

本资料仅供内部使用！

comX100 模块调试及应用

2013 年 8 月 3 日

修改记录

制定日期	生效日期	制定 / 修订 内容摘要	页数	版本	拟稿	审查	批准
2013.08.03		初稿	2	0.01	朱正晶		
2013.09.13		增加 comX100 模块 USB 引脚	10	0.02	朱正晶		
2013.12.06		增加接口说明	10	0.03	朱正晶		

目 录

1	本文档组成部分	1
2	CANOPEN 模块简介	1
3	COMX100 和 STM32F207ZG 接口说明	1
3.1	STM32F207ZG FSMC 接口	1
3.2	COMX100 模块引脚及功能说明	2
3.3	硬件连接	4
3.4	软件设计	5
3.4.1	双端口内存 DPM 结构	5
3.4.2	驱动程序设计	7
3.4.3	时序	8

1 本文档组成部分

主要由以下几个方面组成：

- ① comX100 和 STM32F207ZG 芯片接口说明
- ② FSMC 时序设定
- ③ Toolkit 移植及应用

2 CANOpen 模块简介

CANOpen 协议为 comX100 模块为赫优讯公司的 CANOpen 主站模块。和 STM32F207 采用 FSMC SRAM 接口来通信。MCU 和 comX100 模块通信的媒介为 DPM（双口 SRAM），速度比较快。

模块的具体信息可以参考模块自带的光盘。

3 comX100 和 STM32F207ZG 接口说明

3.1 STM32F207ZG FSMC 接口

图 3-1 为 FSMC 的框架图，由图可以看出 FSMC 能支持很多种类的 RAM 和 ROM。我们使用其中的 Nor/PSRAM 接口。

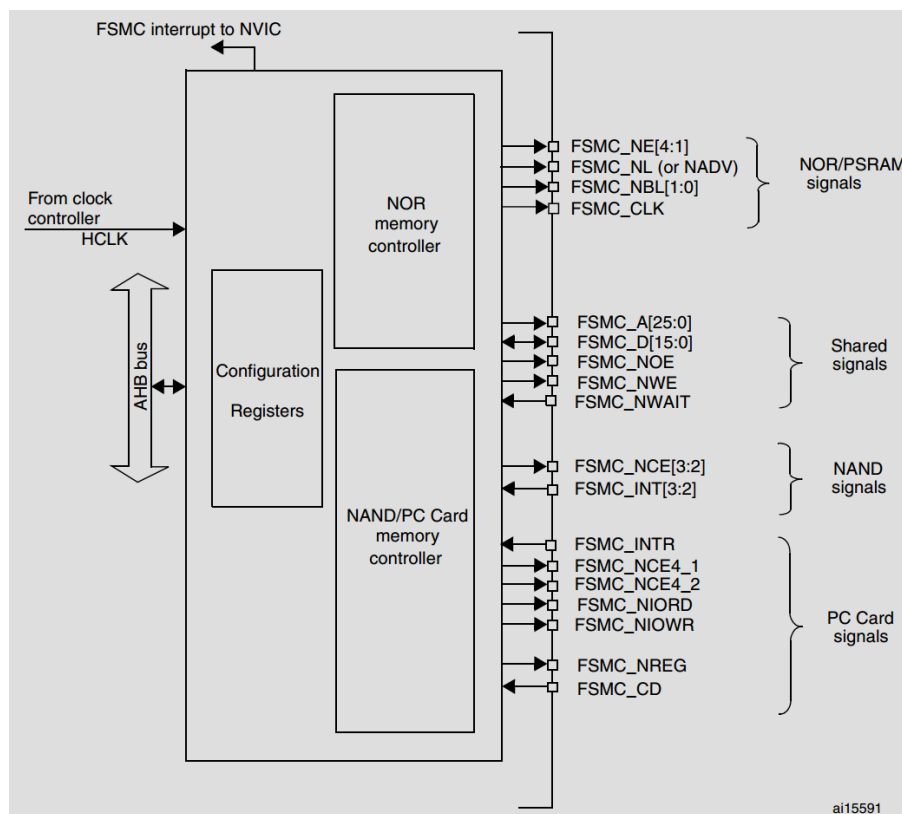


图 3-1 STM32F207ZG FSMC 框图

图 3-2 和图 3-3 分别为 STM32F207ZG 的读时序和写时序。



Figure 409. Asynchronous wait during a read access

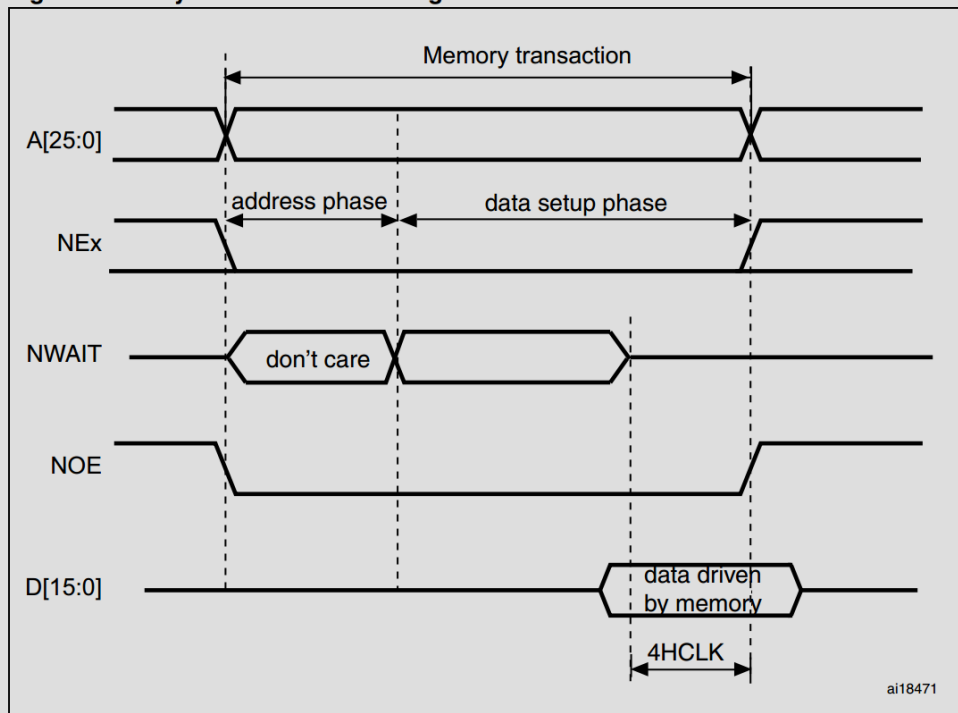


图 3-2 STM32F207ZG 读时序

Figure 410. Asynchronous wait during a write access

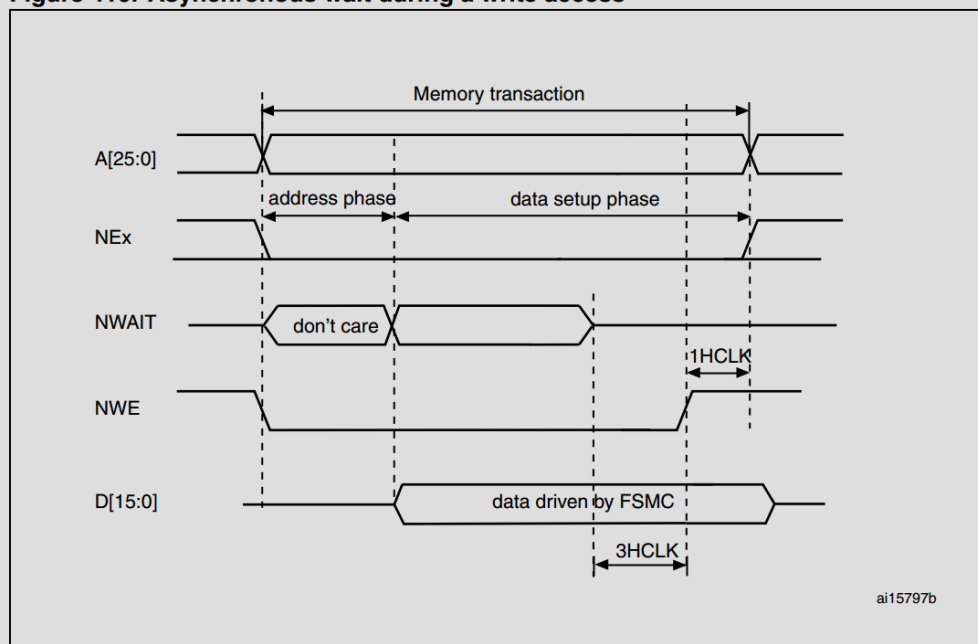


图 3-3 STM32F207ZG 写时序

3.2 comX100 模块引脚及功能说明

图 3-4 为 comX100 模块引脚及功能说明, 这里我们需要重点关注的是 16 位模式下地址 A0 的功能。



3.1.2 COMX Pinning of the System Bus Connector X1 – Parallel Mode

X1	Pin	Signal	COMX 10 PAD Type	COMX 50 PAD Type	COMX 100 PAD Type	Symbol	Type
	1	Word Interface, active low	IOU6	IOU9	IO18C	DPM_SIRQn	LVTTL Input
	2	Bus high enable, active low	IOU6	IOU9	IO18C	DPM_BHEn	LVTTL Input
	3	Data line 15	IOD6	IOU9	IO18C	DPM_D15	LVTTL Input / Output
	4	Data line 14	IOD6	IOU9	IO18C	DPM_D14	LVTTL Input / Output
	5	Data line 13	IOD6	IOU9	IO18C	DPM_D13	LVTTL Input / Output
	6	Data line 12	IOD6	IOU9	IO18C	DPM_D12	LVTTL Input / Output
	7	Data line 11	IOD6	IOU9	IO18C	DPM_D11	LVTTL Input / Output
	8	Data line 10	IOD6	IOU9	IO18C	DPM_D10	LVTTL Input / Output
	9	Data line 9	IOD6	IOU9	IO18C	DPM_D9	LVTTL Input / Output
	10	Data line 8	IOD6	IOU9	IO18C	DPM_D8	LVTTL Input / Output
	11	Ground				GND	
	12	Power Supply				+3V3	
	13	Transmit Data, Serial line	IOUS6	IODS6	IOD6	UART1_TXD	LVTTL Output
	14	Receive Data, Serial line	IOUS6	IODS6	IOD6	UART1_RXD	LVTTL Input
	15	Request to Send, Serial line & SYNC0	IOUS6	IODS6	IOD6	UART1_RTSn / SYNC0	LVTTL Output / SYNC Input / Output Signal XC3_IO0 (Note 1, 2)
	16	Clear to Send, Serial line & SYNC1	IOUS6	IODS6	IOD6	UART1_CTSn / SYNC1	LVTTL Input / SYNC Input / Output Signal XC3_IO1 (Note 1, 2)
	17	USB positive, Diagnostic line	USB	USB	USB	USB+	USB
	18	USB negative, Diagnostic line	USB	USB	USB	USB-	USB
	19	Receive Data, Diagnostic line	IOUS6	IODS6	IOD6	UART0_RXD	LVTTL Input
	20	Transmit Data, Diagnostic line	IOUS6	IODS6	IOD6	UART0_TXD	LVTTL Output
	21	Reset, active low	IUS	IUS	IO18C	DPM_RESETh	LVTTL Input; 10 kΩ pull up at COMX
	22	Busy, active low	IOU6	IOU9	IO18C	DPM_BUSYn	LVTTL Output
	23	During operation: Interrupt, active low COMX 10 at start-up: Host mode selection	IOU6	IOU9	IO18C	DPM_DIRQn	During operation: LVTTL Output At start-up: LVTTL Input
	24	Read, active low	IOU6	IOU9	IO18C	DPM_RDn	LVTTL Input
	25	Write, active low	IOU6	IOU9	IO18C	DPM_WRn	LVTTL Input
	26	Chip select, active low	IOU6	IOU9	IO18C	DPM_CSn	LVTTL Input

Table 18: COMX Pinning of the System Bus Connector X1- Parallel DPM Mode (Part 1)

图 3-4 comX100 模块引脚及功能

X1	Pin	Signal	COMX 10 PAD Type	COMX 50 PAD Type	COMX 100 PAD Type	Symbol	Type
	27	Address line 13	IOD6	IOU9	IO18C	DPM_A13	LVTTL Input
	28	Address line 12	IOD6	IOU9	IO18C	DPM_A12	LVTTL Input
	29	Address line 11	IOD6	IOU9	IO18C	DPM_A11	LVTTL Input
	30	Address line 10	IOD6	IOU9	IO18C	DPM_A10	LVTTL Input
	31	Address line 9	IOD6	IOU9	IO18C	DPM_A9	LVTTL Input
	32	Address line 8	IOD6	IOU9	IO18C	DPM_A8	LVTTL Input
	33	Address line 7	IOD6	IOU9	IO18C	DPM_A7	LVTTL Input
	34	Address line 6	IOD6	IOU9	IO18C	DPM_A6	LVTTL Input
	35	Address line 5	IOD6	IOU9	IO18C	DPM_A5	LVTTL Input
	36	Address line 4	IOD6	IOU9	IO18C	DPM_A4	LVTTL Input
	37	Address line 3	IOD6	IOU9	IO18C	DPM_A3	LVTTL Input
	38	Address line 2	IOD6	IOU9	IO18C	DPM_A2	LVTTL Input
	39	Address line 1	IOD6	IOU9	IO18C	DPM_A1	LVTTL Input
	40	Address line 0	IOD6	IOU9	IO18C	DPM_A0	LVTTL Input
	41	Data line 7	IOD6	IOU9	IO18C	DPM_D7	LVTTL Input / Output
	42	Data line 6	IOD6	IOU9	IO18C	DPM_D6	LVTTL Input / Output
	43	Data line 5	IOD6	IOU9	IO18C	DPM_D5	LVTTL Input / Output
	44	Data line 4	IOD6	IOU9	IO18C	DPM_D4	LVTTL Input / Output
	45	Data line 3	IOD6	IOU9	IO18C	DPM_D3	LVTTL Input / Output
	46	Data line 2	IOD6	IOU9	IO18C	DPM_D2	LVTTL Input / Output
	47	Data line 1	IOD6	IOU9	IO18C	DPM_D1	LVTTL Input / Output
	48	Data line 0	IOD6	IOU9	IO18C	DPM_D0	LVTTL Input / Output
	49	Ground				GND	
	50	Power Supply				+3V3	

Table 19: COMX Pinning of the System Bus Connector X1 – Parallel DPM Mode (Part 2)

图 3-4 comX100 模块引脚及功能（续）

DPM_BHE _n	DPM_A0	Function
0	0	word access
0	1	access high byte
1	0	access low byte
1	1	no access

Table 34: Function Table of the 16 Bit Decode Logic

图 3-5 16 位模式下 A0 引脚的功能

3.3 硬件连接

由 3.1 和 3.2 节我们得出 16 位模式下的连接方式：

FSMC_NBL0 接 DPM_A0

FSMC_NBL1 接 DPM_BHE_n

具体的连接图如下：

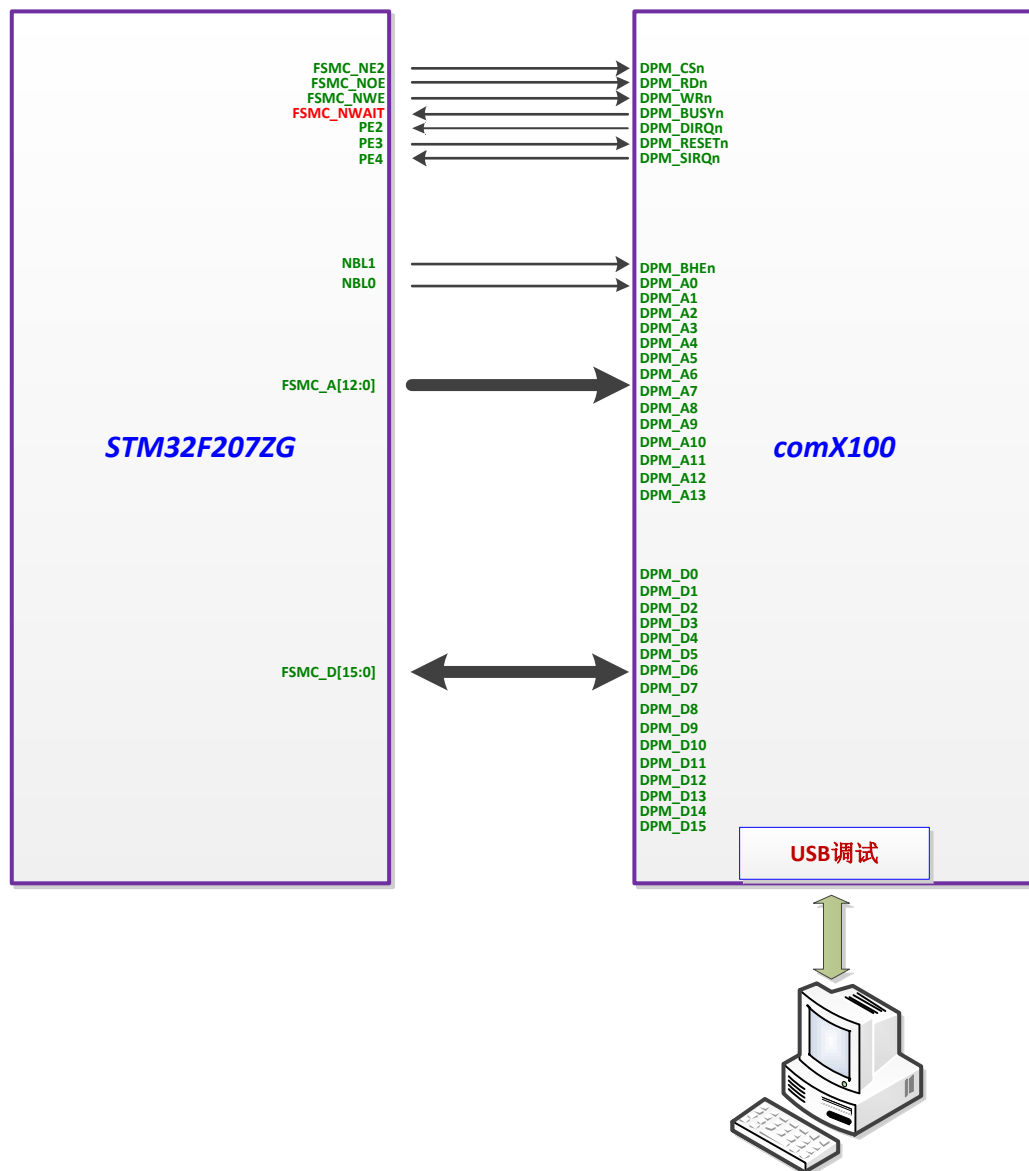


图 3-6 STM32F207ZG 和 comX100 模块硬件连接

3.4 软件设计

嵌入式模块 comX100 提供的主机接口是双端口内存 DPM, 用户应用程序通过 DPM 接口来访问该模块。同时, 为了提高整个系统的实时性和可靠性, 主机系统使用的是实时多任务操作系统 $\mu C/OS-II$ 。因此, 在进行软件设计时, 主要完成驱动程序以及应用程序的编写。

3.4.1 双端口内存 DPM 结构

用户应用程序通过双端口内存 DPM 来访问 CANOpen 嵌入式模块 comX100, 嵌入式模块 comX100 提供的双端口内存 DPM 接口是 16 KByte 的地址空间, 其具体结构如图 3-7 所示。

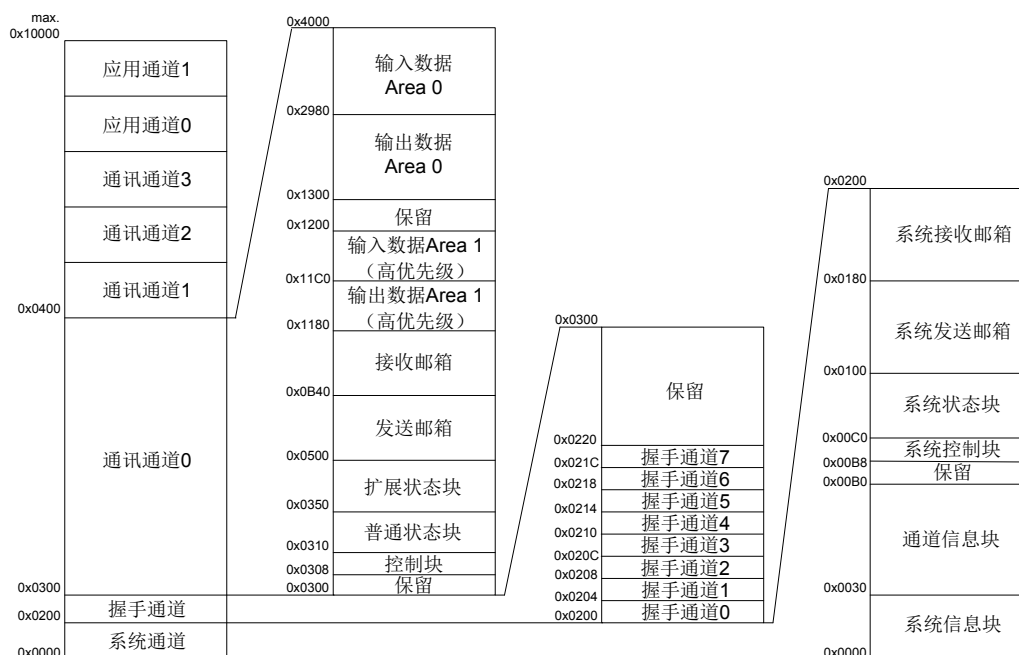


图 3-7 双端口内存地址空间

netX 提供的双端口内存最大地址空间是 64KByte，分为系统通道、握手通道、通讯通道 0~3、应用通道 0~1。对于嵌入式模块 comX，CANOpen 的协议堆栈已经保存在模块的 Flash 中，因此只使用了默认的系统通道、握手通道和通讯通道 0，16KByte 的地址空间。

系统通道 512 个字节，位于双端口内存 DPM 的起始位置，是最重要的通道，总是存在。它包含了系统本身（netX 硬件和实时操作系统 rcX）的关键信息，并且提供一种邮箱的传输机制，用于发送/接收跟系统相关的报文。各功能块如下表所述。

系统通道		
名称	大小	描述
系统信息块	48 Bytes	系统信息，制造商信息，序列号等
通道信息块	128 Bytes	所使用的通讯通道和应用通道的配置信息
保留	8 Bytes	握手单元保留
系统控制块	8 Bytes	系统控制和命令
系统状态块	64 Bytes	系统状态信息
系统邮箱	256 Bytes	系统发送/接收报文邮箱

表 3-1 系统通道结构

握手通道 256 字节，从偏移地址 0x0200 开始，提供主机系统与 netX 固件（运行在 netX 芯片中的协议堆栈）之间的数据传输同步机制。所有通道的握手寄存器都在此区域，每个握手寄存器包含两类握手信息：系统握手信息和通讯握手信息。系统握手信息与“系统设备”相关，由主机应用执行 netX 系统相关的命令，如复位等。通讯握手信息用于同步循环或非循环数据，同时向主机系统提供状态变化信息。

通讯通道 m*256 字节，从偏移地址 0x0300 开始，现场总线协议堆栈使用该通道，与主机系统进行循环数据和非循环数据的数据交换。共有四个通讯通道，每个通讯通道分别对应 netX 芯片的 xMAC/xPEC 端口，其结构都相同。通讯通道 0 的默认大小是 16KByte，其各功能块如表 3-2 所述。

通讯通道（默认布局）			
名称	偏移地址	大小	描述
保留	0x0300	8 Bytes	保留，设为 0
控制块	0x0308	8 Bytes	主机系统控制通讯通道功能块
普通状态块	0x0310	64 Bytes	与协议堆栈相关的状态信息
扩展状态块	0x0350	432 Bytes	网络特殊的状态信息
发送邮箱	0x0500	1600 Bytes	发送非循环数据给 netX
接收邮箱	0x0B40	1600 Bytes	接收来自 netX 的非循环数据
输出数据 Area1	0x1180	64 Bytes	发送给网络的循环数据（高优先级）
输入数据 Area1	0x11C0	64 Bytes	接收来自网络的循环数据（高优先级）
保留	0x1200	256 Bytes	保留，设为 0
输出数据 Area0	0x1300	5760 Bytes	发送给网络的循环数据
输入数据 Area0	0x2980	5760 Bytes	接收来自网络的循环数据

表 3-2 通讯通道结构

应用通道 m*256 字节，根据实际的需求，并不一定需要该通道，该通道实现用户特殊的信息交互，由客户定义使用。

3.4.2 驱动程序设计

赫优讯提供嵌入式模块 comX 的驱动源码 Toolkit，用户可以根据实际的需求，在不同的目标系统中集成该 Toolkit 提供的驱动接口，这样用户应用程序就能很方便的访问嵌入式模块 comX 的双端口内存。Toolkit 提供的驱动接口与实时 CANOpen 协议无关，只与 comX 双端口内存 DPM 相关，因此，如果基于 comX 模块开发多种协议的设备，其驱动程序是通用的。

在本系统中，使用 μ C/OS-II 实时操作系统，设备驱动主要完成对嵌入式模块 comX 双端口内存 DPM 的访问，用户程序通过调用 Toolkit 提供的接口函数来访问实时 CANOpen 嵌入式模块 comX。 μ C/OS-II 设备驱动的框架如图 3-8 所示。

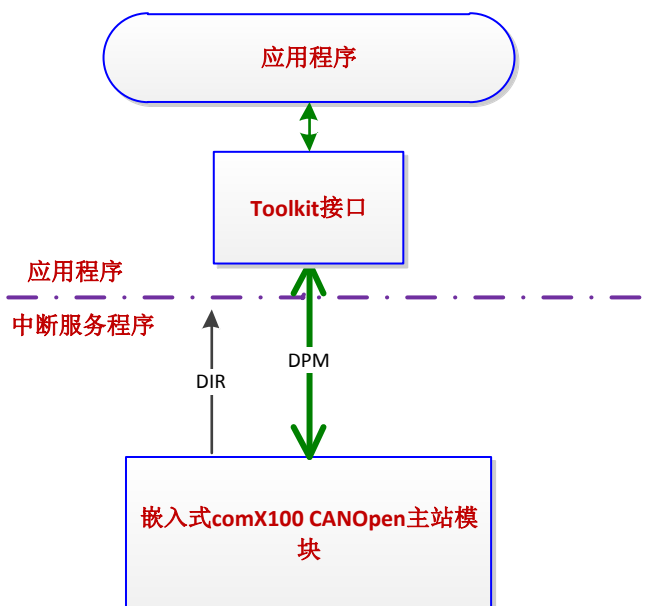


图 3-8 μ C/OS-II 设备驱动框架

μC/OS-II 设备驱动主要实现的函数如下表。

函数	描述
xDriverOpen()	建立应用程序与驱动的连接
xDriverClose()	断开应用程序与驱动的连接
xDriverGetInformation()	获得驱动信息
xDriverGetErrorDescription()	获得驱动错误代码描述
xDriverEnumBoard()	列举驱动程序管理的模块/设备
xDriverEnumChannels()	列举某模块上使用的通讯通道
xChannelIORead ()	读取 DPM 中的 PDO 数据

表 3-3 μC/OS-II 设备驱动函数表

3.4.3 时序

图 3-9 为 comX100 读写时序图

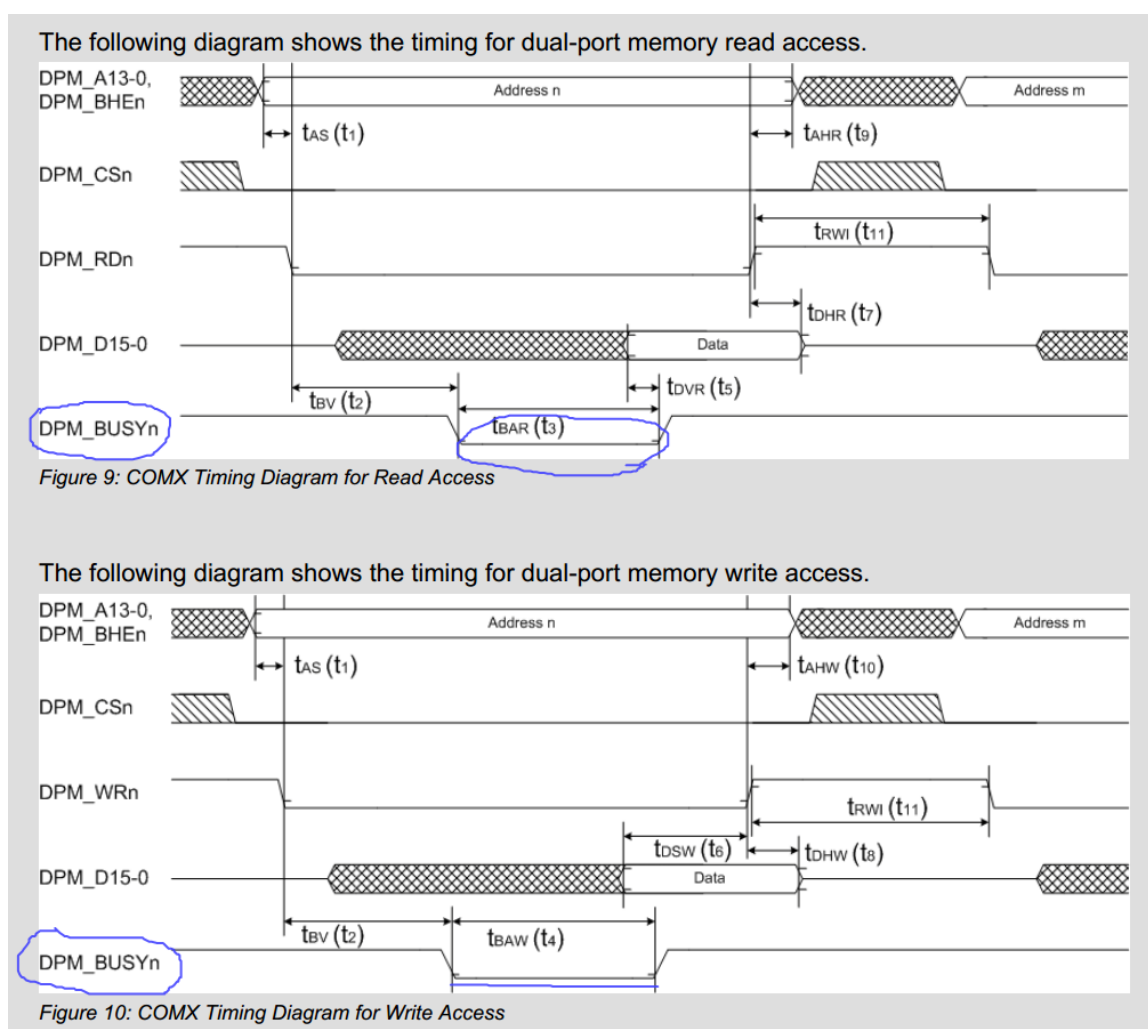


图 3-9 comX100 读写时序图

根据以上时序图和具体的时间（参考文档 Design Guide COMX Communication Modules）设置 STM32F207ZG 的 FSMC 时序如图 3-10。

```

00158:
00159:  /*-- FSMC Configuration -----*/
00160:  p.FSMC_AddressSetupTime = 4;
00161:  p.FSMC_AddressHoldTime = 0;
00162:  p.FSMC_DataSetupTime = 14;
00163:  p.FSMC_BusTurnAroundDuration = 0;
00164:  p.FSMC_CLKDivision = 0;
00165:  p.FSMC_DataLatency = 0;
00166:  p.FSMC_AccessMode = FSMC_AccessMode_A;
    
```

图 3-10 STM32F207ZG 的 FSMC 时序设置

3.4.4 程序流程图

有了 Toolkit，我们程序中需要做的事情不多。这里有个问题需要注意，读 PDO 必须使用查询的方法。比如程序每隔 100ms 查询一次 DPM RPDO，如果发生变化就说明有新的数据收到了，这时需要我们进行进一步处理；如果没有变化说明没有新的数据，不需要做任何处理。

图 3-11 为程序收发 PDO 的流程图。

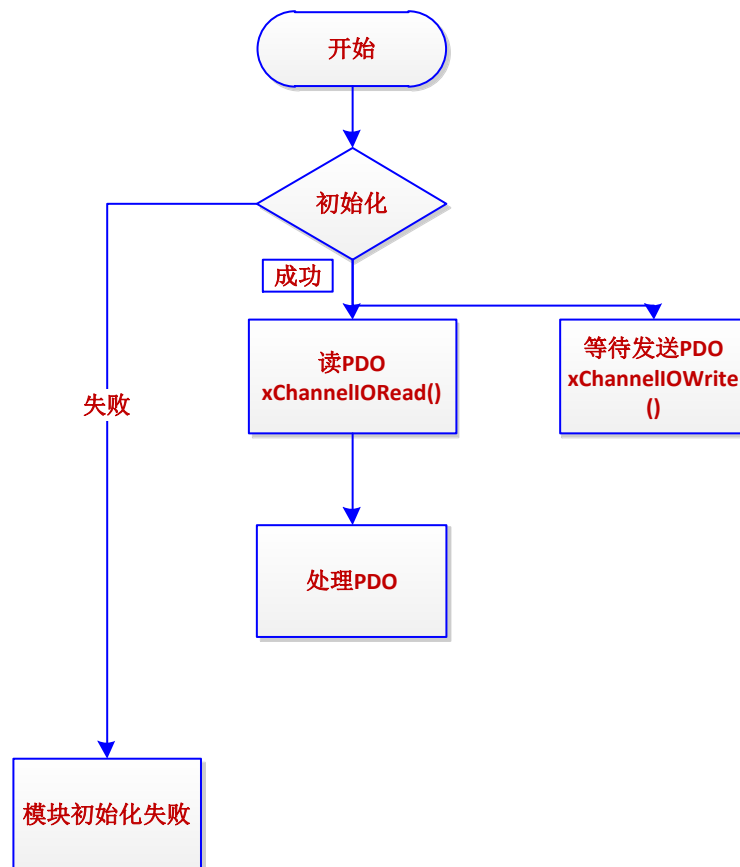


图 3-11 PDO 收发流程图

4 API 封装说明

comX100 使用了一个专门的 task 来管理数据收发，对外提供两个接口。

4.1 初始化 comX100 task API

void init_comX100_module_task(control_pRPDO_function* p_control_callback, u8 call_num)

参数: p_control_callback, 接收到数据 callback 函数
call_num, RPDO 的数量

返回值: void

说明: 这个函数在系统启动时调用一次就行。接收到数据会调用 callback 函数, 注意: callback 函数运行在 comX100 task 环境下, 在内部处理时应简单快速, 否则可能会影响数据的接收。

4.2 发送 PDO API

u8 comX100_send_packet(void* p_data, u8 TPDO_offset, u8 TPDO_num)

参数: p_data, 要发送的数据 buffer 指针
TPDO_offset, TPDO 偏移地址, 这个参数需要和 comX100 模块配置的相对应。
TPDO_num, 一次发送的 PDO 数据包

返回值: u8, 0 成功, 1 失败

说明: 因为 comX100 模块的原因, 如果两次填写的数据完全一样, 那么, 数据不会被发送, 只有当两次发送的数据有差异时才会进行发送。