**光谱仪操作时序使用说明**

**一 多通道光谱扫描**

* SA\_SetMultiChannelIntegrationTime

设置通道的积分时间，一次性设置好所有通道的积分时间

* SA\_GetMultiChannelSpectum

获取当前通道的光谱数据

**二 异步光谱扫描**

* SA\_ScanStartAsyncSoftTrigger

启动异步光谱扫描，积分时间结束函数就返回成功

* SA\_GetStateAsyncSoftTrigger

获取光谱仪目前状态，当返回的状态0x00表示扫描完成可以获取光谱数据

* SA\_GetSpectum

获取光谱

ret = SA\_ScanStartAsyncSoftTrigger(0);

if(ret != SA\_API\_SUCCESS)

{

printf("SA\_AsynchronousScanStart Fail\r\n");

}

while(1)

{

ret = SA\_GetStateAsyncSoftTrigger(0, &State);

if(ret != SA\_API\_SUCCESS)

{

printf("SA\_AsynchronousGetState Fail\r\n");

}

printf("State = %d\r\n", State);

if(State == 0x00)

{

break;

}

}

ret=SA\_GetSpectum(0, dSpectumData, &iSpSpectumDataLength);

if(SA\_API\_SUCCESS != ret)

{

printf("SA\_GetSpectum Fail\r\n");

}

**三 多通道异步光谱扫描**

* SA\_ScanStartMultiChannelAsyncSoftTrigger

启动多通道异步光谱扫描，积分时间结束函数就返回成功

* SA\_GetStateAsyncSoftTrigger

获取光谱仪目前状态

* SA\_GetSpectum

获取光谱

**四 异步存储扫描**

* SA\_ScanStartCacheAsyncTrigger

启动一个通道的异步存储扫描，积分时间结束函数就返回成功，并且把光谱存储在系统内存中。

* SA\_GetCacheAsyncSpectum

获取对应通道的光谱数据