一、GPIO_Config

1、函数 GPIO_INIT

函数名	GPIO_INIT
功能描述	GPIO 初始化函数
函数原型	int GPIO_INIT (GPIO_TypeDef* GPIOx, uint16_t pin, uint8_t Speed, uint8_t mode)
输入参数 1	GPIOx: 端口号 GPIOA GPIOB GPIOC GPIOD GPIOE GPIOF GPIOG
输入参数 2	pin: 引脚号 pin0 pin1 pin2 …… pin15
输入参数 3	Speed: 速度 LOW_Speed 2MHz MID_Speed 10MHz HIGH_Speed 50MHz
输入参数 4	mode: 模式 1: 模拟输入 2: 浮空输入 3: 下拉输入 4: 上拉输出 5: 开漏输出 6: 推挽输出 7: 复用开漏输出 8: 复用推挽输出
输入参数 5	CLK: 时钟线(RCC_APB2Periph) GPIOA GPIOB GPIOC GPIOD GPIOE GPIOF GPIOG
输出参数	无
返回值	0

例:

/*设置 GPIOA 的 8 号管脚为 50MHz 上拉输出模式*/

GPIO_INIT (GPIOA, pin8, HIGH_Speed, 4, GPIOA);

2、函数 Fast_OutputSet

函数名	Fast_OutputSet
功能描述	PORT 快速输出初始化函数，快速设置为 50MHz 推挽
函数原型	int Fast_OutputSet (GPIO_TypeDef* GPIOx, uint pin)
输入参数 1	GPIOx: 端口号 GPIOA GPIOB GPIOC GPIOD GPIOE GPIOF GPIOG
输入参数 2	pin: 引脚号 pin0 pin1 pin2 …… pin15
输出参数	无
返回值	0

例:

/*设置 GPIOA 的 4 号管脚为 50MHz 推挽输出模式*/

Fast_OutputSet (GPIOA, pin4);

3、函数 Fast_InputSet

函数名	Fast_InputSet
功能描述	PORT 快速输入初始化函数，可自行设置输入速度
函数原型	int Fast_OutputSet (GPIO_TypeDef* GPIOx, uint pin)
输入参数 1	GPIOx: 端口号 GPIOA　GPIOB　GPIOC　GPIOD GPIOE　GPIOF　GPIOG
输入参数 2	pin: 引脚号 pin0　pin1　pin2　……　pin15
输入参数 3	Speed: 速度 LOW_Speed　　2MHz MID_Speed　　10MHz HIGH_Speed　　50MHz
输出参数	无
返回值	0

例：

/*设置 GPIOA 的 4 号管脚为 10MHz 浮空输入模式*/

Fast_InputSet (GPIOA, pin4, MID_Speed);

4、函数 Change_Port_Mode

函数名	Change_Port_Mode
功能描述	快速管脚模式改变
函数原型	int Change_Port_Mode (GPIO_TypeDef* GPIOx, uint pin, uint mode, uint Speed)
输入参数 1	GPIOx: 端口号 GPIOA　GPIOB　GPIOC　GPIOD GPIOE　GPIOF　GPIOG
输入参数 2	pin: 引脚号 pin0　pin1　pin2　……　pin15
输入参数 3	mode: 模式 1: 模拟输入　2: 浮空输入　3: 下拉输入　　4: 上拉输出 5: 开漏输出　6: 推挽输出　7: 复用开漏输出　8: 复用推挽输出
输入参数 4	Speed: 速度 LOW_Speed　　2MHz MID_Speed　　10MHz HIGH_Speed　　50MHz
输出参数	无
返回值	0

例：

/*设置 GPIOC 的 3 号管脚为开漏输出模式*/

Chang_Port_Mode (GPIOC, pin3, 5);

5、函数 Cut_PIN_CLK

函数名	Cut_PIN_CLK
功能描述	端口静默函数，断开端口时钟
函数原型	int Cut_PIN_CLK (uint port)
输入参数	port: 端口号 GPIOA_SET　GPIOB_SET　GPIOC_SET　GPIOD_SET GPIOE_SET　GPIOF_SET　GPIOG_SET
输出参数	无
返回值	0

```
/*使 GPIOG 端口断开时钟*/
Cut_PIN_CLK (GPIOG);
```

6、函数 Reconnect_PIN_CLK

函数名	Reconnect_PIN_CLK
功能描述	解除端口静默函数 重连端口时钟
函数原型	int Reconnect_PIN_CLK (uint port)
输入参数	port: 端口号 GPIOA_SET　GPIOB_SET　GPIOC_SET　GPIOD_SET GPIOE_SET　GPIOF_SET　GPIOG_SET
输出参数	无
返回值	0
先决条件	调用 Cut_PIN_CLK(uint port) 之后

```
/*使 GPIOG 端口重连时钟*/
Reconnect_PIN_CLK (GPIOG);
```

7、函数 GPIO_OUT

函数名	GPIO_OUT
功能描述	输出高低电平数据
函数原型	int GPIO_OUT (GPIO_TypeDef* port, uint16_t pin, BitAction data)
输入参数 1	port: 端口号 GPIOA　GPIOB　GPIOC　GPIOD GPIOE　GPIOF　GPIOG
输入参数 2	pin: 引脚号 pin0　pin1　pin2　……　pin15
输入参数 3	data: 高低电平 Bit_SET　　1; Bit_RESET　0;
输出参数	无
返回值	0
先决条件	调用初始化端口之后

```
/*使 GPIOC 端口的 0 号引脚输出高电平*/
GPIO_OUT (GPIOC, pin0, 1);
```

8、函数 GPIO_READ

函数名	GPIO_READ
功能描述	读取输入高低电平数据
函数原型	uint8_t GPIO_READ (GPIO_TypeDef* GPIOx, u16 GPIO_Pin)
输入参数 1	GPIOx: 端口号 GPIOA　GPIOB　GPIOC　GPIOD GPIOE　GPIOF　GPIOG
输入参数 2	pin: 引脚号 pin0　pin1　pin2　……　pin15
输出参数	无
返回值	Read_data, 读取的数据
先决条件	调用初始化端口之后

```
/*读取 GPIOB 端口的 15 号引脚的输入电平存入变量 Level 中*/
Level = GPIO_READ (GPIOB, pin15);
```

9、函数 GPIO_Write_Port

函数名	GPIO_Write_Port
功能描述	端口组输出数据(16 进制)
函数原型	void GPIO_Write_Port (GPIO_TypeDef* GPIOx, u16 Val)
输入参数 1	GPIOx: 端口号 GPIOA　GPIOB　GPIOC　GPIOD GPIOE　GPIOF　GPIOG
输入参数 2	Val: 输出值 16 进制
输出参数	无
返回值	0
先决条件	调用初始化端口之后

```
/*输出数据 0x3f8b 到 GPIOC 端口*/
GPIO_Write_Port (GPIOC, 0x3f8b);
```

10、函数 GPIO_Read_Port

函数名	GPIO_Read_Port
功能描述	读取端口数据(16 位)
函数原型	u16 GPIO_Read_Port (GPIO_TypeDef* GPIOx)
输入参数	GPIOx: 端口号 GPIOA　GPIOB　GPIOC　GPIOD GPIOE　GPIOF　GPIOG
输出参数	无
返回值	Read_Val, 读取的数据
先决条件	调用初始化端口之后

```
/*读取 GPIOE 端口的数据放入变量 Value 中*/
Value = GPIO_Read_Port (GPIOE);
```

11、函数 CLK_SET

函数名	CLK_SET
功能描述	时钟挂载函数
函数原型	void CLK_SET (uint GPIOX)
输入参数	GPIOx: 端口号 GPIOA_SET　GPIOB_SET　GPIOC_SET　GPIOD_SET GPIOE_SET　GPIOF_SET　GPIOG_SET
输出参数	无
返回值	0

```
/*将 GPIOD 端口挂载在时钟线上*/
CLK_SET (GPIOD);
```

二、UART

1、函数 UART_CONFIG

函数名	UART_CONFIG
功能描述	串口初始化函数
函数原型	int UART_CONFIG (uint UARTx, uint BaudRate, uint Word_Length, uint Stop_Bits, uint Parity)
输入参数 1	UARTx: 串口号 UART1_SET　UART2_SET　UART3_SET UART4_SET　UART5_SET
输入参数 2	BaudRate: 波特率 9600　115200　25600 ……
输入参数 3	Word Length: 字长 8　　9
输入参数 4	Stop_Bits: 停止位 1　　2
输入参数 5	Parity: 校验位 0　NO 1　ODD 2　EVEN
输出参数	无
返回值	0

2、函数 Usart_SendByte

函数名	Usart_SendByte
功能描述	串口发送一个字节
函数原型	void Usart_SendByte (USART_TypeDef * pUSARTx, uint8_t ch)
输入参数 1	UARTx: 串口号 UART1　UART2　UART3 UART4　UART5
输入参数 2	ch: 内容
输出参数	无
返回值	无

3、函数 Usart_SendString

函数名	Usart_SendString
功能描述	串口发送字符串
函数原型	void Usart_SendString (USART_TypeDef * pUSARTx, char *str)
输入参数 1	UARTx: 串口号 UART1　UART2　UART3 UART4　UART5
输入参数 2	str: 字符串
输出参数	无

返回值	无

4、函数 Usart_SendHalfWord

函数名	Usart_SendHalfWord
功能描述	串口发送 16 位数
函数原型	void Usart_SendHalfWord (USART_TypeDef * pUSARTx, uint16_t ch)
输入参数 1	UARTx: 串口号 UART1 UART2 UART3 UART4 UART5
输入参数 2	ch: 内容
输出参数	无
返回值	无

5、函数 Usart_SendArray

函数名	Usart_SendArray
功能描述	串口发送一个 8 位数组
函数原型	void Usart_SendArray (USART_TypeDef * pUSARTx, uint8_t *array, uint16_t num)
输入参数 1	UARTx: 串口号 UART1 UART2 UART3 UART4 UART5
输入参数 2	array: 数组
输入参数 3	num: 内容个数
输出参数	无
返回值	无

6、函数 fputc

函数名	fputc
功能描述	重定向后可使用 printf 函数
函数原型	int fputc (int ch, FILE *f)
输出参数	无
返回值	ch

7、函数 fgetc

函数名	fgetc
功能描述	重定向 c 库函数 scanf 到串口，重写向后可使用 scanf、getchar 等函数
函数原型	int fgetc (FILE *f)
输出参数	无

8、函数 Fast_UART_SET

函数名	Fast_UART_SET
功能描述	快速 UART 设置

	默认配置：波特率 115200，字长 8，不进行校验，停止位 1
函数原型	void Fast_UART_SET (uint UARTx)
输入参数	UARTx: 串口号 UART1_SET　UART2_SET　UART3_SET UART4_SET　UART5_SET
输出参数	无
返回值	无

9、函数 set_P_S_VECTOR

函数名	set_P_S_VECTOR
功能描述	重定向 C 库函数
函数原型	void set_P_S_VECTOR (USART_TypeDef * pUSARTx)
输入参数	pUSARTx: 串口号 USART1　USART2　USART3 UART4　　UART5
输出参数	无