



## FT\_USB\_820X\_DAQ\_DEMO\_LABVIEW

版本历史			
版本号	发布者	日期	描述
V20211227A0	曹璐	2021-12-27	首次发布

目录

1. FT\_USB\_820x\_DAQ AO AI 测试..... 2

2. FT\_USB\_820x\_DAQ DIO 测试..... 9

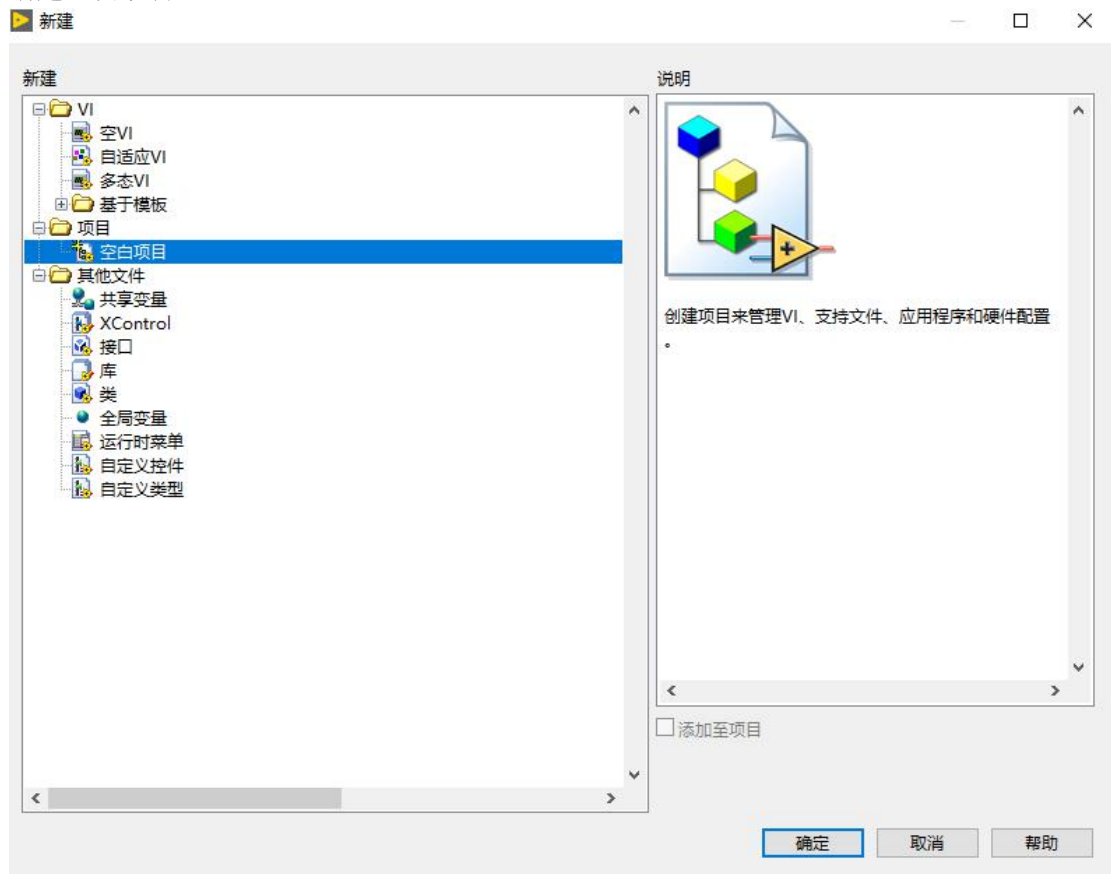
3. FT\_USB\_820x\_DAQ PFI Counter 测试..... 11

## 1. FT\_USB\_820x\_DAQ AO AI 测试

usb DAQ 使用 AO 功能输出一个 1KHz 的正弦波，然后使用 AI 功能采集并展示波形  
使用 usb DAQ 需要调用 dll 函数，并根据驱动使用文档，使用 dll 的中的静态方法

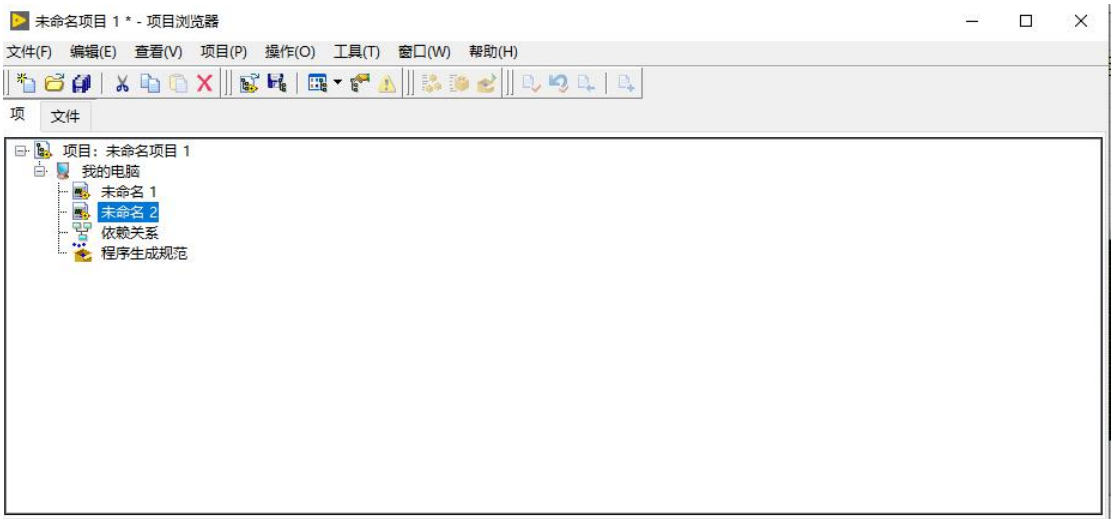


新建空白项目



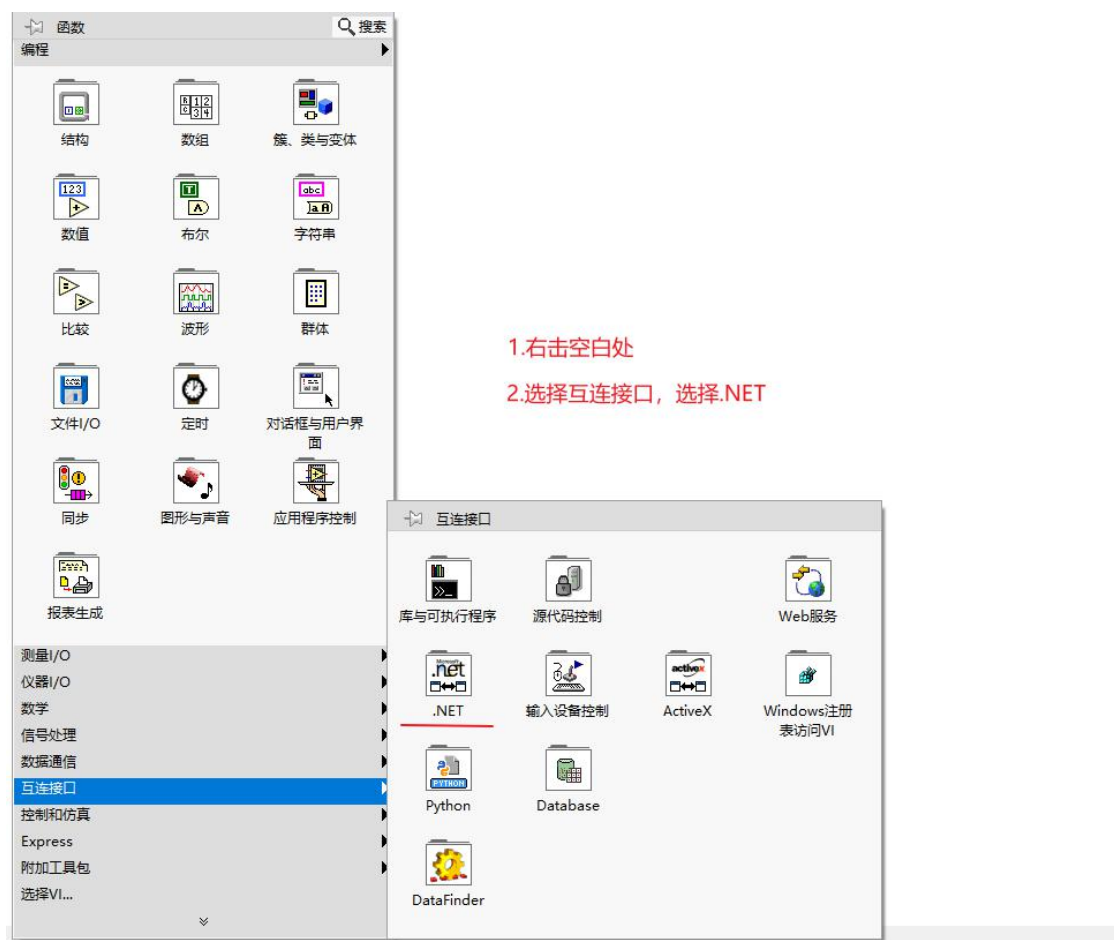
新建 vi 文件

LABVIEW 项目

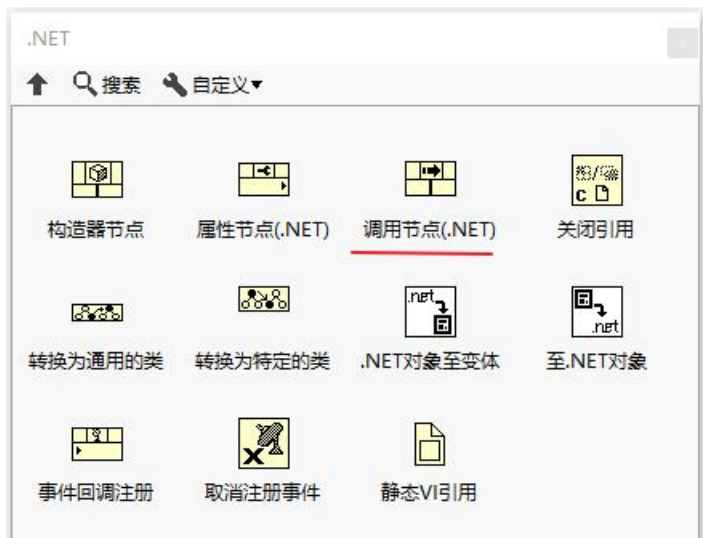


点击进入 vi 文件 显示程序框图

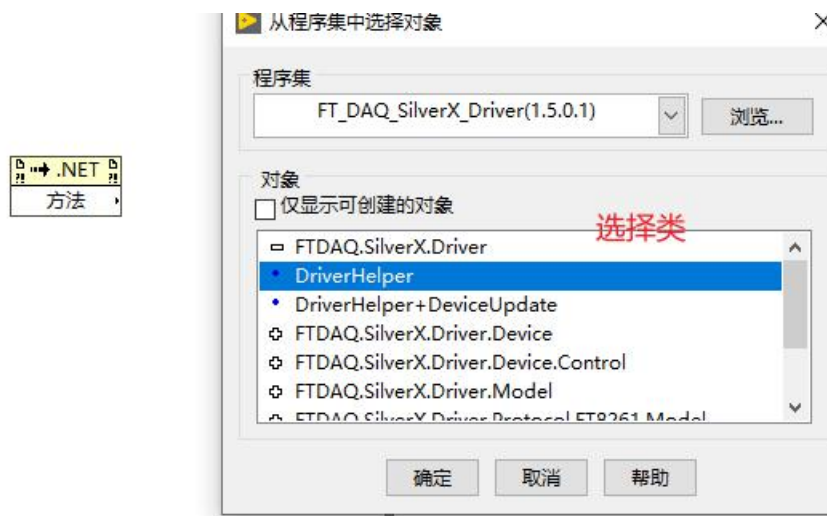
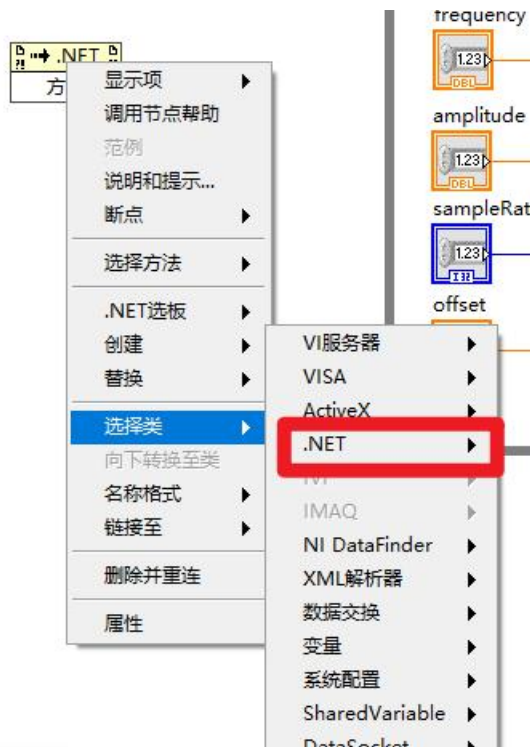


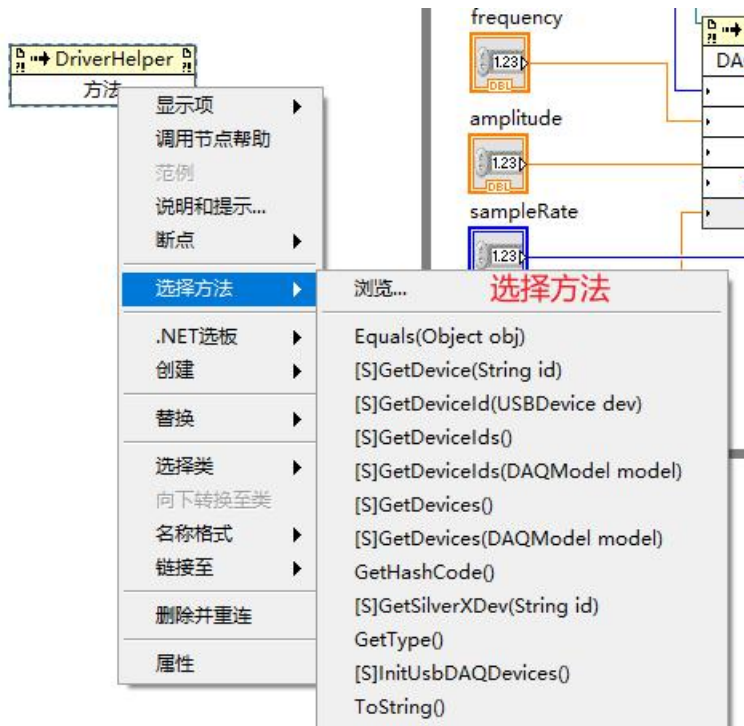


选择调用节点

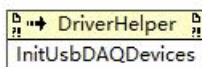


右击调用节点，点击.NET





最终生成静态的使用方法 `DriverHelper.InitUsbDAQDevices`



此 demo 使用到方法

## InitUsbDAQDevices

```
/// <summary>
/// 初始化所有 USB 设备
/// </summary>
public static void InitUsbDAQDevices()
```

## GetSilverXDev

```
/// <summary>
/// 根据 silverX 设备 ID 获取设备信息
/// </summary>
/// <param name="id">silverX 系列 DAQ 的 ID</param>
/// <returns></returns>
public static SilverXDevice GetSilverXDev(string id)
```

## GetSilverXDev

```

/// <summary>
/// 根据 silverX 设备 ID 获取设备信息
/// </summary>
/// <param name="id">silverX 系列 DAQ 的 ID</param>
/// <returns></returns>
public static SilverXDevice GetSilverXDev(string id)

```

## DACGenerateSine

```

/// <summary>
/// DAQ 输出正弦波信号
/// </summary>
/// <param name="channel">需要输出波形的通道</param>
/// <param name="frequency">波形频率</param>
/// <param name="amplitude">波形峰值</param>
/// <param name="sampleRate">采样率</param>
/// <param name="offset">直流偏移</param>
public void DACGenerateSine(int channel, double frequency, double amplitude, int
sampleRate, double offset = 0)

```

## ADCSampleDatas

```

/// <summary>
/// ADC 采集指定长度的数据
/// </summary>
/// <param name="channel">需要采集数据的通道</param>
/// <param name="sampleRate">采样率</param>
/// <param name="samplePoints">需要采集的点数</param>
/// <param name="range">量程</param>
/// <returns></returns>
public List<double> ADCSampleDatas(int channel, int sampleRate, int samplePoints,
VoltRange range)

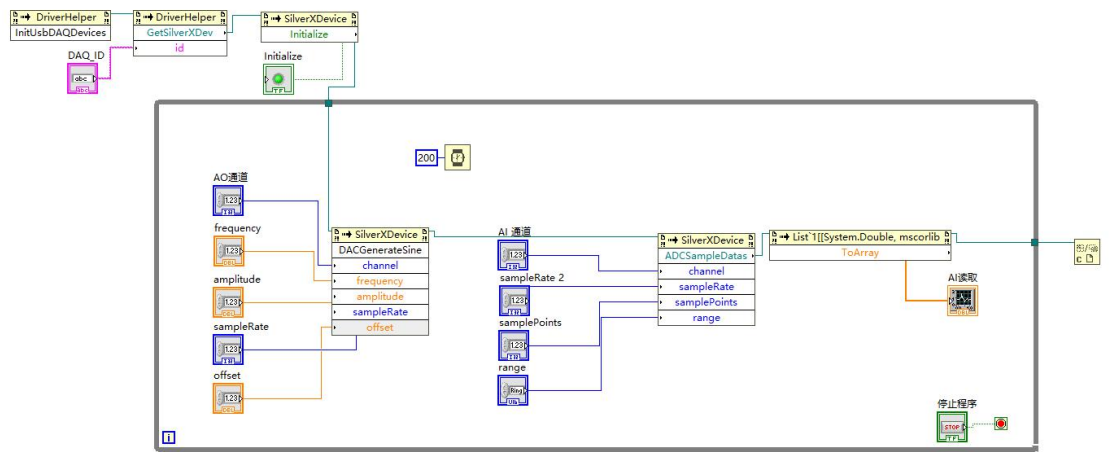
```

程序逻辑:

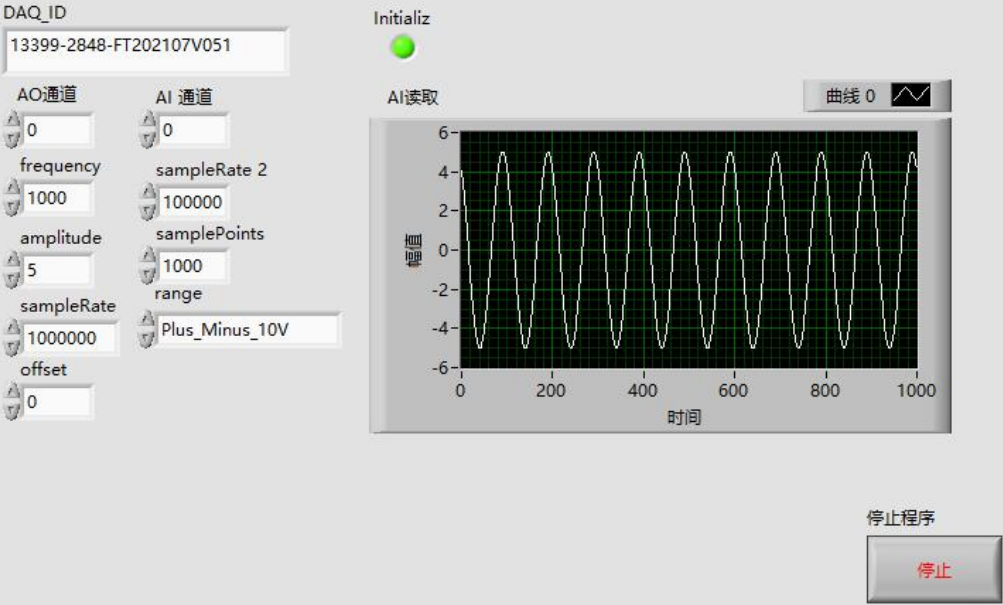
1. USB DAQ 必须初始所有 USB DAQ 设备（用于监听所有 USB 设备），需要调用 DriverHelper.InitUsbDAQDevices
2. 根据 HWSuit 获得的 ID，实例化 DAQ 设备
3. 新设备插入后（或者软件重新打开），**不会**自动初始化，需要调用 SilverXDevice.Initialize() 方法进行初始化
4. 程序进入循环中，输出波形，设置采样率，频率等
5. 采集波形，设置采样率，采集点数等

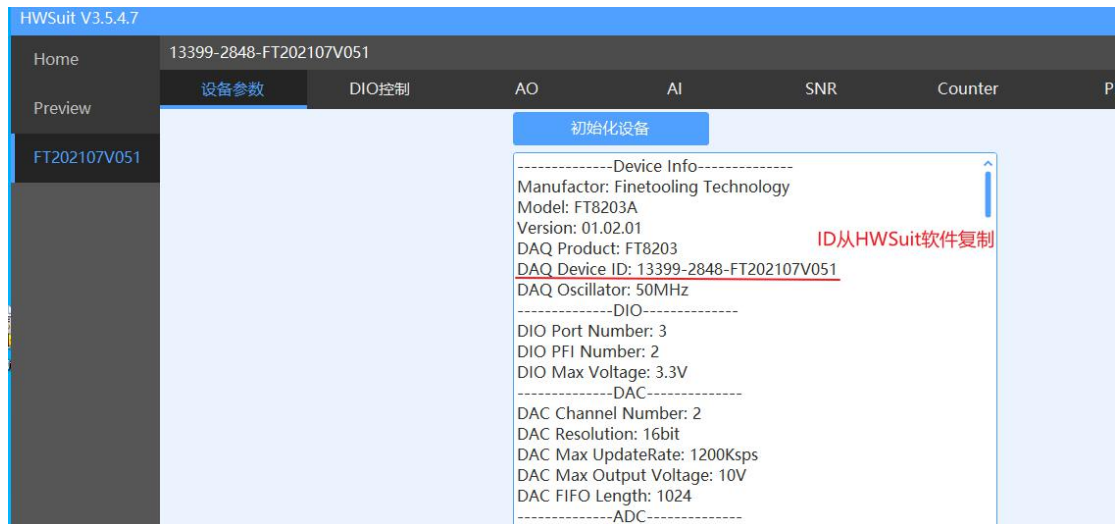


最终的程序框图



说明：使用AO功能输出一个1KHz的正弦波，然后使用AI功能采集并展示波形，按停止按钮后停止测试





## 2. FT\_USB\_820x\_Daq DIO 测试

使用 DIO 功能(Port0.0)输出高电平，使用 Port0.1 读取

操作与第八章相同

使用到方法

### InitUsbDAQDevices

```
/// <summary>
/// 初始化所有 USB 设备
/// </summary>
public static void InitUsbDAQDevices()
```

### GetSilverXDev

```
/// <summary>
/// 根据 silverX 设备 ID 获取设备信息
/// </summary>
/// <param name="id">silverX 系列 DAQ 的 ID</param>
/// <returns></returns>
public static SilverXDevice GetSilverXDev(string id)
```

## DIOWritePortLine

```

/// <summary>
/// DIO 按位输出，该方法会将指定引脚设置为输出
/// </summary>
/// <param name="port">要操作 DIO 的端口</param>
/// <param name="line">要操作的 IO 脚: 0-7</param>
/// <param name="level">要输出的值: Low:低电平, High: 高电平</param>
public void DIOWritePortLine(int port, int line, DIOLevel level)

```

## DIOReadPortLine

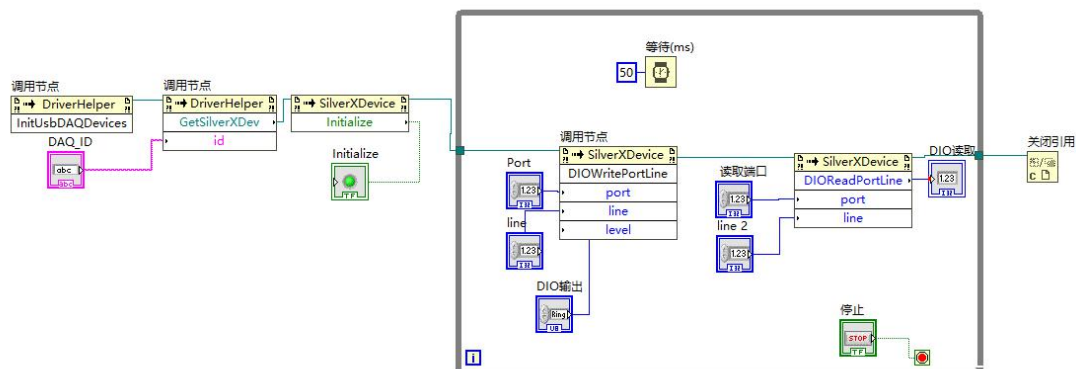
```

/// <summary>
/// 按位读取 DIO 状态，该方法会将指定引脚设置为输入
/// </summary>
/// <param name="port">要操作 DIO 的端口</param>
/// <param name="line">要操作的 IO 脚: 0-7</param>
/// <returns>0: 低电平; 1: 高电平</returns>
public DIOLevel DIOReadPortLine(int port, int line)

```

程序逻辑:

1. USB DAQ 必须初始所有 USB DAQ 设备，需要调用 DriverHelper.InitUsbDAQDevices
2. 根据 HWSuit 获得的 ID，实例化 DAQ 设备
3. 新设备插入后（或者软件重新打开），**不会**自动初始化，需要调用 SilverXDevice.Initialize() 方法进行初始化
4. 使用 DIO 功能，设置高低电平，port 口读取





### 3. FT\_USB\_820x\_DAQ PFI Counter 测试

说明：p0.0/p0.1 PFI 输出方波，CTR0 /CTR1 计数  
使用到方法

#### InitUsbDAQDevices

```
/// <summary>
/// 初始化所有 USB 设备
/// </summary>
public static void InitUsbDAQDevices()
```

#### GetSilverXDev

```
/// <summary>
/// 根据 silverX 设备 ID 获取设备信息
/// </summary>
/// <param name="id">silverX 系列 DAQ 的 ID</param>
/// <returns></returns>
public static SilverXDevice GetSilverXDev(string id)
```

## PFIStartSquare

```
/// <summary>
/// PFI 输出指定频率和占空比的方波
/// </summary>
/// <param name="port">PFI 的 port</param>
/// <param name="line">PFI 的 line</param>
```

---

```
/// <param name="frequency">输出方波频率, 单位 Hz</param>
/// <param name="dutyRate">输出方波频率的占空比, 值范围: 0~1</param>
public void PFIStartSquare(int port, int line, int frequency, double dutyRate = 0.5)
```

## CounterReadEdge

```
/// <summary>
/// 计数器计算沿跳变数量
/// </summary>
/// <param name="channel">计数器通道</param>
/// <param name="edge">需要计数的沿跳变: 上升沿或下降沿</param>
/// <param name="timeMS">计数器计数时长, 单位: 毫秒(ms)</param>
```

---

```
/// <returns>统计到的沿跳变个数</returns>
public uint CounterReadEdge(int channel, CTEdge edge, int timeMS)
```

## CounterReadPeriod

```

/// <summary>
/// 计数器计算波形周期
/// </summary>
/// <param name="channel">计数器通道</param>
/// <param name="timeMS">计数时长, 单位: 毫秒(ms)</param>
/// <param name="periodCount">设置需要计算的周期个数, 该值越大, 计算结果越准确</param>
/// <returns>返回周期长度, 单位是 us</returns>
public double CounterReadPeriod(int channel, int timeMS, uint periodCount = 50)

```

## CounterReadPulseWidth

```

/// <summary>
/// 计数器计算波形脉宽
/// </summary>
/// <param name="channel">计数器通道</param>
/// <param name="ctmode">脉宽类型: 正脉宽/负脉宽</param>
/// <param name="timeMS">计数时长, 单位: 毫秒(ms)</param>
/// <returns>返回脉宽长度, 单位是 us</returns>
public double CounterReadPulseWidth(int channel, CTMode ctmode, int timeMS)

```

程序逻辑:

1. USB DAQ 必须初始所有 USB DAQ 设备, 需要调用 `DriverHelper.InitUsbDAQDevices`
2. 根据 HWSuit 获得的 ID, 实例化 DAQ 设备
3. 新设备插入后 (或者软件重新打开), **不会** 自动初始化, 需要调用 `SilverXDevice.Initialize()` 方法进行初始化
4. Port0.0 使用 PFI 功能输出方波
5. Port0.0 IO 接入 CTR0 口
6. 使用 Counter 方法计数

LABVIEW 项目

