

## 概述

CAN (Controller Area Network) 是一种高性能和高可靠性的通讯协议。SPC2168 内建三个 SIO 单元。使用 SIO 单元的可编程性，基于 CAN2.0-B，实现了 SIO\_CAN 的功能。

SPIN TROL

# 目录

<b>1</b>	<b>SIO_CAN 特性</b> .....	<b>8</b>
<b>2</b>	<b>SIO_CAN 系统概述</b> .....	<b>11</b>
2.1	消息对象以及软件 FIFO 定义 .....	11
<b>3</b>	<b>SIO_CAN 操作方式</b> .....	<b>13</b>
3.1	配置 SIO 时钟 .....	13
3.2	配置 SIO 为 SIO_CAN，配置 PINMUX .....	13
<b>4</b>	<b>API 函数</b> .....	<b>15</b>
<b>5</b>	<b>代码示例</b> .....	<b>17</b>
<b>6</b>	<b>寄存器</b> .....	<b>18</b>
6.1	SIO_CAN 寄存器表 .....	18
6.2	SIO_CAN 寄存器 .....	20

## 图片列表

图 2-1: 基于 SIO 的 CAN 系统架构 ..... 11

SPIN TROL

## 表格列表

表 1-1:	SIO_CAN 波特率和位时间配置 .....	9
表 1-2:	管脚分配.....	10
表 4-1:	API 函数列表 .....	15
表 6-1:	SIO 模块基地址 .....	18
表 6-2:	SIO_CAN 模块寄存器表 .....	19
表 6-3:	Status Register 0 (ST0) Layout .....	20
表 6-4:	Status Register 0 (ST0) Field Description .....	20
表 6-5:	ID Filter 0 Low 16 bit Register (IDF0L) Layout.....	20
表 6-6:	ID Filter 0 Low 16 bit Register (IDF0L) Field Description .....	20
表 6-7:	Transmit Data Register 0 (TXDATA0) Layout .....	21
表 6-8:	Transmit Data Register 0 (TXDATA0) Field Description .....	21
表 6-9:	Transmit Data Register 1 (TXDATA1) Layout .....	21
表 6-10:	Transmit Data Register 1 (TXDATA1) Field Description .....	21
表 6-11:	Transmit Data Register 2 (TXDATA2) Layout .....	22
表 6-12:	Transmit Data Register 2 (TXDATA2) Field Description .....	22
表 6-13:	Transmit Data Register 3 (TXDATA3) Layout .....	22
表 6-14:	Transmit Data Register 3 (TXDATA3) Field Description .....	22
表 6-15:	Transmit Extended ID Register 0 (TXEID0) Layout .....	23
表 6-16:	Transmit Extended ID Register 0 (TXEID0) Field Description.....	23
表 6-17:	Transmit Extended ID Register 1 (TXEID1) Layout.....	23
表 6-18:	Transmit Extended ID Register 1 (TXEID1) Field Description.....	23
表 6-19:	Transmit Standard ID Register (TXSID) Layout.....	24
表 6-20:	Transmit Standard ID Register (TXSID) Field Description .....	24
表 6-21:	Control Register (CTL) Layout .....	25
表 6-22:	Control Register (CTL) Field Description .....	25
表 6-23:	ID Filter 0 High 16 bit Register (IDF0H) Layout .....	26
表 6-24:	ID Filter 0 High 16 bit Register (IDF0H) Filed Description .....	26
表 6-25:	ID Mask 0 Low 16 bit Register (IDM0L) Layout.....	26
表 6-26:	ID Mask 0 Low 16 bit Register (IDM0L) Filed Description.....	26
表 6-27:	ID Mask 0 High 16 bit Register (IDM0H) Layout .....	27
表 6-28:	ID Mask 0 High 16 bit Register (IDM0H) Filed Description .....	27
表 6-29:	ID Filer 1 Low 16 bit Register (IDF1L) Layout .....	27
表 6-30:	ID Filer 1 Low 16 bit Register (IDF1L) Filed Description.....	27
表 6-31:	ID Filer 1 High 16 bit Register (IDF1H) Layout .....	28
表 6-32:	ID Filer 1 High 16 bit Register (IDF1H) Field Description .....	28
表 6-33:	ID Mask 1 Low 16 bit Register (IDM1L) Layout.....	28
表 6-34:	ID Mask 1 Low 16 bit Register (IDM1L) Field Description.....	28
表 6-35:	ID Mask 1 High 16 bit Register (IDM1H) Layout .....	29
表 6-36:	ID Mask 1 High 16 bit Register (IDM1H) Field Description .....	29
表 6-37:	Receive Data Register 0 (RXD0) Layout .....	29
表 6-38:	Receive Data Register 0 (RXD0) Field Description .....	29
表 6-39:	Receive Data Register 1 (RXD1) Layout .....	30
表 6-40:	Receive Data Register 1 (RXD1) Field Description .....	30

表 6-41:	Receive Data Register 2 (RXD2) Layout .....	30
表 6-42:	Receive Data Register 2 (RXD2) Field Description .....	30
表 6-43:	Receive Data Register 3 (RXD3) Layout .....	31
表 6-44:	Receive Data Register 3 (RXD3) Field Description .....	31
表 6-45:	Receive Extended ID Register 0 (RXEID0) Layout .....	31
表 6-46:	Receive Extended ID Register 0 (RXEID0) Field Description .....	31
表 6-47:	Receive Extended ID Register 1 (RXEID1) Layout .....	32
表 6-48:	Receive Extended ID Register 1 (RXEID1) Field Description .....	32
表 6-49:	Receive Standard ID Register (RXSID) Layout .....	32
表 6-50:	Receive Standard ID Register (RXSID) Field Description .....	32
表 6-51:	ID Filer 2 Low 16 bit Register (IDF2L) Layout .....	33
表 6-52:	ID Filer 2 Low 16 bit Register (IDF2L) Field Description .....	33
表 6-53:	ID Filer 2 High 16 bit Register (IDF2H) Layout .....	33
表 6-54:	ID Filer 2 High 16 bit Register (IDF2H) Field Description .....	33
表 6-55:	ID Mask 2 Low 16 bit Register (IDM2L) Layout .....	34
表 6-56:	ID Mask 2 Low 16 bit Register (IDM2L) Field Description .....	34
表 6-57:	ID Mask 2 High 16 bit Register (IDM2H) Layout .....	34
表 6-58:	ID Mask 2 High 16 bit Register (IDM2H) Field Description .....	34
表 6-59:	ID Filer 3 Low 16 bit Register (IDF3L) Layout .....	35
表 6-60:	ID Filer 3 Low 16 bit Register (IDF3L) Field Description .....	35
表 6-61:	ID Filer 3 High 16 bit Register (IDF3H) Layout .....	35
表 6-62:	ID Filer 3 High 16 bit Register (IDF3H) Field Description .....	35
表 6-63:	ID Mack 3 Low 16 bit Register (IDM3L) Layout .....	36
表 6-64:	ID Mack 3 Low 16 bit Register (IDM3L) Field Description .....	36
表 6-65:	ID Mack 3 High 16 bit Register (IDM3H) Layout .....	36
表 6-66:	ID Mack 3 High 16 bit Register (IDM3H) Field Description .....	36
表 6-67:	Status Register 1 (STS1) Layout .....	37
表 6-68:	Status Register 1 (STS1) Field Description .....	37

## 版本历史

版本	日期	作者	状态	变更
1	2019 年 09 月 28 日	韩伟	Outdated	设计 V1.1.0: 初始版本。
2	2021 年 01 月 04 日	韩伟	Outdated	设计 V1.4.0: 修正 SIO 模块最高时钟速率为 100MHz。
A/0	2022 年 08 月 24 日	韩伟	Outdated	设计 V1.6.0: 更新 SIO_Init(), SIO_Deinit() 相关驱动, 修正写 PLA 的时序; 修改文件格式。
A/1	2023 年 03 月 12 日	韩伟	Released	设计 V1.6.0: 添加使用说明, 使用 SIO_CAN 时, 系统必须使用外部晶振作为时钟源。

## 术语或缩写

术语或缩写	描述
SIO	Smart Input Output
CAN	Controller Area Network

SPIN  
TROL

## 1 SIO\_CAN 特性

基于 SIO 所实现的 SIO\_CAN 系统具备如下特性：

- 支持 CAN2.0-B 协议标准；
- 支持标准帧和扩展帧；
- 支持自动重发；
- 支持 Bus-off 后自动恢复 error active 状态；
- 支持 1 个发送信箱和 1 个接收信箱；
- 支持 4 个消息 ID 过滤配置；
- 支持接收数据完成中断；
- 支持发送数据完成中断。

---

**注意** 使用 SIO\_CAN 时，系统时钟需要使用外部晶振作为参考时钟，以满足 CAN 网络对于时钟精准确度的要求。

---



SIO 模块时钟为 100MHz 情况下，支持的波特率和位时间配置如下表所示。

表 1-1: SIO\_CAN 波特率和位时间配置

波特率 (Kbps)	Propagation Segment	Phase Buffer Segment 1	Phase Buffer Segment 2	Re-synchronization Jump Width	采样点	Pla 配置文件后缀
500	3tq	1tq	5tq	2tq	50%	500K_13152
	3tq	1tq	5tq	3tq	50%	500K_13153
	3tq	1tq	5tq	4tq	50%	500K_13154
	3tq	2tq	4tq	2tq	60%	500K_13242
	3tq	2tq	4tq	3tq	60%	500K_13243
	3tq	2tq	4tq	4tq	60%	500K_13244
	3tq	3tq	3tq	2tq	70%	500K_13332
	3tq	3tq	3tq	3tq	70%	500K_13333
250	3tq	4tq	2tq	2tq	80%	500K_13422
	5tq	5tq	5tq	2tq	69%	250K_15552
	5tq	5tq	5tq	3tq	69%	250K_15553
	5tq	5tq	5tq	4tq	69%	250K_15554
	7tq	4tq	4tq	2tq	75%	250K_17442
	7tq	4tq	4tq	3tq	75%	250K_17443
	7tq	4tq	4tq	4tq	75%	250K_17444
	9tq	3tq	3tq	2tq	81%	250K_19332
25	9tq	3tq	3tq	3tq	81%	250K_19333
	11tq	2tq	2tq	2tq	88%	250K_111222
	5tq	5tq	5tq	2tq	69%	25K_15552
	5tq	5tq	5tq	3tq	69%	25K_15553
	5tq	5tq	5tq	4tq	69%	25K_15554
	7tq	4tq	4tq	2tq	75%	25K_17442
	7tq	4tq	4tq	3tq	75%	25K_17443
	7tq	4tq	4tq	4tq	75%	25K_17444
9tq	3tq	3tq	2tq	81%	25K_19332	
9tq	3tq	3tq	3tq	81%	25K_19333	
11tq	2tq	2tq	2tq	88%	25K_111222	

注意事项：

- 不同的波特率和位时间配置需要不同的 SIO 配置文件，不支持通过驱动函数来修改这些参数。需要不同的配置文件，请联系 Spintrol 工程师。
- 可以通过修改 SIO 模块时钟来得到其他波特率。例如，SIO 模块时钟为 100MHz，使用的是 500Kbps 的配置；可以将 SIO 模块时钟配置为 25MHz，则同样配置文件实际使用的波特率为 125Kbps。

表 1-2: 管脚分配

SIO 模块名	SIO 管脚编号	GPIO 管脚编号	功能
SIO0	0	GPIO0	TXD
SIO0	1	GPIO1	RXD

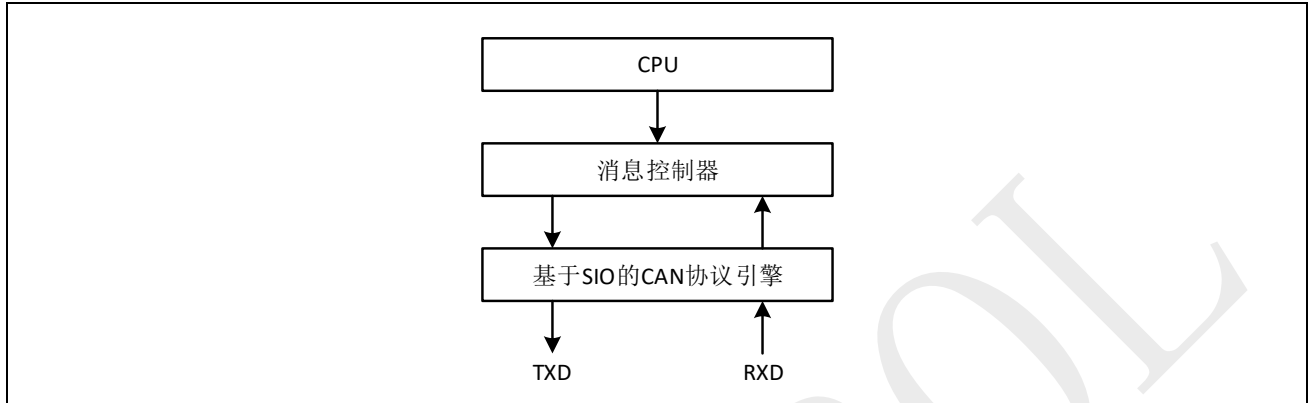
注意事项:

- SIO 时钟频率  $F_{SIO} \leq 100\text{MHz}$ ;
- 不支持通过修改 SIO 配置文件里的 SIO 管脚数组来实现管脚的重新分配。需要不同的配置文件, 请联系 Spintrol 工程师。

## 2 SIO\_CAN 系统概述

如图 2-1 所示，控制器局域网（CAN）系统主要由消息控制器和 CAN 协议引擎组成。

图 2-1: 基于 SIO 的 CAN 系统架构



在发送时，将 CAN 消息写入消息控制器，然后由 CAN 协议引擎发送，发送完成后，触发数据发送完成中断。CAN 协议引擎负责监控总线状态，发送消息到总线，以及检测应答。

在接收时，CAN 协议引擎在被正确配置后开始监控总线上的数据接收，当接收到数据 CRC 校验正确，发送 ACK 信号。接收完整的帧后，根据掩码和过滤器设置，接收符合过滤要求的帧并触发接收到数据中断。

由于 SIO\_CAN 只具有一个接收信箱和一个发送信箱，因此需要软件实现信箱 FIFO 功能来收发批量的数据帧。

TXD 和 RXD 分别为发送通道输出信号和接收通道输入信号。

**注意** 需要外接 CAN 收发器完成 TXD/RXD 逻辑电平到 CAN 总线差分电平的互相转换。

### 2.1 消息对象以及软件 FIFO 定义

每个接收消息对象定义如下：

#### 接收消息定义

```

typedef struct
{
    uint32_t ID : 29; /*!< 帧 ID。
                                标准帧取值范围：[0:0x7FF];
                                扩展帧取值范围：[0:0x1FFFFFFF] */
    uint8_t IDE : 1; /*!< 扩展帧使能。数据类型为：CAN_IDEnumDef */
    uint8_t RTR : 1; /*!< 远程帧使能。数据类型为：CAN_RTREnumDef */
    uint8_t DLC : 4; /*!< 数据长度。取值范围：[0:4]。 */
    uint8_t DATA[8]; /*!< 数据。 */
    uint32_t FilterMatchIndex; /*!< 匹配的过滤器索引。取值范围[0:3]。 */
} CAN_RxFrameTypeDef;
  
```

每个发送消息对象定义如下：

#### 发送消息定义

```
typedef struct
{
    uint32_t ID : 29; /*!< 帧 ID。
                    标准帧取值范围：[0:0x7FF];
                    扩展帧取值范围：[0:0x1FFFFFFF] */
    uint8_t IDE : 1; /*!< 扩展帧使能。数据类型为：CAN_IDEnumDef */
    uint8_t RTR : 1; /*!< 远程帧使能。数据类型为：CAN_RTREnumDef */
    uint8_t DLC : 4; /*!< 数据长度。取值范围：[0:4]。 */
    uint8_t DATA[8]; /*!< 数据。 */
} CAN_TxFrameTypeDef
```

每个接收消息对象的 FIFO 定义如下：

#### 接收消息 FIFO 定义

```
typedef struct
{
    CAN_RxFrameTypeDef * FRAME ; /*!< 接收帧缓存指针 */
    uint16_t DEPTH ; /*!< FIFO 深度。 */
    uint16_t MASK ; /*!< 掩码。 */
    uint16_t BEGIN ; /*!< FIFO 头位置。 */
    uint16_t END ; /*!< FIFO 尾位置。 */
    uint16_t LEVEL ; /*!< 当前 FIFO 深度。 */
    uint16_t OVERFLOW ; /*!< FIFO 溢出。 */
} CAN_RxFIFOTypeDef;
```

每个发送消息对象的 FIFO 定义如下：

#### 发送消息 FIFO 定义

```
typedef struct
{
    CAN_TxFrameTypeDef * FRAME ; /*!< 发送帧缓存指针 */
    uint16_t DEPTH ; /*!< FIFO 深度。 */
    uint16_t MASK ; /*!< 掩码。 */
    uint16_t BEGIN ; /*!< FIFO 头位置。 */
    uint16_t END ; /*!< FIFO 尾位置。 */
    uint16_t LEVEL ; /*!< 当前 FIFO 深度。 */
    uint16_t OVERFLOW ; /*!< FIFO 溢出。 */
} CAN_TxFIFOTypeDef;
```

## 3 SIO\_CAN 操作方式

Spintrol 提供了相应的软件库来简化该系统的使用。

### 3.1 配置 SIO 时钟

用户可以通过 SIOCLKCTL 寄存器来配置 SIO 时钟，包括时钟的使能和分频比。具体可参见《SPC2168 Technical Reference Manual》的第 3 章。当 SIO 被配置用作 SIO\_CAN 时，所允许的 SIO 模块时钟最高频率可达 100MHz（对于部分型号芯片，最高频率会达不到 100MHz，具体参考 TRM）。

### 3.2 配置 SIO 为 SIO\_CAN，配置 PINMUX

提供了 SIO\_CAN\_Program()函数，配置 SIO 模块，将其初始化成 SIO\_CAN；SIO\_CAN\_Init()使能 SIO 运行且配置 PINMUX，将引脚切换至 SIO 通道。用户只需要在代码中直接调用这些函数即可。

注意：

- SIO\_CAN 的引脚重定义需要联系 Spintrol 工程师进行重新配置，目前暂不支持客户自定义。
- 当使用多个 SIO 模块，需要先调用所有的 SIO\_XXX\_Program()函数，确保所有 SIO 模块的 uCode 和 PLA 配置完成后，再调用所有的 SIO\_XXX\_Init()函数，使能各个模块及配置 IO 管脚。

SIO\_CAN 初始化代码如下：

#### 示例代码 3-1: SIO\_CAN 初始化

```
/* Config Flash Timing for 200 MHz */
FLASH_WALLOW();
FLASH_SetTiming(200000000);
FLASH_WDIS();

/* Clock Init */
CLOCK_InitWithRCO(CLOCK_HCLK_200MHZ);

/* Configure SIO Clock, maximum clock is 100MHz */
CLOCK_SetModuleDiv(SIO0_MODULE, 2);
CLOCK_EnableModule(SIO0_MODULE);

/* SIO_CAN RX Filter Init */
// geFilterEnable = ENABLE;
if( ENABLE == geFilterEnable )
{
    // matched id: ( mask & id ) == ( mask & filter_ref_id )
    SIO_CAN_SetMaskFilter(SIOx, 0,
```

```
        CAN_ID_STD, CAN_RTR_NOFILTER, 0xFFFF, 0x0004) ; // ID must be 0x4
SIO_CAN_SetMaskFilter(SIOx, 1,
        CAN_ID_STD, CAN_RTR_NOFILTER, 0xFFFF, 0x0001) ; // ID must be 0x1
SIO_CAN_SetMaskFilter(SIOx, 2,
        CAN_ID_STD, CAN_RTR_NOFILTER, 0xFFFF, 0x0002) ; // ID must be 0x2
SIO_CAN_SetMaskFilter(SIOx, 3,
        CAN_ID_STD, CAN_RTR_NOFILTER, 0xFFFF, 0x0000) ; // ID must be 0x0
}
else
{
    // receive any messages
    SIO_CAN_DisableMaskFilter(SIOx, 0) ;
    SIO_CAN_DisableMaskFilter(SIOx, 1) ;
    SIO_CAN_DisableMaskFilter(SIOx, 2) ;
    SIO_CAN_DisableMaskFilter(SIOx, 3) ;
}

/* SIO program to SIO_CAN function */
SIO_CAN_Program(SIOx);

/* SIO_CAN Init */
SIO_CAN_Init(SIOx);

/* Enable interrupt */
NVIC_EnableIRQ(SIO0A_IRQn);
```

## 4 API 函数

注意：函数详细功能请参考驱动代码注释。

表 4-1: API 函数列表

函数名称	说明
<b>ErrorStatus</b> CAN_TxFIFOPush( CAN_TxFIFOTypeDef * pCanTxFIFO, CAN_TxFrameTypeDef * pCanTxFrame );	TXFIFO 入列。 pCanTxFIFO: TXFIFO 地址指针。 pCanTxFrame: 发送帧数据指针。 TXFIFO 满则返回 ERROR, 否则返回 SUCCESS。
<b>ErrorStatus</b> CAN_TxFIFOPop( CAN_TxFIFOTypeDef * pCanTxFIFO, CAN_TxFrameTypeDef * pCanTxFrame );	TXFIFO 出列。 pCanTxFIFO: TXFIFO 地址指针。 pCanTxFrame: 发送帧数据指针。 TXFIFO 空则返回 ERROR, 否则返回 SUCCESS。
<b>void</b> CAN_TxFIFOClear( CAN_TxFIFOTypeDef * pCanTxFIFO );	TXFIFO 清空。 pCanTxFIFO: TXFIFO 地址指针。
<b>ErrorStatus</b> CAN_RxFIFOPush( CAN_RxFIFOTypeDef * pCanRxFIFO, CAN_RxFrameTypeDef * pCanRxFrame );	RXFIFO 入列。 pCanRxFIFO: RXFIFO 地址指针。 pCanRxFrame: 接收帧数据指针。 RXFIFO 满则返回 ERROR, 否则返回 SUCCESS。
<b>ErrorStatus</b> CAN_RxFIFOPop( CAN_RxFIFOTypeDef * pCanRxFIFO, CAN_RxFrameTypeDef * pCanRxFrame );	RXFIFO 出列。 pCanRxFIFO: RXFIFO 地址指针。 pCanRxFrame: 接收帧数据指针。 RXFIFO 空则返回 ERROR, 否则返回 SUCCESS。
<b>void</b> CAN_RxFIFOClear( CAN_RxFIFOTypeDef * pCanRxFIFO );	RXFIFO 清空。 pCanRxFIFO: RXFIFO 地址指针。
<b>void</b> SIO_CAN_Program(SIOx) ;	SIO 编程为 SIO_CAN 功能
<b>void</b> SIO_CAN_Init(SIOx) ;	SIO_CAN 初始化并等待初始化完成。
<b>void</b> SIO_CAN_SetMaskFilter(SIOx, uint32_t u32Index, CAN_IDEnumDef eFrameType, uint32_t u32FilterRTR, uint32_t u32Mask, uint32_t u32Filter );	SIO_CAN 设置掩码和过滤器。 u32Index: 取值范围 0 到 3。 eFrameType: 选择过滤标准/扩展帧。 u32FilterRTR: 选择远程/数据帧过滤。 u32Mask: ID 掩码。 u32Filter: ID 过滤码。 注意: 收取的 ID 需要满足 (u32Mask & ID) == (u32Mask & u32Filter)

函数名称	说明
<code>void SIO_CAN_DisableMaskFilter(SIOx, uint32_t u32Index);</code>	停用过滤功能。 u32Index: 取值范围 0 到 3。
<code>void SIO_CAN_SendMessage(SIOx, CAN_TxFrameTypeDef * pCanTxFrame);</code>	发送 CAN 消息。 pCanTxFrame: 发送帧数据指针。
<code>ErrorStatus SIO_CAN_GetMessage(SIOx, CAN_RxFrameTypeDef * pCanRxFrame);</code>	读取 CAN 消息。 pCanRxFrame: 接收帧数据指针。 读取接收溢出寄存器置位, 则返回 ERROR, 否则返回 SUCCESS。
<code>ErrorStatus SIO_CAN_PushTxMessage(SIOx, CAN_TxFIFOTypeDef * pCanTxFIFO, CAN_TxFrameTypeDef * pCanTxFrame);</code>	发送消息入列。 pCanTxFIFO: TXFIFO 地址指针。 pCanTxFrame: 发送帧数据指针。 TXFIFO 满则返回 ERROR, 否则返回 SUCCESS。
<code>ErrorStatus SIO_CAN_PopTxMessage(SIOx, CAN_TxFIFOTypeDef * pCanTxFIFO);</code>	发送消息出列。 pCanTxFIFO: TXFIFO 地址指针。 TXFIFO 空则返回 ERROR, 否则返回 SUCCESS。
<code>ErrorStatus SIO_CAN_PushRxMessage(SIOx, CAN_RxFIFOTypeDef * pCanRxFIFO, CAN_RxFrameTypeDef * pCanRxFrame);</code>	读取消息入列。 pCanRxFIFO: RXFIFO 地址指针。 pCanRxFrame: 接收帧数据指针。 RXFIFO 满则返回 ERROR, 否则返回 SUCCESS。
<code>ErrorStatus SIO_CAN_PopRxMessage(SIOx, CAN_RxFIFOTypeDef * pCanRxFIFO, CAN_RxFrameTypeDef * pCanRxFrame);</code>	读取消息出列。 pCanRxFIFO: RXFIFO 地址指针。 pCanRxFrame: 接收帧数据指针。 RXFIFO 空则返回 ERROR, 否则返回 SUCCESS。
<code>void SIO_CAN_IRQHandler(SIOx);</code>	SIO_CAN 函数。
<code>void SIO_CAN_RXCompleteCallback(SIOx);</code>	SIO_CAN 接收完成回调函数。
<code>void SIO_CAN_TXCompleteCallback(SIOx);</code>	SIO_CAN 发送完成回调函数。
<code>SIO_CAN_IsReady(SIOx)</code>	SIO_CAN 初始化完成。 注意, SIO_CAN 启动以及总线自动恢复后需要等待初始化完成。
<code>SIO_CAN_IsRxDone(SIOx)</code>	SIO_CAN 有数据接收到。
<code>SIO_CAN_ClearRxDone(SIOx)</code>	清除数据接收到状态位。
<code>SIO_CAN_IsBusOff(SIOx)</code>	SIO_CAN 掉线。
<code>SIO_CAN_IsRxOverflow(SIOx)</code>	SIO_CAN 接收数据溢出, 该位置位, 已经接收到的数据会被破坏。
<code>SIO_CAN_ClearRxOverflow(SIOx)</code>	清除接收数据溢出状态位。
<code>SIO_CAN_GetRXDLC(SIOx)</code>	读取接收的数据帧中的 DLC。
<code>SIO_CAN_GetRXFilterIndex(SIOx)</code>	读取接收的数据帧匹配的过滤索引。
<code>SIO_CAN_IsTxBusy(SIOx)</code>	SIO_CAN 发送数据忙标志位。
<code>SIO_CAN_IsPassive(SIOx)</code>	SIO_CAN 被动节点状态位。
<code>SIO_CAN_GetTEC(SIOx)</code>	读取 TEC。
<code>SIO_CAN_GetREC(SIOx)</code>	读取 REC。



## 5 代码示例

以 SIO0 为例子，参考 demos 目录下的例程。

SPIN TROL

## 6 寄存器

### 6.1 SIO\_CAN 寄存器表

表 6-1: SIO 模块基地址

外设模块	基地址
SIO0	0x4000B000
SIO1	0x4000C000
SIO2	0x4000D000

注意 对 SPC2168，SIO 的 FIFO 深度为 8，宽度为 16-bits。

表 6-2: SIO\_CAN 模块寄存器表

寄存器	偏移地址	说明	默认值
STS0	0x0	Status Register 0	0x00000000
IDF0L	0x8	ID Filer 0 Low 16 bit Register	0x00000000
TXD0	0xC	Transmit Data Register 0	0x00000000
TXD1	0x10	Transmit Data Register 1	0x00000000
TXD2	0x14	Transmit Data Register 2	0x00000000
TXD3	0x18	Transmit Data Register 3	0x00000000
TXEID0	0x1C	Transmit Extended ID Register 0	0x00000000
TXEID1	0x20	Transmit Extended ID Register 1	0x00000000
TXSID	0x20	Transmit Standard ID Register	0x00000000
CTL	0x24	Control Register	0x00000000
IDF0H	0x28	ID Filer 0 High 16 bit Register	0x00000000
IDM0L	0x2C	ID Mask 0 Low 16 bit Register	0x00000000
IDM0H	0x30	ID Mask 0 High 16 bit Register	0x00000000
IDF1L	0x34	ID Filer 1 Low 16 bit Register	0x00000000
IDF1H	0x38	ID Filer 1 High 16 bit Register	0x00000000
IDM1L	0x3C	ID Mask 1 Low 16 bit Register	0x00000000
IDM1H	0x40	ID Mask 1 High 16 bit Register	0x00000000
RXD0	0x44	Receive Data Register 0	0x00000000
RXD1	0x48	Receive Data Register 1	0x00000000
RXD2	0x4C	Receive Data Register 2	0x00000000
RXD3	0x50	Receive Data Register 3	0x00000000
RXEID0	0x54	Receive Extended ID Register 0	0x00000000
RXEID1	0x58	Receive Extended ID Register 1	0x00000000
RXSID	0x58	Receive Standard ID Register	0x00000000
IDF2L	0x5C	ID Filer 2 Low 16 bit Register	0x00000000
IDF2H	0x60	ID Filer 2 High 16 bit Register	0x00000000
IDM2L	0x64	ID Mask 2 Low 16 bit Register	0x00000000
IDM2H	0x68	ID Mask 2 High 16 bit Register	0x00000000
IDF3L	0x6C	ID Filer 3 Low 16 bit Register	0x00000000
IDF3H	0x70	ID Filer 3 High 16 bit Register	0x00000000
IDM3L	0x74	ID Mask 3 Low 16 bit Register	0x00000000
IDM3H	0x78	ID Mask 3 High 16 bit Register	0x00000000
STS1	0x7C	Status Register 1	0x00002101

## 6.2 SIO\_CAN 寄存器

表 6-3 到表 6-46 列举了 SIO\_CAN 寄存器的所有细节。

表 6-3: Status Register 0 (ST0) Layout

ST0 (Status Register 0) Offset: 0x0 Default: 0x00000000								
Access: SIO_CAN -> ST0.all								
31	30	29	28	27	26	25	24	
RESERVED_31_16								
23	22	21	20	19	18	17	16	
RESERVED_31_16								
15	14	13	12	11	10	9	8	
REC								
7	6	5	4	3	2	1	0	
PASSIVE							TEC	

表 6-4: Status Register 0 (ST0) Field Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:8	REC	RO	0x0	RX error counter
7	PASSIVE	RO	0x0	Node passive error status 0: CAN is in active mode 1: CAN is in passive mode
6:0	TEC	RO	0x0	TX error counter

表 6-5: ID Filter 0 Low 16 bit Register (IDF0L) Layout

IDF0L (ID Filer 0 Low 16 bit) Offset: 0x8 Default: 0x00000000							
Access: SIO_CAN -> IDF0L.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 6-6: ID Filter 0 Low 16 bit Register (IDF0L) Field Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	

**表 6-7: Transmit Data Register 0 (TXDATA0) Layout**

TXD0 (Transmit Data Register 0) Offset: 0xC Default: 0x00000000							
Access: SIO_CAN -> TXD0.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
BYTE1							
7	6	5	4	3	2	1	0
BYTE0							

**表 6-8: Transmit Data Register 0 (TXDATA0) Field Description**

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:8	BYTE1	RW	0x0	Data byte 1
7:0	BYTE0	RW	0x0	Data byte 0

**表 6-9: Transmit Data Register 1 (TXDATA1) Layout**

TXD1 (Transmit Data Register 1) Offset: 0x10 Default: 0x00000000							
Access: SIO_CAN -> TXD1.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
BYTE3							
7	6	5	4	3	2	1	0
BYTE2							

**表 6-10: Transmit Data Register 1 (TXDATA1) Field Description**

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:8	BYTE3	RW	0x0	Data byte 3
7:0	BYTE2	RW	0x0	Data byte 2

表 6-11: Transmit Data Register 2 (TXDATA2) Layout

TXD2 (Transmit Data Register 2) Offset: 0x14 Default: 0x00000000							
Access: SIO_CAN -> TXD2.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
BYTE5							
7	6	5	4	3	2	1	0
BYTE4							

表 6-12: Transmit Data Register 2 (TXDATA2) Field Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:8	BYTE5	RW	0x0	Data byte 5
7:0	BYTE4	RW	0x0	Data byte 4

表 6-13: Transmit Data Register 3 (TXDATA3) Layout

TXD3 (Transmit Data Register 3) Offset: 0x18 Default: 0x00000000							
Access: SIO_CAN -> TXD3.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
BYTE7							
7	6	5	4	3	2	1	0
BYTE6							

表 6-14: Transmit Data Register 3 (TXDATA3) Field Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:8	BYTE7	RW	0x0	Data byte 7
7:0	BYTE6	RW	0x0	Data byte 6

**表 6-15: Transmit Extended ID Register 0 (TXEID0) Layout**

TXEID0 (Transmit Extended ID Register 0) Offset: 0x1C Default: 0x00000000							
Access: SIO_CAN -> TXEID0.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
ID14T0							
7	6	5	4	3	2	1	0
ID14T0							RTR

**表 6-16: Transmit Extended ID Register 0 (TXEID0) Field Description**

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:1	ID14T0	RW	0x0	Transmit extended ID[14:0]
0	RTR	RW	0x0	Transmit extended RTR

**表 6-17: Transmit Extended ID Register 1 (TXEID1) Layout**

TXEID1 (Transmit Extended ID Register 1) Offset: 0x20 Default: 0x00000000							
Access: SIO_CAN -> TXEID1.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
ID28T18							
7	6	5	4	3	2	1	0
ID28T18			SRR	IDE	ID17T15		

**表 6-18: Transmit Extended ID Register 1 (TXEID1) Field Description**

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:5	ID28T18	RW	0x0	Transmit extended ID[28:18]
4	SRR	RW	0x0	Transmit extended RTR
3	IDE	RW	0x0	Transmit extended IDE
2:0	ID17T15	RW	0x0	Transmit extended ID[17:15]

**表 6-19: Transmit Standard ID Register (TXSID) Layout**

TXSID (Transmit Standard ID Register) Offset: 0x20 Default: 0x00000000							
Access: SIO_CAN -> TXSID.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
ID							
7	6	5	4	3	2	1	0
ID			RTR	IDE	RESERVED_2_0		

**表 6-20: Transmit Standard ID Register (TXSID) Field Description**

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:5	ID	RW	0x0	Transmit standard ID[10:0]
4	RTR	RW	0x0	Transmit standard RTR
3	IDE	RW	0x0	Transmit standard IDE
2:0	RESERVED_2_0	RW	0x0	Reserved.



**表 6-21: Control Register (CTL) Layout**

CTL (Control Register) Offset: 0x24 Default: 0x00000000							
Access: SIO_CAN -> CTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
TXRUN	RESERVED_14_4						
7	6	5	4	3	2	1	0
RESERVED_14_4				TXDLC			

**表 6-22: Control Register (CTL) Field Description**

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15	TXRUN	RW	0x0	Transmit engine running control and indicator 0: Write 0 is not permitted. Read 0 indicates the engine is idle and ready to take new message. 1: Write 1 will start the engine, which will transmit the message once the bus is available. NOTE: All information on the transmitting message should be completed before set this bit. Read 1 indicates the message is pending for transmitting.
14:4	RESERVED_14_4	RW	0x0	Reserved.
3:0	TXDLC	RW	0x0	Transmit Data length counter

**表 6-23: ID Filter 0 High 16 bit Register (IDF0H) Layout**

IDF0H (ID Filer 0 High 16 bit) Offset: 0x28 Default: 0x00000000							
Access: SIO_CAN -> IDF0H.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

**表 6-24: ID Filter 0 High 16 bit Register (IDF0H) Filed Description**

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	

**表 6-25: ID Mask 0 Low 16 bit Register (IDM0L) Layout**

IDM0L (ID Mask 0 Low 16 bit) Offset: 0x2C Default: 0x00000000							
Access: SIO_CAN -> IDM0L.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

**表 6-26: ID Mask 0 Low 16 bit Register (IDM0L) Filed Description**

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	

**表 6-27: ID Mask 0 High 16 bit Register (IDM0H) Layout**

IDM0H (ID Mask 0 High 16 bit) Offset: 0x30 Default: 0x00000000							
Access: SIO_CAN -> IDM0H.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

**表 6-28: ID Mask 0 High 16 bit Register (IDM0H) Filed Description**

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	

**表 6-29: ID Filer 1 Low 16 bit Register (IDF1L) Layout**

IDF1L (ID Filer 1 Low 16 bit) Offset: 0x34 Default: 0x00000000							
Access: SIO_CAN -> IDF1L.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

**表 6-30: ID Filer 1 Low 16 bit Register (IDF1L) Filed Description**

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	

**表 6-31: ID Filer 1 High 16 bit Register (IDF1H) Layout**

IDF1H (ID Filer 1 High 16 bit) Offset: 0x38 Default: 0x00000000							
Access: SIO_CAN -> IDF1H.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

**表 6-32: ID Filer 1 High 16 bit Register (IDF1H) Field Description**

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	

**表 6-33: ID Mask 1 Low 16 bit Register (IDM1L) Layout**

IDM1L (ID Mask 1 Low 16 bit) Offset: 0x3C Default: 0x00000000							
Access: SIO_CAN -> IDM1L.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

**表 6-34: ID Mask 1 Low 16 bit Register (IDM1L) Field Description**

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	

**表 6-35: ID Mask 1 High 16 bit Register (IDM1H) Layout**

IDM1H (ID Mask 1 High 16 bit) Offset: 0x40 Default: 0x00000000							
Access: SIO_CAN -> IDM1H.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

**表 6-36: ID Mask 1 High 16 bit Register (IDM1H) Field Description**

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	

**表 6-37: Receive Data Register 0 (RXD0) Layout**

RXD0 (Receive Data Register 0) Offset: 0x44 Default: 0x00000000							
Access: SIO_CAN -> RXD0.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
BYTE1							
7	6	5	4	3	2	1	0
BYTE0							

**表 6-38: Receive Data Register 0 (RXD0) Field Description**

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:8	BYTE1	RW	0x0	Receive byte 1
7:0	BYTE0	RW	0x0	Receive byte 0

表 6-39: Receive Data Register 1 (RXD1) Layout

RXD1 (Receive Data Register 1) Offset: 0x48 Default: 0x00000000							
Access: SIO_CAN -> RXD1.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
BYTE3							
7	6	5	4	3	2	1	0
BYTE2							

表 6-40: Receive Data Register 1 (RXD1) Field Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:8	BYTE3	RW	0x0	Receive byte 3
7:0	BYTE2	RW	0x0	Receive byte 2

表 6-41: Receive Data Register 2 (RXD2) Layout

RXD2 (Receive Data Register 2) Offset: 0x4C Default: 0x00000000							
Access: SIO_CAN -> RXD2.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
BYTE5							
7	6	5	4	3	2	1	0
BYTE4							

表 6-42: Receive Data Register 2 (RXD2) Field Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:8	BYTE5	RW	0x0	Receive byte 5
7:0	BYTE4	RW	0x0	Receive byte 4

**表 6-43: Receive Data Register 3 (RXD3) Layout**

RXD3 (Receive Data Register 3) Offset: 0x50 Default: 0x00000000							
Access: SIO_CAN -> RXD3.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
BYTE7							
7	6	5	4	3	2	1	0
BYTE6							

**表 6-44: Receive Data Register 3 (RXD3) Field Description**

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:8	BYTE7	RW	0x0	Receive byte 7
7:0	BYTE6	RW	0x0	Receive byte 6

**表 6-45: Receive Extended ID Register 0 (RXEID0) Layout**

RXEID0 (Receive Extended ID Register 0) Offset: 0x54 Default: 0x00000000							
Access: SIO_CAN -> RXEID0.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
ID14T0							
7	6	5	4	3	2	1	0
ID14T0							RTR

**表 6-46: Receive Extended ID Register 0 (RXEID0) Field Description**

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:1	ID14T0	RW	0x0	Receive extended ID[14:0]
0	RTR	RW	0x0	Receive extended RTR

**表 6-47: Receive Extended ID Register 1 (RXEID1) Layout**

RXEID1 (Receive Extended ID Register 1) Offset: 0x58 Default: 0x00000000							
Access: SIO_CAN -> RXEID1.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
ID28T18							
7	6	5	4	3	2	1	0
ID28T18			SRR	IDE	ID17T15		

**表 6-48: Receive Extended ID Register 1 (RXEID1) Field Description**

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:5	ID28T18	RW	0x0	Receive extended ID[28:18]
4	SRR	RW	0x0	Receive extended RTR
3	IDE	RW	0x0	Receive extended IDE
2:0	ID17T15	RW	0x0	Receive extended ID[17:15]

**表 6-49: Receive Standard ID Register (RXSID) Layout**

RXSID (Receive Standard ID Register) Offset: 0x58 Default: 0x00000000							
Access: SIO_CAN -> RXSID.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
ID							
7	6	5	4	3	2	1	0
ID			RTR	IDE	RESERVED_2_0		

**表 6-50: Receive Standard ID Register (RXSID) Field Description**

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:5	ID	RW	0x0	Receive standard ID[10:0]
4	RTR	RW	0x0	Receive standard RTR
3	IDE	RW	0x0	Receive extended IDE
2:0	RESERVED_2_0	RO	0x0	Reserved.



**表 6-51: ID Filer 2 Low 16 bit Register (IDF2L) Layout**

IDF2L (ID Filer 2 Low 16 bit) Offset: 0x5C Default: 0x00000000							
Access: SIO_CAN -> IDF2L.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

**表 6-52: ID Filer 2 Low 16 bit Register (IDF2L) Field Description**

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	

**表 6-53: ID Filer 2 High 16 bit Register (IDF2H) Layout**

IDF2H (ID Filer 2 High 16 bit) Offset: 0x60 Default: 0x00000000							
Access: SIO_CAN -> IDF2H.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

**表 6-54: ID Filer 2 High 16 bit Register (IDF2H) Field Description**

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	

**表 6-55: ID Mask 2 Low 16 bit Register (IDM2L) Layout**

IDM2L (ID Mask 2 Low 16 bit) Offset: 0x64 Default: 0x00000000							
Access: SIO_CAN -> IDM2L.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

**表 6-56: ID Mask 2 Low 16 bit Register (IDM2L) Field Description**

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	

**表 6-57: ID Mask 2 High 16 bit Register (IDM2H) Layout**

IDM2H (ID Mask 2 High 16 bit) Offset: 0x68 Default: 0x00000000							
Access: SIO_CAN -> IDM2H.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

**表 6-58: ID Mask 2 High 16 bit Register (IDM2H) Field Description**

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	

**表 6-59: ID Filer 3 Low 16 bit Register (IDF3L) Layout**

IDF3L (ID Filer 3 Low 16 bit) Offset: 0x6C Default: 0x00000000							
Access: SIO_CAN -> IDF3L.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

**表 6-60: ID Filer 3 Low 16 bit Register (IDF3L) Field Description**

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	

**表 6-61: ID Filer 3 High 16 bit Register (IDF3H) Layout**

IDF3H (ID Filer 3 High 16 bit) Offset: 0x70 Default: 0x00000000							
Access: SIO_CAN -> IDF3H.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

**表 6-62: ID Filer 3 High 16 bit Register (IDF3H) Field Description**

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	

**表 6-63: ID Mack 3 Low 16 bit Register (IDM3L) Layout**

IDM3L (ID Mask 3 Low 16 bit) Offset: 0x74 Default: 0x00000000							
Access: SIO_CAN -> IDM3L.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

**表 6-64: ID Mack 3 Low 16 bit Register (IDM3L) Field Description**

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	

**表 6-65: ID Mack 3 High 16 bit Register (IDM3H) Layout**

IDM3H (ID Mask 3 High 16 bit) Offset: 0x78 Default: 0x00000000							
Access: SIO_CAN -> IDM3H.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

**表 6-66: ID Mack 3 High 16 bit Register (IDM3H) Field Description**

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	

**表 6-67: Status Register 1 (STS1) Layout**

STS1 (Status Register 1) Offset: 0x7C Default: 0x00002101							
Access: SIO_CAN -> STS1.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
RXDONE	BUSOFF	RDY	RXOVLD	RESERVED_11_8			
7	6	5	4	3	2	1	0
RXDLC				MFI		RESERVED_1_0	

**表 6-68: Status Register 1 (STS1) Field Description**

Bits	Field Name	Type	Reset	Description
15	RXDONE	RW	0x0	RX complete receive a message 0: Read 0 indicates the data in the receive registers are not valid. Write 0 will clear the bit. 1: Read 1 indicates the engine has received a complete message. Write 1 is not permitted.
14	BUSOFF	RO	0x0	CAN bus off indicator. Once CAN TEC reaches its limit of 256, the CAN will enter BusOff mode and BusOff indicator will become 1. The CAN will monitor the bus and once it sees 128 occurrences of 11 consecutive 'recessive' bot, the error counter will be reset to 0 and it goes back to active mode and the indicator will become 0. 0: Bus is normal. 1: Bus off status
13	RDY	RO	0x1	CAN initial ready 0: CAN is not ready. 1: CAN is ready.
12	RXOVLD	RO	0x0	Receive new frame when RXDONE is 1 0: RX message box is not corrupted. 1: RX message box is corrupted.
11:8	RESERVED_11_8	RW	0x1	Reserved.
7:4	RXDLC	RO	0x0	Receive data length counter
3:2	MFI	RO	0x0	Matched filter index
1:0	RESERVED_1_0	RW	0x1	Reserved.