HT9316 数据采集卡使用手册

Ver 1.0

西安方解石信息技术有限责任公司

Revision

时间	原因	版本	作者	备注
2019-10-15	V1.0 版本生成	V1.0	冯江宏	建立文档
2019-12-20	V1.1 版本生成	V1.1	冯江宏	

1. 概述

1.1. 产品描述

HT9316-PCI-T1R2 是一款包含 1 路发送 2 路接收配置的 ARINC9316 通讯板卡，其功能能够满足用户的工业测量和自动化控制需求，良好的兼容性适用于各类系统配置。

1.2. 特性

- 33MHz/32Bit PCI 接口；
- 最大 4 路发送通道， 每路带 （16M-1）×32 位 FIFO；
- 最大 8 路接收通道，每路带（4M-1）×32 位 FIFO；
- 发送和接收 FIFO 复位；
- 具有内外同步时钟触发功能；
- 在内外触发时随时更新数据；
- 触发具有可设的消息间隔、字间隔和发送帧的预定数量；
- 添加时间标签功能；
- 接收标号过滤；
- 接收 FIFO 触发深度可设功能；
- 波特率 100Kbps、 50Kbps、 48Kbps、 12.5Kbps 可设置；
- 标准 ARINC9316 字格式转换与否可选择；
- 支持包模式发送与接收功能。

1.3. 详细描述

1.3.1. 发送通道

- 字模式下，发送缓冲区可存放（16M-1）个 32 位数据；包模式下，发送缓冲区可存放 4095 个数据包，每个数据包最大 4095 个 32 位数据；
- 可支持外部触发、内部触发、软件触发三个触发源；
- 支持高电平触发、低电平触发、上升沿触发、下降沿触发、双沿触发；
- 触发延迟可设置，分辨率为 1us；
- 触发数据个数可设置；
- 发送数据间隔可设置；
- 包模式下，可设置包与包之间的间隔，每个发送包独立设置，分辨率为 1us；
- 发送波特率 100Kbps、 50Kbps、 48Kbps、 12.5Kbps 可设置；
- 标准 ARINC9316 字格式转换与否可选择。

1.3.2. 接收通道

- 字模式下每路接收 FIFO 大小（4M-1）×32 位；包模式下，能缓存的数据包最大个数为 1023，每个数据包能够能够存取 4095 个 32 位；
- 包模式下，包间隔判定可设，分辨率为 1us；
- 接收使能和接收 FIFO 复位功能；
- 接收标号过滤功能；
- 接收添加时标功能；
- 接收波特率 100Kbps、 50Kbps、 48Kbps、 12.5Kbps 可设置；
- 标准 ARINC9316 字格式转换与否可选择。

1.4. 一般规格

- 物理尺寸：长×宽： 174.63×106.68mm
- 连 接 器： SCSI-68CN
- 工作电源： 5V
- 相对湿度： 5～95%，无凝结

1.5. 产品安装

1.5.1. 硬件安装

第一步： 打开板卡的防静电包装袋，取出板卡。
第二步： 关闭计算机设备的电源，将板卡安装到您的计算机机箱内。
第三步： 将配套的连接器或连接电缆插到板卡的连接器接口上。
关于连接电缆的制作请参照节 2.3 信号定义的内容。
开启计算机，系统提示发现新硬件，然后安装产品的驱动

1.5.2. 驱动安装

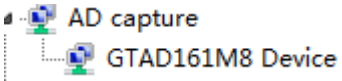
驱动程序包含下列文件：

驱动安装步骤如下：
(1) 双击安装程序：

(2) 点击“Install”，安装驱动程序。安装过程中若出现以下显示：

选择“始终安装此驱动程序软件”，等待安装程序完成安装。

(3) 驱动安装完成后，可以在设备管理器中看到如下图所示：



至此，设备可以正常使用。

1.5.3. SDK 文件

SDK 库文件包含下列文件：
gtad161m8api_win32.dll
gtad161m8api_win32.lib
这两个文件用于应用程序开发。

2. 硬件说明

本章描述了 HT9316-PCI-T1R2 板卡硬件信息，包括硬件设置及信号定义等。

2.1. 功能结构框图

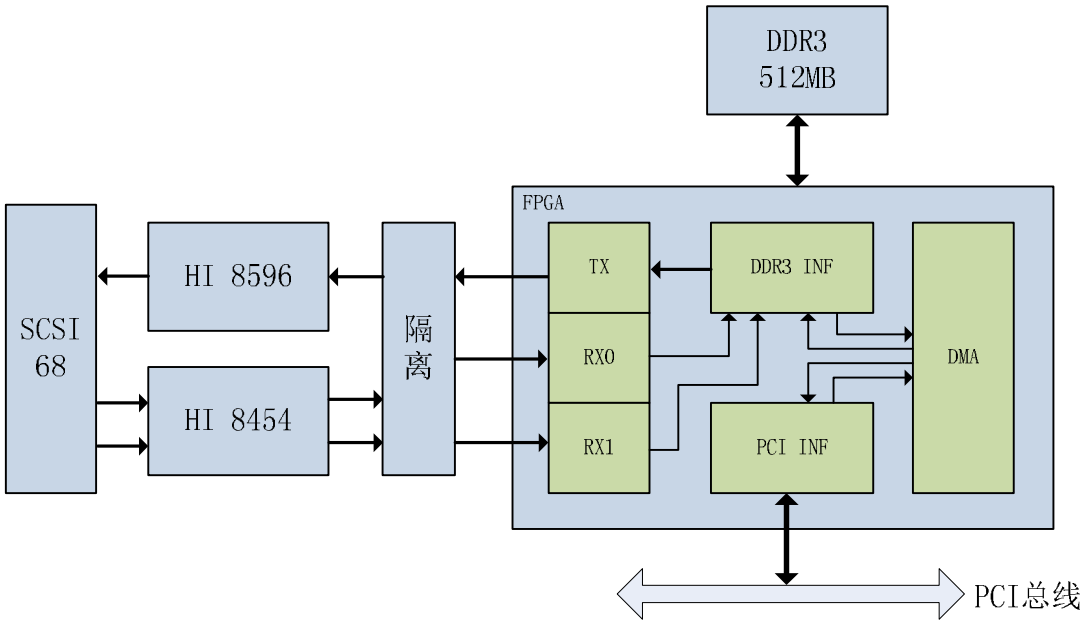


图 2-1 HT9316-PCI-T1R2 功能结构框图

2.2. 印制板示意图

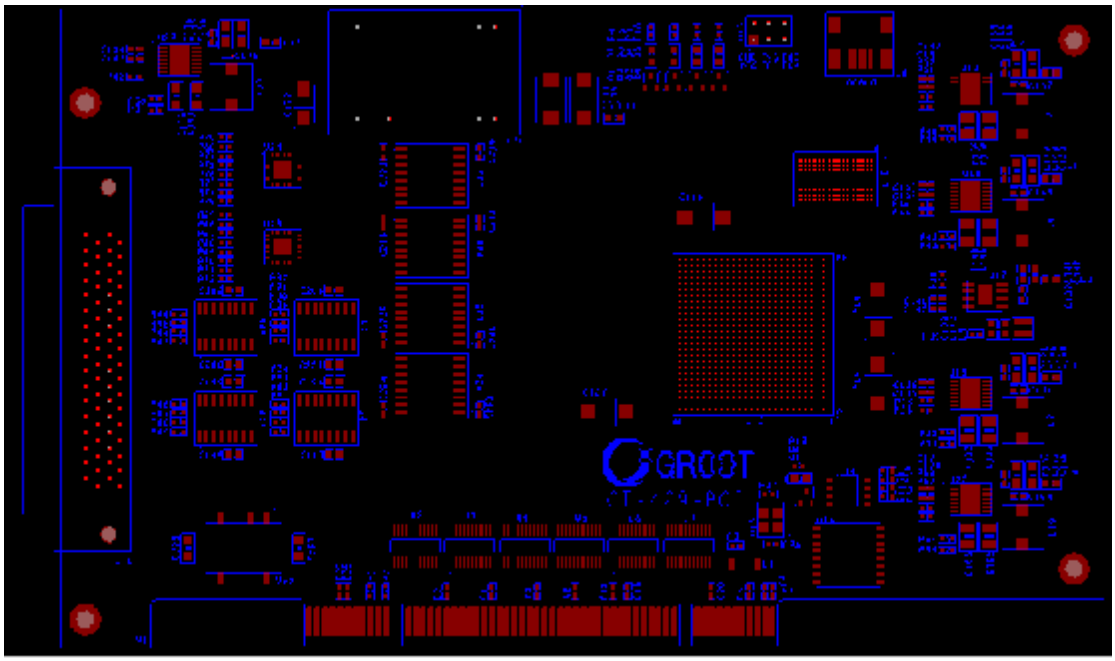


图 2-2 HT9316-PCI-T1R2 印制板示意图

图 2-3 HT9316-PCI-T1R2 实物图

2.3. 连接器和信号定义

连接器为标准 SCSI-68CN 型接头。

表 2-1 SCSI-68 信号定义表

引脚	信号名	信号说明	引脚	信号名	信号说明
1	TRI_IN_0	外部触发 0	35	TRI_IN_2	外部触发 2
2	TRI_IN_1	外部触发 1	36	TRI_IN_3	外部触发 3
3	GND_TRI	触发参考地	37	GND_TRI	触发参考地
4	GND_TRI	触发参考地	38	GND_TRI	触发参考地
5			39		
6			40		
7			41		
8			42		
9	GND_9316	9316 参考地	43	GND_9316	9316 参考地
10	GND_9316	9316 参考地	44	GND_9316	9316 参考地
11	9316_TXB_0	9316 CH0 发送负	45		
12	9316_TXA_0	9316 CH0 发送正	46		
13	GND_9316	9316 参考地	47	GND_9316	9316 参考地
14	GND_9316	9316 参考地	48	GND_9316	9316 参考地
15			49		
16			50		
17	GND_9316	9316 参考地	51	GND_9316	9316 参考地
18	GND_9316	9316 参考地	52	GND_9316	9316 参考地
19	9316_RXB_0	9316 CH0 接收负	53	9316_RXB_1	9316 CH1 接收负
20	9316_RXA_0	9316 CH0 接收正	54	9316_RXA_1	9316 CH1 接收正
21	GND_9316	9316 参考地	55	GND_9316	9316 参考地
22	GND_9316	9316 参考地	56	GND_9316	9316 参考地
23			57		
24			58		
25	GND_9316	9316 参考地	59	GND_9316	9316 参考地
26	GND_9316	9316 参考地	60	GND_9316	9316 参考地
27			61		
28			62		
29	GND_9316	9316 参考地	63	GND_9316	9316 参考地
30	GND_9316	9316 参考地	64	GND_9316	9316 参考地
31			65		
32			66		
33	GND_9316	9316 参考地	67	GND_9316	9316 参考地
34	GND_9316	9316 参考地	68	GND_9316	9316 参考地

3. 宏定义

3.1. AD 最大通道数

```
#define MAX_AD_CHANNEL_NUM        8
```

3.2. 函数返回值

Api 函数返回值统一使用 GT9316_STATUS 枚举值定义。
返回值定义如下：
GT9316_SUCCESS
成功
GT9316_OS_ERROR
系统错误
GT9316_LOW_MEMORY
内存不足
GT9316_INVALID_MSG_ID
无效的消息 ID
GT9316_BAD_PARAMETER_1
GT9316_BAD_PARAMETER_2
GT9316_BAD_PARAMETER_3
GT9316_BAD_PARAMETER_4
GT9316_BAD_PARAMETER_5
GT9316_BAD_PARAMETER_6
GT9316_BAD_PARAMETER_7
GT9316_BAD_PARAMETER_8
GT9316_BAD_PARAMETER_9
无效的参数 1...9
GT9316_NULL_DESCRIPTOR
空描述符
GT9316_NO_BOARD
板卡打开失败
GT9316_NOT_OPEN
板卡未打开
GT9316_READ_TIMEOUT
读操作超时
GT9316_WRITE_TIMEOUT
写操作超时
GT9316_SIGNALED
收到系统信号

4. 枚举

4.1. 触发源定义

```
typedef enum {
    EN_TRIGGER_SRC_SOFTWARE = 0,
    EN_TRIGGER_SRC_IO_0,
    EN_TRIGGER_SRC_IO_1,
    EN_TRIGGER_SRC_IO_2,
    EN_TRIGGER_SRC_IO_3,
    EN_TRIGGER_SRC_IO_4,
    EN_TRIGGER_SRC_IO_5,
    EN_TRIGGER_SRC_IO_6,
    EN_TRIGGER_SRC_IO_7,
    EN_TRIGGER_SRC_IO_8,
    EN_TRIGGER_SRC_IO_9,
    EN_TRIGGER_SRC_IO_10,
    EN_TRIGGER_SRC_IO_11,
    EN_TRIGGER_SRC_IO_12,
    EN_TRIGGER_SRC_IO_13,
    EN_TRIGGER_SRC_IO_14,
    EN_TRIGGER_SRC_IO_15,
    EN_TRIGGER_SRC_IO_16,
    EN_TRIGGER_SRC_IO_17,
    EN_TRIGGER_SRC_IO_18,
    EN_TRIGGER_SRC_IO_19,
    EN_TRIGGER_SRC_IO_20,
    EN_TRIGGER_SRC_IO_21,
    EN_TRIGGER_SRC_IO_22,
    EN_TRIGGER_SRC_IO_23,
    EN_TRIGGER_SRC_IO_24,
    EN_TRIGGER_SRC_IO_25,
    EN_TRIGGER_SRC_IO_26,
    EN_TRIGGER_SRC_IO_27,
    EN_TRIGGER_SRC_IO_28,
    EN_TRIGGER_SRC_IO_29,
    EN_TRIGGER_SRC_IO_30,
    EN_TRIGGER_SRC_IO_31,
    EN_TRIGGER_SRC_ACMP_0,
    EN_TRIGGER_SRC_ACMP_1,
    EN_TRIGGER_SRC_ACMP_2,
    EN_TRIGGER_SRC_ACMP_3,
    EN_TRIGGER_SRC_ACMP_4,
```

```
    EN_TRIGGER_SRC_ACMP_5,
    EN_TRIGGER_SRC_ACMP_6,
    EN_TRIGGER_SRC_ACMP_7,

    EN_TRIGGER_SRC_BOTTOM,
}EN_TRIGGER_SRC;
```

成员定义

EN_TRIGGER_SRC_SOFTWARE：软件触发，立即触发，设置了软件触发之后，软件启动转换立即触发数据采集操作。
EN_TRIGGER_IO_0~31：IO 触发，对应 I0 管脚接收到对应触发模式信号后，触发数据采集。
EN_TRIGGER_ACMP_0~7：模拟通道比较触发。上升沿高于比较电平，下降沿低于比较电平触发数据采集。

4.2. 触发模式枚举定义

```
typedef enum {
    EN_TRIGGER_MODE_HIGHLEVEL,              //高电平触发
    EN_TRIGGER_MODE_LOWLEVEL,           //低电平触发
    EN_TRIGGER_MODE_POSEDGE,                //上升沿触发
    EN_TRIGGER_MODE_NEGEDGE,                //下降沿触发
    EN_TRIGGER_MODE_BOTHEDGE,               //双沿触发
    EN_TRIGGER_MODE_BOTTOM,
}EN_TRIGGER_MODE;
```

成员定义

EN_TRIGGER_MODE_HIGHLEVEL：高电平触发
EN_TRIGGER_MODE_LOWLEVEL：低电平触发
EN_TRIGGER_MODE_POSEDGE：上升沿触发
EN_TRIGGER_MODE_NEGEDGE：下降沿触发
EN_TRIGGER_MODE_BOTHEDGE：双沿触发

4.3. 转换范围枚举定义

```
typedef enum {
    EN_RANGE_NP5 = 0,
    EN_RANGE_NP10,
    EN_RANGE_NP6,
    EN_RANGE_NP12,
}EN_ADC_RANGE;
```

成员定义

EN_RANGE_NP5：±5V 转换范围
EN_RANGE_NP10：±10V 转换范围
EN_RANGE_NP6：±6V 转换范围
EN_RANGE_NP12：±12V 转换范围

5. API 说明

5.1. 通用接口

5.1.1. GT9316_OpenCard

原型

```
STDGT9316CALL GT9316_OpenCard(GT9316HANDLE  * ph, GT9316_UINT8 CardId)
```

功能

打开板卡，并分配板卡资源。

参数说明

输入参数

ph：板卡句柄指针
CardId：板卡编号，取值为 1~255。

输出参数

无。

返回值

成功时返回 GT9316_SUCCESS。
失败时可能返回的返回值列表：
GT9316_BAD_PARAMETER_1
GT9316_LOW_MEMORY
GT9316_NOT_OPEN

Code example

```
    GT9316_STATUS status = GT9316_SUCCESS;
    GT9316HANDLE ph = NULL;
    status = GT9316_OpenCard(&ph, 1);
    if (status != GT9316_SUCCESS) {
        debug_printf("GT9316_OpenCard failed. Return value = %d\r\n", status);
        return status;
    }
```

5.1.2. GT9316_CloseCard

原型

```
STDGT9316CALL GT9316_CloseCard(GT9316HANDLE * ph)
```

功能

关闭板卡，并释放打开板卡时申请的资源。

参数说明

输入参数

ph：输入参数，板卡句柄指针

输出参数

无

返回值

成功时返回 GT9316_SUCCESS。

失败时可能返回的返回值列表：

GT9316_BAD_PARAMETER_1

GT9316_NULL_DESCRIPTOR

GT9316_NO_BOARD

GT9316_OS_ERROR

Code example

```
    GT9316_STATUS status = GT9316_SUCCESS;
    status = GT9316_CloseCard(ph);
    if (status != GT9316_SUCCESS) {
        debug_printf("GT9316_CloseCard failed. Return value = %d\r\n", status);
        return status;
    }
```

5.1.3. GT9316_ResetCard

原型

```
STDGT9316CALL GT9316_ResetCard(GT9316HANDLE ph)
```

功能

复位板卡。

参数说明

输入参数

ph：板卡句柄，由函数 GT9316_OpenCard 得到。

输出参数

无。

返回值

成功时返回 GT9316_SUCCESS。

失败时可能返回的返回值列表：

GT9316_BAD_PARAMETER_1

GT9316_NULL_DESCRIPTOR

GT9316_NO_BOARD

GT9316_OS_ERROR

函数也可能返回其它错误码

Code example

```
    GT9316_STATUS status = GT9316_SUCCESS;
```

```
    status = GT9316_ResetCard(ph);
    if (status != GT9316_SUCCESS) {
        debug_printf("GT9316_ResetCard failed. Return value = %d\r\n", status);
        return status;
    }
```

5.2. ADC 接口

5.2.1. GT9316_AdcSetEnable

原型

```
STDGT9316CALL GT9316_AdcSetEnable(GT9316HANDLE ph, GT9316_BOOL enable)
```

功能

设置 ADC 是否使能。

参数说明

输入参数

ph：板卡句柄，由函数 GT9316_OpenCard 得到。
enable：GT9316_TRUE -使能 ADC，GT9316_FALSE-不使能 ADC。

输出参数

无

返回值:

成功时返回 GT9316_SUCCESS。
失败时可能返回的返回值列表：
GT9316_BAD_PARAMETER_1
GT9316_NULL_DESCRIPTOR
GT9316_NO_BOARD
GT9316_OS_ERROR
函数也可能返回其它错误码。

Code example

```
GT9316_STATUS status = GT9316_SUCCESS;
status = GT9316_AdcSetEnable(ph, GT9316_TRUE);
if (status != GT9316_SUCCESS) {
    debug_printf("GT9316_AdcSetEnable failed, return status = 0x%08x!!!\r\n", status);
    return status;
}
```

5.2.2. GT9316_AdcGetEnable

原型

```
STDGT9316CALL GT9316_AdcGetEnable(GT9316HANDLE ph, GT9316_BOOL * enable)
```

功能

获取 ADC 是否使能。

参数说明

输入参数

ph：板卡句柄，由函数 GT9316_OpenCard 得到。

输出参数

enable：GT9316_TRUE - ADC 使能， GT9316_FALSE - ADC 不使能。

返回值

成功时返回 GT9316_SUCCESS。

失败时可能返回的返回值列表：

GT9316_BAD_PARAMETER_1

GT9316_BAD_PARAMETER_2

GT9316_NULL_DESCRIPTOR

GT9316_NO_BOARD

GT9316_OS_ERROR

Code example

```
GT9316_STATUS status = GT9316_SUCCESS;
GT9316_BOOL enable;
status = GT9316_AdcGetEnable(ph, &enable);
if (status != GT9316_SUCCESS) {
    debug_printf("GT9316_AdcGetEnable, return status = 0x%08x!!!\r\n", status);
    return status;
}
```

5.2.3. GT9316_AdcStart

原型

```
STDGT9316CALL GT9316_AdcStart(GT9316HANDLE ph)
```

功能

启动 ADC 采集。

参数说明

输入参数

ph：板卡句柄，由函数 GT9316_OpenCard 得到。

输出参数

无。

返回值

成功时返回 GT9316_SUCCESS。

失败时可能返回的返回值列表：

GT9316_BAD_PARAMETER_1

GT9316_NULL_DESCRIPTOR
GT9316_NO_BOARD
GT9316_OS_ERROR
函数也可能返回其它错误码

Code example

```
GT9316_STATUS status = GT9316_SUCCESS;
status = GT9316_AdcStart(ph);
if (status != GT9316_SUCCESS) {
    debug_printf("GT9316_AdcStart failed. Return value = %d\r\n", status);
    return status;
}
```

5.2.4. GT9316_AdcStop

原型

```
STDGT9316CALL GT9316_AdcStop(GT9316HANDLE ph)
```

功能

停止 ADC 采集。

参数说明

输入参数

ph：板卡句柄，由函数 GT9316_OpenCard 得到。

输出参数

无。

返回值

成功时返回 GT9316_SUCCESS。
失败时可能返回的返回值列表：
GT9316_BAD_PARAMETER_1
GT9316_NULL_DESCRIPTOR
GT9316_NO_BOARD
GT9316_OS_ERROR
函数也可能返回其它错误码

Code example

```
GT9316_STATUS status = GT9316_SUCCESS;
status = GT9316_AdcStop(ph);
if (status != GT9316_SUCCESS) {
    debug_printf("GT9316_AdcStop failed. Return value = %d\r\n", status);
    return status;
}
```

5.2.5. GT9316_AdcGetRunningState

原型

```
STDGT9316CALL GT9316_AdcGetRunningState(GT9316HANDLE ph, GT9316_BOOL * state)
```

功能

ADC 获取当前运行状态。

参数说明

输入参数

ph：板卡句柄，由函数 GT9316_OpenCard 得到。

输出参数

无。

返回值

成功时返回 GT9316_SUCCESS。

失败时可能返回的返回值列表：

GT9316_BAD_PARAMETER_1

GT9316_BAD_PARAMETER_2

GT9316_NULL_DESCRIPTOR

GT9316_NO_BOARD

GT9316_OS_ERROR

函数也可能返回其它错误码

Code example

```
GT9316_STATUS status = GT9316_SUCCESS;
GT9316_BOOL enable;
status = GT9316_AdcGetRunningState(ph, &enable);
if (status != GT9316_SUCCESS) {
    debug_printf("GT9316_AdcGetRunningState failed. Return value = %d\r\n", status);
    return status;
}
```

5.2.6. GT9316_AdcSetRange

原型

```
STDGT9316CALL GT9316_AdcSetRange(GT9316HANDLE ph, EN_ADC_RANGE range)
```

功能

设置 ADC 的转换范围。

参数说明

输入参数

ph：板卡句柄，由函数 GT9316_OpenCard 得到。
range：转换范围，参见 EN_ADC_RANGE 定义。

输出参数

无。

返回值

成功时返回 GT9316_SUCCESS。
失败时可能返回的返回值列表：
GT9316_BAD_PARAMETER_1
GT9316_BAD_PARAMETER_2
GT9316_NULL_DESCRIPTOR
GT9316_NO_BOARD
GT9316_OS_ERROR
函数也可能返回其它错误码

Code example

```
GT9316_STATUS status = GT9316_SUCCESS;
status = GT9316_AdcSetRange(ph, EN_RANGE_NP5);
if (status != GT9316_SUCCESS) {
    debug_printf("GT9316_AdcSetRange failed. Return value = %d\r\n", status);
    return status;
}
```

5.2.7. GT9316_AdcGetRange

原型

```
STDGT9316CALL GT9316_AdcGetRange(GT9316HANDLE ph, EN_ADC_RANGE * range)
```

功能

获取 ADC 当前设置的转换范围。

参数说明

输入参数

ph：板卡句柄，由函数 GT9316_OpenCard 得到。

输出参数

range：转换范围，参见 EN_ADC_RANGE 定义。

返回值

成功时返回 GT9316_SUCCESS。
失败时可能返回的返回值列表：
GT9316_BAD_PARAMETER_1
GT9316_BAD_PARAMETER_2
GT9316_NULL_DESCRIPTOR

GT9316_NO_BOARD
GT9316_OS_ERROR
函数也可能返回其它错误码

Code example

```
GT9316_STATUS status = GT9316_SUCCESS;
EN_ADC_RANGE range;
status = GT9316_AdcGetRange(ph, &range);
if (status != GT9316_SUCCESS) {
    debug_printf("GT9316_AdcGetRange failed. Return value = %d\r\n", status);
    return status;
}
```

5.2.8. GT9316_AdcSetSampleRate

原型

```
STDGT9316CALL GT9316_AdcSetSampleRate(GT9316HANDLE ph, GT9316_UINT16 div)
```

功能

设置 ADC 的采样率分频因子, ADC 主时钟使用 23M，分频因子设置范围为 22~65535， 采样率为 23M/(div+1), div = 22 时， 采样率为 1M。

参数说明

输入参数

ph：板卡句柄，由函数 GT9316_OpenCard 得到。
div：分频因子。

输出参数

无。

返回值

成功时返回 GT9316_SUCCESS。
失败时可能返回的返回值列表：
GT9316_BAD_PARAMETER_1
GT9316_BAD_PARAMETER_2
GT9316_NULL_DESCRIPTOR
GT9316_NO_BOARD
GT9316_OS_ERROR
函数也可能返回其它错误码

Code example

```
GT9316_STATUS status = GT9316_SUCCESS;
status = GT9316_AdcSetSampleRate(ph, 22);
if (status != GT9316_SUCCESS) {
    debug_printf("GT9316_AdcSetSampleRate failed. Return value = %d\r\n", status);
    return status;
}
```

5.2.9. GT9316_AdcGetSampleRate

原型

```
STDGT9316CALL GT9316_AdcGetSampleRate(GT9316HANDLE ph, GT9316_UINT16 * div)
```

功能

获取 ADC 采样率分频因子。

参数说明

输入参数

ph：板卡句柄，由函数 GT9316_OpenCard 得到。

输出参数

div：分频因子。

返回值

成功时返回 GT9316_SUCCESS。

失败时可能返回的返回值列表：

GT9316_BAD_PARAMETER_1

GT9316_BAD_PARAMETER_2

GT9316_NULL_DESCRIPTOR

GT9316_NO_BOARD

GT9316_OS_ERROR

函数也可能返回其它错误码

Code example

```
GT9316_STATUS status = GT9316_SUCCESS;
GT9316_UINT16 div;
status = GT9316_AdcGetSampleRate(ph, &div);
if (status != GT9316_SUCCESS) {
    debug_printf("GT9316_AdcGetSampleRate failed. Return value = %d\r\n", status);
    return status;
}
```

5.2.10. GT9316_AdcSetTriggerSrc

原型

```
STDGT9316CALL GT9316_AdcSetTriggerSrc(GT9316HANDLE ph, EN_TRIGGER_SRC src)
```

功能

设置 ADC 触发源。

参数说明

输入参数

ph：板卡句柄，由函数 GT9316_OpenCard 得到。

src：触发源，定义参见 EN_TRIGGER_SRC。

输出参数

无

返回值:

成功时返回 GT9316_SUCCESS。

失败时可能返回的返回值列表:

GT9316_BAD_PARAMETER_1

GT9316_NULL_DESCRIPTOR

GT9316_NO_BOARD

GT9316_OS_ERROR

函数也可能返回其它错误码。

Code example

```
GT9316_STATUS status = GT9316_SUCCESS;
status = GT9316_AdcSetTriggerSrc(ph, EN_TRIGGER_SRC_SOFTWARE);
if (status != GT9316_SUCCESS) {
    debug_printf("GT9316_AdcSetTriggerSrc failed, return status = 0x%08x!!!\r\n", status);
    return status;
}
```

5.2.11. GT9316_AdcGetTriggerSrc

原型

```
STDGT9316CALL GT9316_AdcGetTriggerSrc(GT9316HANDLE ph, EN_TRIGGER_SRC * src)
```

功能

获取当前 ADC 触发源。

参数说明

输入参数

ph：板卡句柄，由函数 GT9316_OpenCard 得到。

输出参数

src：当前 ADC 触发源。

返回值

成功时返回 GT9316_SUCCESS。

失败时可能返回的返回值列表:

GT9316_BAD_PARAMETER_1

GT9316_BAD_PARAMETER_2

GT9316_NULL_DESCRIPTOR

GT9316_NO_BOARD

GT9316_OS_ERROR

Code example

```
GT9316_STATUS status = GT9316_SUCCESS;
EN_TRIGGER_SRC src;
status = GT9316_AdcGetTriggerSrc(ph, &src);
```

```
if (status != GT9316_SUCCESS) {
    debug_printf("GT9316_AdcGetTriggerSrc, return status = 0x%08x!!!\r\n", status);
    return status;
}
```

5.2.12. GT9316_AdcSetTriggerMode

原型

```
STDGT9316CALL GT9316_AdcSetTriggerMode(GT9316HANDLE ph, EN_TRIGGER_MODE mode)
```

功能

设置 ADC 触发模式，如果触发源为软件触发，则忽略触发模式。

参数说明

输入参数

ph：板卡句柄，由函数 GT9316_OpenCard 得到。
mode：触发模式，参见 `EN_TRIGGER_MODE` 定义。

输出参数

无

返回值:

成功时返回 GT9316_SUCCESS。
失败时可能返回的返回值列表：
GT9316_BAD_PARAMETER_1
GT9316_NULL_DESCRIPTOR
GT9316_NO_BOARD
GT9316_OS_ERROR
函数也可能返回其它错误码。

Code example

```
GT9316_STATUS status = GT9316_SUCCESS;
status = GT9316_AdcSetTriggerMode(ph, EN_TRIGGER_MODE_HIGHLEVEL);
if (status != GT9316_SUCCESS) {
    debug_printf("GT9316_AdcSetTriggerMode failed, return status = 0x%08x!!!\r\n", status);
    return status;
}
```

5.2.13. GT9316_AdcGetTriggerMode

原型

```
STDGT9316CALL GT9316_AdcGetTriggerMode(GT9316HANDLE ph, EN_TRIGGER_MODE * mode)
```

功能

获取 ADC 触发模式。

参数说明

输入参数

ph：板卡句柄，由函数 GT9316_OpenCard 得到。

输出参数

mode：ADC 触发模式。

返回值

成功时返回 GT9316_SUCCESS。

失败时可能返回的返回值列表：

GT9316_BAD_PARAMETER_1

GT9316_BAD_PARAMETER_2

GT9316_NULL_DESCRIPTOR

GT9316_NO_BOARD

GT9316_OS_ERROR

Code example

```
GT9316_STATUS status = GT9316_SUCCESS;
EN_TRIGGER_MODE mode;
status = GT9316_AdcGetTriggerMode(ph, &mode);
if (status != GT9316_SUCCESS) {
    debug_printf("GT9316_AdcGetTriggerMode, return status = 0x%08x!!!\r\n", status);
    return status;
}
```

5.2.14. GT9316_AdcSetTriggerDelay

原型

```
STDGT9316CALL GT9316_AdcSetTriggerDelay(GT9316HANDLE ph, GT9316_UINT32 delay)
```

功能

设置 ADC 触发延时。

参数说明

输入参数

ph：板卡句柄，由函数 GT9316_OpenCard 得到。

delay：延迟值，单位 1 us。

输出参数

无

返回值:

成功时返回 GT9316_SUCCESS。

失败时可能返回的返回值列表：

GT9316_BAD_PARAMETER_1

GT9316_NULL_DESCRIPTOR

GT9316_NO_BOARD

GT9316_OS_ERROR
函数也可能返回其它错误码。

Code example

```
GT9316_STATUS status = GT9316_SUCCESS;
status = GT9316_AdcSetTriggerDelay(ph, 1000);
if (status != GT9316_SUCCESS) {
    debug_printf("GT9316_TxSetTriggerDelay failed, return status = 0x%08x!!!\r\n", status);
    return status;
}
```

5.2.15. GT9316_AdcGetTriggerDelay

原型

```
STDGT9316CALL GT9316_AdcGetTriggerDelay(GT9316HANDLE ph, GT9316_UINT32 * delay)
```

功能

获取 ADC 触发延时。

参数说明

输入参数

ph：板卡句柄，由函数 GT9316_OpenCard 得到。

输出参数

delay：延时值，单位 1 us。

返回值

成功时返回 GT9316_SUCCESS。
失败时可能返回的返回值列表：
GT9316_BAD_PARAMETER_1
GT9316_BAD_PARAMETER_2
GT9316_NULL_DESCRIPTOR
GT9316_NO_BOARD
GT9316_OS_ERROR

Code example

```
GT9316_STATUS status = GT9316_SUCCESS;
GT9316_UINT32 delay;
status = GT9316_AdcGetTriggerDelay(ph, &delay);
if (status != GT9316_SUCCESS) {
    debug_printf("GT9316_AdcGetTriggerDelay, return status = 0x%08x!!!\r\n", status);
    return status;
}
```

5.2.16. GT9316_AdcSetTriggerPreCount

原型

```
STDGT9316CALL    GT9316_AdcSetTriggerPreCount(GT9316HANDLE    ph,GT9316_UINT32
count)
```

功能

设置 ADC 预触发点数，即在触发之前要采样的点数。

参数说明

输入参数

ph：板卡句柄，由函数 GT9316_OpenCard 得到。
count：预触发点数，最大可设置为（0x4000000 - 1）。

输出参数

无。

返回值

成功时返回 GT9316_SUCCESS。
失败时可能返回的返回值列表：
GT9316_BAD_PARAMETER_1
GT9316_NULL_DESCRIPTOR
GT9316_NO_BOARD
GT9316_OS_ERROR

Code example

```
GT9316_STATUS status = GT9316_SUCCESS;
status = GT9316_AdcSetTriggerPreCount(ph, 0x1000);
if (status != GT9316_SUCCESS) {
    debug_printf("GT9316_AdcSetTriggerPreCount, return status = 0x%08x!!!\r\n", status);
    return status;
}
```

5.2.17. GT9316_AdcGetTriggerPreCount

原型

```
STDGT9316CALL  GT9316_AdcGetTriggerPreCount(GT9316HANDLE  ph,GT9316_UINT32  *
count)
```

功能

获取 ADC 预触发点数，即在触发之前要采样的点数。

参数说明

输入参数

ph：板卡句柄，由函数 GT9316_OpenCard 得到。

输出参数

count：预触发点数。

返回值

成功时返回 GT9316_SUCCESS。
失败时可能返回的返回值列表：
GT9316_BAD_PARAMETER_1
GT9316_BAD_PARAMETER_2
GT9316_NULL_DESCRIPTOR
GT9316_NO_BOARD
GT9316_OS_ERROR

Code example

```
GT9316_STATUS status = GT9316_SUCCESS;
GT9316_UINT32 count;
status = GT9316_AdcGetTriggerPreCount(ph, &count);
if (status != GT9316_SUCCESS) {
    debug_printf("GT9316_AdcGetTriggerPreCount, return status = 0x%08x!!!\r\n", status);
    return status;
}
```

5.2.18. GT9316_AdcSetTriggerDeepCount

原型

```
STDGT9316CALL   GT9316_AdcSetTriggerDeepCount(GT9316HANDLE   ph,GT9316_UINT32 count)
```

功能

设置 ADC 触发深度点数，其中也包含预触发点数，所以此触发深度点数要大于预触发点数，例如预触发点数设置为 1000，触发深度设置为 4000，则在触发前采集 1000 个数据点，触发后采集 3000 个数据点。

参数说明

输入参数

ph：板卡句柄，由函数 GT9316_OpenCard 得到。
count: 触发深度点数，最大可设置为（0x8000000 - 1）。

输出参数

无。

返回值

成功时返回 GT9316_SUCCESS。
失败时可能返回的返回值列表：
GT9316_BAD_PARAMETER_1
GT9316_NULL_DESCRIPTOR
GT9316_NO_BOARD
GT9316_OS_ERROR

Code example

```
GT9316_STATUS status = GT9316_SUCCESS;
status = GT9316_AdcSetTriggerDeepCount(ph, 0x4000);
if (status != GT9316_SUCCESS) {
```

```
    debug_printf("GT9316_AdcSetTriggerDeepCount, return status = 0x%08x!!!\r\n", status);
    return status;
}
```

5.2.19. GT9316_AdcGetTriggerDeepCount

原型

```
STDGT9316CALL GT9316_AdcGetTriggerDeepCount(GT9316HANDLE ph,GT9316_UINT32 *count)
```

功能

获取 ADC 触发深度点数。

参数说明

输入参数

ph：板卡句柄，由函数 GT9316_OpenCard 得到。

输出参数

count: 触发深度点数。

返回值

成功时返回 GT9316_SUCCESS。

失败时可能返回的返回值列表：

GT9316_BAD_PARAMETER_1

GT9316_BAD_PARAMETER_2

GT9316_NULL_DESCRIPTOR

GT9316_NO_BOARD

GT9316_OS_ERROR

Code example

```
GT9316_STATUS status = GT9316_SUCCESS;
GT9316_UINT32 count;
status = GT9316_AdcGetTriggerDeepCount(ph, &count);
if (status != GT9316_SUCCESS) {
    debug_printf("GT9316_AdcGetTriggerDeepCount, return status = 0x%08x!!!\r\n", status);
    return status;
}
```

5.2.20. GT9316_AdcSetAnalogCmpThr

原型

```
STDGT9316CALL GT9316_AdcSetAnalogCmpThr(GT9316HANDLE ph, GT9316_UINT8 channel, GT9316_INT16 thr)
```

功能

设置 ADC 触发模拟信号比较阈值。

参数说明

输入参数

ph：板卡句柄，由函数 GT9316_OpenCard 得到。
channel: ADC 通道号，共 8 个通道： 0~7。
thr: 比较阈值，16 位有符号数，与 ADC 通道转换后的数据值进行比较。

输出参数

无。

返回值

成功时返回 GT9316_SUCCESS。
失败时可能返回的返回值列表：
GT9316_BAD_PARAMETER_1
GT9316_BAD_PARAMETER_2
GT9316_NULL_DESCRIPTOR
GT9316_NO_BOARD
GT9316_OS_ERROR

Code example

```
GT9316_STATUS status = GT9316_SUCCESS;
status = GT9316_AdcSetAnalogCmpThr(ph, 0, 2000);
if (status != GT9316_SUCCESS) {
    debug_printf("GT9316_AdcSetAnalogCmpThr, return status = 0x%08x!!!\r\n", status);
    return status;
}
```

5.2.21. GT9316_AdcGetAnalogCmpThr

原型

```
STDGT9316CALL GT9316_AdcGetAnalogCmpThr(GT9316HANDLE ph, GT9316_UINT8 channel,
GT9316_INT16 * thr)
```

功能

获取 ADC 触发模拟信号比较阈值。

参数说明

输入参数

ph：板卡句柄，由函数 GT9316_OpenCard 得到。
channel: ADC 通道号，共 8 个通道： 0~7。

输出参数

thr: 比较阈值，16 位有符号数。

返回值

成功时返回 GT9316_SUCCESS。
失败时可能返回的返回值列表：

GT9316_BAD_PARAMETER_1
GT9316_BAD_PARAMETER_2
GT9316_BAD_PARAMETER_3
GT9316_NULL_DESCRIPTOR
GT9316_NO_BOARD
GT9316_OS_ERROR

Code example

```
GT9316_STATUS status = GT9316_SUCCESS;
GT9316_INT16 thr;
status = GT9316_AdcGetAnalogCmpThr(ph, 0, &thr);
if (status != GT9316_SUCCESS) {
    debug_printf("GT9316_AdcGetAnalogCmpThr, return status = 0x%08x!!!\r\n", status);
    return status;
}
```

5.2.22. GT9316_AdcGetDataCount

原型

```
STDGT9316CALL GT9316_AdcGetDataCount(GT9316HANDLE ph, GT9316_UINT8 channel, GT9316_UINT32 * count)
```

功能

获取 ADC 当前通道数据点数。

参数说明

输入参数

ph：板卡句柄，由函数 GT9316_OpenCard 得到。
channel：ADC 通道，取值 0~7。

输出参数

count: ADC 当前通道转换的数据点数。

返回值

成功时返回 GT9316_SUCCESS。
失败时可能返回的返回值列表：
GT9316_BAD_PARAMETER_1
GT9316_BAD_PARAMETER_2
GT9316_BAD_PARAMETER_3
GT9316_NULL_DESCRIPTOR
GT9316_NO_BOARD
GT9316_OS_ERROR
函数也可能返回其它错误码

Code example

```
GT9316_STATUS status = GT9316_SUCCESS;
GT9316_UINT32 count;
status = GT9316_AdcGetDataCount(ph, 1, &count);
```

```
if (status != GT9316_SUCCESS) {
    debug_printf("GT9316_AdcGetDataCount failed. Return value = %d\r\n", status);
    return status;
}
```

5.2.23. GT9316_AdcReadData

原型

```
STDGT9316CALL GT9316_AdcReadData(GT9316HANDLE ph, GT9316_UINT8 channel, GT9316_INT16 * buf, GT9316_UINT32 length, GT9316_UINT32 * olength GT9316_UINT32 timeout)
```

功能

读 ADC 当前通道数据，一个数据点为双字节有符号数。

参数说明

输入参数

ph：板卡句柄，由函数 GT9316_OpenCard 得到。
channel：ADC 通道，取值 0~7。
length：要读取的数据点数，一个数据点为双字节有符号数。
timeout：读超时时间，单位 ms。

输出参数

buf：存放读出数据的缓冲区，由调用者分配。
olength：实际读出的数据点数。

返回值:

成功时返回 GT9316_SUCCESS。
失败时可能返回的返回值列表：
GT9316_BAD_PARAMETER_1
GT9316_BAD_PARAMETER_2
GT9316_BAD_PARAMETER_3
GT9316_BAD_PARAMETER_5
GT9316_NULL_DESCRIPTOR
GT9316_NO_BOARD
GT9316_OS_ERROR
函数也可能返回其它错误码。

Code example

```
GT9316_STATUS status = GT9316_SUCCESS;
GT9316_INT32 ret = 0;
GT9316_INT16 buf[8];
GT9316_UINT32 i = 0, j = 0;
GT9316_UINT32 rcnt = 0;
for (i=0; i<1;i++) {
    status = GT9316_AdcReadData(ph, (GT9316_UINT8)i, buf, 8, &rcnt, 10000);
    printf("########################################################################\r\n");
```

```
    for (j = 0; j < rcnt; j++) {
        printf("ch %d,  wi = %d, data = %08x\r\n", i, j, buf[j]);
    }
    printf("#########################################################################\r\n");
}
```