

Instruments Playground SDK 说明书

目录

自定义常量:	2
函数:	5
设备函数:	5
模拟输入函数:	5
模拟输出函数:	8
电源函数:	10
校准模块:	10
静态数字 IO 模块:	11
数字输入模块:	11
数字输出模块:	14
通讯协议模块:	16
基础读写函数:	19

自定义常量：

```
//Function Return
typedef int FT_STATUS
```

FT_STATUS	Int	设备状态
FT_OK	0	指令成功
FT_INVALID_HANDLE	1	无效的句柄
FT_DEVICE_NOT_FOUND	2	设备未找到
FT_DEVICE_NOT_OPENED	3	设备未打开
FT_IO_ERROR	4	设备 IO 错误
FT_INSUFFICIENT_RESOURCES	5	数字输入通道
FT_INVALID_PARAMETER	6	无效的参数
FT_INVALID_BAUD_RATE	7	无效的波特率
FT_OTHER_ERROR	>=8	其他问题

```
// instrument states:
typedef BYTE RDState
```

RDState	Int	采集状态
RDStateReady	0	准备完成，可以进行各种操作
RDStateArmed	1	等待触发中
RDStateDone	2	完成采集
RDStateTriggered	3	完成触发
RDStateRunning	3	运行采集中
RDStateConfig	4	配置中
RDStateWait	7	等待触发

```
// acquisition modes:
typedef int RDACQMODE
```

RDACQMODE	Int	采集模式
RDACQMODESingle	0	单次采集
RDACQMODEScanShift	1	滑动采集
RDACQMODEScanScreen	2	扫描采集

```
// trigger source
typedef BYTE RDTRIGSRC
```

RDTRIGSRC	Int	触发源
RDTRIGSRCNone	0	无触发源=不触发
RDTRIGSRCDetectorAnalogInCH1	1	模拟输入通道 1
RDTRIGSRCDetectorAnalogInCH2	2	模拟输入通道 2
RDTRIGSRCAnalogOut1	3	模拟输出通道 1
RDTRIGSRCAnalogOut2	4	模拟输出通道 2
RDTRIGSRCDetectorDigitalIn	5	数字输入通道

RDTRIGSRCDigitalOut	6	数字输出通道
RDTRIGSRCExternal1	8	外部触发 1
RDTRIGSRCExternal2	9	外部触发 2

```
//trigger slope
```

```
typedef int RDTriggerSlope
```

RDTriggerSlope	Int	边沿触发模式
RDTriggerSlopeRise	0	上升沿触发模式
RDTriggerSlopeFall	1	下降沿触发模式
RDTriggerSlopeEdge	2	双边沿触发模式

```
// trigger mode:
```

```
typedef int RDTRIGTYPE
```

RDTRIGTYPE	Int	触发类型
RDTRIGTYPEEdge	0	边沿触发

```
// analog out signal types
```

```
typedef BYTE RDFUNC
```

RDFUNC	Int	模拟输出类型
RDFUNCDC	0	直流
RDFUNCSine	1	正弦
RDFUNCSquare	2	方波
RDFUNCTriangle	3	三角波
RDFUNCRampUp	4	上升锯齿波
RDFUNCRampDown	5	下降锯齿波
RDFUNCNoise	6	噪声
RDFUNCPulse	7	脉冲
RDFUNCTrapezium	8	梯形波
RDFUNCSinePower	9	幂正弦
RDFUNCCustom	10	自定义波形
RDFUNCPlay	31	连续波

```
// analog out channel node types
```

```
typedef int RDAAnalogOutNode
```

RDAAnalogOutNode	Int	输出节点
RDAAnalogOutNodeCarrier	0	载波
RDAAnalogOutNodeFM	1	FM 调制
RDAAnalogOutNodeAM	2	AM 调制

```
// digital io channel node types
```

```
typedef int RRDigitalOutType
```

RRDigitalOutType	Int	数字信号输出模式
RRDigitalOutTypePulse	0	脉冲模式
RRDigitalOutTypeCustom	1	自定义模式
RRDigitalOutTypeRandom	2	随机模式

```
typedef int RRDigitalOutIdle
```

RRDigitalOutIdle	Int	空闲电平
RRDigitalOutIdleInit	0	初始化
RRDigitalOutIdleLow	1	低电平
RRDigitalOutIdleHigh	2	高电平
RRDigitalOutIdleZet	3	高阻

函数：

设备函数：

Int RDEnumDeviceCount(DWORD* count)
描述： 扫描所有设备，返回设备数量。
参数： - count – 查找到的设备数量

int RDEnumDeviceInfo (int i, char* Description, char* SerialNumber, FT_HANDLE *handle)
描述： 根据设备索引返回对应设备的 SN 号和描述信息。
参数： - i – 设备索引，从 0 开始 - Description – 设备描述信息 - SerialNumber – 设备序列号 - handle – 设备句柄

int RDdeviceOpen(int i, FT_HANDLE *ftHandle)
描述： 根据设备索引打开对应设备，返回句柄。
参数： - i – 设备索引，从 0 开始 - ftHandle – 设备句柄

int RDdeviceClose(FT_HANDLE ftHandle)
描述： 根据设备句柄关闭设备。
参数： - ftHandle – 设备句柄

模拟输入函数：

int RDAalogInStatus(FT_HANDLE handle, RDState* sts)
描述： 获取设备的模拟输入模块的状态。
参数： - handle – 设备句柄

- sts – 状态类型

int RDAAnalogInFrequencySet(FT_HANDLE handle,int fre)

描述:

设置设备的模拟输入模块的采样频率。

参数:

- handle – 设备句柄
- fre – 频率(1~40MHz) 单位为 Hz。

int RDAAnalogInChannelRangeSet(FT_HANDLE handle,int channel,double range)

描述:

设置设备的模拟输入模块每个通道的电压量程。

参数:

- handle – 设备句柄
- channel – 通道号, 0 为通道 A, 1 为通道 B
- range – 量程, 5V 或者 25V,单位为 V

int RDAAnalogInChannelEnableSet(FT_HANDLE handle,int channel,bool enable)

描述:

设置设备的模拟输入模块每个通道的采样使能。

参数:

- handle – 设备句柄
- channel – 通道号, 0 为通道 A, 1 为通道 B
- enable – 0 为关闭采样, 1 为开启采样

int RDAAnalogInTriggerAutoTimeoutSet(FT_HANDLE handle, int timeout)

描述:

设置设备的模拟输入模块的自动触发超时功能。

参数:

- handle – 设备句柄
- timeout – 0 为永不超时, 1 为超时 1s… 单位为 s

int RDAAnalogInTriggerSourceSet(FT_HANDLE handle,RDTRIGSRC trigsrc)

描述:

设置设备的模拟输入模块的触发源。

参数:

- handle – 设备句柄
- trigsrc – 触发源参数, 常用 RDTRIGSRCNone ,RDTRIGSRCDetectorAnalogInCH1(2)

int RDAAnalogInTriggerTypeSet(FT_HANDLE handle, RDTRIGTYPE trigtype)

描述:

设置设备的模拟输入模块的触发类型。

参数:

- handle – 设备句柄
- trigtype – 触发类型参数, 常用边沿触发, RDTRIGTYPEEdge

int RDAAnalogInTriggerLevelSet(FT_HANDLE handle, double voltsLevel, double range)

描述:

设置设备的模拟输入模块的触发电压。

参数:

- handle – 设备句柄
- voltsLevel – 触发电压设置, 范围为: -5~5V 或 -25~+25 V
- range - 当前通道的量程 5V 或 25V

int RDAAnalogInTriggerConditionSet(FT_HANDLE handle, RDTriggerSlope trigcond)

描述:

设置设备的模拟输入模块的触发边沿。

参数:

- handle – 设备句柄
- trigcond – 触发边沿参数, 包括上边沿, 下边沿, 双向边沿

int RDAAnalogInBufferSizeSet(FT_HANDLE handle, int buffersize)

描述:

设置设备的模拟输入模块的缓存大小。

参数:

- handle – 设备句柄
- buffersize – 单次采样的每个通道的缓存区域大小, 可选值: 32, 64, 128...2048

int RDAAnalogInConfigure(FT_HANDLE handle, bool run)

描述:

设置设备的模拟输入模块的开启和关闭。

参数:

- handle – 设备句柄
- run – 0: 关闭模拟输入模块, 1: 开启模拟输入

int RDAAnalogInRead(FT_HANDLE handle, int ch, int size, double range, double *RxBuffer, uchar* CalArray, int *backsize)

描述:

设置设备的模拟输入模块的开启和关闭。

参数:

- handle – 设备句柄
- ch – 通道索引 0: 通道 A, 1: 通道 B

- size – 缓存大小 数值和 RDAAnalogInBufferSizeSet 设置值相同
- range – 量程大小 数值和 RDAAnalogInRangeSet 设置值相同
- RxBuffer – 采样结果数组 单位为 V
- CalArray – 校准数组，通过 RDCalibrationRead 获取
- backsize – 设备返回的采样结果数组的实际大小

模拟输出函数：

```
int RDAAnalogOutNodeEnableSet(FT_HANDLE handle,int ch, RDAAnalogOutNode node, bool enable)
```

描述：

设置设备的模拟输出模块的对应通道和节点的使能。

参数：

- handle – 设备句柄
- ch – 通道索引，0：模拟输出通道 A，1：模拟输出通道 B
- node – 节点索引
- enable – 0：使能模拟输出通道，1：关闭模拟输出通道

```
int RDAAnalogOutNodeFunctionSet(FT_HANDLE handle, int ch, RDAAnalogOutNode node, RDFUNC func)
```

描述：

设置设备的模拟输出模块的对应通道和节点的波形函数。

参数：

- handle – 设备句柄
- ch – 通道索引，0：模拟输出通道 A，1：模拟输出通道 B
- node – 节点索引
- func – 波形函数索引，分为简单波形函数，自定义波形函数和连续波

```
int RDAAnalogOutNodeFrequencySet(FT_HANDLE handle, int ch, RDAAnalogOutNode node, int hzFrequency)
```

描述：

设置设备的模拟输出模块的对应通道和节点的波形频率。

参数：

- handle – 设备句柄
- ch – 通道索引，0：模拟输出通道 A，1：模拟输出通道 B
- node – 节点索引
- hzFrequency – 波形频率，单位 Hz，范围 0-20MHz，对于简单函数为整个波形的周期频率，对于自定义和连续波为两个采样点的间隔频率

```
int RDAAnalogOutNodeOffsetAmpSet(FT_HANDLE handle, int ch, RDAAnalogOutNode node, double vOffset,double amp,uchar* CalArray)
```


描述:

设置设备的模拟输出模块的对应通道和节点的波形偏置和幅值。

参数:

- handle – 设备句柄
- ch – 通道索引, 0: 模拟输出通道 A, 1: 模拟输出通道 B
- node – 节点索引
- vOffset– 输出波形的直流偏置, 单位为 V。范围-5~5V
- amp – 输出波形的幅值, 单位为 V。范围-5~5V
- CalArray – 通过 RDCalibrationRead 获取的校准数组

int RDAAnalogOutNodeSymmetrySet(FT_HANDLE handle,int ch,RDAAnalogOutNode node,double percentageSymmetry)

描述:

设置设备的模拟输出模块的对应通道和节点的对称性。

参数:

- handle – 设备句柄
- ch – 通道索引, 0: 模拟输出通道 A, 1: 模拟输出通道 B
- node – 节点索引
- percentageSymmetry– 波形对称性百分比参数, 范围 0~100。大多数情况合适 50%。

int RDAAnalogOutNodePhaseSet(FT_HANDLE handle,int ch,RDAAnalogOutNode node,double degreePhase)

描述:

设置设备的模拟输出模块的对应通道和节点的初始相位。

参数:

- handle – 设备句柄
- ch – 通道索引, 0: 模拟输出通道 A, 1: 模拟输出通道 B
- node – 节点索引
- degreePhase – 波形初始相位角度, 范围 0~360°。

int RDAAnalogOutNodeDataSet(FT_HANDLE handle, int ch, RDAAnalogOutNode node, double *rgdData, int cdData)

描述:

设置设备的模拟输出模块的对应通道和节点的自定义波形数据。

参数:

- handle – 设备句柄
- ch – 通道索引, 0: 模拟输出通道 A, 1: 模拟输出通道 B
- node – 节点索引
- rgdData – 最大支持 255 个数据点, 范围-1~1 之间。
幅值和偏置通过 RDAAnalogOutNodeOffsetAmpSet 调节。
- cdData – rgdData 的实际大小

int RDAAnalogOutConfigure(FT_HANDLE handle, int ch, bool output)

描述:

设置设备的模拟输出模块的对应通道和节点的使能。

参数:

- handle – 设备句柄
- ch – 通道索引, 0: 模拟输出通道 A, 1: 模拟输出通道 B, 2: AB 通道同时设置
- output – 0: 关闭模拟输出通道输出, 1: 打开模拟输出通道输出

电源函数:

int RDAAnalogIOChannelNodeSet(FT_HANDLE handle, int ch, double value, uchar* CalArray)

描述:

设置设备的电源模块的对应通道的电压值。

参数:

- handle – 设备句柄
- ch – 通道索引, 0: 电源正向输出通道 A, 1: 电源负向输出通道 B
- value – 电压值, 正向输出为 0~5V, 负向输出为 -5~0V
- CalArray – 校准值, 通过 RDCalibrationsRead 获取

int RDAAnalogIOChannelEnableSet (FT_HANDLE handle, int ch, bool enable)

描述:

设置设备的电源输出模块的对应通道和节点的使能。

参数:

- handle – 设备句柄
- ch – 通道索引, 0: 电源正向输出通道 A, 1: 电源负向输出通道 B
- enable – 通道输出使能 0: 关闭输出, 1: 打开输出

校准模块:

int RDCalibrationWrite(FT_HANDLE handle, uchar *CalArray)

描述:

写入设备的校准模块的参数。写入成功后保存在 EEPROM 中。*通常不建议客户使用

参数:

- handle – 设备句柄
- CalArray – 校准参数数组, 范围 0~255, 大小为 14。具体排列顺序如下:
0~3: 模拟输出 A, B 通道的 Offset, 模拟通道 A, B 的 Amplitude。
4~5: 电源输出 A, B 通道的 Offset
6~9: 5V 量程下的模拟输入 A, B 通道的 Offset, 模拟通道 A, B 的 Amplitude。
10~13: 25V 量程下的模拟输入 A, B 通道的 Offset, 模拟通道 A, B 的 Amplitude。

int RDCalibrationRead(FT_HANDLE handle,uchar *CalArray)

描述:

读取设备的校准模块的参数。在进行模拟输入输出和电源控制时务必先使用该函数获取对应参数，并将参数数组整体传递给需要的函数即可。

参数:

- handle – 设备句柄
- CalArray – 校准参数数组，范围 0~255，大小为 14。

静态数字 IO 模块:

int RDDigitalIOOutputEnableSet (FT_HANDLE handle,uint outputs)

描述:

设置设备的数字 IO 的各个通道是否为输出通道。

参数:

- handle – 设备句柄
- outputs – 使能通道，LSB, 0xFF 表示全部设置为输出通道。

int RDDigitalIOOutputSet (FT_HANDLE handle,uint sets)

描述:

设置设备的数字 IO 的各个通道输出电平状态。

参数:

- handle – 设备句柄
- sets – 设置状态，LSB, 0xFF 表示全高电平。

int RDDigitalIOInputStatus (FT_HANDLE handle,uint *reads)

描述:

回读数字 IO 的 16 个通道电平状态。

参数:

- handle – 设备句柄
- reads – 回读结果，LSB, 0xFF 表示全高电平。

数字输入模块:

int RDDigitalInDividerSet(FT_HANDLE handle,uint div)

描述:

设置设备的数字输入的分频参数。

参数:

- handle – 设备句柄
- div – 分频。实际频率等于最大频率（40MHz）除以分频率。

int RDDigitalInBufferSizeSet(FT_HANDLE handle,int buffersize)

描述:

设置设备的数字输入的缓存大小。

参数:

- handle – 设备句柄
- buffersize – 缓存大小，可用值：32，64，128…2048。

int RDDigitalInChannelSet(FT_HANDLE handle,int channel)

描述:

设置设备的数字输入的使能通道一共 16 个，和数字输出，静态 IO 共用。

- handle – 设备句柄
- channel – 数字通道，LSB，范围：0x00~0xFF，0：关闭输入，1：打开输入。

int RDDigitalInTriggerSourceSet(FT_HANDLE handle,RDTRIGSRC trigsr)

描述:

设置设备的数字输入的触发源。

参数:

- handle – 设备句柄
- trigsr – 触发源，常用 RDTRIGSRCDetectorDigitalIn。

int RDDigitalInTriggerTypeSet(FT_HANDLE handle, RDTRIGTYPE type)

描述:

设置设备的数字输入的触发类型。

参数:

- handle – 设备句柄
- type – 触发类型，常用边沿触发，RDTRIGTYPEEdge。

int RDDigitalInTriggerSlopeSet(FT_HANDLE handle,RDTriggerSlope slope)

描述:

设置设备的数字输入的触发边沿方向。（用于非 RDTRIGSRCDetectorDigitalIn 的情况）

参数:

- handle – 设备句柄
- slope – 触发边沿方向。

int RDDigitalInTriggerSet(FT_HANDLE handle,uint Rais,uint Fall)

描述:

设置设备的数字输入的 RDTRIGSRCDetectorDigitalIn 时候的各个通道触发边沿开关。

参数:

- handle – 设备句柄
- Rais – 上边沿开关。LSB, 0x11 使能数字 1 和 8 通道的上升沿触发。
- Fall – 下边沿开关。LSB, 0x11 使能数字 1 和 8 通道的下降沿触发。

int RDDigitalInTriggerTimeoutSet(FT_HANDLE handle, int enable)

描述:

设置设备的数字输入的触发超时。

参数:

- handle – 设备句柄
- enable– 超时使能, 0: 永不超时, N>1: 超时 N 秒。

int RDDigitalInConfigure(FT_HANDLE handle, bool enable)

描述:

设置设备的数字输入的采集开关。

参数:

- handle – 设备句柄
- enable – 0: 关闭采集, 1: 开始采集。

int RDDigitalInStatus(FT_HANDLE handle, RDState* sts)

描述:

设置设备的数字输入的状态读取。

参数:

- handle – 设备句柄
- sts – 状态, 0x02 为可以读取数据。

int RDDigitalInRead(FT_HANDLE handle, int size, UINT16* RxBuffer, int *backsize)

描述:

设置设备的数字输入的缓存大小。

参数:

- handle – 设备句柄
- size – 缓存大小 数值和 RDAalogInBufferSizeSet 设置值相同
- RxBuffer – 数字信号结果, LSB, 每一位代表一个通道, 1 为高电平, 0 为低电平。
- backsize – 设备返回的 RxBuffer 的实际大小

数字输出模块：

int RDDigitalOutRun(FT_HANDLE handle, bool enable)

描述：

设置设备的数字输出的输出使能。

参数：

- handle – 设备句柄
- enable – 输出使能，0：关闭输出，1：开启输出。

int RDDigitalOutTriggerSourceSet(FT_HANDLE handle, RDTRIGSRC trigsr)

描述：

设置设备的数字输出的触发源。

参数：

- handle – 设备句柄
- trigsr – 触发源。

int RDDigitalOutTriggerSlopeSet(FT_HANDLE handle, RDTriggerSlope slope)

描述：

设置设备的数字输出的触发边沿方向。

参数：

- handle – 设备句柄
- slope – 边沿类型，上边沿，下边沿，双向边沿。

int RDDigitalOutTypeSet(FT_HANDLE handle, int ch, RDDigitalOutType type)

描述：

设置设备的数字输出的触发类型。

参数：

- handle – 设备句柄
- type – 触发类型，常用边沿触发，RDTRIGTYPEEdge。

int RDDigitalOutEnableSet(FT_HANDLE handle, uint chs)

描述：

设置设备的数字输出的通道使能。

参数：

- handle – 设备句柄
- chs – 通道使能，共 16 个，LSB，0x00~0xFF。例如 0x0F,使能 1~8 通道

int RDDigitalOutIdleSet(FT_HANDLE handle, int ch, RDDigitalOutIdle idle)

描述：

设置设备的数字输出的空闲状态。

参数：

- handle – 设备句柄

- idle – 空闲状态。常用低电平。

int RDDigitalOutDividerInitSet(FT_HANDLE handle,int ch, uint divinit)

描述:

设置设备的数字输出的分频初始数。用于延时信号输出，单位为最大频率的倒数。

参数:

- handle – 设备句柄
- ch – 通道号，LSB，0x00~0xFF。
- divinit – 分频初始数。N: N/MaxFre 秒后发送信号。

int RDDigitalOutDividerSet(FT_HANDLE handle,int ch, uint div)

描述:

设置设备的数字输出的分屏。

参数:

- handle – 设备句柄
- ch – 通道号，LSB，0x00~0xFF。
- div – 分频。实际输出的数值信号频率等于最大频率除以分频率。

int RDDigitalOutCounterInitSet(FT_HANDLE handle,int ch,int initlevel,uint counter)

描述:

设置设备的数字输出的初始电平和持续时间。

参数:

- handle – 设备句柄
- ch – 通道号，LSB，0x00~0xFF。
- initlevel – 初始电平，0: 低电平，1: 高电平。
- counter – 持续时间的个数，最小周期（25ns）乘以个数为实际持续时间。

int RDDigitalOutCounterSet(FT_HANDLE handle,int ch, uint counter_l,uint counter_h)

描述:

设置设备的数字输出的高低电平的持续时间。

参数:

- handle – 设备句柄
- ch – 通道号，LSB，0x00~0xFF。
- counter_l – 低电平持续时间的个数，最小周期（25ns）乘以个数为实际持续时间。
- counter_h – 高电平持续时间的个数，最小周期（25ns）乘以个数为实际持续时间。

通讯协议模块：

int RDDigitalUartRateSet(FT_HANDLE handle, double rate)

描述：

设置设备的串口通信速率。

参数：

- handle – 设备句柄
- rate – 串口波特率，可选：110,150,300,600,1200,2400,4800,9600,14400,19200,28800,38400,56000,57600,115200,128000,153600,230400,256000,460800,921600。

int RDDigitalUartTxRxIOSet(FT_HANDLE handle, int TxDIO, int RxDIO)

描述：

设置设备的串口的 TX,RX 接口通道号。

参数：

- handle – 设备句柄
- TxDIO – 发送接口，0：ch1…15:ch16。
- RxDIO – 接收接口，0：ch1…15:ch16。

int RDDigitalUartTx(FT_HANDLE handle, uchar* data, int length)

描述：

设置设备的串口的 TX 输出数据，并发送。

参数：

- handle – 设备句柄
- data – 发送数据，单次最长支持 255 个。
- length – 数据长度。

int RDDigitalUartRx(FT_HANDLE handle, uchar* rgRX, int length, int * rxcount, int *fParity)

描述：

回读设备的串口 RX 数据。

参数：

- handle – 设备句柄
- length – 缓存 rgRX 的长度
- rgRX – 串口数据结果。
- rxcount– 设备返回的 rgRX 的实际长度
- fParity – 奇偶校验


```
int RDDigitalI2CRateSet(FT_HANDLE handle, double rate)
```

描述:

设置设备的 I2C 通信速率。

参数:

- handle – 设备句柄
- rate – 串口波特率，可选：1,4,10,40,100,400,1000,4000,1e4,4e4,1e5,4e5,1e6,4e6,1e7。

```
int RDDigitalI2CTxRxIOSet(FT_HANDLE handle, int SCLDIO, int SDADIO)
```

描述:

设置设备的 I2C 的 SCL 和 SDA 接口通道号。

参数:

- handle – 设备句柄
- SCLDIO – 时钟接口，0: ch1…15:ch16。
- SDADIO – 信号接口，0: ch1…15:ch16。

```
int RDDigitalI2CTx(FT_HANDLE handle, uchar addr, uchar* data, int length)
```

描述:

设置设备的 I2C 的 TX 输出数据，并发送。

参数:

- handle – 设备句柄
- data – 发送数据，单次最长支持 255 个。
- length – 数据长度。

```
int RDDigitalI2CRx(FT_HANDLE handle, uchar addr, uchar* rgRX, int length, int *  
rxcount, int *fParity)
```

描述:

回读设备的 I2C 的 RX 数据。

参数:

- handle – 设备句柄
- length – 缓存 rgRX 的长度
- rgRX – I2C 数据结果。
- rxcount– 设备返回的 rgRX 的实际长度
- fParity – 奇偶校验

int RDDigitalSPIRateSet(FT_HANDLE handle, double rate)
<p>描述： 设置设备的 SPI 通信速率。</p> <p>参数：</p> <ul style="list-style-type: none"> - handle – 设备句柄 - rate – 串口波特率，可选：1,4,10,40,100,400,1000,4000,1e4,4e4,1e5,4e5,1e6,4e6,1e7
int RDDigitalSPITxRxIOSet(FT_HANDLE handle, int CSDIO, int ClockDIO, int DQ0, int DQ1)
<p>描述： 设置设备的 SPI 的 CS,Clock,DQ0,DQ1 接口通道号。</p> <p>参数：</p> <ul style="list-style-type: none"> - handle – 设备句柄 - CSDIO – 片选接口，0: ch1…15:ch16。 - ClockDIO – 时钟接口，0: ch1…15:ch16。 - DQ0 – 数据 0 接口，0: ch1…15:ch16。 - DQ1 – 数据 1 接口，0: ch1…15:ch16。
int RDDigitalSPITx(FT_HANDLE handle, uchar* data, int length)
<p>描述： 设置设备的 SPI 的 TX 输出数据，并发送。</p> <p>参数：</p> <ul style="list-style-type: none"> - handle – 设备句柄 - data – 发送数据，单次最长支持 255 个。 - length – 数据长度。
int RDDigitalSPIRx(FT_HANDLE handle, uchar* rgRX, int length, int * rxcount, int *fParity)
<p>描述： 回读设备的 SPI 的 RX 数据。</p> <p>参数：</p> <ul style="list-style-type: none"> - handle – 设备句柄 - length – 缓存 rgRX 的长度 - rgRX – SPI 数据结果。 - rxcount – 设备返回的 rgRX 的实际长度 - fParity – 奇偶校验

基础读写函数：

```
int write(FT_HANDLE handle, uchar *data, int count)
```

描述：

对设备进行基础读写操作，需要对应指令才会有响应。

参数：

- handle – 设备句柄
- data – 指令数组
- count – 指令数组长度。

```
int read(FT_HANDLE handle, uchar *data, int *count, int msTimeout=0)
```

描述：

对设备进行基础读操作。

参数：

- handle – 设备句柄
- data – 指令数组
- count – 指令数组长度。
- msTimeout – 延时等待，0 为不延时，单位 ms。