HT9118 使用手册

Ver 1.0

Revision

时间	原因	版本	作者	备注
2019-10-13	V1.0 版本生成	V1.0	冯江宏	建立文档
2019-12-01	V1.1 版本生成	V1.1	冯江宏	更改部分描述

1. 概述

1.1. 产品描述

略。

1.2. 特性

略。

1.3. 详细描述

略。

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

略。

1.5.2. 驱动安装

略。

1.5.3. SDK 文件

SDK 库文件包含下列文件：
gt9118_api_win7_32bit.dll
gt9118_api_win7_32bit.lib

这两个文件用于应用程序开发。

2. 硬件说明

略。

3. 宏定义

3.1. AD 最大通道数

```
#define MAX_AD_CHANNEL_NUM        2
```

3.2. 函数返回值

Api 函数返回值统一使用 GT9118_STATUS 枚举值定义。
返回值定义如下：
GT9118_SUCCESS
成功
GT9118_OS_ERROR
系统错误
GT9118_LOW_MEMORY
内存不足
GT9118_INVALID_MSG_ID
无效的消息 ID
GT9118_BAD_PARAMETER_1
GT9118_BAD_PARAMETER_2
GT9118_BAD_PARAMETER_3
GT9118_BAD_PARAMETER_4
GT9118_BAD_PARAMETER_5
GT9118_BAD_PARAMETER_6
GT9118_BAD_PARAMETER_7
GT9118_BAD_PARAMETER_8
GT9118_BAD_PARAMETER_9
无效的参数 1...9
GT9118_NULL_DESCRIPTOR
空描述符
GT9118_NO_BOARD
板卡打开失败
GT9118_NOT_OPEN
板卡未打开
GT9118_READ_TIMEOUT
读操作超时

GT9118_WRITE_TIMEOUT
写操作超时
GT9118_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_EXTERNAL,
    EN_TRIGGER_SRC_ACMP_0,
    EN_TRIGGER_SRC_ACMP_1,
    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_SRC_EXTERNAL：外部。
EN_TRIGGER_ACMP_0~1：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_X2 = 0,
    EN_RANGE_X1,
    EN_RANGE_BOTTOM,
}EN_ADC_RANGE;
```

成员定义

EN_RANGE_X2：X2 转换范围
EN_RANGE_X1：X1 转换范围

4.4. 时钟选择定义

```
typedef enum {
    EN_CLK_60M = 0,
    EN_CLK_EXTERNAL,
    EN_CLK_BOTTOM,
}EN_CLK;
```

成员定义

EN_CLK_60M：60M 时钟
EN_CLK_EXTERNAL：外部时钟

5. API 说明

5.1. 通用接口

5.1.1. GT9118_OpenCard

原型

```
STDGT9118CALL GT9118_OpenCard(GT9118HANDLE  * ph, GT9118_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 GT9118_SUCCESS。
失败时可能返回的返回值列表：
GT9118_BAD_PARAMETER_1
GT9118_LOW_MEMORY
GT9118_NOT_OPEN

Code example

```
    GT9118_STATUS status = GT9118_SUCCESS;
    GT9118HANDLE ph = NULL;
    status = GT9118_OpenCard(&ph, 1);
    if (status != GT9118_SUCCESS) {
        debug_printf("GT9118_OpenCard failed. Return value = %d\r\n", status);
        return status;
    }
```

5.1.2. GT9118_CloseCard

原型

```
STDGT9118CALL GT9118_CloseCard(GT9118HANDLE * 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 GT9118_SUCCESS。

失败时可能返回的返回值列表：

GT9118_BAD_PARAMETER_1

GT9118_NULL_DESCRIPTOR

GT9118_NO_BOARD

GT9118_OS_ERROR

Code example

```
    GT9118_STATUS status = GT9118_SUCCESS;
    status = GT9118_CloseCard(ph);
    if (status != GT9118_SUCCESS) {
        debug_printf("GT9118_CloseCard failed. Return value = %d\r\n", status);
        return status;
    }
```

5.1.3. GT9118_ResetCard

原型

```
STDGT9118CALL GT9118_ResetCard(GT9118HANDLE ph)
```

功能

复位板卡。

参数说明

输入参数

ph：板卡句柄，由函数 GT9118_OpenCard 得到。

输出参数

无。

返回值

成功时返回 GT9118_SUCCESS。

失败时可能返回的返回值列表：

GT9118_BAD_PARAMETER_1

GT9118_NULL_DESCRIPTOR

GT9118_NO_BOARD

GT9118_OS_ERROR

函数也可能返回其它错误码

Code example

```
    GT9118_STATUS status = GT9118_SUCCESS;
```

```
    status = GT9118_ResetCard(ph);
    if (status != GT9118_SUCCESS) {
        debug_printf("GT9118_ResetCard failed. Return value = %d\r\n", status);
        return status;
    }
```

5.2. ADC 接口

5.2.1. GT9118_AdcSetEnable

原型

```
STDGT9118CALL GT9118_AdcSetEnable(GT9118HANDLE ph, GT9118_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 GT9118_OpenCard 得到。
enable：GT9118_TRUE -使能 ADC，GT9118_FALSE-不使能 ADC。

输出参数

无

返回值:

成功时返回 GT9118_SUCCESS。
失败时可能返回的返回值列表：
GT9118_BAD_PARAMETER_1
GT9118_NULL_DESCRIPTOR
GT9118_NO_BOARD
GT9118_OS_ERROR
函数也可能返回其它错误码。

Code example

```
GT9118_STATUS status = GT9118_SUCCESS;
status = GT9118_AdcSetEnable(ph, GT9118_TRUE);
if (status != GT9118_SUCCESS) {
    debug_printf("GT9118_AdcSetEnable failed, return status = 0x%08x!!!\r\n", status);
    return status;
}
```

5.2.2. GT9118_AdcGetEnable

原型

```
STDGT9118CALL GT9118_AdcGetEnable(GT9118HANDLE ph, GT9118_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 GT9118_OpenCard 得到。

输出参数

enable：GT9118_TRUE - ADC 使能， GT9118_FALSE - ADC 不使能。

返回值

成功时返回 GT9118_SUCCESS。

失败时可能返回的返回值列表：

GT9118_BAD_PARAMETER_1

GT9118_BAD_PARAMETER_2

GT9118_NULL_DESCRIPTOR

GT9118_NO_BOARD

GT9118_OS_ERROR

Code example

```
GT9118_STATUS status = GT9118_SUCCESS;
GT9118_BOOL enable;
status = GT9118_AdcGetEnable(ph, &enable);
if (status != GT9118_SUCCESS) {
    debug_printf("GT9118_AdcGetEnable, return status = 0x%08x!!!\r\n", status);
    return status;
}
```

5.2.3. GT9118_AdcStart

原型

```
STDGT9118CALL GT9118_AdcStart(GT9118HANDLE ph)
```

功能

启动 ADC 采集。

参数说明

输入参数

ph：板卡句柄，由函数 GT9118_OpenCard 得到。

输出参数

无。

返回值

成功时返回 GT9118_SUCCESS。

失败时可能返回的返回值列表：

GT9118_BAD_PARAMETER_1

GT9118_NULL_DESCRIPTOR
GT9118_NO_BOARD
GT9118_OS_ERROR
函数也可能返回其它错误码

Code example

```
GT9118_STATUS status = GT9118_SUCCESS;
status = GT9118_AdcStart(ph);
if (status != GT9118_SUCCESS) {
    debug_printf("GT9118_AdcStart failed. Return value = %d\r\n", status);
    return status;
}
```

5.2.4. GT9118_AdcStop

原型

```
STDGT9118CALL GT9118_AdcStop(GT9118HANDLE ph)
```

功能

停止 ADC 采集。

参数说明

输入参数

ph：板卡句柄，由函数 GT9118_OpenCard 得到。

输出参数

无。

返回值

成功时返回 GT9118_SUCCESS。
失败时可能返回的返回值列表：
GT9118_BAD_PARAMETER_1
GT9118_NULL_DESCRIPTOR
GT9118_NO_BOARD
GT9118_OS_ERROR
函数也可能返回其它错误码

Code example

```
GT9118_STATUS status = GT9118_SUCCESS;
status = GT9118_AdcStop(ph);
if (status != GT9118_SUCCESS) {
    debug_printf("GT9118_AdcStop failed. Return value = %d\r\n", status);
    return status;
}
```

5.2.5. GT9118_AdcGetRunningState

原型

```
STDGT9118CALL GT9118_AdcGetRunningState(GT9118HANDLE ph, GT9118_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 GT9118_OpenCard 得到。

输出参数

无。

返回值

成功时返回 GT9118_SUCCESS。

失败时可能返回的返回值列表：

GT9118_BAD_PARAMETER_1

GT9118_BAD_PARAMETER_2

GT9118_NULL_DESCRIPTOR

GT9118_NO_BOARD

GT9118_OS_ERROR

函数也可能返回其它错误码

Code example

```
GT9118_STATUS status = GT9118_SUCCESS;
GT9118_BOOL enable;
status = GT9118_AdcGetRunningState(ph, &enable);
if (status != GT9118_SUCCESS) {
    debug_printf("GT9118_AdcGetRunningState failed. Return value = %d\r\n", status);
    return status;
}
```

5.2.6. GT9118_AdcSetAcEnable

原型

```
STDGT9118CALL GT9118_AdcSetAcEnable(GT9118HANDLE ph, GT9118_BOOL enable)
```

功能

设置 ADC 是否使能 AC 耦合。

参数说明

输入参数

ph：板卡句柄，由函数 GT9118_OpenCard 得到。
enable：GT9118_TRUE - 使能 AC 输入耦合， GT9118_FALSE - 禁用 AC 输入耦合。

输出参数

无。

返回值

成功时返回 GT9118_SUCCESS。
失败时可能返回的返回值列表：
GT9118_BAD_PARAMETER_1
GT9118_NULL_DESCRIPTOR
GT9118_NO_BOARD
GT9118_OS_ERROR
函数也可能返回其它错误码

Code example

```
GT9118_STATUS status = GT9118_SUCCESS;
status = GT9118_AdcSetAcEnable(ph, GT9118_TRUE);
if (status != GT9118_SUCCESS) {
    debug_printf("GT9118_AdcSetAcEnable failed. Return value = %d\r\n", status);
    return status;
}
```

5.2.7. GT9118_AdcGetAcEnable

原型

```
STDGT9118CALL GT9118_AdcGetAcEnable(GT9118HANDLE ph, GT9118_BOOL * enable)
```

功能

获取 ADC 当前是否使能 AC 耦合。

参数说明

输入参数

ph：板卡句柄，由函数 GT9118_OpenCard 得到。

输出参数

enable：GT9118_TRUE - 使能 AC 输入耦合， GT9118_FALSE - 禁用 AC 输入耦合。

返回值

成功时返回 GT9118_SUCCESS。
失败时可能返回的返回值列表：
GT9118_BAD_PARAMETER_1
GT9118_BAD_PARAMETER_2
GT9118_NULL_DESCRIPTOR
GT9118_NO_BOARD
GT9118_OS_ERROR
函数也可能返回其它错误码

Code example

```
GT9118_STATUS status = GT9118_SUCCESS;
GT9118_BOOL enable;
status = GT9118_AdcGetAcEnable(ph, &enable);
if (status != GT9118_SUCCESS) {
    debug_printf("GT9118_AdcGetAcEnable failed. Return value = %d\r\n", status);
    return status;
}
```

5.2.8. GT9118_AdcSetClkSel

原型

```
STDGT9118CALL GT9118_AdcSetClkSel(GT9118HANDLE ph, EN_CLK clk)
```

功能

设置 ADC 时钟选择。

参数说明

输入参数

ph：板卡句柄，由函数 GT9118_OpenCard 得到。
clk：时钟源选择，参见 EN_CLK 定义。

输出参数

无。

返回值

成功时返回 GT9118_SUCCESS。
失败时可能返回的返回值列表：
GT9118_BAD_PARAMETER_1
GT9118_NULL_DESCRIPTOR
GT9118_NO_BOARD
GT9118_OS_ERROR
函数也可能返回其它错误码

Code example

```
GT9118_STATUS status = GT9118_SUCCESS;
status = GT9118_AdcSetClkSel(ph, EN_CLK_60M);
if (status != GT9118_SUCCESS) {
    return status;
}
```

5.2.9. GT9118_AdcGetClkSel

原型

```
STDGT9118CALL GT9118_AdcGetClkSel(GT9118HANDLE ph, EN_CLK * clk)
```

功能

获取 ADC 时钟源类型。

参数说明

输入参数

ph：板卡句柄，由函数 GT9118_OpenCard 得到。

输出参数

clk：时钟源输入定义，参见 EN_CLK 定义。

返回值

成功时返回 GT9118_SUCCESS。

失败时可能返回的返回值列表：

GT9118_BAD_PARAMETER_1

GT9118_BAD_PARAMETER_2

GT9118_NULL_DESCRIPTOR

GT9118_NO_BOARD

GT9118_OS_ERROR

函数也可能返回其它错误码

Code example

```
GT9118_STATUS status = GT9118_SUCCESS;
EN_CLK clk;
status = GT9118_AdcGetClkSel(ph, &clk);
if (status != GT9118_SUCCESS) {
    return status;
}
```

5.2.10. GT9118_AdcSetRange

原型

```
STDGT9118CALL GT9118_AdcSetRange(GT9118HANDLE 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 GT9118_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 GT9118_SUCCESS。

失败时可能返回的返回值列表：

GT9118_BAD_PARAMETER_1

GT9118_BAD_PARAMETER_2
GT9118_NULL_DESCRIPTOR
GT9118_NO_BOARD
GT9118_OS_ERROR
函数也可能返回其它错误码

Code example

```
GT9118_STATUS status = GT9118_SUCCESS;
status = GT9118_AdcSetRange(ph, EN_RANGE_NP5);
if (status != GT9118_SUCCESS) {
    debug_printf("GT9118_AdcSetRange failed. Return value = %d\r\n", status);
    return status;
}
```

5.2.11. GT9118_AdcGetRange

原型

```
STDGT9118CALL GT9118_AdcGetRange(GT9118HANDLE 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 GT9118_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 GT9118_SUCCESS。
失败时可能返回的返回值列表：
GT9118_BAD_PARAMETER_1
GT9118_BAD_PARAMETER_2
GT9118_NULL_DESCRIPTOR
GT9118_NO_BOARD
GT9118_OS_ERROR
函数也可能返回其它错误码

Code example

```
GT9118_STATUS status = GT9118_SUCCESS;
EN_ADC_RANGE range;
status = GT9118_AdcGetRange(ph, &range);
if (status != GT9118_SUCCESS) {
    debug_printf("GT9118_AdcGetRange failed. Return value = %d\r\n", status);
    return status;
}
```

5.2.12. GT9118_AdcSetClkDiv

原型

```
STDGT9118CALL GT9118_AdcSetClkDiv(GT9118HANDLE ph, GT9118_UINT16 div)
```

功能

设置 ADC 的采样频率对于时钟频率的分频因子，0 表示 1 分频，1 表示 2 分频，依次类推。

参数说明

输入参数

ph：板卡句柄，由函数 GT9118_OpenCard 得到。
div：分频因子。

输出参数

无。

返回值

成功时返回 GT9118_SUCCESS。
失败时可能返回的返回值列表：
GT9118_BAD_PARAMETER_1
GT9118_BAD_PARAMETER_2
GT9118_NULL_DESCRIPTOR
GT9118_NO_BOARD
GT9118_OS_ERROR
函数也可能返回其它错误码

Code example

```
GT9118_STATUS status = GT9118_SUCCESS;
status = GT9118_AdcSetClkDiv(ph, 22);
if (status != GT9118_SUCCESS) {
    return status;
}
```

5.2.13. GT9118_AdcGetClkDiv

原型

```
STDGT9118CALL GT9118_AdcGetClkDiv(GT9118HANDLE ph, GT9118_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 GT9118_OpenCard 得到。

输出参数

div：分频因子。

返回值

成功时返回 GT9118_SUCCESS。

失败时可能返回的返回值列表：

GT9118_BAD_PARAMETER_1

GT9118_BAD_PARAMETER_2

GT9118_NULL_DESCRIPTOR

GT9118_NO_BOARD

GT9118_OS_ERROR

函数也可能返回其它错误码

Code example

```
GT9118_STATUS status = GT9118_SUCCESS;
GT9118_UINT16 div;
status = GT9118_AdcGetClkDiv(ph, &div);
if (status != GT9118_SUCCESS) {
    return status;
}
```

5.2.14. GT9118_AdcSetTriggerSrc

原型

```
STDGT9118CALL GT9118_AdcSetTriggerSrc(GT9118HANDLE 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 GT9118_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 GT9118_SUCCESS。

失败时可能返回的返回值列表：

GT9118_BAD_PARAMETER_1

GT9118_NULL_DESCRIPTOR

GT9118_NO_BOARD

GT9118_OS_ERROR

函数也可能返回其它错误码。

Code example

```
GT9118_STATUS status = GT9118_SUCCESS;
status = GT9118_AdcSetTriggerSrc(ph, EN_TRIGGER_SRC_SOFTWARE);
if (status != GT9118_SUCCESS) {
    debug_printf("GT9118_AdcSetTriggerSrc failed, return status = 0x%08x!!!\r\n", status);
```

```
    return status;
}
```

5.2.15. GT9118_AdcGetTriggerSrc

原型

```
STDGT9118CALL GT9118_AdcGetTriggerSrc(GT9118HANDLE 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 GT9118_OpenCard 得到。

输出参数

src：当前 ADC 触发源。

返回值

成功时返回 GT9118_SUCCESS。

失败时可能返回的返回值列表：

GT9118_BAD_PARAMETER_1

GT9118_BAD_PARAMETER_2

GT9118_NULL_DESCRIPTOR

GT9118_NO_BOARD

GT9118_OS_ERROR

Code example

```
GT9118_STATUS status = GT9118_SUCCESS;
EN_TRIGGER_SRC src;
status = GT9118_AdcGetTriggerSrc(ph, &src);
if (status != GT9118_SUCCESS) {
    debug_printf("GT9118_AdcGetTriggerSrc, return status = 0x%08x!!!\r\n", status);
    return status;
}
```

5.2.16. GT9118_AdcSetTriggerMode

原型

```
STDGT9118CALL GT9118_AdcSetTriggerMode(GT9118HANDLE 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 GT9118_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 GT9118_SUCCESS。

失败时可能返回的返回值列表：

GT9118_BAD_PARAMETER_1

GT9118_NULL_DESCRIPTOR

GT9118_NO_BOARD

GT9118_OS_ERROR

函数也可能返回其它错误码。

Code example

```
GT9118_STATUS status = GT9118_SUCCESS;
status = GT9118_AdcSetTriggerMode(ph, EN_TRIGGER_MODE_HIGHLEVEL);
if (status != GT9118_SUCCESS) {
    debug_printf("GT9118_AdcSetTriggerMode failed, return status = 0x%08x!!!\r\n", status);
    return status;
}
```

5.2.17. GT9118_AdcGetTriggerMode

原型

```
STDGT9118CALL GT9118_AdcGetTriggerMode(GT9118HANDLE 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 GT9118_OpenCard 得到。

输出参数

mode：ADC 触发模式。

返回值

成功时返回 GT9118_SUCCESS。

失败时可能返回的返回值列表：

GT9118_BAD_PARAMETER_1

GT9118_BAD_PARAMETER_2

GT9118_NULL_DESCRIPTOR

GT9118_NO_BOARD

GT9118_OS_ERROR

Code example

```
GT9118_STATUS status = GT9118_SUCCESS;
EN_TRIGGER_MODE mode;
```

```
status = GT9118_AdcGetTriggerMode(ph, &mode);
if (status != GT9118_SUCCESS) {
    debug_printf("GT9118_AdcGetTriggerMode, return status = 0x%08x!!!\r\n", status);
    return status;
}
```

5.2.18. GT9118_AdcSetTriggerDelay

原型

```
STDGT9118CALL GT9118_AdcSetTriggerDelay(GT9118HANDLE ph, GT9118_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 GT9118_OpenCard 得到。
delay：延迟值，单位 1 us。

输出参数

无

返回值:

成功时返回 GT9118_SUCCESS。
失败时可能返回的返回值列表：
GT9118_BAD_PARAMETER_1
GT9118_NULL_DESCRIPTOR
GT9118_NO_BOARD
GT9118_OS_ERROR
函数也可能返回其它错误码。

Code example

```
GT9118_STATUS status = GT9118_SUCCESS;
status = GT9118_AdcSetTriggerDelay(ph, 1000);
if (status != GT9118_SUCCESS) {
    debug_printf("GT9118_TxSetTriggerDelay failed, return status = 0x%08x!!!\r\n", status);
    return status;
}
```

5.2.19. GT9118_AdcGetTriggerDelay

原型

```
STDGT9118CALL GT9118_AdcGetTriggerDelay(GT9118HANDLE ph, GT9118_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 GT9118_OpenCard 得到。

输出参数

delay：延时值，单位 1 us。

返回值

成功时返回 GT9118_SUCCESS。

失败时可能返回的返回值列表：

GT9118_BAD_PARAMETER_1

GT9118_BAD_PARAMETER_2

GT9118_NULL_DESCRIPTOR

GT9118_NO_BOARD

GT9118_OS_ERROR

Code example

```
GT9118_STATUS status = GT9118_SUCCESS;
GT9118_UINT32 delay;
status = GT9118_AdcGetTriggerDelay(ph, &delay);
if (status != GT9118_SUCCESS) {
    debug_printf("GT9118_AdcGetTriggerDelay, return status = 0x%08x!!!\r\n", status);
    return status;
}
```

5.2.20. GT9118_AdcSetTriggerPreCount

原型

```
STDGT9118CALL    GT9118_AdcSetTriggerPreCount(GT9118HANDLE    ph,GT9118_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 GT9118_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 GT9118_SUCCESS。

失败时可能返回的返回值列表：

GT9118_BAD_PARAMETER_1

GT9118_NULL_DESCRIPTOR

GT9118_NO_BOARD

GT9118_OS_ERROR

Code example

```
GT9118_STATUS status = GT9118_SUCCESS;
status = GT9118_AdcSetTriggerPreCount(ph, 0x1000);
if (status != GT9118_SUCCESS) {
    debug_printf("GT9118_AdcSetTriggerPreCount, return status = 0x%08x!!!\r\n", status);
    return status;
}
```

5.2.21. GT9118_AdcGetTriggerPreCount

原型

```
STDGT9118CALL  GT9118_AdcGetTriggerPreCount(GT9118HANDLE  ph,GT9118_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 GT9118_OpenCard 得到。

输出参数

count: 预触发点数。

返回值

成功时返回 GT9118_SUCCESS。

失败时可能返回的返回值列表：

GT9118_BAD_PARAMETER_1

GT9118_BAD_PARAMETER_2

GT9118_NULL_DESCRIPTOR

GT9118_NO_BOARD

GT9118_OS_ERROR

Code example

```
GT9118_STATUS status = GT9118_SUCCESS;
GT9118_UINT32 count;
status = GT9118_AdcGetTriggerPreCount(ph, &count);
if (status != GT9118_SUCCESS) {
    debug_printf("GT9118_AdcGetTriggerPreCount, return status = 0x%08x!!!\r\n", status);
    return status;
}
```

5.2.22. GT9118_AdcSetTriggerDeepCount

原型

```
STDGT9118CALL   GT9118_AdcSetTriggerDeepCount(GT9118HANDLE   ph,GT9118_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 GT9118_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 GT9118_SUCCESS。
失败时可能返回的返回值列表：
GT9118_BAD_PARAMETER_1
GT9118_NULL_DESCRIPTOR
GT9118_NO_BOARD
GT9118_OS_ERROR

Code example

```
GT9118_STATUS status = GT9118_SUCCESS;
status = GT9118_AdcSetTriggerDeepCount(ph, 0x4000);
if (status != GT9118_SUCCESS) {
    debug_printf("GT9118_AdcSetTriggerDeepCount, return status = 0x%08x!!!\r\n", status);
    return status;
}
```

5.2.23. GT9118_AdcGetTriggerDeepCount

原型

```
STDGT9118CALL  GT9118_AdcGetTriggerDeepCount(GT9118HANDLE  ph,GT9118_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 GT9118_OpenCard 得到。

输出参数

count: 触发深度点数。

返回值

成功时返回 GT9118_SUCCESS。

失败时可能返回的返回值列表：

GT9118_BAD_PARAMETER_1

GT9118_BAD_PARAMETER_2

GT9118_NULL_DESCRIPTOR

GT9118_NO_BOARD

GT9118_OS_ERROR

Code example

```
GT9118_STATUS status = GT9118_SUCCESS;
GT9118_UINT32 count;
status = GT9118_AdcGetTriggerDeepCount(ph, &count);
if (status != GT9118_SUCCESS) {
    debug_printf("GT9118_AdcGetTriggerDeepCount, return status = 0x%08x!!!\r\n", status);
    return status;
}
```

5.2.24. GT9118_AdcSetAnalogCmpThr

原型

```
STDGT9118CALL GT9118_AdcSetAnalogCmpThr(GT9118HANDLE ph, GT9118_UINT8 channel,
GT9118_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 GT9118_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 GT9118_SUCCESS。

失败时可能返回的返回值列表：

GT9118_BAD_PARAMETER_1

GT9118_BAD_PARAMETER_2

GT9118_NULL_DESCRIPTOR

GT9118_NO_BOARD
GT9118_OS_ERROR

Code example

```
GT9118_STATUS status = GT9118_SUCCESS;
status = GT9118_AdcSetAnalogCmpThr(ph, 0, 2000);
if (status != GT9118_SUCCESS) {
    debug_printf("GT9118_AdcSetAnalogCmpThr, return status = 0x%08x!!!\r\n", status);
    return status;
}
```

5.2.25. GT9118_AdcGetAnalogCmpThr

原型

```
STDGT9118CALL GT9118_AdcGetAnalogCmpThr(GT9118HANDLE ph, GT9118_UINT8 channel, GT9118_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 GT9118_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 GT9118_SUCCESS。
失败时可能返回的返回值列表：
GT9118_BAD_PARAMETER_1
GT9118_BAD_PARAMETER_2
GT9118_BAD_PARAMETER_3
GT9118_NULL_DESCRIPTOR
GT9118_NO_BOARD
GT9118_OS_ERROR

Code example

```
GT9118_STATUS status = GT9118_SUCCESS;
GT9118_INT16 thr;
status = GT9118_AdcGetAnalogCmpThr(ph, 0, &thr);
if (status != GT9118_SUCCESS) {
    debug_printf("GT9118_AdcGetAnalogCmpThr, return status = 0x%08x!!!\r\n", status);
    return status;
}
```

5.2.26. GT9118_AdcGetDataCount

原型

```
STDGT9118CALL GT9118_AdcGetDataCount(GT9118HANDLE ph, GT9118_UINT8 channel, GT9118_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 GT9118_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 GT9118_SUCCESS。
失败时可能返回的返回值列表：
GT9118_BAD_PARAMETER_1
GT9118_BAD_PARAMETER_2
GT9118_BAD_PARAMETER_3
GT9118_NULL_DESCRIPTOR
GT9118_NO_BOARD
GT9118_OS_ERROR
函数也可能返回其它错误码

Code example

```
GT9118_STATUS status = GT9118_SUCCESS;
GT9118_UINT32 count;
status = GT9118_AdcGetDataCount(ph, 1, &count);
if (status != GT9118_SUCCESS) {
    debug_printf("GT9118_AdcGetDataCount failed. Return value = %d\r\n", status);
    return status;
}
```

5.2.27. GT9118_AdcReadData

原型

```
STDGT9118CALL GT9118_AdcReadData(GT9118HANDLE ph, GT9118_UINT8 channel, GT9118_INT16 * buf, GT9118_UINT32 length, GT9118_UINT32 * olength GT9118_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 GT9118_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 GT9118_SUCCESS。
失败时可能返回的返回值列表：
GT9118_BAD_PARAMETER_1
GT9118_BAD_PARAMETER_2
GT9118_BAD_PARAMETER_3
GT9118_BAD_PARAMETER_5
GT9118_NULL_DESCRIPTOR
GT9118_NO_BOARD
GT9118_OS_ERROR
函数也可能返回其它错误码。

Code example

```
GT9118_STATUS status = GT9118_SUCCESS;
GT9118_INT32 ret = 0;
GT9118_INT16 buf[8];
GT9118_UINT32 i = 0, j = 0;
GT9118_UINT32 rcnt = 0;
for (i=0; i<1;i++) {
    status = GT9118_AdcReadData(ph, (GT9118_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");
}
```