

LM402-DEMO 板 使用说明

易智联（西安）科技有限公司

地址：西安市高新区高新六路 52 号

电话：15829686916

◆ 说明

本文档描述了 LM402-DEMO 板的硬件接口及其 Ping-Pong 例程。

◆ 修订历史

日期	版本	作者	说明
2022-3-9	V0.1	王伟	初始版本

◆ 免责声明和版权公告

由于产品版本升级或其它原因，本手册内容会不定期进行更新。除非另有约定，本手册仅作为使用指导，本手册中的所有陈述、信息和建议不构成任何明示或暗示的担保。本文档不负任何责任，包括使用本文档内信息产生的侵犯任何专利权行为的责任。手册中信息修改，恕不另行通知。

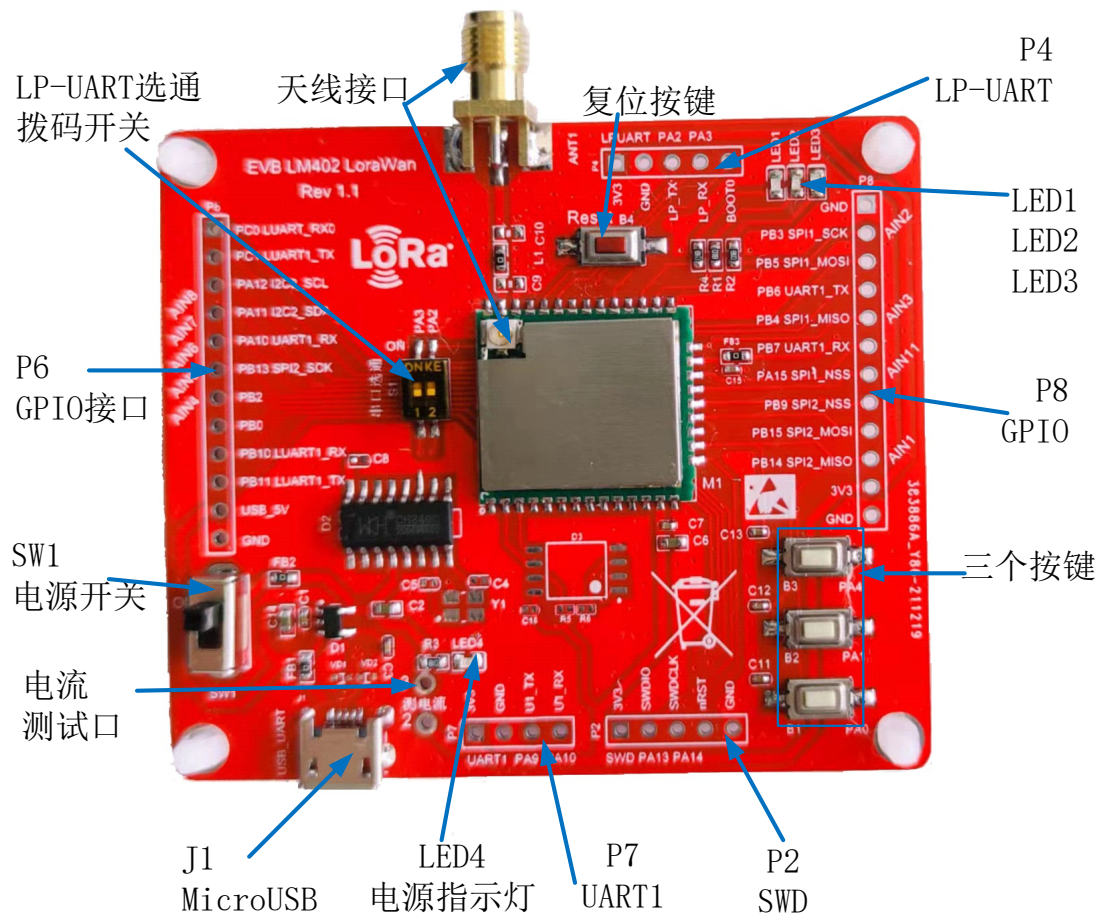
文中提到的所有商标名称、商标和注册商标均属其各自所有者的财产，特此声明。

本手册包含易智联（西安）科技有限公司的专利技术信息。除非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本手册内容的部分或全部，并不得以任何形式传播，犯规者可被追究支付赔偿金。对专利或者实用新型或者外观设计的版权所有，易智联（西安）科技保留一切权利。

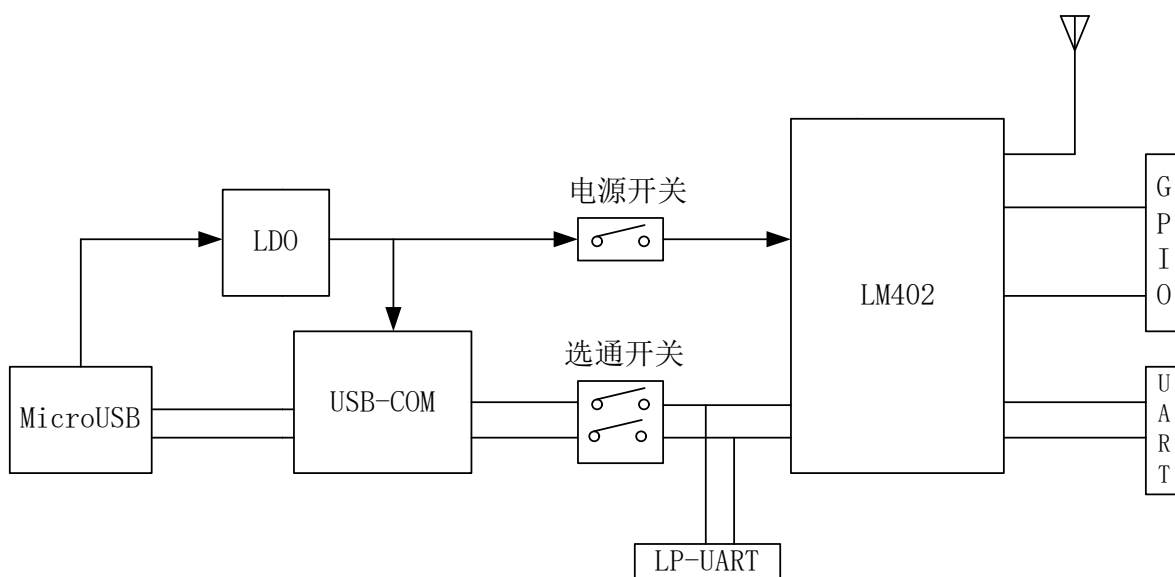
目录

一、 LM402-Demo 板简介.....	4
1. 功能说明.....	4
2. 接口定义.....	5
(1) 电源开关.....	5
(2) LP-UART 选通开关.....	5
(3) 扩展 GPIO	5
(4) MicroUSB 接口.....	6
(5) UART1 接口.....	6
(6) SWD 接口.....	6
(7) LP-UART4 接口.....	6
3. LM402 功耗测试	7
二、 Ping-Pong 例程说明	7
1. 简介.....	7
2. 业务流程图.....	8
(1) Main 主流程	8
(2) PingPong 流程.....	9
3. Ping-Pong 例程的参数配置	10
三、下载器连接方式.....	11
1. J-OB V2 下载器与 DEMO 板下载接口连接示意图.....	11
2. J-LINK V9 下载器与 DEMO 板下载接口连接示意图	11
四、附录.....	11
1. LoRa 相关函数.....	11
2. 定时服务.....	12
3. 任务调度.....	13
4. 低功耗.....	14

一、LM402-Demo 板简介



1. 功能说明



如上图所示，LM402-Demo 板为测试评估 LM402 模组提供了基本的外围硬件电路，并提供了板子 USB 转 COM 以及 LDO。USB 连接电脑既可以实现供电、打印输出、输入操作。

2. 接口定义

（1）电源开关

控制 LM402 模组的 3.3V 通断。

（2）LP-UART 选通开关

控制 LP-UART 接口是否接通 USB-COM。拨到 ON 侧，LP-UART 连接到 USB-COM 的转换芯片。上位机可以通过 USB-COM 与 LM401 模组进行交互。当需要通过 LP-UART 接口控制 LM401 模组或者做功耗测试时断开该拨码开关。

（3）扩展 GPIO

P6 信号定义

序号	名称	复用说明	备注
1	PC0	LPUART1_RXD	
2	PC1	LPUART1_TXD	
3	PA12	I2C2_SCL	
4	PA11	I2C2_SDA	
5	PA10	UART1_RX	
6	PB13	SPI2_SCK	
7	PB2		
8	PB0		
9	PB10	LPUART1_RXD	
10	PB11	LPUART1_TXD	
11	VBUS_5V	USB 接口的 5V	
12	GND	接地	

P8 信号定义

序号	名称	复用说明	备注
1	GND	接地	

2	PB3	SPI1_SCK	板上连接 LED3
3	PB5	SPI1_MOSI	板上连接 LED1
4	PB6	UART1_TX	
5	PB4	SPI1_MISO	板上连接 LED2
6	PB7	UART1_RX	
7	PA15	SPI1_NSS	
8	PB9	SPI2_NSS	板上连接按键 B1
9	PB15	SPI2_MOSI	板上连接按键 B2
10	PB14	SPI2_MISO	板上连接按键 B3
11	LORA_3V3	LM402 的 3.3V 供电	
12	GND		

(4) MicroUSB 接口

板上内置 USB 转 COM 芯片，对应型号为 CH340C。安装 CH340C 驱动后生成 COM 口。该 COM 口连接 LM402 的 PA2、PA3（可配置 LP-UART1 或者 UART2）。

(5) UART1 接口

P7 信号定义

序号	名称	说明	备注
1	3V3	连接 LM402 的 3.3V	
2	GND	接地	
3	U1_TX	连接 PB6	
4	U1_RX	连接 PB7	

(6) SWD 接口

P2 信号定义

序号	名称	说明	备注
1	Lora3V3	连接 LM402 的 3.3V	
2	SWDIO	SWD 数据	
3	SWDCLK	SWD 时钟	
4	nRST	MCU 复位	例程中有休眠需连接下载器
5	GND	接地	

(7) LP-UART1 接口

P4 信号定义

序号	名称	说明	备注
1	3V3	连接 LM402 的 3.3V	
2	GND	接地	
3	LP_TX	连接 PA2	
4	LP_RX	连接 PA3	
5	BOOT0	连接 BOOT0	

3. LM402 功耗测试

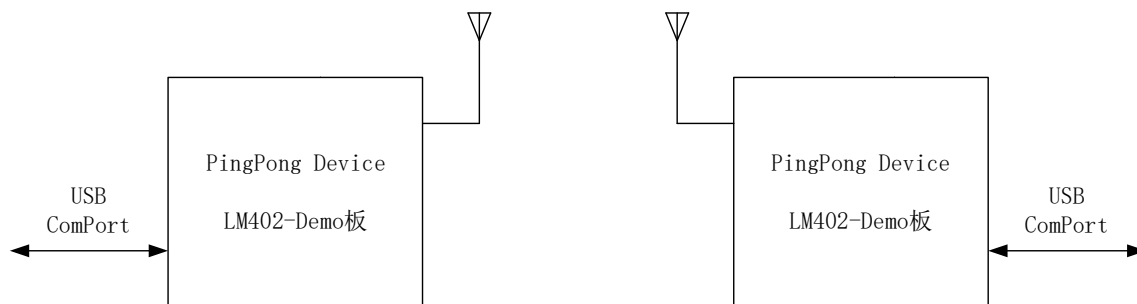
测试 LM402 模组功耗时需断开 LP-UART 选通开关和电源开关，通过 LP-UART 或者 UART1 插座上的 3V3 与 GND 对 LM402 供电，排除 USB-COM 部分的干扰。

二、Ping-Pong 例程说明

1. 简介

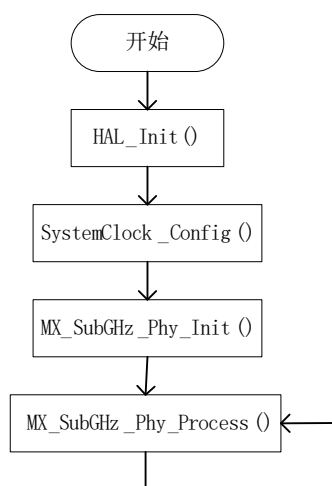
Ping-Pong 例程是两块 LM402-DEMO 板之间的一个简单 Lora 收发例程。默认情况下，LM402-DEMO 评估板作为主设备，发送一个“Ping”消息然后等待回复。第一个收到“Ping”信息的设备将变为从设备，并以“Pong”信息回复主设备。当主设备收到“Pong”信息之后继续发送“Ping”信息。这样就建立了一个持续的 Ping-Pong 收发过程。

正常情况下两个评估板建立 Ping-Pong 收发关系，只有一个 LED 闪烁。接收到 Ping 消息时，LED1 闪烁，接收到 Pong 消息时，LED2 闪烁。当两个板完全同步时，即两个主板的 RX 窗口同步(TX 窗口也是)时，无法建立 Ping-Pong 收发关系（两个 LED 长时间同时闪烁）。这时只需要重新启动其中的一块板即可。



2. 业务流程图

(1) Main 主流程



参见 main.c

MX_SubGHz_Phy_Init()调用 SystemApp_Init()初始化定时器，复位外部端口，准备调测输出、配置低功耗等完成基本工作环境。调用 SubghzApp_Init()完成 Lora 的 PingPong 配置，注册 PingPong_Process 任务。

MX_SubGHz_Phy_Process()调用 UTIL_SEQ_Run(UTIL_SEQ_DEFAULT)完成基本的任务调度。

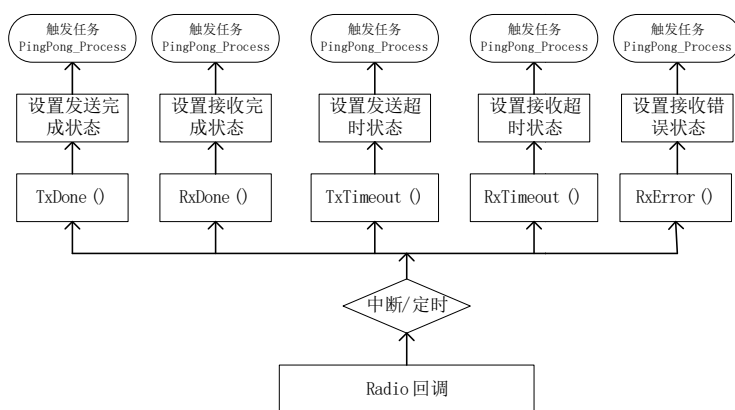
例程中提供了一个类似于 RTOS 的，事件驱动的调度机制。具体内容参考附录及 stm32_seq.c。

(2) PingPong 流程

整个 PingPong 的执行由 Lora 射频回调事件驱动，实现于 subghz_phy_app.c。主要有初始化 SubghzApp_Init()，PingPong 处理 PingPong_Process() 以及 Lora 射频回调函数 TxDone()、RxDone()、TxTimeout()、RxTimeout()、RxError() 组成。

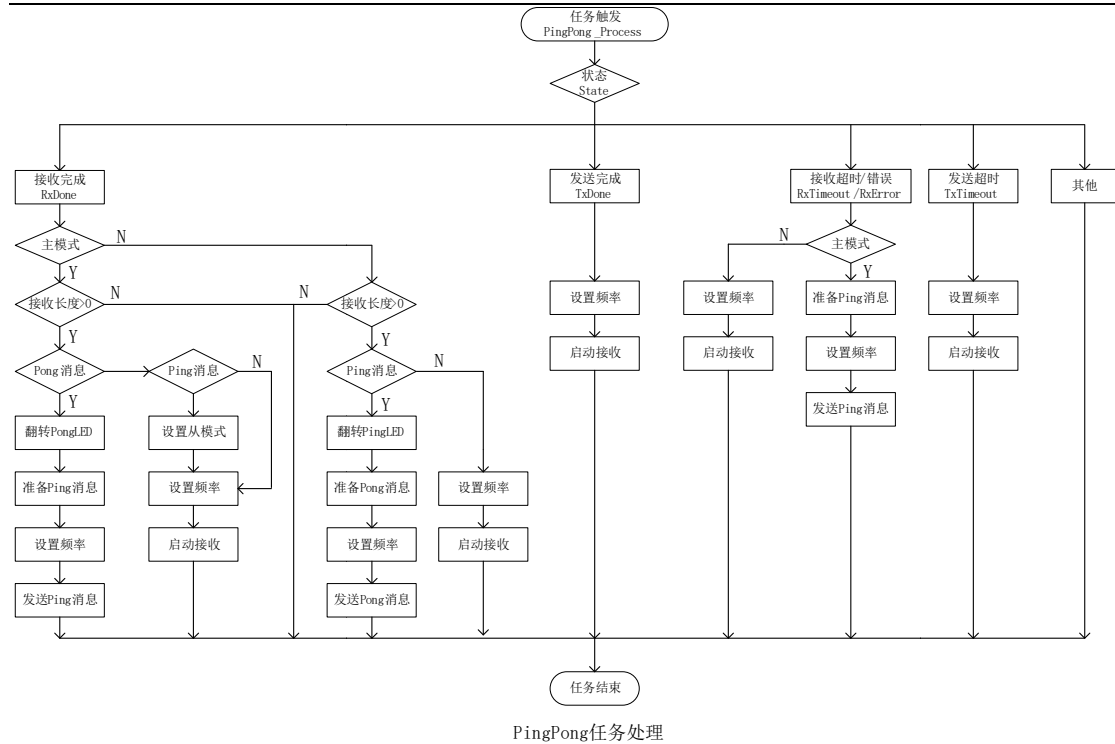


PingPong任务的初始化



Lora射频的回调函数

PingPong 的处理流程如下图所示



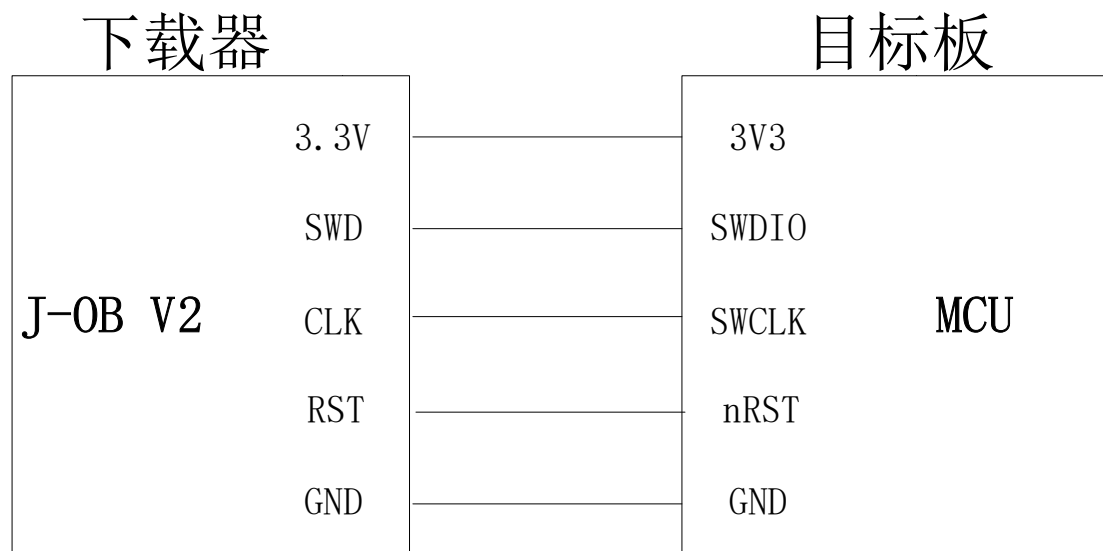
3. Ping-Pong 例程的参数配置

Ping-Pong 例程的参数配置如下表所示

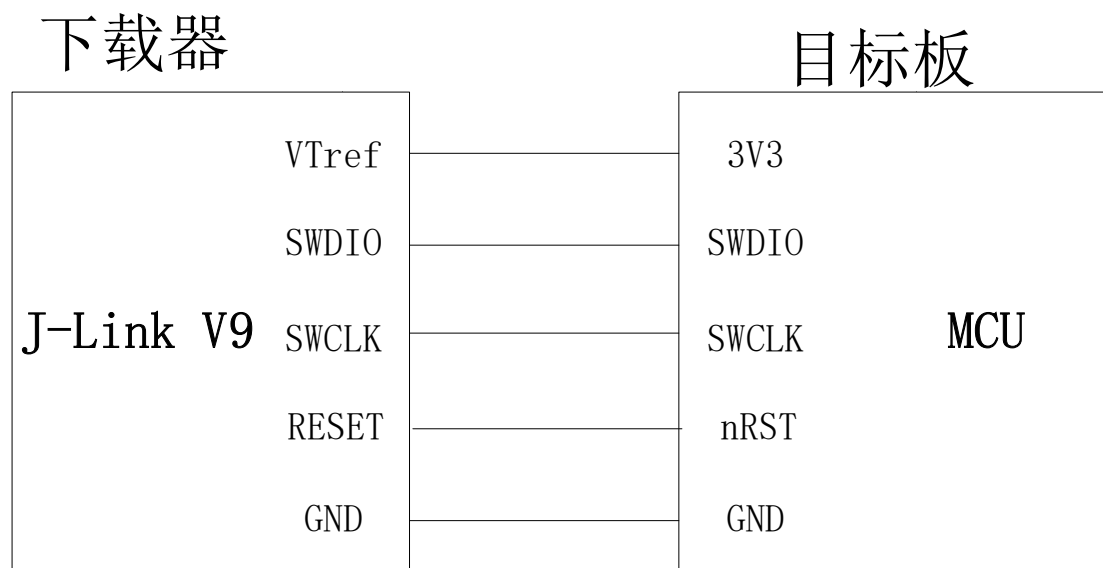
分类	参数	描述	位置
Tx/Rx 配置	BUFFER_SIZE	数据 Buf 大小	subghz_phy_app.h
	RF_FREQUENCY	发送频率	
	RX_TIMEOUT_VALUE	接收超时时间	
	TX_TIMEOUT_VALUE	发送超时时间	
Lora SetTXConfig 参数	TX_OUTPUT_POWER	发送功率	
	LORA_BANDWIDTH	信号带宽	
	LORA_SPREADING_FACTOR	扩频因子	
	LORA_CODINGRATE	编码率	
	LORA_PREAMBLE_LENGTH	前导长度	
	LORA_SYMBOL_TIMEOUT	超时符号数	
	LORA_FIX_LENGTH_PAYLOAD_ON	负载长度	
	LORA_IQ_INVERSION_ON	IQ 采样反转	
Low power	LOW_POWER_DISABLE	关闭低功耗	sys_conf.h
Trace	VERBOSE_LEVEL	调试等级	
	APP_LOG_ENABLED	LOG 使能	

三、下载器连接方式

1. J-OB V2 下载器与 DEMO 板下载接口连接示意图



2. J-LINK V9 下载器与 DEMO 板下载接口连接示意图



四、附录

1. LoRa 相关函数

Lora 相关操作 Semtech 已经进行了相关封装，详见 `radio.c`

函数名	功能
-----	----

RadioInit	初始化 Lora
RadioGetStatus	返回当前的 Lora 状态
RadioSetModem	配置 Lora 模式
RadioSetChannel	设置频率
RadioIsChannelFree	检查该频段是否空闲
RadioSetRxConfig	设置接收参数
RadioSetTxConfig	设置发送参数
RadioCheckRfFrequeunc	检查该频率硬件是否支持
RadioSend	发送缓冲区
RadioSleepLora	休眠
RadioStandby	将 Lora 设置为待机模式
RadioRx	将 Lora 设置为接收等待模式
RadioStartCad	设置为 CAD 模式
RadioSetTxContinuousWave	设置 Lora 为连续波传输模式
RadioRssi	读取当前 RSSI 值
RadioSetMaxPayloadLength	设置最大负载长度

2. 定时服务

基于 RTC 的定时服务在低功耗模式下依然有效。

相关接口函数如下表所示，具体实现参见 `stm32_timer.c`

函数	简介
<code>UTIL_TIMER_Status_t UTIL_TIMER_Init(void)</code>	Initializes the timer server.
<code>UTIL_TIMER_Status_t UTIL_TIMER_Create (UTIL_TIMER_Object_t *TimerObject, uint32_t PeriodValue, UTIL_TIMER_Mode_t Mode, void (*Callback) (void *), void *Argument)</code>	Creates the timer object and associates a callback function when timer elapses.
<code>UTIL_TIMER_Status_t</code>	Updates the period and starts

UTIL_TIMER_SetPeriod(UTIL_TIMER_Object_t *TimerObject, uint32_t NewPeriodValue)	the timer with a timeout value (milliseconds).
UTIL_TIMER_Status_tUTIL_TIMER_Start (UTIL_TIMER_Object_t *TimerObject)	Starts and adds the timer object to the list of timer events.
UTIL_TIMER_Status_tUTIL_TIMER_Stop (UTIL_TIMER_Object_t *TimerObject)	Stops and removes the timer object from the list of timer events.

3. 任务调度

The sequencer provides a robust and easy framework to execute tasks in the background and enters low-power mode when there is no more activity. The sequencer implements a mechanism to prevent race conditions.

In addition, the sequencer provides an event feature allowing any function to wait for an event (where particular event is set by interrupt) and MIPS and power to be easily saved in any application that implements “run to completion” command.

The utilities_conf.h file located in the project sub-folder is used to configure the task and event IDs. The ones already listed must not be removed.

The sequencer is not an OS. Any task is run to completion and can not switch to another task like a RTOS would do on RTOS tick. Moreover, one single-memory stack is used. The sequencer is an advanced ‘while loop’ centralizing task and event bitmap flags.

The sequencer provides the following features:

- Advanced and packaged while loop system
- Support up to 32 tasks and 32 events
- Task registration and execution
- Waiting event and set event
- Task priority setting

To use the sequencer, the application must perform the following:

- Set the number of maximum of supported functions, by defining a value for

UTIL_SEQ_CONF_TASK_NBR.

- Register a function to be supported by the sequencer with UTIL_SEQ_RegTask().
- Start the sequencer by calling UTIL_SEQ_Run() to run a background while loop.
- Call UTIL_SEQ_SetTask()

相关接口函数如下表所示，具体实现参见 stm32_seq.c

函数	简介
void UTIL_SEQ_Idle(void)	Called (in critical section - PRIMASK) when there is nothing to execute.
void UTIL_SEQ_Run(UTIL_SEQ_bm_t mask_bm)	Requests the sequencer to execute functions that are pending and enabled in the mask mask_bm.
void UTIL_SEQ_RegTask(UTIL_SEQ_bm_t task_id_bm, uint32_t flags, void (*task)(void))	Registers a function (task) associated with a signal (task_id_bm) in the sequencer. The task_id_bm must have a single bit set.
void UTIL_SEQ_SetTask(UTIL_SEQ_bm_t taskId_bm , uint32_t task_Prio)	Requests the function associated with thetask_id_bm to be executed. The task_prio is evaluated by the sequencer only when a function has finished. If several functions are pending at any one time, the one with the highest priority (0) is executed.

4. 低功耗

每个应用模块实现自己的低功耗设计，统一由任务调度的在空闲状态下执行。例如,控制台使用 DMA 进行数据输出,这时的系统只能进入 Sleep 模式，不能进入 Stop 模式。Stop 模式下 DMA 时钟关闭。

相关接口函数

函数	简介
----	----

void UTIL_LPM_EnterLowPower(void)	Enters the selected low-power mode. Called by idle state of the system
void LPM_SetStopMode(LPM_Id_t id, LPM_SetMode_t mode)	Sets Stop mode. id defines the process mode requested: LPM_Enable or LPM_Disable. (1)
void LPM_SetOffMode(LPM_Id_t id, LPM_SetMode_t mode)	Sets Stop mode. id defines the process mode requested: LPM_Enable or LPM_Disable.
UTIL_LPM_Mode_t UTIL_LPM_GetMode(void)	Returns the selected low-power mode.

1. LPM_Id_t are bitmaps. Their shift values are defined in utilities_def.h of project sub-folder.

默认的低功耗模式是 Off mode，如果需要 Standby 或者 Shutdown mode（通过 void PWR_EnterOffMode (void)进行修改）

- If Stop mode is disabled and low-power is entered, Sleep mode is selected.
- If Stop mode is not disabled, Off mode is disabled and low-power is entered, the LPStop mode is selected.
- If Stop mode is not disabled, Off mode is not disabled and low-power is entered, low-power Standby or Shutdown mode is selected.

低功耗真值表

Low-power idle mode	LPM_SetStopMode	LPM_OffStopMode
LPSleep	UTIL_LPM_DISABLE	Enable or disable
LPStop	UTIL_LPM_ENABLE	UTIL_LPM_DISABLE
LP Off	UTIL_LPM_ENABLE	