

# DSO2250/2150/2090 Second Design Manual

## 1 . Functions introduction

### Define struct:

```
struct Control_Data1_Struct{  
    WORD Trig_Source;           //Trigger Source  
    WORD Ch_select;             //Channel Select  
    WORD TIMEBASE;              //TimeBase  
    WORD Trig_Addr;             //Address of trigger point  
    WORD Data_Length;           //Length of data  
    WORD Data_BufferSize;       //Buffer Size  
    WORD Alt_flag;              //Remark of ALT  
};
```

```
struct Levers_Struct{  
    WORD Ch1Lever;              //Position of Channel 1  
    WORD Ch2Lever;              //Position of Channel 2  
    WORD Ch1TrigLever;          //Position of Channel 1 trigger  
    WORD Ch2TrigLever;          //Position of Channel 2 trigger  
    WORD ExtTrigLever;          //Position of extern trigger  
};
```

### Functions for hardware

#### 1.

```
WORD __stdcall dsoSetTriggerAndSampleRate (  
    WORD DeviceIndex, //Device Index  
    Struct Control_Data1_Struct *Control_Data1 )
```

Note: Init and set trigger and sample rate. If you change TimeBase, you must use this function. If successful return 1, else return 0.

2.

```
WORD __stdcall dsoSetFilt ( WORD DeviceIndex,  
                           WORD Ch1_Filt,      //Channel 1 filter  
                           WORD Ch2_Filt,      //Channel 2 filter  
                           WORD Trigger_Filt)  // Trigger filter
```

Note: Init and set channel or trigger filter. If successful return 1, else return 0.

3.

```
WORD __stdcall dsoSetVoltageAndCoupling (  
        WORD DeviceIndex,  
        WORD Ch1_Att,      //Channel 1 voltage  
        WORD Ch2_Att,      //Channel 2 voltage  
        WORD Ch1_ACDC,     //Channel 1 AC/DC  
        WORD Ch2_ACDC      //Channel 2 AC/DC,  
        WORD TriggerSource ) //Trigger source
```

Note: Init and set channel voltage and coupling. If successful return 1, else return 0.

4.

```
bool __stdcall dsoSetOffset ( WORD DeviceIndex,  
                             struct Levers_Struct * Levers, /* Channels and trigger  
                                                             lever data */  
                             WORD Ch1_Att,  
                             WORD Ch2_Att,  
                             WORD TriggerSource,  
                             WORD * level ) //Channels level data
```

Note: Init and set trigger lever and level. If successful return TRUE, else return FALSE.

5.

```
WORD __stdcall dsoGetCaptureState ( WORD DeviceIndex,  
                                     int *Ram_Addr ) //Address of trigger point
```

Note: Get address of trigger point. If successful return **2**, else return **0** or **1**.

6.

```
WORD __stdcall dsoGetChannelData ( WORD DeviceIndex,  
                                   WORD Ch1_Att,  
                                   WORD Ch2_Att,  
                                   WORD pos_trig, //Position of trigger point  
                                   WORD * ch1_data, //Channel 1 data  
                                   WORD * ch2_data, //Channel 2 data  
                                   struct Control_Data1_Struct *Control_Data1,  
                                   int Ram_Addr,  
                                   int calData ) //Calibration Data Offset
```

Note:

Variable: '**calData**': Calibration Data Offset. Range(0~10).

Get Channel1 and Channel 2 data. If successful return **1**, else return **0**.

7.

```
WORD __stdcall dsoCaptureStart ( WORD DeviceIndex )
```

Note: Start capturing data. If successful return **1**, else return **0**.

8.

```
long __stdcall dsoTriggerEnabled ( WORD DeviceIndex,  
                                   WORD Ch1_Att,  
                                   WORD Ch2_Att )
```

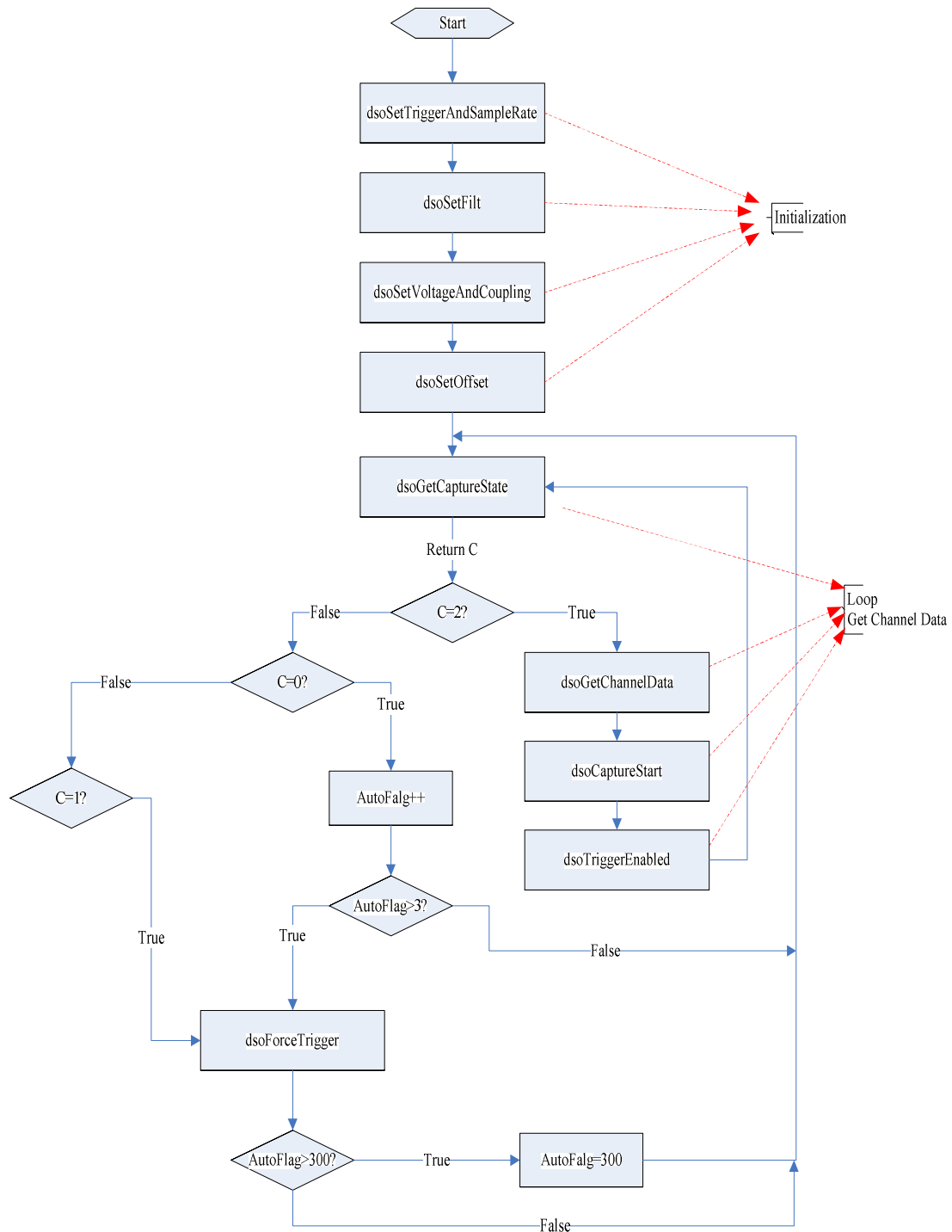
Note: Allow of next trigger. If the function succeeds, the return value is **nonzero**. If the function fails, the return value is **zero**.

9.

BOOL \_\_stdcall dsoForceTrigger ( WORD DeviceIndex )

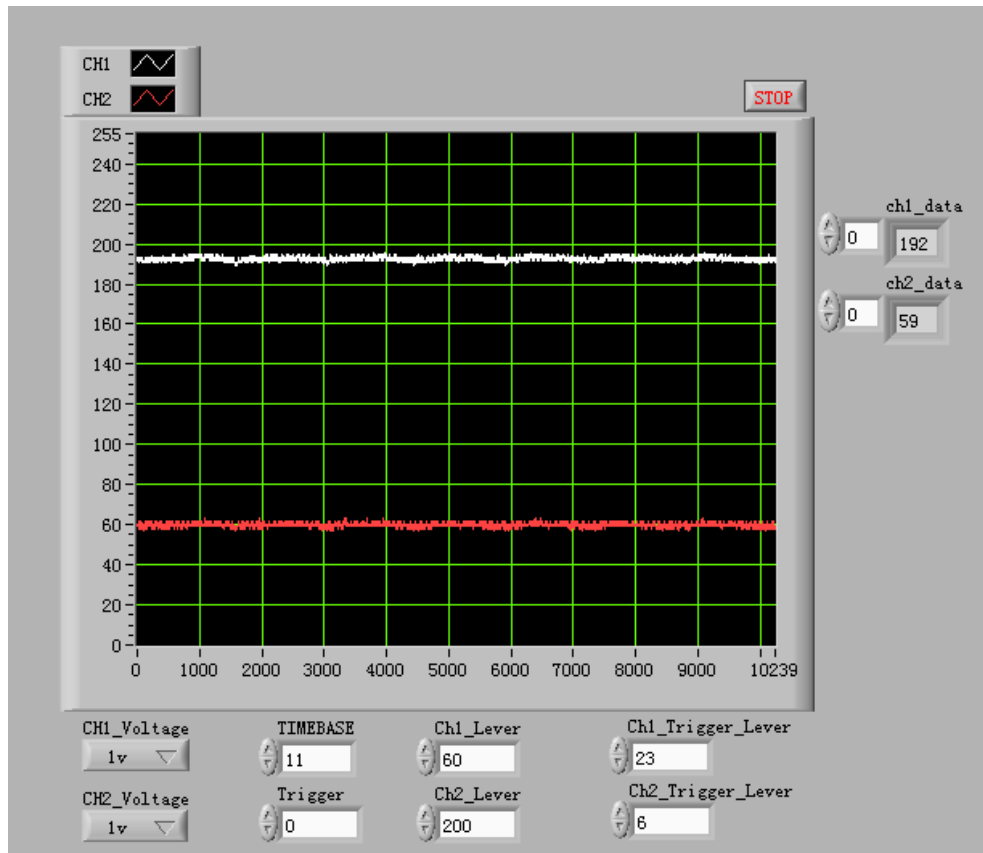
Note: If hardware does not trigger long time, this function will trigger one time. If the function succeeds, the return value is **nonzero**. If the function fails, the return value is **zero**.

## 2. Flow chart



### 3. LabVIEW Example

Image:

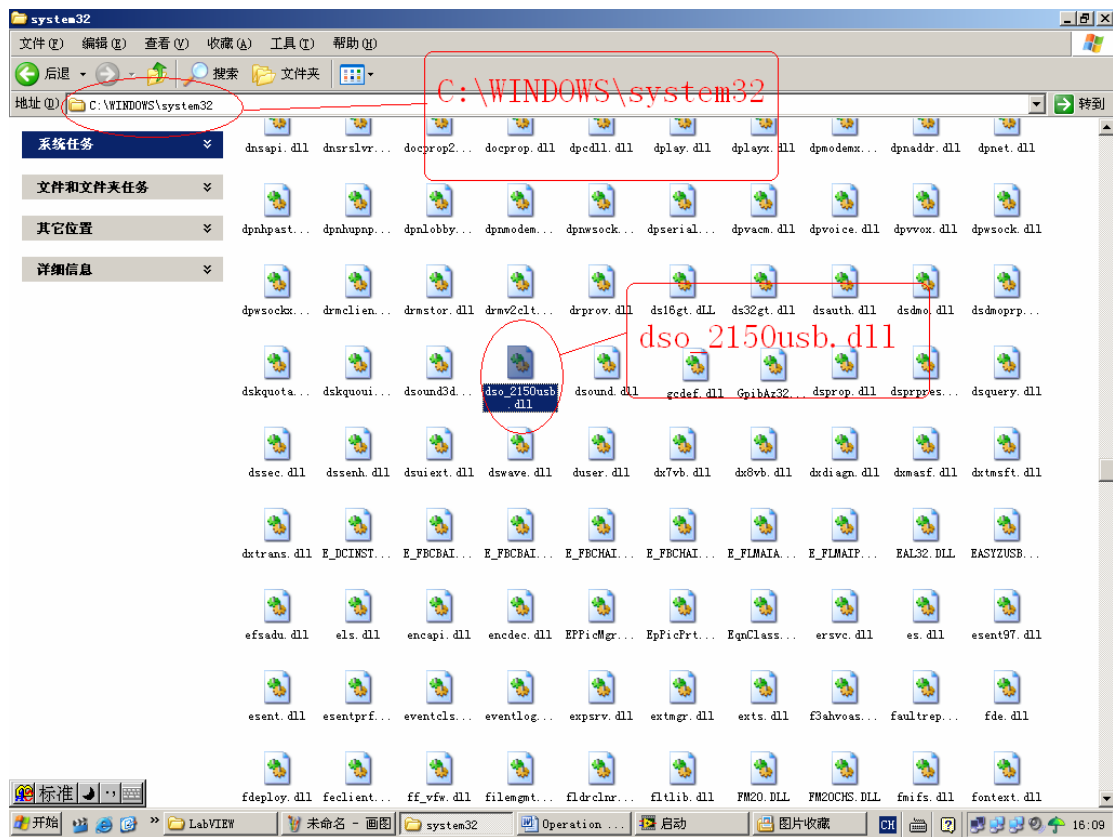


### Operation Manual

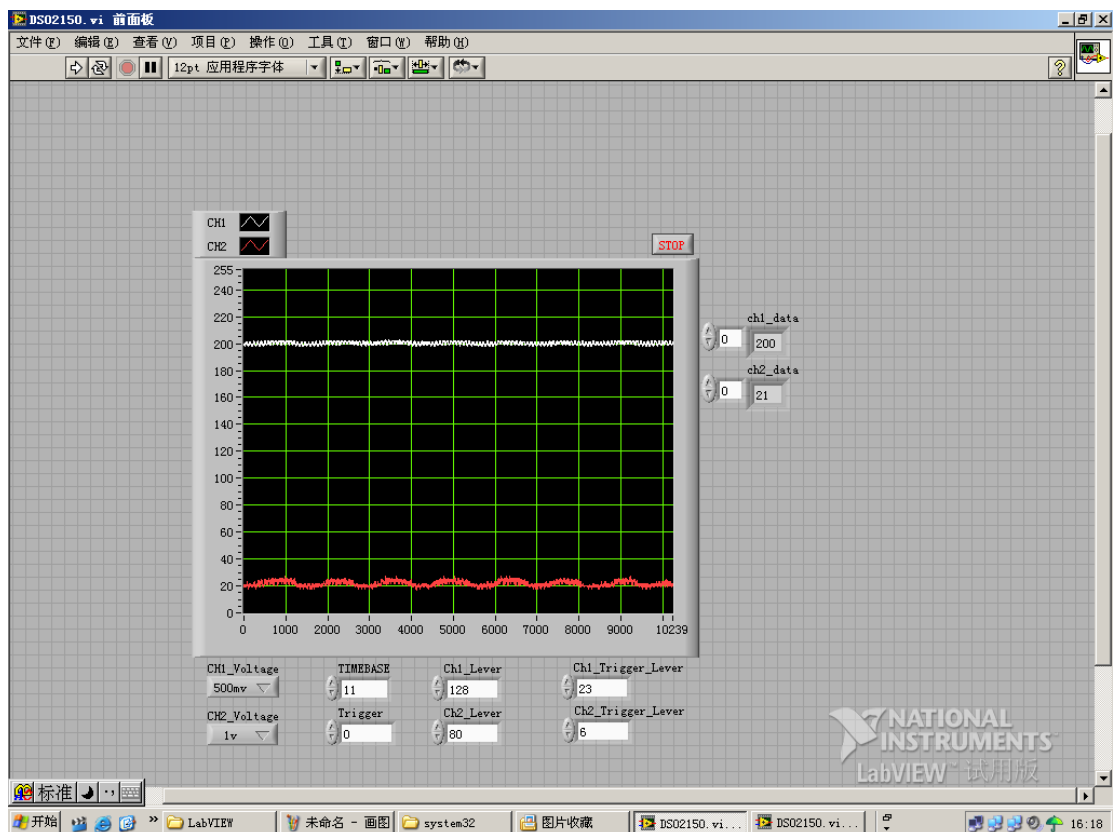
For example: **DSO2150 USB**

Running Step:

- 1 . Install labVIEW.
- 2 . Copy the file "**dso\_2150usb.dll**" to the path "C:\WINDOWS\system32".

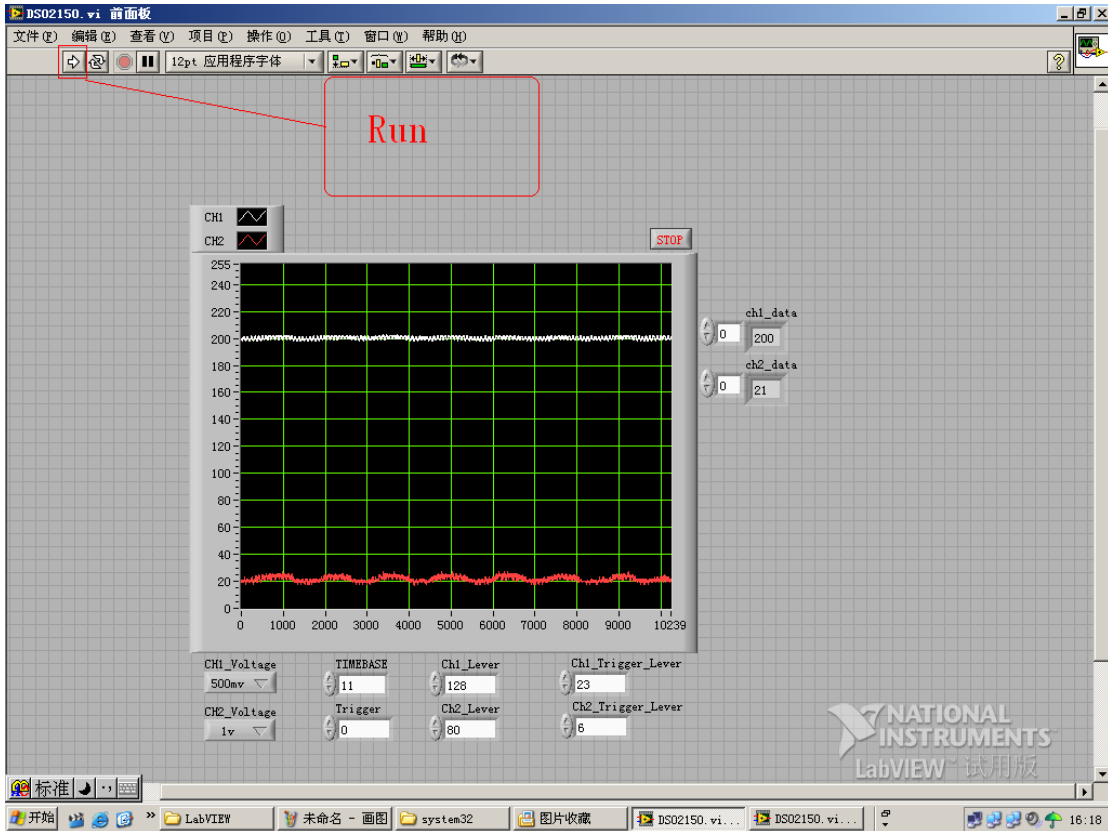


3 . Click "DSO2150.vi".

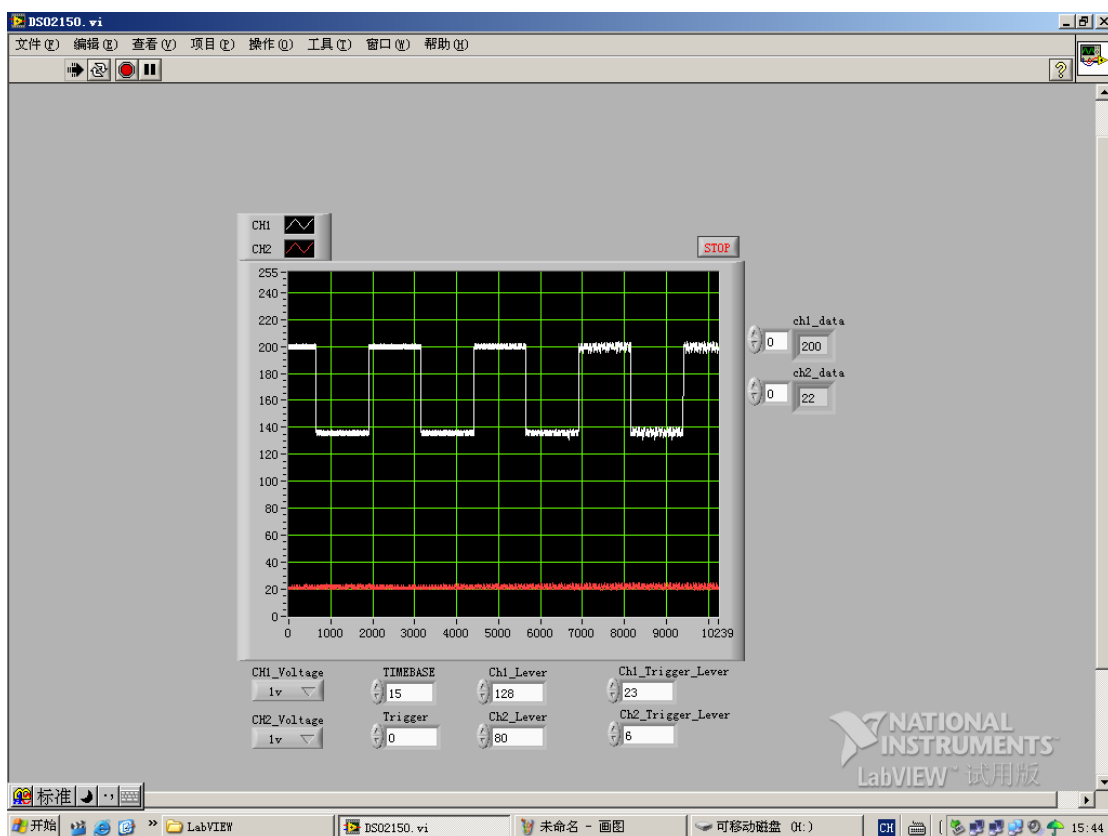


4 . Connect **DSO2150USB** to your PC.

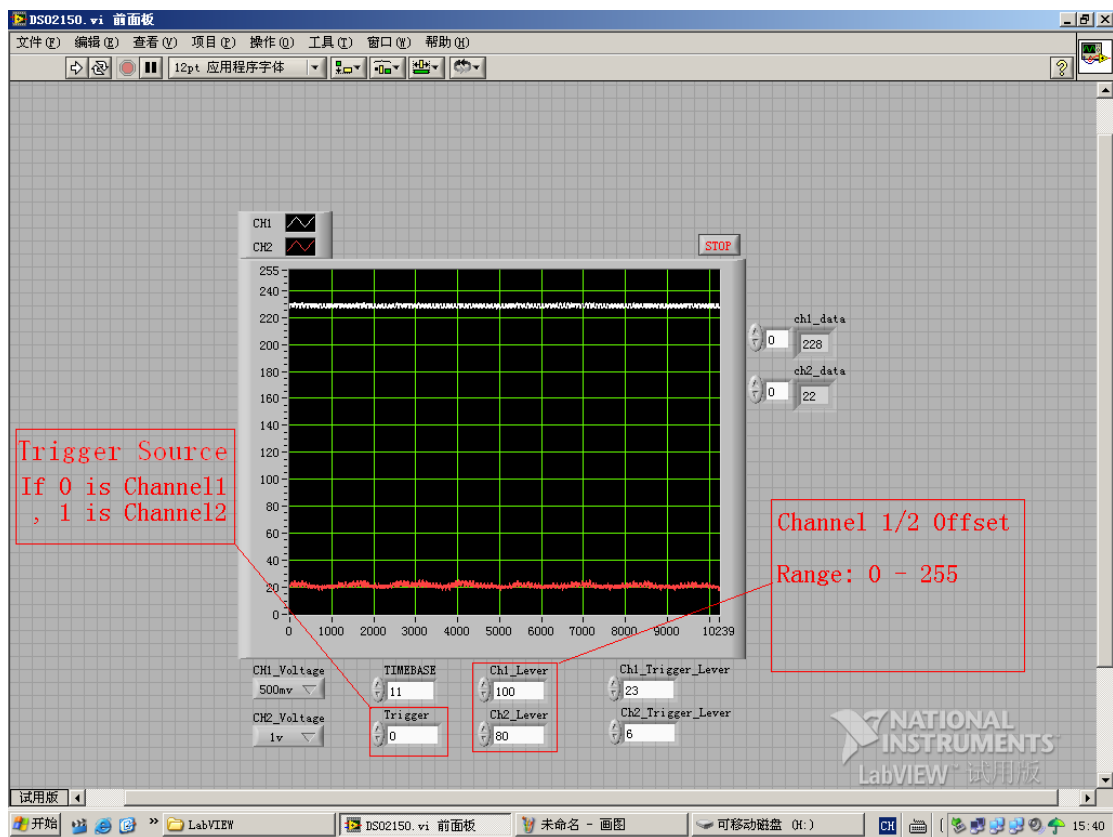
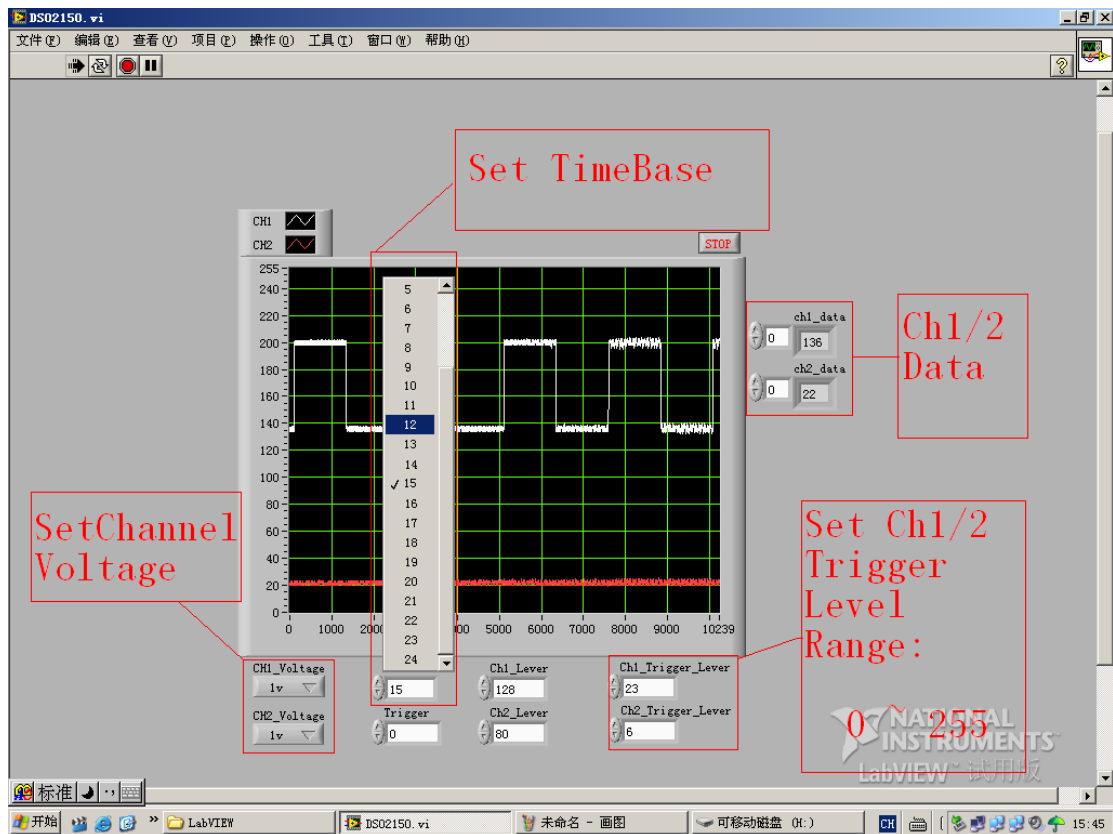
5 . Run "**DSO2150.vi**".



6 . Waveform display.



## Operation:





**Note:**

**CH1\_Voltage and CH2\_Volatge:**

Set CH1/CH2 Voltage.

**Trigger:**

If value is 0, it is Ch1 trigger.

If value is 1, it is Ch2 trigger.

**CH1\_Lever and CH2\_Lever:**

Set CH1/CH2 waveform position. (0 ~ 255).

**CH1\_Trigger\_Lever and CH2\_Trigger\_Lever:**

Set CH1/CH2 trigger position. (0 ~ 255).