入、输出设备功能实现  dev_gpio.c

*********************************************************************************/

void dev_gio_up(unsigned char num);

void dev_gio_down(unsigned char num);

void dev_gio_turnover(unsigned char num);

void dev_gio_set(unsigned char num, int state);

int dev_gio_get(unsigned char num);

void dev_gio_rlc_init(void);

void dev_pwm_init(void);

void dev_pwm_set(unsigned char dev_num, unsigned short duty);

int dev_gio_open(unsigned char subdev_num, int option);

int    dev_gio_ioctl(unsigned char subdev_num, int op, void* param);

void dev_gio_close(unsigned char subdev_num);

void Dev_Gio_Init(void);

/*********************************************************************************

* 功　能 : GPIO 输入、输出设备功能实现  dev_gpio.c     end

*********************************************************************************/



/*********************************************************************************

* 功　能 : iic 输入、输出设备功能实现  dev_iic.c

*********************************************************************************/

I2Cx_PARAM *BSP_I2C_Get_Param(unsigned char i2c_num);
```

```
void BSP_I2C_Init(unsigned char i2c_num);

void BSP_I2C_Reset(unsigned char i2c_num);

void BSP_I2C_Start(unsigned char i2c_num);

int BSP_I2C_Timeout_UserCallback(unsigned char i2c_num, int error);

int BSP_I2C_SendData(unsigned char i2c_num, char *buff, int size);

int BSP_I2C_ReadData(unsigned char i2c_num, char* buff, int size);

int BSP_I2C_Ioctl(unsigned char i2c_num, int op, void* param);

int BSP_I2C_open(unsigned char i2c_num, int option);

void BSP_I2C_close(unsigned char i2c_num);

void Dev_IIC_Init(void);

/*****************************************************************************

* 功  能 : iic 输入、输出设备功能实现  dev_iic.c     end

*****************************************************************************/




/*****************************************************************************

* 功  能 : 片内 flash 读、写设备功能实现  dev_flash.c

*****************************************************************************/

unsigned char BSP_GetFlashSector(unsigned int pos);

FLASH_PARAM* BSP_Flash_Get_Param(unsigned char subdev_num);

int BSP_Flash_ReadData(unsigned char subdev_num, char* buff, int data_len);

int BSP_Flash_WriteData(unsigned char subdev_num, char* buff, int data_len);
```

```
int BSP_Flash_Erase(FLASH_PARAM* FlashParam);

int    BSP_Flash_ioctl(unsigned char subdev_num, int op, void* param);

int BSP_Flash_Seek(unsigned char subdev_num, int offset);

int    BSP_Flash_open(unsigned char subdev_num, int option);

void BSP_Flash_close(unsigned char subdev_num);

void Dev_Flash_Init(void);

/*******************************************************************************

* 功  能 : 片内 flash 读、写设备功能实现  dev_flash.c    end

*******************************************************************************/
```