# 驱动软件对外接口

1. DIO卡
   1. 参数类型枚举

typedef enum **\_EnableStatus**

{

UNABLE\_STATUS = 0, 不使能

ENABLE\_STATUS = 1 使能

} **EnableStatus**;

说明：使能枚举

typedef enum **\_Outp**utMode{

NULL\_OUTPUT\_MODE = 0x00, 无模式

LEVEL\_OUTPUT\_MODE = 0x01, 电平模式

PWM\_OUTPUT\_MODE = 0x02, PWM模式

BIT\_OUTPUT\_MODE = 0x03 BIT数据流模式

} OutputMode;

说明：DIO输出模式枚举

typedef enum **\_OutputImpedanceMode**

{

HIGH\_IMPEDANCE\_MODE = 0, 高阻模式

PULL\_MODE = 0X01, 拉模式

PUSH\_MODE = 0X02, 推模式

PUSH\_AND\_PULL\_MODE = 0X03 推拉模式

} **OutputImpedanceMode**;

说明：输出阻抗模式

typedef enum **\_VoltageLevel**

{

LOW\_LEVEL = 0x00, 低电平

HIGH\_LEVEL = 0x01, 高电平

INVALID\_LEVEL

} **VoltageLevel**;

说明：电平枚举

typedef enum **\_OutputReferenceVoltage**

{

OUTPUT\_5V\_REF = 0X01, 输出5v

OUTPUT\_12V\_REF = 0X02, 输出12v

OUTPUT\_OUTSIDE\_REF = 0X03 输出外部参考

} **OutputReferenceVoltage**;

说明：输出参考电压

typedef enum **\_ReferenceClock**

{

REFCLK\_100\_MHZ = 0, 100M参考时钟

REFCLK\_20\_MHZ = 1 20M参考时钟

} **ReferenceClock**;

说明：参考时钟

* 1. 参数数据结构体

typedef struct **\_DOLevelConfigure**

{

OutputImpedanceMode eMode; 输出阻抗

OutputReferenceVoltage eRef; 输出参考电压

} **DOLevelConfigure**;

说明：输出电平配置

typedef struct **\_DOPWMConfigure**

{

double dFreq; 频率Hz

double dDuty; 占空比

OutputImpedanceMode eMode; 输出阻抗

OutputReferenceVoltage eRef; 输出参考电压

} **DOPWMConfigure**;

说明：输出PWM配置

typedef struct **\_DIPWMConfigure**

{

double dDurationTime; 采样时间

ReferenceClock eRefClk; 参考时钟（保留，目前为自适应频率）

} **DIPWMConfigure**;

说明：输入PWM配置

typedef struct **\_PWMProperty**

{

double dFreq; 频率Hz

double dDuty; 占空比

} **PWMProperty**;

说明：PWM属性

typedef struct **\_DOBitConfigure**

{

OutputImpedanceMode eMode; 输出阻抗

long lRate; 速率

OutputReferenceVoltage eRef; 输出参考电压

} **DOBitConfigure**;

说明：输出BIT流配置

* 1. 函数接口

int **OpenDio** (const char\* strDev, void \*\*handle);

说明： 打开DIO对应设备文件，获取对应设备文件指针

参数： strDev DIO对应设备文件，如“/dev/dio\_3\_in”，表示打卡3槽DIO输入部分

handle对应设备文件指针

返回值：正常为0

备注：

int **GetDioVersion** (void \*handle,

char\* pStr,

int iLength,

int\* pActLength);

说明： 获取当前DIO卡逻辑固件版本信息

参数： handle 资源设备对应指针

pStr日期版本字符串首地址，用户申请字符数组之数组地址

iLength用户申请的字节数组长度

pActLength获取的版本信息实际有效字节数组长度

返回值：正常为0

备注：

int **CloseDio** (void\* handle);

说明： 关闭DIO对应设备文件

参数： handle资源设备对应指针

返回值：正常为0

备注：

int **SetDoChannelEnable** (void \*handle,

int iChannel,

EnableStatus eStatus);

说明： 设置DIO卡指定通道输出使能

参数： handle资源设备对应指针

iChannel 输出通道号,范围0~23

eStatus使能状态，ENABLE\_STATUS使能输出

返回值：正常为0

备注：

int **SetDoAllChannelEnable** (void \*handle, unsigned int iStatus);

说明： 设置DIO卡全通道输出使能

参数： handle资源设备对应指针

iStatus全通道使能状态，通道按位表示，0bit位为0通道，1为使能，0为未使能，如参数0x03表示输出通道0与1使能输出，其余30个输出通道均未使能输出

返回值：正常为0

备注：

int **GetDoAllChannelEnable** (void \*handle, unsigned int\* pStatus);

说明： 获取全通道使能状态

参数： handle资源设备对应指针

pStatus全通道使能状态，通道按位表示，0bit位为0通道，1为使能，0为未使能，如获取0x03表示输出通道0与1为使能输出状态，其余30个输出通道均未使能输出

返回值：正常为0

备注：

int **SetDoMode** (void \*handle, int iChannel, OutputMode eMode);

说明： 设置DIO卡指定输出通道模式

参数： handle资源设备对应指针

iChannel输出通道号，范围0~23

eMode输出模式，主要有电平模式，PWM模式以及BIT数据流模式

返回值：正常为0

备注：

int **GetDoAllMode** (void \*handle,

unsigned int\* pHiMode,

unsigned int\* pLoMode);

说明： 获取所有输出通道模式

参数： handle资源设备对应指针

pHiMode 通道16~23输出模式指针，如获取0x01，表示输出通道16模式为电平模式

pLoMode 通道0~15输出模式指针 如获取0x02,表示输出通道0模式为PWM模式

返回值：正常为0

备注：

int **SetDiReferenceVoltage** (void \*handle,

int iChannel,

int iVoltage);;

说明： 设置输入通道比较参考电压

参数： handle资源设备对应指针

iChannel通道号，范围0~31，0~7为一组比较电压参考值，8~15为一组比较电压参考值，16~23为一组比较电压参考值，24~31为一组比较电压参考值，如输入通道参数1与7，其实下发的是同一组比较电压值

iVoltage参考电压，单位毫伏，数值在0~25V之间

返回值：正常为0

备注：

int **GetDiLevel** (void \*handle, int iChannel, int\* iVoltage);

说明： 获取指定输入通道电平值

参数： handle资源设备对应指针

iChannel通道号

iVoltage输入电平，0为低，1为高

返回值：正常为0

备注：

int **GetDiAllLevel** (void \*handle, int\* iLevel);

说明： 获取所有输入通道输入电平

参数： handle资源设备对应指针

iLevel所有通道的输入电平，通道按位表示，0bit位为0通道，1为高电平，0为低电平，如获取0x01表示0通道为高电平输入，其余31个通道为低电平输入

返回值：正常为0

备注：

int **SetDoLevelConfigure** (void \*handle,

int iChannel,

DOLevelConfigure stDOLevelCfg);

说明： 设置指定输出通道电平配置

参数： handle资源设备对应指针

iChannel输出通道号，0~23

stDOLevelCfg输出电平配置结构

返回值：正常为0

备注：

int **SetDoLevel** (void \*handle, int iChannel, VoltageLevel eLevel);

说明： 设置指定输出通道电平值

参数： handle资源设备对应指针

iChannel通道号，范围0~23

eLevel 输出电平

返回值：正常为0

备注：

int **SetDiPWMCaptureEnable** (void \*handle,

int iChannel,

EnableStatus eStatus);

说明： 设置指定输入通道PWM捕获使能

参数： handle资源设备对应指针

iChannel通道号，范围0~31

eStatus 使能标志

返回值：正常为0

备注：

int **SetDiPWMAllCaptureEnable** (void \*handle, unsigned int iStatus);

说明： 设置所有输入通道PWM捕获使能

参数： handle资源设备对应指针

iStatus所有输入通道PWM采集使能，0bit位为0通道，1为使能，0为不使能，如0x03表示输入通道0、1PWM捕获使能，其余30个通道捕获未使能

返回值：正常为0

备注：

int **GetDiPWMAllCaptureEnable** (void \*handle, unsigned int\* pStatus);

说明： 获取所有输入通道PWM捕获使能状态

参数： handle资源设备对应指针

pStatus所有输入通道PWM采集使能状态指针，0Bit位0通道，1位使能，0为不使能，如0x01表示输入通道0为捕获使能状态，其余输入通道未捕获使能

返回值：正常为0

备注：

int **SetDoPWMConfigure** (void \*handle,

int iChannel,

DOPWMConfigure stDOPWMCfg);

说明： 设置指定输出通道PWM波形配置

参数： handle资源设备对应指针

iChannel输出通道号，范围0~23

stDOPWMCfg输出pwm配置数据结构

返回值：正常为0

备注：

int **SetDiPWMConfigure** (void \*handle,

int iChannel,

DIPWMConfigure stDIPWMCfg);

说明： 设置指定输入通道PWM波形采集设置

参数： handle资源设备对应指针

iChannel通道号，范围0~31

stDIPWMCfg输入pwm配置数据结构

返回值：正常为0

备注：

int **GetDiPWMCapture** (void \*handle, PWMProperty arrPWMProper[32]);

说明： 获取所有输入通道PWM波形采集数据

参数： handle资源设备对应指针

arrPWMProper波形数据结构体

返回值：正常为0

备注：

int **ClearDoOverProtectionStatus** (void \*handle);

说明： 清除板卡过流保护状态标志，过流指示灯恢复为绿色正常

参数： handle资源设备对应指针

返回值：正常为0

备注：

int **SetDoBitConfigure** (void \*handle,

int iChannel,

DOBitConfigure stDOBitCfg);

说明： 设置指定输出通道BIT数据流配置

参数： handle资源设备对应指针

iChannel通道号，范围0~23

stDOBitCfg数据流配置结构

返回值：正常为0

备注：

int **WriteDoBITData** (void \*handle, uint32\_t\* pArr, int iLength);

说明： 指定DIO卡发送BIT数据流

参数： handle资源设备对应指针

pArr数据流缓冲区首地址

iLength数据流缓冲区长度，长度须为2的整数倍,

返回值：正常为0

备注：数据流为uint32\_t类型数组，每uint32\_t数据，表示在指定通道配置为bit流模式下，24路输出通道同时输出1bit数据。数据从低uint32\_t元素至高 uint32\_t依次输出，如已配置好0、1通道输出模式为bit流模式，下发数据0x03，0x01，0x02，0x01，表示通道0输出BIT流信息1101，通道1输出BIT流信息1010，如下图所示：



图 1Bit流

int **SetDioPpsUpdateCount**(void\* handle, uint32\_t sec);

说明： 设置PPS秒记数配置

参数： handle资源设备对应指针

sec pps秒计数值

返回值：正常为0

备注：

int **GetDioBitStreamSendTime** (void\* handle, uint64\_t \*mic\_sec);

说明： 获取BIT流发送时间戳

参数： handle资源设备对应指针

mic\_sec 时间戳微秒计数值

返回值：正常为0

备注：

int **GetDioPpsSecCount** (void\* handle, uint32\_t \*sec);

说明： 获取当前pps秒计数值

参数： handle资源设备对应指针

sec pps秒计数值

返回值：正常为0

备注：

* 1. 工作流程操作
     1. 设备初始化与反初始化



1

图 2设备初始化流程图

·

图 3设备反初始化流程图

* + 1. DIO电平输出

设置输出电平模式

SetDoMode

设置输出电平配置参数

，

包括阻抗和电压参考

SetDoLevelConfigure

设置输出电平高低数值

SetDoLevel

设置使能输出

SetDoChannelEnable

开始

结束

图 4电平输出流程图

* + 1. DIO电平输入

设置输入比较参考电压

SetDiReferenceVoltage

开始

结束

获取通道输入电平信号数值

getDiLevel

或

getDiAllLevel

成功

？

Y

N

图 5电平输入流程图

* + 1. DIO PWM波形输入

设置输入比较参考电压

SetDiReferenceVoltage

设置输入

PWM

配置参数

SetDiPWMConfigure

开始

结束

设置输入

PWM

捕获使能

SetDiPWMAllCaptureEnable

获取输入

PWM

波形信息

GetDiPWMCapture

成功

？

Y

N

图 6PWM输入流程图

* + 1. DIO PWM波形输出

设置输出

PWM

模式

setDoMode

设置输出

PWM

配置参

数

，

包括阻抗和电压参

考

，

占空比

，

频率等

setDoPWMConfigure

设置使能输出

setDoChannelEnable

开始

结束

图 7PWM输出流程图

* + 1. DIO Bit流信息输出

设置输出

BIT

流模式

setDoMode

设置输出

BIT

流配置参

数

，

包括阻抗等

setDoBitConfigure

设置输出

BIT

流数据

writeDoBITData

开始

结束

图 8BIT流输出流程图