UM800Y API 参考手册

版本：V1.0

UNICMICRO
广芯微电子

广芯微电子（广州）股份有限公司

http://www.unicmicro.com/

条款协议

本文档的所有部分，其著作产权归广芯微电子（广州）股份有限公司（以下简称广芯微电子）所有，未经广芯微电子授权许可，任何个人及组织不得复制、转载、仿制本文档的全部或部分组件。本文档没有任何形式的担保、立场表达或其他暗示，若有任何因本文档或其中提及的产品所有资讯所引起的直接或间接损失，广芯微电子及所属员工恕不为其担保任何责任。除此以外，本文档所提到的产品规格及资讯仅供参考，内容亦会随时更新，恕不另行通知。

1. 本文档中所记载的关于电路、软件和其他相关信息仅用于说明半导体产品的操作和应用实例。用户如在设备设计中应用本文档中的电路、软件和相关信息，请自行负责。对于用户或第三方因使用上述电路、软件或信息而遭受的任何损失，广芯微电子不承担任何责任。
2. 在准备本文档所记载的信息的过程中，广芯微电子已尽量做到合理注意，但是，广芯微电子并不保证这些信息都是准确无误的。用户因本文档中所记载的信息的错误或遗漏而遭受的任何损失，广芯微电子不承担任何责任。
3. 对于因使用本文档中的广芯微电子产品或技术信息而造成的侵权行为或因此而侵犯第三方的专利、版权或其他知识产权的行为，广芯微电子不承担任何责任。本文档所记载的内容不应视为对广芯微电子或其他人所有的专利、版权或其他知识产权作出任何明示、默示或其它方式的许可及授权。
4. 使用本文档中记载的广芯微电子产品时，应在广芯微电子指定的范围内，特别是在最大额定值、电源工作电压范围、热辐射特性、安装条件以及其他产品特性的范围内使用。对于在上述指定范围之外使用广芯微电子产品而产生的故障或损失，广芯微电子不承担任何责任。
5. 虽然广芯微电子一直致力于提高广芯微电子产品的质量和可靠性，但是，半导体产品有其自身的具体特性，如一定的故障发生率以及在某些使用条件下会发生故障等。此外，广芯微电子产品均未进行防辐射设计。所以请采取安全保护措施，以避免当广芯微电子产品在发生故障而造成火灾时导致人身事故、伤害或损害的事故。例如进行软硬件安全设计（包括但不限于冗余设计、防火控制以及故障预防等）、适当的老化处理或其他适当的措施等。

版本修订

版本	日期	描述
V1.0	2022.03.12	初始版

目录

1 应用程序接口（API）

1.1 ADC接口

#include "adc.h"

1.1.1 adc_clear_buf_state

功能	adc_clear_buf_state
描述	adc_clear_buf_state
函数定义	void adc_clear_buf_state (void)
参数	none
返回	none

函数的调用关系图：

1.1.2 adc_clk_config

功能	adc_clk_config
描述	ADC时钟源配置
函数定义	void adc_clk_config (uint8_t clksource, uint8_t vrefsource, uint16_t clksource_div, uint8_t newstate)
参数	uint8_t clksource ： ADC CLK SOURCE uint8_t vrefsource ： ADC VREF SOURCE uint16_t clksource_div ： fadc_clk = fpclk / clkdiv，clkdiv = {clkdiv1, clkdiv0} uint8_t newstate ： ADC ENABLE/DISABLE
返回	none

1.1.3 adc_controller_config

功能	adc_controller_config
描述	ADC ENABLE/DISABLE配置
函数定义	void adc_controller_config (uint8_t newstate)
参数	uint8_t newstate ： ADC ENABLE/DISABLE
返回	none

1.1.4 adc_convert_start

功能	adc_convert_start
描述	ADC转换开始
函数定义	void adc_convert_start (uint8_t ch)
参数	uint8_t ch ：ADC Channel return
返回	none

函数的调用关系图:

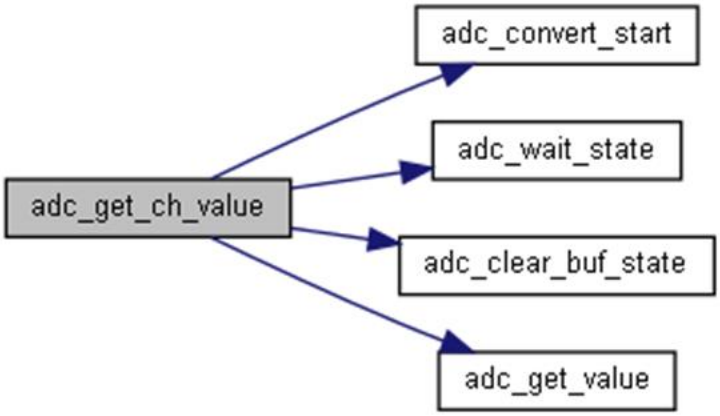
1.1.5 adc_convert_stop

功能	adc_convert_stop
描述	ADC转换完成
函数定义	void adc_convert_stop (void)
参数	none
返回	none

1.1.6 adc_get_ch_value

功能	adc_get_ch_value
描述	adc_get_ch_value
函数定义	uint16_t adc_get_ch_value (uint8_t ch)
参数	uint8_t ch：ADC Channel
返回	12bit convert value of adc

函数调用关系图：



1.1.7 adc_get_value

功能	adc_get_value
描述	ADC速率补偿
函数定义	uint16_t adc_get_value (void)
参数	none
返回	12bit convert value of adc

函数的调用关系图:

1.1.8 adc_io_config

功能	adc_io_config
描述	ADC IO配置
函数定义	void adc_io_config (uint8_t ch)
参数	uint8_t ch：ADC Channel
返回	none

1.1.9 adc_irq_config

功能	adc_irq_config
描述	ADC中断配置
函数定义	void adc_irq_config (void(*)() adc_irq_pro, uint8_t newstate)
参数	void (*adc_irq_pro)() ： call back function uint8_t newstate ： ADC ENABLE/DISABLE
返回	none

1.1.10 ADC_IRQHandler

功能	ADC_IRQHandler
描述	ADC中断处理
函数定义	void ADC_IRQHandler (void)
参数	none
返回	none

1.1.11 adc_power_config

功能	adc_power_config
描述	ADC电源配置
函数定义	void adc_power_config (uint8_t newstate)

参数	uint8_t newstate ： ADC ENABLE/DISABLE
返回	none

1.1.12 adc_priority_level_config

功能	adc_priority_level_config
描述	ADC优先级配置
函数定义	void adc_priority_level_config (uint8_t level)
参数	uint8_t level：ADC PRIORITY LEVEL
返回	none

1.1.13 adc_sample_clk_config

功能	adc_sample_clk_config
描述	ADC采样时钟脉冲宽度配置
函数定义	void adc_sample_clk_config (uint8_t sampclk)
参数	uint8_t sampclk ： ADC SAMPLE CLK
返回	none

1.1.14 adc_scan_mode_config

功能	adc_scan_mode_config
描述	ADC扫描模式配置
函数定义	void adc_scan_mode_config (uint8_t mode)
参数	uint8_t mode ： ADC CONVERT MODE
返回	none

1.1.15 adc_wait_state

功能	adc_wait_state
描述	ADC等待状态
函数定义	uint8_t adc_wait_state (uint8_t state, uint16_t timeout)
参数	uint8_t state ： ADC STATE uint16_t timeout ： ADC TIME OUT
返回	1：FAIL 0：SUCCESS

函数的调用关系图:

1.2 BEEPER接口

#include "beeper.h"

1.2.1 void beeper_init

功能	beeper_init
描述	Beeper初始化
函数定义	void beeper_init(void)
参数	none
返回	none

1.2.2 beeper_level_set

功能	beeper_level_set
描述	set beeper output polarity
函数定义	void beeper_level_set(uint8_t level,uint8_t freq)
参数	uint8_t level：0 低电平 1 高电平 uint8_t freq：频率
返回	none

1.3 CAN接口

#include "can.h"

1.3.1 can_filter_config

功能	can_filter_config
描述	can过滤器配置
函数定义	void can_filter_config(S_Can_Filter_Msg *can_filter_msg,uint8_t filter_mode, uint8_t filter_en)
参数	S_Can_Filter_Msg *can_filter_msg：*can_filter_msg uint8_t filter_mode：CAN工作复位模式 uint8_t filter_en：使能can过滤器
返回	none

1.3.2 can_get_rmc_reg

功能	can_get_rmc_reg
描述	获取can rx fifo data 个数
函数定义	uint8_t can_get_rmc_reg(void)
参数	none
返回	none

1.3.3 can_get_sr_reg

功能	can_get_sr_reg
描述	获取can状态寄存器值
函数定义	uint8_t can_get_sr_reg(void)
参数	none
返回	none

函数的调用关系图:

1.3.4 can_init

功能	can_init
描述	can初始化
函数定义	void can_init(uint8_t baudrate)
参数	uint8_t baudrate： baudrate 速率可以选择为1M/500k/250k/125k bps Fpclk=16Mhz, CAN波特率 BitRate = Fpclk/2*((BRP+1)*(TS1+TS2+3)), 有个约定:TS1>=TS2 设波特率为1M的参数： 设置BRP=0(2分频), BitRate = 1M = 16M/2((0+1)*(TS1+TS2+3)),所以可以设置TS1=3,TS2=2 设波特率为500K的参数： 设置BRP=1(4分频), BitRate = 0.5M = 16M/2((1+1)*(TS1+TS2+3)),所以可以设置TS1=3,TS2=2 设波特率为250K的参数： 设置BRP=3(8分频), BitRate = 0.25M = 16M/2((3+1)*(TS1+TS2+3)),所以可以设置TS1=3,TS2=2 设波特率为125K的参数： 设置BRP=7(16分频), BitRate = 0.125M = 16M/2((7+1)*(TS1+TS2+3)),所以可以设置TS1=3,TS2=2
返回	none

1.3.5 can_irq_init

功能	can_irq_init
描述	can中断初始化
函数定义	void can_irq_init(uint8_t irq_able, void (*pfunc)())
参数	uint8_t irq_able：filter_value void(*)() pfunc：回调函数
返回	none

1.3.6 CAN_IRQHandler

功能	CAN_IRQHandler
描述	can中断处理
函数定义	void CAN_IRQHandler (void)
参数	none
返回	none

1.3.7 can_recv_data

功能	can_recv_data
描述	can接收数据
函数定义	void can_recv_data(S_Can_Rx_Msg *can_rx_msg)
参数	S_Can_Rx_Msg *can_rx_msg：*can_rx_msg
返回	none

1.3.8 can_send_data

功能	can_send_data
描述	can发送数据
函数定义	void can_send_data(S_Can_Tx_Msg *can_tx_msg)
参数	S_Can_Tx_Msg *can_tx_msg：can_tx_msg
返回	none

1.4 EFLASH接口

#include "eflash.h"

1.4.1 eflash_erase_chip

功能	eflash_erase_chip
描述	eflash擦除
函数定义	void eflash_erase_chip (void)
参数	none
返回	none

函数调用图:

1.4.2 eflash_erase_page

功能	eflash_erase_page
描述	Page擦除
函数定义	void eflash_erase_page (uint16_t page_addr)
参数	uint16_t page_addr：page地址
返回	none

1.4.3 eflash_init

功能	eflash_init
描述	Eflash初始化
函数定义	void eflash_init (uint32_t system_clk_hz)
参数	uint32_t system_clk_hz：系统时钟频率
返回	none

1.4.4 eflash_read_byte

功能	eflash_read_byte
描述	Eflash读一个字节
函数定义	uint8_t eflash_read_byte (uint16_t addr)
参数	uint16_t addr：数据地址
返回	none

1.4.5 eflash_read_bytes

功能	eflash_read_bytes
描述	Eflash读多个字节
函数定义	uint32_t eflash_read_bytes (uint16_t addr, uint8_t * buff, uint32_t length)
参数	uint16_t addr：数据地址 uint8_t * buff：buff数据 uint32_t length：数据长度
返回	none

1.4.6 eflash_read_dword

功能	eflash_read_dword
描述	Eflash读dword
函数定义	uint32_t eflash_read_dword (uint16_t addr)
参数	uint16_t addr：数据地址
返回	none

1.4.7 eflash_read_word

功能	eflash_read_word
描述	Eflash读word
函数定义	uint16_t eflash_read_word (uint16_t addr)
参数	uint16_t addr：数据地址
返回	none

1.4.8 eflash_write_byte

功能	eflash_write_byte
描述	Eflash写一个字节
函数定义	void eflash_write_byte (uint16_t addr, uint8_t value)
参数	uint16_t addr：数据地址 uint8_t value：数据值
返回	none

1.4.9 eflash_write_bytes

功能	eflash_write_bytes
描述	Eflash写多个字节
函数定义	void eflash_write_bytes (uint16_t addr, uint8_t * buff, uint32_t length)
参数	uint16_t addr：数据地址 uint8_t * buff：buff数据 uint32_t length：buff长度
返回	none

1.4.10 eflash_write_dword

功能	eflash_write_dword
描述	Eflash写dword
函数定义	void eflash_write_dword (uint16_t addr, uint32_t value)
参数	uint16_t addr：数据地址 uint32_t value：数据值
返回	none

1.4.11 eflash_write_word

功能	eflash_write_word
描述	Eflash写word
函数定义	void eflash_write_word (uint16_t addr, uint16_t value)
参数	uint16_t addr：数据地址 uint16_t value：数据值
返回	none

1.5 GPIO接口

#include "gpio.h"

1.5.1 gpio_cs_set

功能	gpio_cs_set
描述	set gpio input type，GPIO管脚输入类型配置
函数定义	void gpio_cs_set(uint8_t pin, uint8_t mode)
参数	uint8_t pin： P00，P01，P02 ... uint8_t mode ： GPIO_CS_CMOS：CMOS input buffer; GPIO_CS_SCHMITT：Schmitt input buffer
返回	none

1.5.2 gpio_dir_set

功能	gpio_dir_set
描述	set direction of gpio pin，GPIO管脚数据流方向配置
函数定义	void gpio_dir_set(uint8_t pin, uint8_t dir)
参数	uint8_t pin：P00，P01，P02... uint8_t dir： GPIO_DIR_OUT：输出 GPIO_DIR_IN：输入
返回	none

1.5.3 gpio_dr_set

功能	gpio_dr_set
描述	set gpio driving power，GPIO管脚驱动能力配置
函数定义	void gpio_dr_set(uint8_t pin, uint8_t mode)
参数	uint8_t pin：P00，P01，P02 ... uint8_t mode： GPIO_DR_HIGH：高驱动能力 GPIO_DR_LOW：低驱动能力
返回	none

1.5.4 gpio_in_enable

功能	gpio_in_enable
描述	GPIO输入使能
函数定义	void gpio_in_enable(uint8_t pin, uint8_t mode)
参数	uint8_t pin：P00，P01，P02... uint8_t mode： IN_ENABLE：输入使能 IN_DISABLE：输入禁止
返回	none

1.5.5 gpio_init

功能	gpio_init
描述	GPIO初始化，包括开时钟，模块正常工作
函数定义	void gpio_init (uint8_t pin)
参数	uint8_t pin：P00，P01，P02...
返回	none

1.5.6 gpio_io_get

功能	gpio_io_get
描述	get gpio pin value，GPIO输入电平获取
函数定义	uint8_t gpio_io_get(uint8_t pin)
参数	uint8_t pin： P00，P01，P02...
返回	管脚电平状态

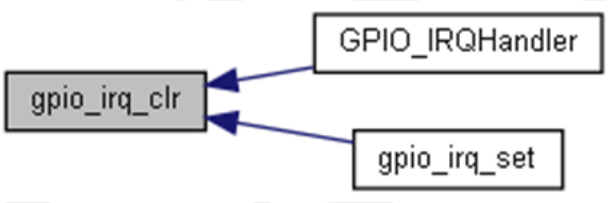
1.5.7 gpio_io_set

功能	gpio_io_set
描述	gpio set 1 or 0，GPIO电平输出
函数定义	void gpio_io_set(uint8_t pin, uint8_t level)
参数	uint8_t pin：P00，P01，P02... uint8_t level： GPIO_HIGH：1 GPIO_LOW：0
返回	none

1.5.8 gpio_irq_clr

功能	gpio_irq_clr
描述	clear the interrupt flag of gpio pin，GPIO中断清除
函数定义	void gpio_irq_clr(uint8_t pin)
参数	uint8_t pin：P00，P01，P02 ...
返回	none

函数的调用关系图:



1.5.9 gpio_irq_get

功能	gpio_irq_get
描述	get the interrupt flag of gpio pin，GPIO中断状态获取
函数定义	uint8_t gpio_irq_get(uint8_t pin)
参数	uint8_t pin：P00，P01，P02 ...
返回	管脚中断状态

函数的调用关系图:

1.5.10 gpio_irq_set

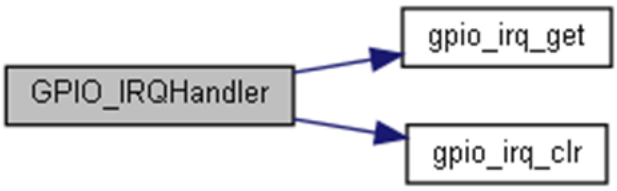
功能	gpio_irq_set
描述	gpio interrupt configuration，GPIO中断使能，失能
函数定义	void gpio_irq_set(uint8_t pin, uint8_t mode, void (*pfunc)())
参数	uint8_t pin：P00，P01，P02... uint8_t mode： GPIO_IRQ_ENABLE: 使能; GPIO_IRQ_DISABLE：失能 void (*pfunc)() ：中断处理回调函数
返回	none

函数调用图:

1.5.11 GPIO_IRQHandler

功能	GPIO_IRQHandler
描述	GPIO interrupt handling
函数定义	void GPIO_IRQHandler(void)
参数	none
返回	none

函数调用图:



1.5.12 gpio_od_set

功能	gpio_od_set
描述	set gpio output open drain，GPIO管脚开漏输出配置
函数定义	void gpio_od_set(uint8_t pin, uint8_t mode)
参数	uint8_t pin： P00，P01，P02 ... uint8_t mode： GPIO_OD_ENABLE：开漏使能 GPIO_OD_DISABLE：开漏禁止
返回	none

1.5.13 gpio_pd_set

功能	gpio_pd_set
描述	set pulldown function，GPIO管脚下拉配置
函数定义	void gpio_pd_set (uint8_t pin, uint8_t mode)
参数	uint8_t pin： P00，P01，P02... uint8_t mode： GPIO_PD_ENABLE：下拉使能 GPIO_PD_DISABLE：下拉禁止
返回	none

1.5.14 gpio_pu_set

功能	gpio_pu_set
描述	GPIO管脚上拉配置
函数定义	void gpio_pu_set(uint8_t pin, uint8_t mode)
参数	uint8_t pin：P00，P01，P02... uint8_t mode： GPIO_PU_ENABLE：上拉使能 GPIO_PU_DISABLE：上拉禁止
返回	none

1.5.15 gpio_sr_set

功能	gpio_sr_set
描述	set gpio speed，GPIO管脚速度配置
函数定义	void gpio_sr_set(uint8_t pin, uint8_t mode)
参数	uint8_t pin： P00，P01，P02 ... uint8_t mode： GPIO_SR_HIGH：快速 GPIO_SR_LOW：慢速
返回	none

1.6 GTIMER接口

#include "gtimer.h"

1.6.1 GTIMER0

1.6.1.1 GTIM0_IRQHandler

功能	GTIM0_IRQHandler
描述	GTIM0中断处理
函数定义	void GTIM0_IRQHandler(void)
参数	none
返回	none

1.6.1.2 gtimer0_bke_init

功能	gtimer0_bke_init
描述	GTIMER0刹车初始化
函数定义	void gtimer0_bke_init(void)
参数	none
返回	none

1.6.1.3 gtimer0_capture_init

功能	gtimer0_capture_init
描述	GTIMER0输入捕获
函数定义	void gtimer0_capture_init(uint16_t arr,uint8_t capture_div,uint8_t capssel)
参数	uint16_t arr：装载值 uint8_t capture_div：捕捉源预分频位 uint8_t capssel：捕捉源选择位
返回	none

1.6.1.4 gtimer0_count_init

功能	gtimer0_count_init
描述	GTIMER0 count初始化
函数定义	void gtimer0_count_init(uint16_t arr,uint16_t psc)
参数	Uint16_t arr：GTIMER0重载值 Uint16_t psc：GTIMER0分频值
返回	none

1.6.1.5 gtimer0_irq_init

功能	gtimer0_irq_init
描述	GTIMER0 中断初始化
函数定义	void gtimer0_irq_init(uint8_t irq_enable,uint8_t gtimer_irq_type,void (*pfunc)())
参数	uint8_t irq_enable：GTIMER0中断使能开关 1：使能 0：禁止 uint8_t gtimer_irq_type：GTIMER0中断类型选择 void (*pfunc)()：GTIMER0中断回调函数
返回	none

1.6.1.6 gtimer0_pwm_init

功能	gtimer0_pwm_init
描述	GTIMER0 PWM初始化
函数定义	void gtimer0_pwm_init(uint16_t arr, uint16_t psc, uint16_t ccr)
参数	uint16_t arr：GTIMER0重载值 uint16_t psc：GTIMER0分频值 uint16_t ccr：GTIMER0比较值
返回	none

1.6.1.7 gtimer0_start

功能	gtimer0_start
描述	启动GTIMER0计数
函数定义	void gtimer0_start(void)
参数	none
返回	none

1.6.1.8 gtimer0_stop

功能	gtimer0_stop
描述	停止GTIMER0计数器
函数定义	void gtimer0_stop(void)
参数	none
返回	none

1.6.2 GTIMER1

1.6.2.1 GTIM1_IRQHandler

功能	GTIM1_IRQHandler
描述	GTIM1 interrupt handling

函数定义	void GTIM1_IRQHandler(void)
参数	none
返回	none

1.6.2.2 gtimer1_bke_init

功能	gtimer1_bke_init
描述	GTIMER1刹车初始化
函数定义	void gtimer1_bke_init(void)
参数	none
返回	none

1.6.2.3 gtimer1_capture_init

功能	gtimer1_capture_init
描述	GTIMER1输入捕获
函数定义	void gtimer1_capture_init(uint16_t arr,uint8_t capture_div,uint8_t capssel)
参数	uint16_t arr：装载值 uint8_t capture_div：捕捉源预分频位 uint8_t capssel：捕捉源选择位
返回	none

1.6.2.4 gtimer1_count_init

功能	gtimer1_count_init
描述	GTIMER1 count初始化
函数定义	void gtimer1_count_init(uint16_t arr,uint16_t psc)
参数	uint16_t arr：GTIMER1重载值 uint16_t psc：GTIMER1分频值
返回	none

1.6.2.5 gtimer1_irq_init

功能	gtimer1_irq_init
描述	GTIMER1 中断初始化
函数定义	void gtimer1_irq_init(uint8_t irq_enable,uint8_t gtimer_irq_type,void (*pfunc)())
参数	uint8_t irq_enable：GTIMER1中断使能开关 1：使能 0：禁止 uint8_t gtimer_irq_type：GTIMER1中断类型选择 void (*pfunc)() ：GTIMER1中断回调函数
返回	none

1.6.2.6 gtimer1_pwm_init

功能	gtimer1_pwm_init
描述	GTIMER1 PWM初始化
函数定义	void gtimer1_pwm_init(uint16_t arr, uint16_t psc, uint16_t ccr)
参数	uint16_t arr：GTIMER1重载值 uint16_t psc：GTIMER1分频值 uint16_t ccr：GTIMER1比较值
返回	none

1.6.2.7 gtimer1_start

功能	gtimer1_start
描述	启动GTIMER1计数
函数定义	void gtimer1_start(void)
参数	none
返回	none

1.6.2.8 gtimer1_stop

功能	gtimer1_stop
描述	停止GTIMER1计数器
函数定义	void gtimer1_stop(void)
参数	none
返回	none

1.6.3 GTIMER2

1.6.3.1 GTIM2_IRQHandler

功能	GTIM2_IRQHandler
描述	GTIMER2 interrupt handling
函数定义	void GTIM2_IRQHandler(void)
参数	none
返回	none

1.6.3.2 gtimer2_bke_init

功能	gtimer2_bke_init
描述	GTIMER2刹车初始化
函数定义	void gtimer2_bke_init(void)
参数	none
返回	none

1.6.3.3 gtimer2_capture_init

功能	Gtimer2_capture_init
描述	GTIMER2输入捕获
函数定义	void gtimer2_capture_init(uint16_t arr,uint8_t capture_div,uint8_t capssel)
参数	uint16_t arr：装载值 uint8_t capture_div：捕捉源预分频位 uint8_t capssel：捕捉源选择位
返回	none

1.6.3.4 gtimer2_count_init

功能	gtimer2_count_init
描述	GTIMER2 count初始化
函数定义	void gtimer2_count_init(uint16_t arr,uint16_t psc)
参数	uint16_t arr：GTIMER2重载值 uint16_t psc：GTIMER2分频值
返回	none

1.6.3.5 gtimer2_irq_init

功能	gtimer2_irq_init
描述	GTIMER2 中断初始化
函数定义	void gtimer2_irq_init(uint8_t irq_enable,uint8_t gtimer_irq_type,void (*pfunc)())
参数	uint8_t irq_enable：GTIMER2中断使能开关 1：使能 0：禁止 uint8_t gtimer_irq_type：GTIMER2中断类型选择 void (*pfunc)()：GTIMER2中断回调函数
返回	none

1.6.3.6 gtimer2_pwm_init

功能	gtimer2_pwm_init
描述	GTIMER2 PWM初始化
函数定义	void gtimer2_pwm_init(uint16_t arr, uint16_t psc, uint16_t ccr)
参数	uint16_t arr：GTIMER2重载值 uint16_t psc：GTIMER2分频值 uint16_t ccr：GTIMER2比较值
返回	none

1.6.3.7 gtimer2_start

功能	gtimer2_start
描述	启动GTIMER2计数
函数定义	void gtimer2_start(void)
参数	none
返回	none

1.6.3.8 gtimer2_stop

功能	gtimer2_stop
描述	停止GTIMER2计数器
函数定义	void gtimer2_stop(void)
参数	none
返回	none

1.7 I2C接口

#include "i2c.h"

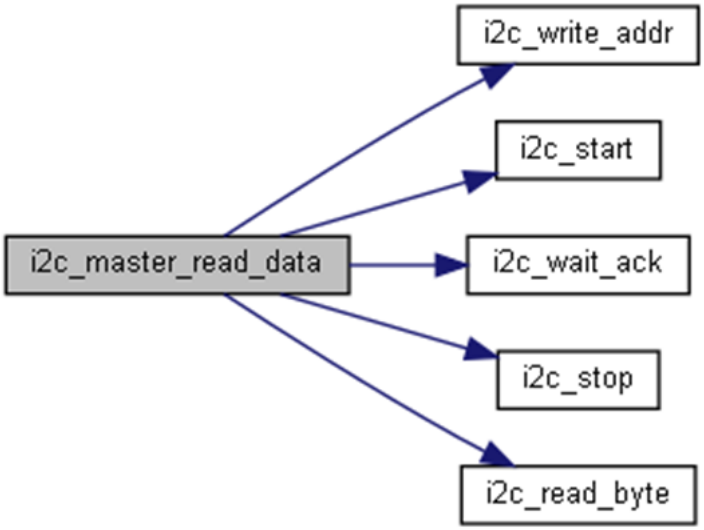
1.7.1 i2c_master_init

功能	i2c_master_init
描述	I2C主机初始化
函数定义	void i2c_master_init(uint32_t i2c_speed)
参数	uint32_t i2c_speed：I2C速率 = (fSYSCLK)/(4*(CLK_DIV+1))
返回	none

1.7.2 i2c_master_read_data

功能	i2c_master_read_data
描述	i2c读主机数据
函数定义	uint8_t i2c_master_read_data(uint8_t slave_addr, uint8_t *buffer, uint16_t len)
参数	uint8_t slave_addr：设备地址 uint8_t* buffer：要接收的数据 uint16_t len：数据传输长度
返回	1：fail 0:success

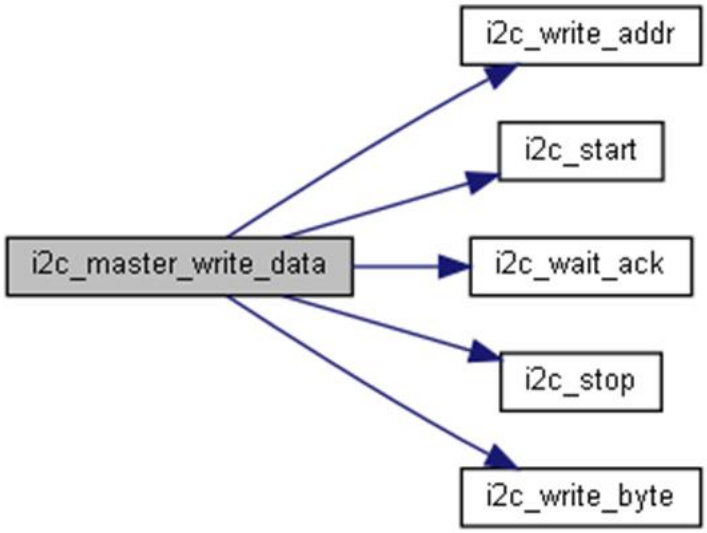
函数调用图:



1.7.3 i2c_master_write_data

功能	i2c_master_write_data
描述	i2c写主机数据
函数定义	uint8_t i2c_master_write_data(uint8_t slave_addr, uint8_t *buffer, uint16_t len)
参数	uint8_t slave_addr：从机地址 uint8_t *buffer：要发送的数据 uint16_t len：数据传输长度
返回	1：fail 0:success

函数调用图:



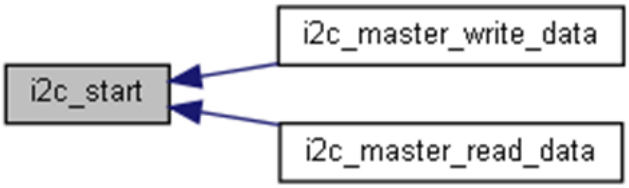
1.7.4 i2c_restart

功能	i2c_restart
描述	产生restart条件
函数定义	void i2c_restart(void)
参数	none
返回	none

1.7.5 i2c_start

功能	i2c_start
描述	产生start条件
函数定义	void i2c_start(void)
参数	none
返回	none

函数的调用关系图:



1.7.6 i2c_stop

功能	i2c_stop
描述	产生stop条件
函数定义	void i2c_stop(void)
参数	none
返回	none

函数的调用关系图:

1.7.7 i2c_ack

功能	i2c_ack
描述	i2c应答ack
函数定义	void i2c_ack(void)
参数	none
返回	none

1.7.8 i2c_deinit

功能	i2c_deinit
描述	关闭I2C模块
函数定义	void i2c_deinit(void)
参数	none
返回	none

1.7.9 i2c_irq_disable

功能	i2c_irq_disable
描述	关闭中断
函数定义	void i2c_irq_disable(uint8_t irq_type)
参数	uint8_t irq_type：i2c中断源
返回	none

1.7.10 i2c_irq_enable

功能	i2c_irq_enable
描述	使能I2C中断
函数定义	void i2c_irq_enable(uint8_t irq_type,void (*pfunc)())
参数	uint8_t irq_type：i2c中断源 void (*pfunc)()：产生中断之后执行的回调函数
返回	none

1.7.11 I2C_IRQHandler

功能	I2C_IRQHandler
描述	I2C中断处理
函数定义	void I2C_IRQHandler (void)
参数	none
返回	none

1.7.12 i2c_no_ack

功能	i2c_no_ack
描述	i2c不应答ack
函数定义	void i2c_no_ack(void)
参数	none
返回	none

1.7.13 i2c_read_byte

功能	i2c_read_byte
描述	读一个byte
函数定义	uint8_t i2c_read_byte(void)
参数	none
返回	uint8_t byte

1.7.14 i2c_set_slave_addr1

功能	i2c_set_slave_addr1
描述	设置从机匹配地址
函数定义	void i2c_set_slave_addr1(uint8_t slave_addr)
参数	uint8_t slave_addr：从机地址
返回	none

1.7.15 i2c_set_slave_addr2

功能	i2c_set_slave_addr2
描述	设置从机匹配地址
函数定义	void i2c_set_slave_addr2(uint8_t slave_addr)
参数	uint8_t slave_addr：从机地址
返回	none

1.7.16 i2c_slave_init

功能	i2c_slave_init
描述	I2C从机初始化
函数定义	void i2c_slave_init(void)
参数	none
返回	none

1.7.17 i2c_slave_is_match_addr1

功能	i2c_slave_is_match_addr1
描述	用于判断是否匹配地址1
函数定义	uint8_t i2c_slave_is_match_addr1(void)
参数	none
返回	0-否 1-是

1.7.18 i2c_slave_is_match_addr2

功能	i2c_slave_is_match_addr3
描述	用于判断是否匹配地址3
函数定义	uint8_t i2c_slave_is_match_addr2(void)
参数	none
返回	0-否 1-是

1.7.19 i2c_slave_read

功能	i2c_slave_read
描述	i2c读从机数据
函数定义	uint8_t i2c_slave_read(uint8_t *rxdata)
参数	uint8_t *rxdata: 要接收的数据
返回	datalen 数据传输长度

函数调用图:

1.7.20 i2c_slave_write

功能	i2c_slave_write
描述	i2c写从机数据
函数定义	uint8_t i2c_slave_write(uint8_t *txdata)
参数	uint8_t* txdata：要发送的数据
返回	datalen 数据传输长度

1.7.21 i2c_wait_ack

功能	i2c_wait_ack
描述	等待应答
函数定义	uint8_t i2c_wait_ack(void)
参数	none
返回	1：NACK 0：ACK

1.7.22 i2c_write_addr

功能	i2c_write_addr
描述	发送I2C地址+R/W标志
函数定义	void i2c_write_addr(uint8_t addr)
参数	uint8_t addr：数据地址
返回	none

1.7.23 i2c_write_byte

功能	i2c_write_byte
描述	写一个byte
函数定义	void i2c_write_byte(uint8_t byte)

参数	uint8_t byte：数据值
返回	none

1.8 LPTIMER接口

#include "lptimer.h"

1.8.1 lptim_get_cnt_value

功能	lptim_get_cnt_value
描述	获取计数值
函数定义	uint16_t lptim_get_cnt_value(void)
参数	none
返回	none

1.8.2 lptimer_capture_init

功能	lptimer_capture_init
描述	Lptimer捕获初始化
函数定义	void lptimer_capture_init(uint8_t channel, uint8_t edge)
参数	uint8_t channel：PWM通道 uint8_t edge：上升沿/下降沿
返回	none

1.8.3 lptimer_get_lptcmp

功能	lptimer_get_lptcmp
描述	lptimer_get_lptcmp
函数定义	uint16_t lptimer_get_lptcmp(uint8_t channel)
参数	uint8_t channel：PWM通道
返回	none

1.8.4 lptimer_init

功能	lptimer_init
描述	lptimer 初始化
函数定义	void lptimer_init(uint8_t tmode, uint8_t clock_sel, uint8_t div)
参数	tmode： 0：普通定时器模式 1：Trigger脉冲触发计数模式 2：外部异步脉冲计数模式 3：Timeout模式

	uint8_t clock_sel： 时钟源设置 0：RCL低速时钟 1：clk_1hz时钟 2：系统时钟 3：外部输入时钟 uint8_t div：计数时钟分频
返回	none

1.8.5 lptimer_io_config

功能	lptimer_outio
描述	lptimer 输出管脚配置
函数定义	void lptimer_io_config(uint8_t io)
参数	uint8_t io：io端口
返回	none

1.8.6 lptimer_irq_init

功能	lptimer_irq_init
描述	lptimer 中断初始化
函数定义	void lptimer_irq_init(uint8_t irqstate,uint8_t ie,void (*pfunc)())
参数	uint8_t irqstate： 0：中断失能 1：中断使能 uint8_t ie： 0：比较匹配中断使能 1：计数器溢出中断使能 2：外部触发到来中断使能 void (*pfunc)()：中断回调函数
返回	none

1.8.7 LPTIMER_IRQHandler

功能	lptimer_IRQHandler
描述	lptimer interrupt handler
函数定义	void LPTIMER_IRQHandler(void)
参数	none
返回	none

1.8.8 lptimer_lptin_edge_set

功能	lptimer_lptin_edge_set
描述	lptimer_lptin边沿选择
函数定义	void lptimer_lptin_edge_set(uint8_t edge)
参数	uint8_t edge：上升沿/下降沿
返回	none

1.8.9 lptimer_pwm_channel_config

功能	lptimer_pwm_channel_config
描述	Lptimer pwm通道配置
函数定义	void lptimer_pwm_channel_config (uint8_t channel, uint8_t polar)
参数	uint8_t channel：PWM通道 uint8_t polar：PWM极性选择
返回	none

1.8.10 lptimer_pwm_init

功能	lptimer_pwm_init
描述	lptimer PWM输出初始化
函数定义	void lptimer_pwm_init(uint8_t clock_sel, uint8_t div)
参数	uint8_t clock_sel： 时钟源设置 0：RCL低速时钟 1：clk_1hz时钟 2：系统时钟 3：外部输入时钟 uint8_t div： 计数时钟分频
返回	none

1.8.11 lptimer_pwm_set

功能	lptimer_pwm_set
描述	Lptimer PWM 设置
函数定义	void lptimer_pwm_set(uint8_t changesel, uint32_t target, uint16_t duty)
参数	uint8_t changesel：PWM通道 uint32_t target： xms:PWM波周期为xms = 时钟频率/target(时钟源为RCL低速时钟下最大计时时长为65535ms) uint16_t duty：占空比（1~99，标识占空比为1%~99%）
返回	none

1.8.12 lptimer_set_time

功能	lptimer_set_s
描述	Lptimer设置时钟源
函数定义	void lptimer_set_time(uint16_t target)
参数	uint16_t target： xms：定时xms(时钟源为RCL低速时钟下最大计时时长为65535ms)
返回	none

1.8.13 lptimer_start

功能	lptimer_on
描述	lptimer 计数启动
函数定义	void lptimer_start(void)
参数	none
返回	none

1.8.14 lptimer_stop

功能	lptimer_off
描述	lptimer 计数停止
函数定义	void lptimer_stop(void)
参数	none
返回	none

1.8.15 lptimer_trigger_edge_set

功能	lptimer_trigger_edge_set
描述	lptimer_trigger边沿选择
函数定义	void lptimer_trigger_edge_set(uint8_t edge)
参数	uint8_t edge：上升沿/下降沿
返回	none

1.9 PWM接口

#include "pwm.h"

1.9.1 pwm0_init

功能	pwm0_init
描述	初始化PWM0

函数定义	void pwm0_init(uint16_t cycle, uint16_t duty, uint8_t level)
参数	uint16_t cycle：周期（周期时长计算 = 系统时钟/cycle） uint16_t duty：占空比 uint8_t level： HIGH：占空比期间输出高电平 LOW：占空比期间输出低电平
返回	none

1.9.2 pwm0_irq_init

功能	pwm0_irq_init
描述	初始化PWM0中断
函数定义	void pwm0_irq_init(em_pwm_irq mode,void (*pfunc)())
参数	em_pwm_irq mode：中断使能 void (*pfunc)()：中断回调函数
返回	none

1.9.3 pwm0_start

功能	pwm0_start
描述	启动PWM0输出
函数定义	void pwm0_start()
参数	none
返回	none

1.9.4 pwm0_stop

功能	pwm0_stop
描述	停止PWM输出
函数定义	void pwm0_stop()
参数	none
返回	none

1.9.5 pwm1_init

功能	pwm1_init
描述	pwm1初始化
函数定义	void pwm1_init(uint16_t cycle, uint16_t duty, uint8_t level)
参数	uint16_t cycle：周期（周期时长计算 = 系统时钟/cycle） uint16_t duty：占空比 uint8_t level： HIGH：占空比期间输出高电平 LOW：占空比期间输出低电平
返回	none

1.9.6 pwm1_irq_init

功能	pwm1_irq_init
描述	初始化PWM1中断
函数定义	void pwm1_irq_init(em_pwm_irq mode,void (*pfunc)())
参数	em_pwm_irq mode：中断使能 void (*pfunc)()：中断回调函数
返回	none

1.9.7 pwm1_start

功能	pwm1_start
描述	启动PWM1输出
函数定义	void pwm1_start()
参数	none
返回	none

1.9.8 pwm1_stop

功能	pwm1_stop
描述	停止PWM1输出
函数定义	void pwm1_stop()
参数	none
返回	none

1.9.9 pwm2_init

功能	pwm2_init
描述	pwm2_init
函数定义	void pwm2_init(uint16_t cycle, uint16_t duty, uint8_t level)
参数	uint16_t cycle：周期（周期时长计算 = 系统时钟/cycle） uint16_t duty：占空比 uint8_t level： HIGH：占空比期间输出高电平 LOW：占空比期间输出低电平
返回	none

1.9.10 pwm2_irq_init

功能	pwm2_irq_init
描述	pwm2中断初始化
函数定义	void pwm2_irq_init(em_pwm_irq mode,void (*pfunc)())
参数	em_pwm_irq mode：中断使能

	void (*pfunc)()：中断回调函数
返回	none

1.9.11 pwm2_start

功能	pwm2_start
描述	启动PWM2输出
函数定义	void pwm2_start()
参数	none
返回	none

1.9.12 pwm2_stop

功能	pwm2_stop
描述	停止PWM2输出
函数定义	void pwm2_stop()
参数	none
返回	none

1.9.13 PWM_IRQHandler

功能	PWM_IRQHandler
描述	PWM中断处理函数
函数定义	void PWM_IRQHandler(void)
参数	none
返回	none

1.10 SPI接口

#include "spi.h"

1.10.1 spi0_master_full_duplex_tranfer

功能	spi0_master_full_duplex_tranfer
描述	全双工传输
函数定义	uint8_t spi0_master_full_duplex_tranfer(uint8_t *send_buff, uint8_t *rec_buff, uint32_t length)
参数	uint8_t *send_buff：发送数据 uint8_t *rec_buff：接收数据 uint32_t length：数据长度
返回	none

1.10.2 spi_cs_disable

功能	spi_cs_disable
描述	CS片选拉高停止工作
函数定义	void spi_cs_disable(void)
参数	none
返回	none

1.10.3 spi_cs_enable

功能	spi_cs_enable
描述	CS片选拉低开始工作
函数定义	void spi_cs_enable(void)
参数	none
返回	none

1.10.4 spi_deinit

功能	spi_deinit
描述	关闭SPI模块
函数定义	void spi_deinit(void)
参数	none
返回	none

1.10.5 spi_irq_disable

功能	spi_irq_disable
描述	Spi中断disable
函数定义	void spi_irq_disable(void)
参数	none
返回	none

1.10.6 spi_irq_init

功能	spi_irq_enable
描述	spi中断使能
函数定义	void spi_irq_init(uint8_t irq_enable, uint8_t spi_irq_type,void (*pfunc)())
参数	uint8_t irq_enable：spi中断使能 uint8_t spi_irq_type：中断类型 void (*pfunc)()：中断回调函数
返回	none

1.10.7 SPI_IRQHandler

功能	SPI_IRQHandler
描述	SPI中断处理函数
函数定义	void SPI_IRQHandler(void)
参数	none
返回	none

1.10.8 spi_master_half_duplex_receive_bytes

功能	spi_master_half_duplex_receive_bytes
描述	半双工传输，接收数据
函数定义	uint8_t spi_master_half_duplex_receive_bytes (uint8_t * buff, uint32_t length)
参数	uint8_t *buff：接收数据 uint32_t length：接收数据长度
返回	none

1.10.9 spi_master_half_duplex_send_bytes

功能	spi_master_half_duplex_send_bytes
描述	半双工传输发送数据
函数定义	uint8_t spi_master_half_duplex_send_bytes(uint8_t *buff, uint32_t length)
参数	uint8_t *buff：发送数据 uint32_t length：发送数据长度
返回	none

1.10.10 spi_master_init

功能	spi_master_init
描述	SPI主机初始化
函数定义	void spi_master_init (uint8_t work_mode, uint8_t spi_baudrate_psc)
参数	uint8_t work_mode：选择SPI工作模式0/1/2/3 uint8_t spi_baudrate_psc：选择SPI波特率分频 SPI通信速率=fSYSCLK/波特率分频
返回	none

1.10.11 spi_receive_byte

功能	spi_receive_byte
描述	spi读取一字节
函数定义	uint8_t spi_receive_byte(void)
参数	none
返回	REG_SPI_RXBUF，读取一字节数据

1.10.12 spi_send_byte

功能	spi_send_byte
描述	spi写入一字节
函数定义	void spi_send_byte(uint8_t txdata)
参数	uint8_t txdata：写入一字节数据
返回	none

1.10.13 spi_slave_init

功能	spi_slave_init
描述	SPI从机初始化
函数定义	void spi_slave_init (uint8_t work_mode)
参数	uint8_t work_mode：选择SPI工作模式0/1/2/3
返回	none

1.10.14 spi_write_read_byte

功能	spi_write_read_byte
描述	spi读取一字节
函数定义	uint8_t spi_write_read_byte(uint8_t byte)
参数	uint8_t byte：写入的一个字节
返回	REG_SPI_RXBUF，读取到的一个字节

1.11 TIMER0&1接口

#include "timer.h"。

1.11.1 timer0_delay1ms

功能	timer0_delay1ms
描述	timer0延时ms函数
函数定义	void timer0_delay1ms(uint16_t ms)
参数	uint16_t ms：0~65535
返回	none

函数调用图:

1.11.2 timer0_init

功能	timer0_init
描述	timer0初始化
函数定义	void timer0_init(uint8_t clksrc,uint8_t psc,uint16_t load_value)
参数	uint8_t clksrc：timer0时钟源选择 uint8_t psc：timer0分频选择 uint16_t load_value：timer0定时器设定初值 fclk = fsrc/ psc 定时时间t=load_value/fclk (s)
返回	none

函数的调用关系图:

1.11.3 timer0_irq_config

功能	timer0_irq_config
描述	timer0中断使能配置
函数定义	void timer0_irq_config(uint8_t irq_enable,void (*pfunc)())
参数	uint8_t irq_enable：timer0中断使能控制 void (*pfunc)()：timer0中断回调函数
返回	none

1.11.4 TIMER0_IRQHandler

功能	TIMER0_IRQHandler
描述	TIMER0中断处理
函数定义	void TIMER0_IRQHandler(void)
参数	none
返回	none

1.11.5 timer0_start

功能	timer0_start
描述	timer0定时开始
函数定义	void timer0_start(void)
参数	none
返回	none

1.11.6 timer0_stop

功能	timer0_stop
描述	timer0定时停止
函数定义	void timer0_stop(void)
参数	none
返回	none

1.11.7 timer1_delay1ms

功能	timer1_delay1ms
描述	timer1延时ms函数
函数定义	void timer1_delay1ms(uint16_t ms)
参数	uint16_t ms：0~65535
返回	none

1.11.8 timer1_init

功能	timer1_init
描述	timer1初始化
函数定义	void timer1_init(uint8_t clksrc,uint8_t psc,uint16_t load_value)
参数	uint8_t clksrc：timer1时钟源选择 uint8_t psc：timer1分频选择 uint16_t load_value：timer1定时器设定初值 fclk = fsrc/ psc 定时时间t=load_value/fclk (s)
返回	none

函数的调用关系图:

1.11.9 timer1_irq_config

功能	timer1_irq_config
描述	timer1中断使能配置
函数定义	void timer1_irq_config(uint8_t irq_enable,void (*pfunc)())
参数	uint8_t irq_enable：timer1中断使能控制 void (*pfunc)()：timer1中断回调函数
返回	none

1.11.10 TIMER1_IRQHandler

功能	TIMER1_IRQHandler
描述	TIMER1中断处理
函数定义	void TIMER1_IRQHandler(void)
参数	none
返回	none

1.11.11 timer1_start

功能	timer1_start
描述	timer1定时开始
函数定义	void timer1_start(void)
参数	none
返回	none

1.11.12 timer1_stop

功能	timer1_stop
描述	timer1定时停止
函数定义	void timer1_stop(void)
参数	none
返回	none

1.11.13 timer_deinit

功能	timer_deinit
描述	关闭timer模块
函数定义	void timer_deinit(void)
参数	none
返回	none

1.12 UART0接口

#include "uart0.h"

#include "config.h"

1.12.1 putchar

功能	putchar
描述	打印函数重映射

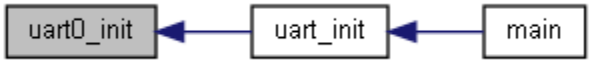
函数定义	char putchar(char c)
参数	char c：一个字符
返回	none

函数调用图:

1.12.2 uart0_init

功能	uart0_init
描述	uart0 initial for baud_rate
函数定义	void uart0_init(uint32_t baud_rate)
参数	uint32_t baud_rate: Series rate
返回	none

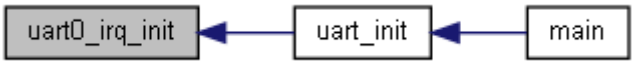
函数的调用关系图:



1.12.3 uart0_irq_init

功能	uart0_irq_init
描述	uart0 interrupt enable
函数定义	void uart0_irq_init(uint8_t irqstate,void (*pfunc_recv)())
参数	uint8_t irqstate：0 中断失能，1 中断使能 (*pfunc_recv)()：接收处理回调函数
返回	none

函数的调用关系图:



1.12.4 UART0_IRQHandler

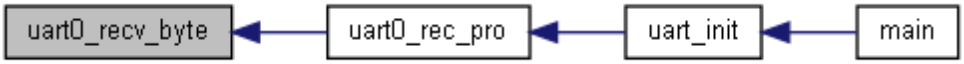
功能	UART0_IRQHandler
描述	uart interrupt handling
函数定义	void UART0_IRQHandler(void)
输出参数	none
返回	none

1.12.5 uart0_recv_byte

功能	uart0_recv_byte

描述	uart0 接收一个字节数据
函数定义	uint8_t uart0_recv_byte(void)
参数	none
返回	S0BUF：返回一个byte

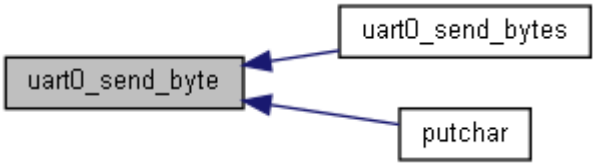
函数的调用关系图:



1.12.6 uart0_send_byte

功能	uart0_send_byte
描述	uart0 发送一个字节数据
函数定义	void uart0_send_byte(char c)
参数	char c：out byte
返回	none

函数的调用关系图:



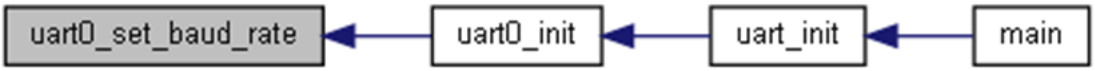
1.12.7 uart0_send_bytes

功能	uart0_send_bytes
描述	发送多个字节
函数定义	void uart0_send_bytes(uint8_t *buff, uint32_t length)
参数	uint8_t* buff：buffer数据 uint32_t length：buffer长度
返回	none

1.12.8 uart0_set_baud_rate

功能	uart0_set_baud_rate
描述	UART0波特率设置
函数定义	static void uart0_set_baud_rate (uint32_t clk_hz, uint32_t baud_rate)
参数	uint32_t clk_hz：cpu frequency uint32_t baud_rate：Series rate Baud Rate = SYSCK/(16*(1024-S0REL)) SYSCK:RC16M
输出	none
返回	none

函数的调用关系图:



1.13 UART1接口

#include "uart1.h"

#include "config.h"

1.13.1 putchar

功能	putchar
描述	打印函数重映射
函数定义	char putchar(char c)
参数	char c：一个字符
返回	none

函数调用图:

1.13.2 uart1_init

功能	uart1_init
描述	uart1 initial for baud_rate
函数定义	void uart1_init(uint32_t baud_rate)
参数	uint32_t baud_rate: Series rate
返回	none

函数调用图:

1.13.3 uart1_irq_init

功能	uart1_irq_init
描述	uart1 interrupt enable
函数定义	void uart1_irq_init(uint8_t irqstate,void (*pfunc_recv)())
参数	uint8_t irqstate：0 中断失能，1 中断使能 void (*pfunc_recv)()：接收处理回调函数
返回	none

1.13.4 UART1_IRQHandler

功能	UART1_IRQHandler
描述	uart interrupt handling
函数定义	void UART1_IRQHandler(void)
参数	none
返回	none

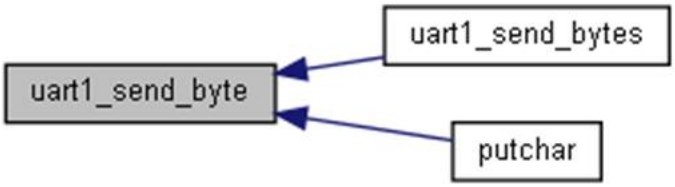
1.13.5 uart1_recv_byte

功能	uart1_recv_byte
描述	uart1 接收一个字节数据
函数定义	uint8_t uart1_recv_byte(void)
参数	none
返回	S1BUF：返回一个byte

1.13.6 uart1_send_byte

功能	uart1_send_byte
描述	uart1 发送一个字节数据
函数定义	void uart1_send_byte(char c)
参数	char c：out byte
返回	none

这是这个函数的调用关系图:



1.13.7 uart1_send_bytes

功能	uart1_send_bytes
描述	uart1 发送多个字节数据
函数定义	void uart1_send_bytes(uint8_t *buff, uint32_t length)
参数	uint8_t* buff：out buffer uint32_t length：buffer length
返回	none

函数调用图:

1.13.8 uart1_set_baud_rate

功能	uart1_set_baud_rate
描述	uart1波特率设置
函数定义	void uart1_set_baud_rate (uint32_t clk_hz, uint32_t baud_rate)
参数	uint32_t clk_hz：cpu frequency uint32_t baud_rate：Series rate Baud Rate = SYSCK/(16*(1024-S1REL)) SYSCK:RC16M
返回	none

函数的调用关系图:

1.14 UART2接口

#include "uart2.h"

#include "config.h"

1.14.1 putchar

功能	putchar
描述	打印函数重映射
函数定义	char putchar(char c)
参数	char c：一个字符
返回	none

函数调用图:

1.14.2 uart2_init

功能	uart2_init
描述	uart2 initial for baud_rate
函数定义	void uart2_init(uint32_t baud_rate)
参数	uint32_t baud_rate: Series rate
返回	none

函数调用图:

1.14.3 uart2_irq_init

功能	uart2_irq_init
描述	uart2 interrupt enable
函数定义	void uart2_irq_init(uint8_t irqstate,void (*pfunc_recv)())
参数	uint8_t irqstate: 0 中断失能，1 中断使能 void (*pfunc_recv)(): 接收处理回调函数
返回	none

1.14.4 UART2_IRQHandler

功能	UART2_IRQHandler
描述	uart interrupt handling
函数定义	void UART2_IRQHandler(void)
参数	none
返回	none

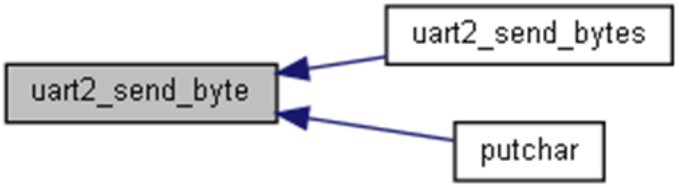
1.14.5 uart2_recv_byte

功能	uart2_recv_byte
描述	uart2 接收一个字节数据
函数定义	uint8_t uart2_recv_byte(void)
参数	none
返回	S1BUF：返回一个byte

1.14.6 uart2_send_byte

功能	uart2_send_byte
描述	uart2 发送一个字节数据
函数定义	void uart2_send_byte(char c)
参数	char c：out byte
返回	none

函数的调用关系图:



1.14.7 uart2_send_bytes

功能	uart2_send_bytes
描述	uart2 发送多个字节数据
函数定义	void uart2_send_bytes(uint8_t *buff, uint32_t length)
参数	uint8_t* buff：out buffer uint32_t length：buffer length
返回	none

函数调用图:

1.14.8 uart2_set_baud_rate

功能	uart2_set_baud_rate
描述	uart2波特率设置
函数定义	void uart2_set_baud_rate (uint32_t clk_hz, uint32_t baud_rate)
参数	uint32_t clk_hz：cpu frequency uint32_t baud_rate：Series rate Baud Rate = SYSCK/(16*(1024-S1REL)) SYSCK:RC16M
返回	none

函数的调用关系图:

1.15 UART3接口

#include "uart3.h"

#include "config.h"

1.15.1 putchar

功能	putchar
描述	打印函数重映射
函数定义	char putchar(char c)
参数	char c：一个字符
返回	none

函数调用图:

1.15.2 uart3_init

功能	uart3_init
描述	uart3 initial for baud_rate
函数定义	void uart3_init(uint32_t baud_rate)
参数	uint32_t baud_rate：Series rate
返回	none

函数调用图:

1.15.3 uart3_irq_init

功能	uart3_irq_init
描述	uart3 interrupt enable
函数定义	void uart3_irq_init(uint8_t irqstate,void (*pfunc_recv)())
参数	uint8_t irqstate： 0：中断失能 1：中断使能 void (*pfunc_recv)()：接收处理回调函数
返回	none

1.15.4 UART3_IRQHandler

功能	UART3_IRQHandler
描述	uart interrupt handling
函数定义	void UART3_IRQHandler(void)
参数	none
返回	none

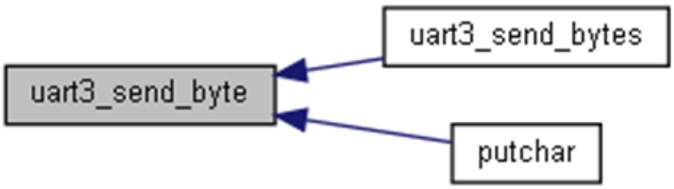
1.15.5 uart3_recv_byte

功能	uart3_recv_byte
描述	uart3 接收一个字节数据
函数定义	uint8_t uart3_recv_byte(void)
参数	none
返回	S1BUF：返回一个byte

1.15.6 uart3_send_byte

功能	uart3_send_byte
描述	uart3 发送一个字节数据
函数定义	void uart3_send_byte(char c)
参数	char c：out byte
返回	none

函数的调用关系图:



1.15.7 uart3_send_bytes

功能	uart3_send_bytes
描述	uart3 发送多个字节数据
函数定义	void uart3_send_bytes(uint8_t *buff, uint32_t length)
参数	uint8_t* buff：out buffer uint32_t length：buffer length
返回	none

函数调用图:

1.15.8 void uart3_set_baud_rate

功能	uart3_set_baud_rate
描述	uart3波特率设置
函数定义	void uart3_set_baud_rate (uint32_t clk_hz, uint32_t baud_rate)
参数	uint32_t clk_hz：cpu frequency uint32_t baud_rate：Series rate Baud Rate = SYSCK/(16*(1024-S1REL)) SYSCK:RC16M
返回	none

函数的调用关系图:

1.16 WDT接口

#include "wdt.h"

1.16.1 wdt_feed

功能	wdt_feed
描述	喂狗函数
函数定义	void wdt_feed(uint8_t arr)
参数	uint8_t arr：arr_1_ms，arr_4_ms...
返回	none

1.16.2 wdt_init

功能	wdt_init
描述	Wdt初始化
函数定义	void wdt_init(void)
参数	none
返回	none

1.16.3 wdt_load

功能	wdt_load
描述	设置溢出周期最小值
函数定义	void wdt_load(uint8_t arr)
参数	uint8_t arr：arr_1_ms，arr_4_ms...
返回	none