密级状态:绝密(　　)　秘密()　内部资料(√)　　公开()

文档编号:(芯片型号) –ASR6501/ASR6502(英文、数字)

ASR6501/ASR6502 芯片 API 使用手册

文件状态: [√] 正在修改 [　] 正式发布	当前版本:	V0.1
	作者:	Ao Ye
	启动日期:	2019-04-04
	审核:	
	完成日期:	2019-04-11

ASR Confidential

翱捷科技（上海）有限公司

ASR Microelectronics Co., Ltd

(版本所有,翻版必究)

版本历史

版本号	修改日期	作 者	修 改 说 明
V0.1	2019.04.11	Ye Ao	Initial version

ASR Confidential

适用芯片型号

序号	芯片型号	简介
1	ASR6501	150MHz to 960MHz，6mm x 6mm x 0.9mm
2	ASR6502	150MHz to 960MHz，7mm x 7mm x 0.9mm

ASR Confidential

目录

ASR Confidential

ASR Confidential

1. 概述

本文旨在帮助基于使用 ASR6501 和 ASR6502 芯片作为 MCU 进行开发的用户，让其能快速使用各种外设接口（GPIO、I2C、UART、SPI、ADC）、LoRaMac 相关接口和 AliOS 操作系统相关接口。

对于使用其它 MCU 来控制本芯片进行 LORA 相关业务处理时，可以使用 AT 命令进行控制，参考文档《ASR650X-AT-Commands-Introduction》。

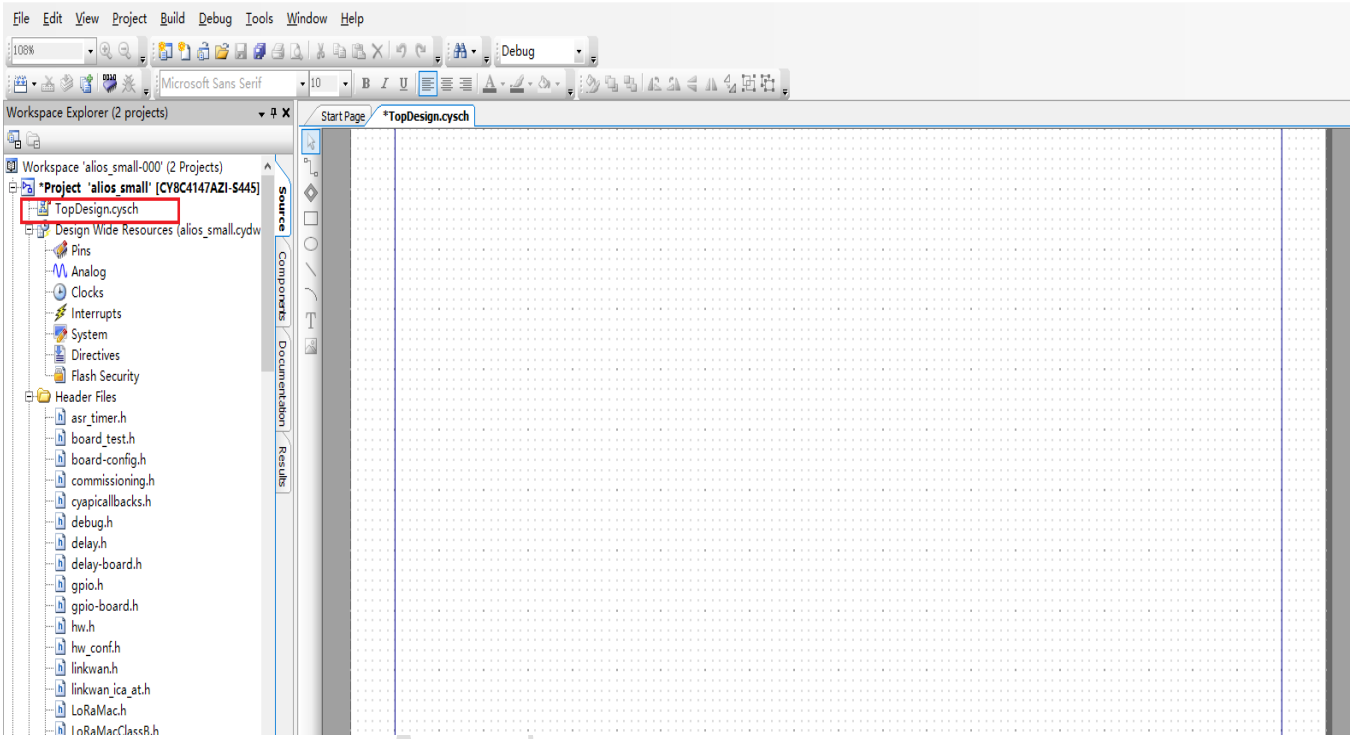
ASR Confidential

2. 外设接口

2.1. GPIO

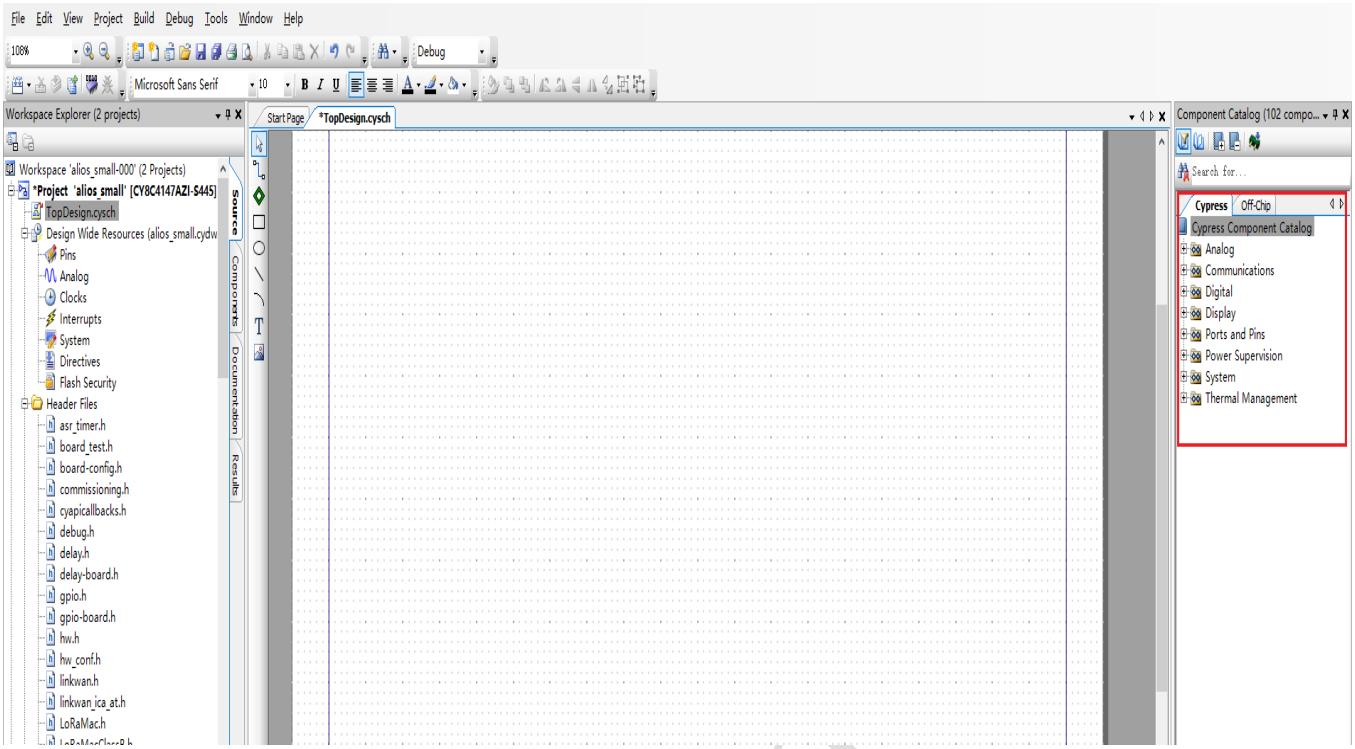
2.1.1. 图形配置

首先进入 TopDesign 配置界面，如下图红色标红部分：

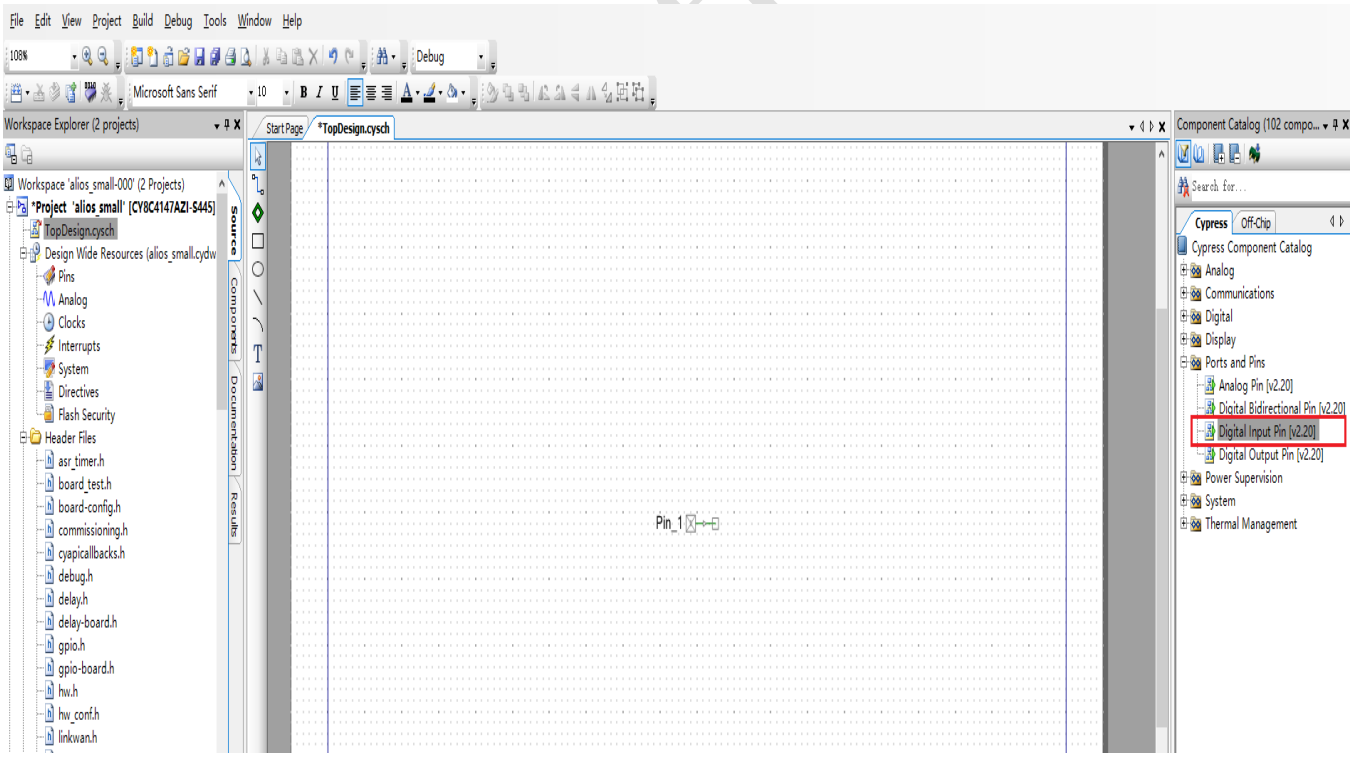


双击上图中红色部分 TopDesign.cysch，界面最右侧出现如下图所示配置：

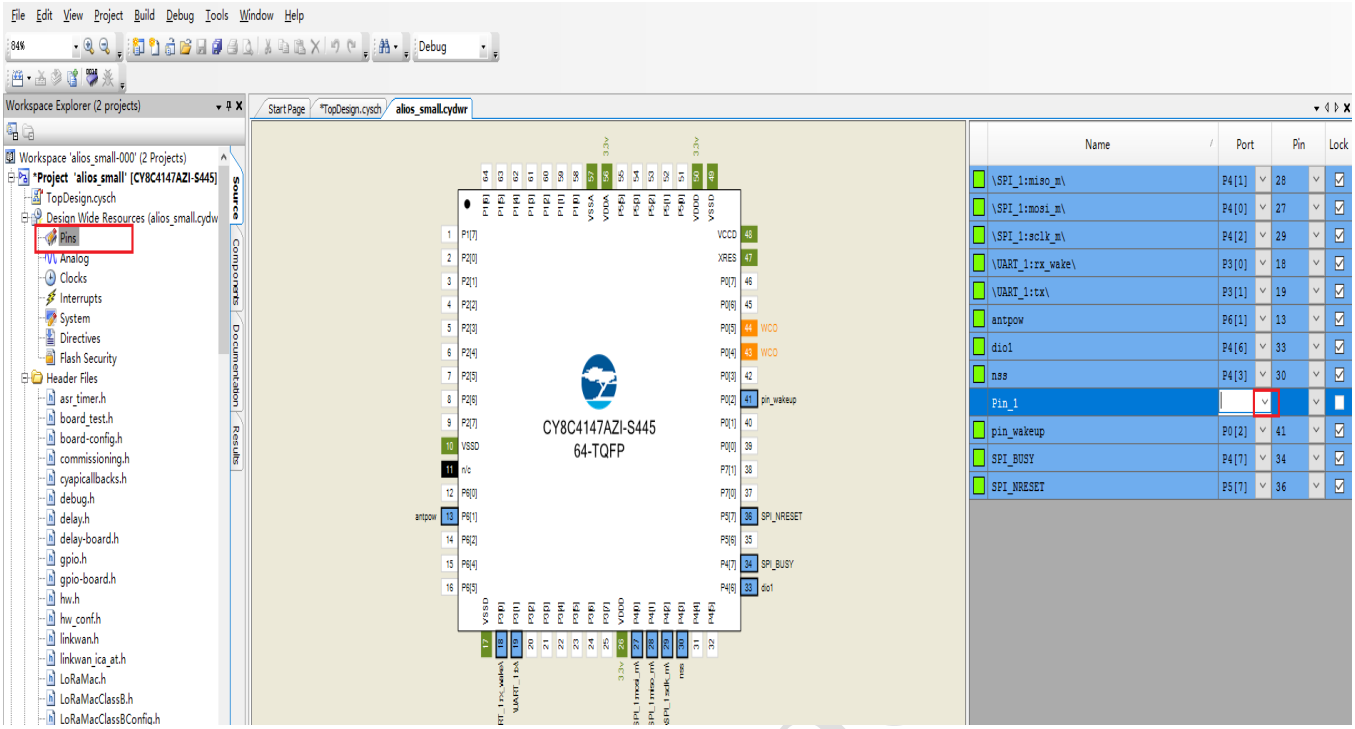
ASR C



以 GPIO 类型为 Digital Input Pin 为例，点击下图中红色部分，按下左键后拖拽到软件界面中间部分。然后双击图标可以配置 Drive mode 和 Interrupt，以及可以更改名称。



另外在 Pins 配置界面中可以配置 GPIO 的 PIN 脚，如下图中双击最左侧的红色部分，会出现界面最右侧配置部分。点击右侧红色标记的倒三角可以选择 PIN 脚。



2.1.2. GPIO 接口函数

2.1.2.1. GPIO 驱动模式

功能分类	内容	描述
GPIO 驱动模式	XXX _SetDriveMode	
函数原型	void XXX _SetDriveMode(uint8 mode)	如果通过 PSOC Creator 配置就不需要调用此函数配置。XXX 为对应 PIN 脚名称
参数说明	mode：GPIO 模式	
返回值	无	

ASR Confidential

2.1.2.2. GPIO 电平设置

功能分类	内容	描述
GPIO 电平设置	XXX_Write	
函数原型	void XXX _Write(uint8 value)	XXX 为对应 PIN 脚名称
参数说明	value：要写入的电平值	
返回值	无	

2.1.2.3. GPIO 电平读取

功能分类	内容	描述
GPIO 电平读取	XXX _Read	
函数原型	uint8 XXX _Read(void)	XXX 为对应 PIN 脚名称
参数说明	无	
返回值	PIN 脚的电平值	

2.1.2.4. GPIO 中断设置

功能分类	内容	描述
GPIO 中断设置	XXX _SetInterruptMode	
函数原型	void XXX_SetInterruptMode(uint16 position, uint16 mode)	如果通过 PSOC Creator 配置就不需要调用此函数配置。XXX 为对应 PIN 脚名称
参数说明	position：PIN 脚中断所在的位置，如 PIN0.2 脚，则此参数为 CYREG_GPIO_PRT0_INTR_CFG，PIN1.2 脚则为 CYREG_GPIO_PRT1_INTR_CFG，依此类推 mode：中断触发模式	
返回值	无	

ASR Confidential

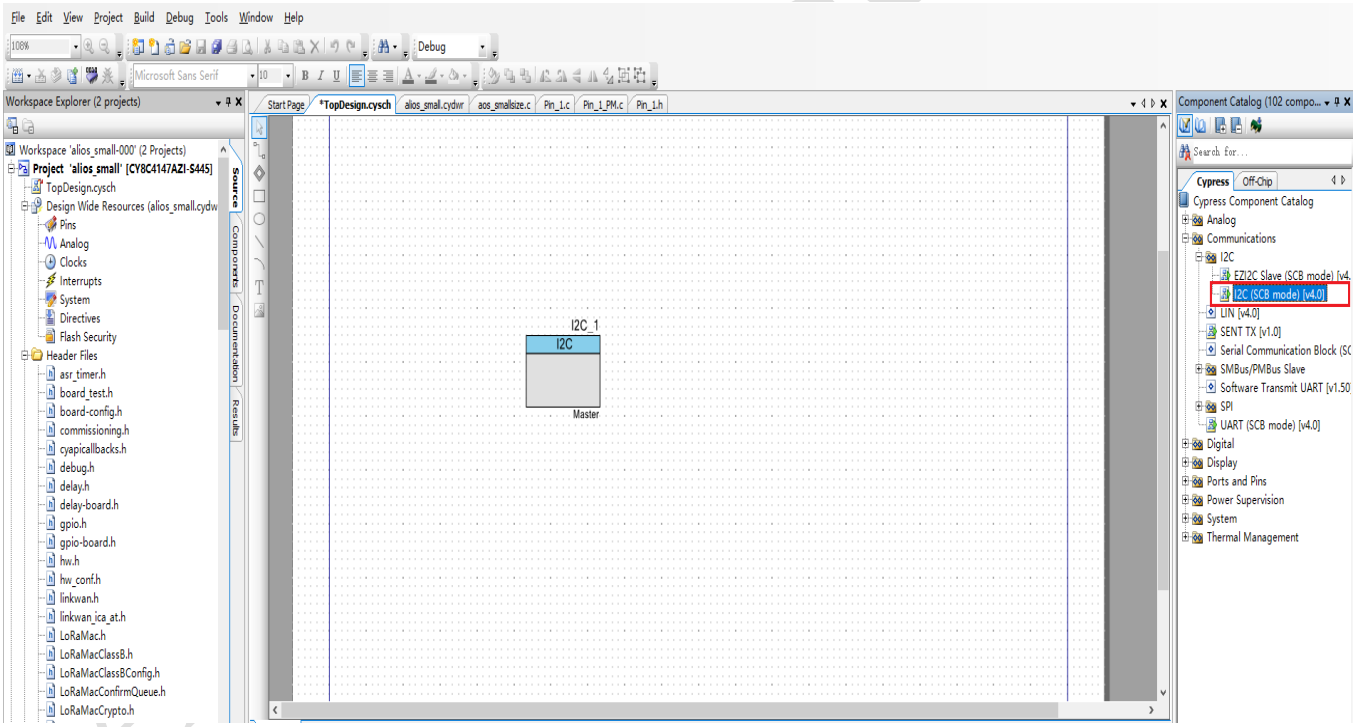
2.1.2.5. GPIO 中断清除

功能分类	内容	描述
GPIO 中断清除	XXX _ ClearInterrupt	
函数原型	uint8 XXX_ClearInterrupt(void)	XXX 为对应 PIN 脚名称
参数说明	无	
返回值	中断状态	

2.2. I2C

2.2.1. 图形配置

首先进入 TopDesign 配置界面，进入 TopDesign 的方式如 2.1.1 所述。点击下图中红色部分，按下左键后拖拽到软件界面中间部分。然后双击图标可以配置 Slave/Mater Mode、速率，以及可以更改名称。配置 PIN 脚如 2.1.1 所述。



2.2.2. I2C 接口函数

2.2.2.1.I2C 操作启动接口

功能分类	内容	描述
I2C 操作启动接口	I2C_1_I2CMasterSendStart	
函数原型	uint32 I2C_1_I2CMasterSendStart(uint32 slaveAddress, uint32 bitRnW, uint32 timeoutMs)	
参数说明	slaveAddress：从机地址 bitRnW：数据方向，读或写 timeoutMs：阻塞时间	
返回值	Error status	

2.2.2.2.I2C 写接口

功能分类	内容	描述
I2C 写接口	I2C_1_I2CMasterWriteByte	
函数原型	uint32 I2C_1_I2CMasterWriteByte(uint32 wrByte, uint32 timeoutMs)	
参数说明	wrByte：从机地址 timeoutMs：阻塞时间	
返回值	Error status	

2.2.2.3.I2C 读接口

功能分类	内容	描述
I2C 读接口	I2C_1_I2CMasterReadByte	
函数原型	uint32 I2C_1_I2CMasterReadByte(uint32 ackNack, uint8 *rdByte, uint32 timeoutMs)	
参数说明	ackNack：应答类型 rdByte：接收的数据 buffer timeoutMs：阻塞时间	
返回值	Error status	

2.2.2.4.I2C 停止接口

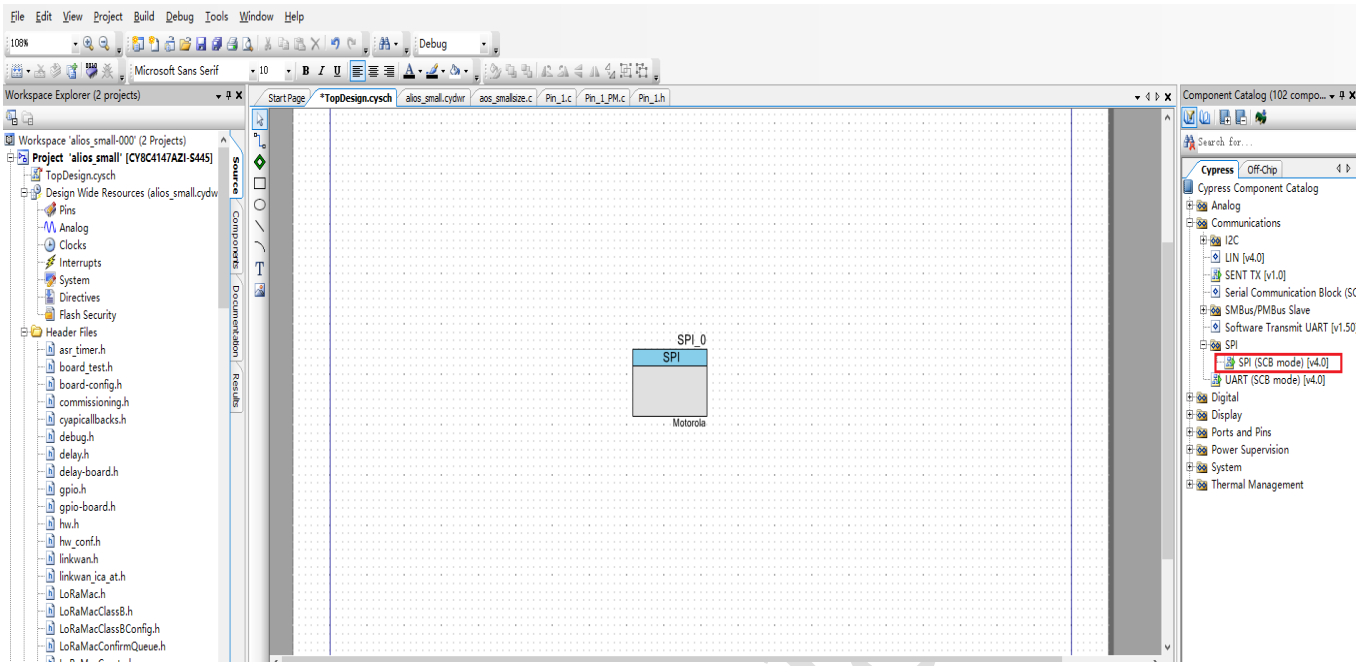
功能分类	内容	描述
I2C 停止接口	I2C_1_I2CMasterSendStop	
函数原型	uint32 I2C_1_I2CMasterSendStop(uint32 timeoutMs)	
参数说明	timeoutMs：阻塞时间	
返回值	Error status	

2.3. SPI

2.3.1. 图形配置

首先进入 TopDesign 配置界面，进入 TopDesign 的方式如 2.1.1 所述。点击下图中红色部分，按下左键后拖拽到软件界面中间部分。然后双击图标可以配置 Slave/Master Mode、速率、TX&RX 的 buffer 大小以及可

以更改名称。配置 PIN 脚如 2.1.1 所述。



2.3.2. SPI 接口函数

2.3.2.1.SPI 启动接口

功能分类	内容	描述
SPI 启动接口	XXX_Start	
函数原型	void XXX_Start(void)	XXX 为对应的 SPI 名称
参数说明	无	
返回值	无	

2.3.2.2.SPI 停止启用接口

功能分类	内容	描述
SPI 停止启用接口	XXX_ Stop	
函数原型	void XXX_Stop(void)	XXX 为对应的 SPI 名称
参数说明	无	
返回值	无	

2.3.2.3.SPI 发送接口

功能分类	内容	描述
SPI 发送接口	XXX_SpiUartWriteTxData	
函数原型	void XXX_SpiUartWriteTxData(uint32 txData)	XXX 为对应的 SPI 名称
参数说明	txData：需要发送的数据	
返回值	无	

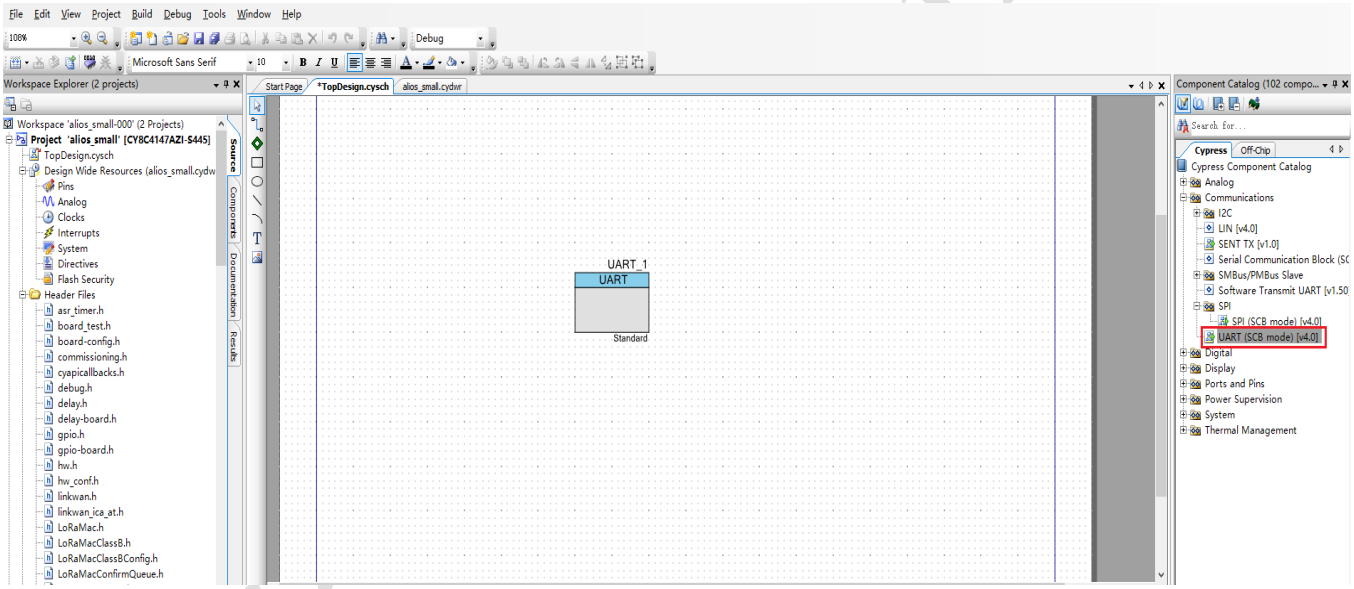
2.3.2.4. SPI 接收接口

功能分类	内容	描述
SPI 接收接口	XXX_ SpiUartReadRxData	
函数原型	uint32 XXX_ SpiUartReadRxData (void)	XXX 为对应的 SPI 名称
参数说明	无	
返回值	接收到的数据	

2.4. UART

2.4.1. 图形配置

首先进入 TopDesign 配置界面，进入 TopDesign 的方式如 2.1.1 所述。点击下图中红色部分，按下左键后拖拽到软件界面中间部分。然后双击图标可以配置 Slave/Mater Mode、速率、TX&RX 的 buffer 大小以及可以更改名称。配置 PIN 脚如 2.1.1 所述。



2.4.2. UART 接口函数

2.4.2.1. UART 启动接口

功能分类	内容	描述
UART 启动接口	XXX_Start	
函数原型	void XXX_Start(void)	XXX 为对应的 UART 名称
参数说明	无	
返回值	无	

2.4.2.2. UART 停止接口

功能分类	内容	描述
UART 启动接口	XXX_Stop	
函数原型	void XXX_Stop(void)	XXX 为对应的 UART 名称
参数说明	无	
返回值	无	

2.4.2.3. UART 接收

功能分类	内容	描述
UART 接收	XXX_UartGetChar	
函数原型	uint32 XXX_UartGetChar(void)	XXX 为对应的 UART 名称
参数说明	无	
返回值	接收的字符	

2.4.2.4. UART 发送

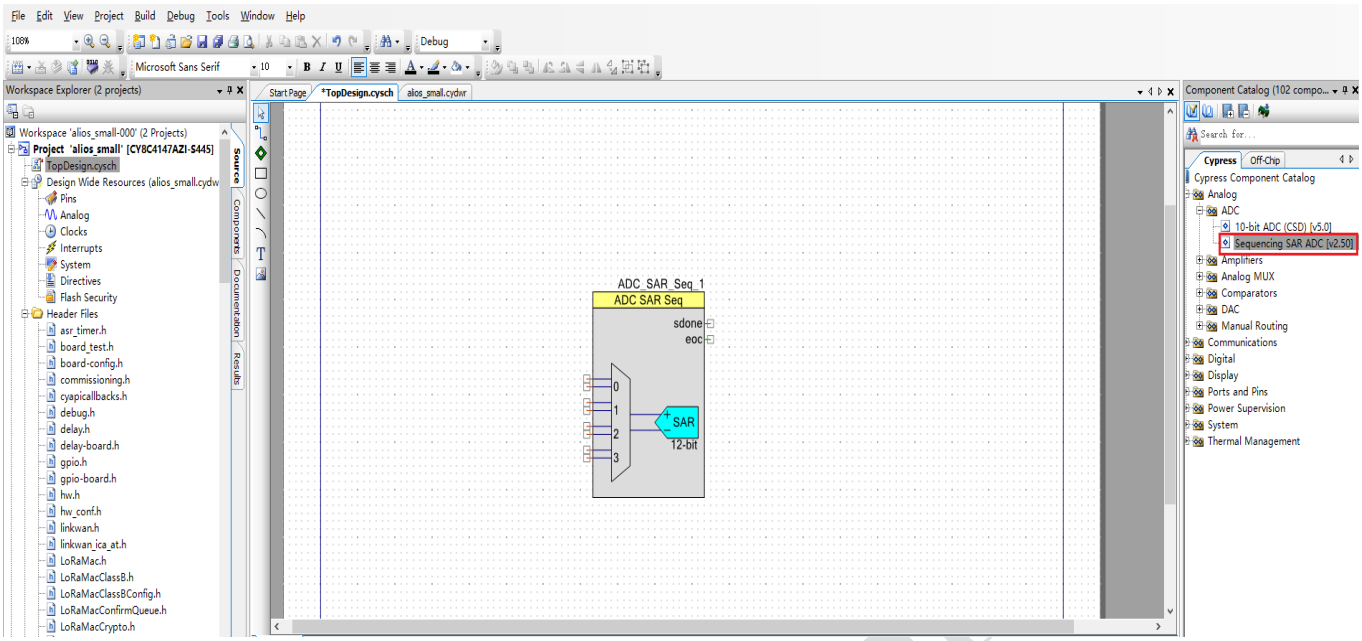
功能分类	内容	描述
UART 发送	XXX_ SpiUartPutArray	
函数原型	void XXX _SpiUartPutArray(const uint8 wrBuf[], uint32 count)	XXX 为对应的 UART 名称
参数说明	wrBuf：要发送的数据 count：要发送的数据长度	
返回值	无	

2.5. ADC

2.5.1. 图形配置

首先进入 TopDesign 配置界面，进入 TopDesign 的方式如 2.1.1 所述。点击下图中红色部分，按下左键后拖拽到软件界面中间部分。然后双击图标可以配置 channel 数目、sample rate、vref、单端或差分输入模式以及可以更改名称。配置 PIN 脚如 2.1.1 所述。

ASR Confidential



2.5.2. ADC 接口函数

2.5.2.1. ADC 启动接口

功能分类	内容	描述
ADC 启动接口	XXX_Start	
函数原型	void XXX_Start(void)	XXX 为对应的 ADC 名称
参数说明	无	
返回值	无	

2.5.2.2. ADC 停止接口

功能分类	内容	描述
ADC 停止接口	XXX_ Stop	
函数原型	void XXX_Stop(void)	XXX 为对应的 ADC 名称
参数说明	无	
返回值	无	

2.5.2.3. ADC 转换接口

功能分类	内容	描述
ADC 转换接口	XXX_ GetResult16	
函数原型	int16 XXX _GetResult16(uint32 chan)	XXX 为对应的 ADC 名称
参数说明	Chan：channel 编号	
返回值	ADC 转换数据，即 SAR DATA 寄存器中的数据	

2.5.2.4. ADC 获取电压接口

功能分类	内容	描述
ADC 获取电压接口	XXX_ CountsTo_Volts	
函数原型	float32 XXX _CountsTo_Volts(uint32 chan, int16 adcCounts)	XXX 为对应的 ADC 名称
参数说明	Chan：channel 编号 adcCounts：接口 XXX _GetResult16 的返回值	
返回值	电压值	

2.6. Flash

2.6.1. **Flash 读**

Flash 的读，直接对地址指针进行读取即可，例如：

uint8_t data = *(uint8_t *)addr;

2.6.2. **Flash 写**

功能分类	内容	描述
Flash 写	CySysFlashWriteRow	
函数原型	uint32 CySysFlashWriteRow(uint32 rowNum, const uint8 rowData[])	
参数说明	rowNum：flash 行号，每行 256 个字节 rowData：待写入数据	
返回值	写入状态	

ASR Confidential

3. LoRaMac 接口

3.1. 初始化接口

功能分类	内容	描述
初始化 LoRaMac	LoRaMacInitialization	
函数原型	LoRaMacStatus_t LoRaMacInitialization(LoRaMacPrimitives_t *primitives, LoRaMacCallback_t *callbacks, LoRaMacRegion_t region)	
参数说明	primitives：mlme 和 mcps 的回调函数 callbacks：app 的回调函数 region：地区参数	
返回值	操作状态	

3.2. 发送报文

功能分类	内容	描述
发送 lora 报文	LoRaMacMcpsRequest	
函数原型	LoRaMacStatus_t LoRaMacMcpsRequest(McpsReq_t *mcpsRequest)	
参数说明	mcpsRequest：要发报文的相关信息和存放 buffer	
返回值	操作状态	

3.3. 网络控制

功能分类	内容	描述
网络控制	LoRaMacMlmeRequest	
函数原型	LoRaMacStatus_t LoRaMacMlmeRequest(MlmeReq_t *mlmeRequest)	控制终端节点 join 和发送 MAC 命令
参数说明	mlmeRequest：管理网络相关请求信息	
返回值	操作状态	

3.4. MAC 信息设置

功能分类	内容	描述
MAC 信息设置	LoRaMacMibSetRequestConfirm	
函数原型	LoRaMacStatus_t LoRaMacMibSetRequestConfirm(MibRequestConfirm_t *mibSet)	
参数说明	mibSet：MAC 信息	
返回值	操作状态	

ASR Confidential

3.5. MAC 信息获取

功能分类	内容	描述
MAC 信息获取	LoRaMacMibGetRequestConfirm	
函数原型	LoRaMacStatus_t LoRaMacMibGetRequestConfirm(MibRequestConfirm_t *mibGet)	
参数说明	mibGet：MAC 信息	
返回值	操作状态	

3.6. 回调函数

3.6.1. `MacMlmeConfirm`

功能分类	内容	描述
MAC 层管理回调	MacMlmeConfirm	
函数原型	void (*MacMlmeConfirm)(MlmeConfirm_t *MlmeConfirm)	此回调函数需要用户实现，通过函数接口 LoRaMacInitialization 回调
参数说明	MlmeConfirm：Mlme 确认参数	
返回值	无	

3.6.2. `MacMlmeIndication`

功能分类	内容	描述
MAC 层管理回调	MacMlmeIndication	
函数原型	void (*MacMlmeIndication)(MlmeIndication_t *MlmeIndication);	此回调函数需要用户实现，通过函数接口 LoRaMacInitialization 回调
参数说明	MlmeIndication：Mlme 指示参数	
返回值	无	

ASR Confidential

3.6.3. MacMcpsConfirm

功能分类	内容	描述
报文收发回调	MacMcpsConfirm	
函数原型	void (*MacMcpsConfirm)(McpsConfirm_t *McpsConfirm);	此回调函数需要用户实现，通过函数接口 LoRaMacInitialization 回调
参数说明	McpsConfirm：mcps 确认参数	
返回值	无	

3.6.4. MacMcpsIndication

功能分类	内容	描述
报文收发回调	MacMcpsIndication	
函数原型	Void (*MacMcpsIndication) (McpsIndication_t *McpsIndication)	此回调函数需要用户实现，通过函数接口 LoRaMacInitialization 回调
参数说明	McpsIndication：接收到的数据相关信息和存放 buffer	
返回值	无	

3.7. Timer

3.7.1. Timer 初始化

功能分类	内容	描述
Timer 初始化	TimerInit	
函数原型	void TimerInit(TimerEvent_t *obj, void (*callback)(void))	
参数说明	obj：Timer 对象 callback：用户的回调函数	
返回值	无	

3.7.2. Timer 启动

功能分类	内容	描述
Timer 启动	TimerStart	
函数原型	void TimerStart(TimerEvent_t *obj)	
参数说明	obj：Timer 对象	
返回值	无	

ASR Confidential

3.7.3. Timer 停止

功能分类	内容	描述
Timer 停止	TimerStop	
函数原型	void TimerStop(TimerEvent_t *obj)	
参数说明	obj：Timer 对象	
返回值	无	

3.7.4. Timer 时间设置

功能分类	内容	描述
Timer 时间设置	TimerSetValue	
函数原型	void TimerSetValue(TimerEvent_t *obj, uint32_t value)	
参数说明	obj：Timer 对象 value：定时器时间间隔	
返回值	无	

ASR Confidential

4. AliOS 操作系统接口

4.1. KV

4.1.1. KV 初始化

功能分类	内容	描述
KV 初始化	aos_kv_init	
函数原型	int aos_kv_init(void)	
参数说明	无	
返回值	无	

4.1.2. KV 键值存储

功能分类	内容	描述
KV 键值存储	aos_kv_set	
函数原型	int aos_kv_set(const char *key, const void *val, int len, int sync)	
参数说明	key：键值中的 key val：键值中的值 len：值的长度 sync：键值是否立即存储到 flash 中（需要一直为 true）	
返回值	成功或失败	

4.1.3. KV 值获取

功能分类	内容	描述
KV 值获取	aos_kv_get	
函数原型	int aos_kv_get(const char *key, void *buffer, int *buffer_len)	
参数说明	key：键值中的 key buffer：存储键值中值的 buffer buffer_len：buffer 长度	
返回值	成功或失败	

ASR Confidential