

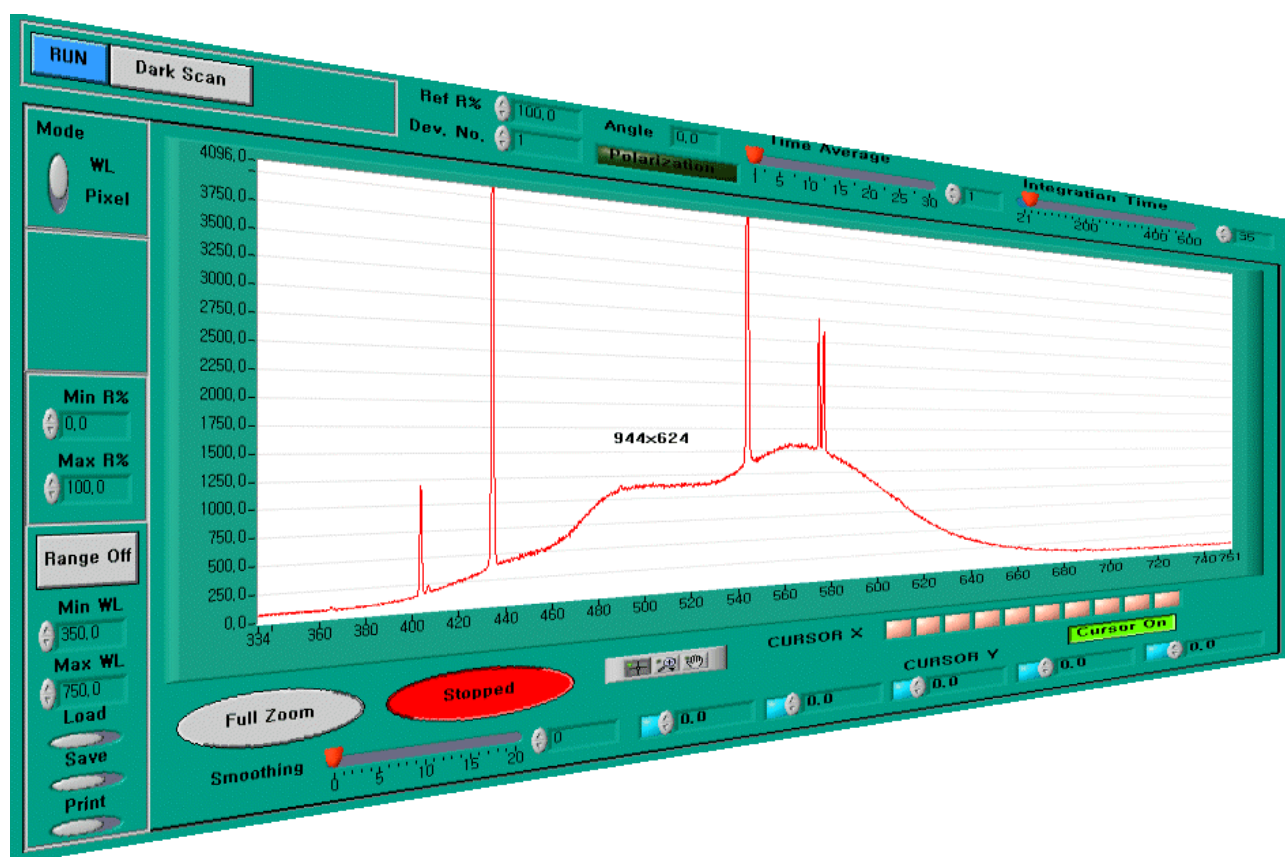
# SM Series Spectrometer LabVIEW VI Library Manual

[ Ver1.0 ]

March 21, 2002  
CVI Spectral Products

## Contents

1. Overview
2. LLB Includings
3. VI Function Description
- 4.. Training Course : Getting Simple Spectral Data



# 1. OVERVIEW

A lot of customers who purchased CVI SM Series Spectrometers are interested in developing their own application system by use of our spectrometers.

To reduce the time for developing an analysis software, many customer uses the LabVIEW.

So we started to distribute the LabVIEW VI libraries to customer free of charge Now!

In our SM32Pro Installation CD, we included the followings for LabVIEW users

1-1. SM Series Spectrometer LabVIEW LLB Libraries (for PCI1200/PCI6023 interfaces)

1-2. Example source code for training courses

1-3. Example SM32Lab.ini for LabVIEW users

All VIs are developed for LabVIEW 6.0 or later version and these libraries are all for CVI Spectrometers SM200 / SM240 / SM241 only.

## 2. LLB Includings

### 2-1. For PCI 1200 Interface Users

Calibration. vi  
Counter\_Setup(1200). vi  
Generate pulse train (8253). Vi  
get timebase (8253) .vi  
timebase generator (8253) .vi  
Load. vi  
Save. vi  
Smoothing\_Array. vi  
Spectrum\_Acquisition(1200). vi  
Spectrum\_Graph. vi

### 2-2. For PCI 6023E Interface Users

AI Start\_for\_PCI6023E .vi  
Calibration .vi  
Counter\_Setup(6023) .vi  
load .vi  
save .vi  
Smoothing\_Array. vi  
Spectrum\_Acquisition(6023). vi  
Spectrum\_Graph. vi

## 3. VI Function Description

### 3-1. Calibration. vi



Opens the dialog box and select \*.ini file and calibration data is transferred. This ini file is different from general ini file.

Customer can make their own ini file as follows (Example)

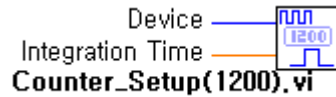
```
12
3650  164
4047  368
4078  384
4358  525
4916  802
5461  1068
5770  1217
5791  1227
6965  1788
7067  1836
7273  1934
7384  1987
```

12 means calibration wavelength number.

3650, 4047, 4078 ... are wavelength (Angstrom) and 164, 368, 384... are calibrated pixel number. You can make this ini file by reference of basic SM32Pro.ini file which is included in your diskettes.

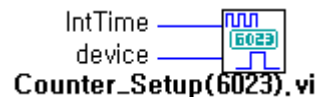
Pixel-WL Array is the 2-dimensional array includes pixel-wavelength matching value. When you use Spectrum\_Graph.vi for showing the data in X-Y Graph, you can connect this array to P-W terminal.

### 3-2. Counter\_Setup(1200). vi



Assign Device number and initial integration time. Then counter is automatically setup (PCI1200). Before using the Spectrum\_Acquisition.vi you should excute this vi first.

### 3-3. Counter\_Setup(6023) .vi



Assign Device number and initial integration time. Then counter is automatically setup (PCI6023). Before using the Spectrum\_Acquisition.vi you should execute this vi first.

### 3-4. Spectrum\_Acquisition(1200). vi



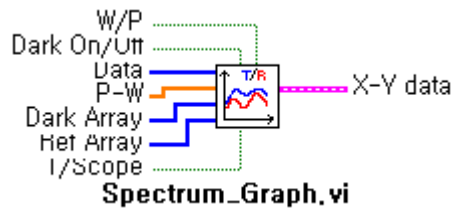
Assign device number and time average. If you input 1 to Time Average terminal, you can acquire data fastest. If you increased this number, you can get time-averaged Intensity data. Intensity data is an integer array for each 2048 pixels (PCI1200)

### 3-5. Spectrum\_Acquisition(6023). vi



Assign device number and time average. If you input 1 to Time Average terminal, you can acquire data fastest. If you increased this number, you can get time-averaged Intensity data. Intensity data is an integer array for each 2048 pixels (PCI6023)

### 3-6. Spectrum\_Graph. vi

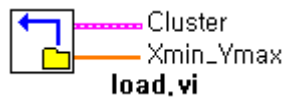


First, connect spectrum data to Data. This data may be intensity being currently measured. Then, connect Pixel-WL Array (output of Calibration.vi) to P-W terminal. After measuring the Dark and Reference (by Spectrum\_Acquisition.vi in Dark or Reference scan time), connect these values to Dark Array and Ref Array terminal. Then the Output X-Y data can be intensity or ratio(Transmittance, Reflectance, or Absorbance) If you want to change the pixel-wavelength mode, you have only to connect the Boolean to W/P terminal. Dark On/Off and T/Scope are similar with W/P terminal. T/Scope mode is the change between intensity and ratio. Once Dark array and Ref array are wired, ratio is automatically calculated.

Then, in Front Panel of LabVIEW, just make the X-Y Graph and in Diagram, wire the output X-Y data terminal to the X-Y Graph. Then you can see the data.

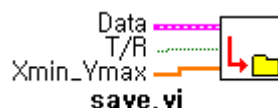
If you use while loop and Boolean buttons, you can easily see the real-time data by this X-Y graph within only 5 minutes programming.

### 3-7. load .vi



Once this vi is executed, you can select the saved file and the cluster is the saved data cluster. You can see this saved intensity or ratio by wiring to X-Y graph. If extended file name is int, all save values are intensity values, and if extended file name is trs, all save values are ratio values. You can distinguish data type by extended file name.

### 3-8. save .vi

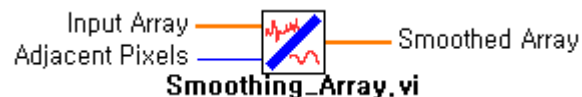


Once this vi is executed, you can save the intensity data or ratio data. If T/R value is false, the data is saved as \*.int (intensity data) and if T/R value is true, the data is saved as \*.trs(ratio data).

Xmin-Ymax is an 1-dimensional array of the lowest wavelength of pixel(index 0), the highest wavelength or pixel(index 1), the low limitation of intensity or ratio(index 2), the high limitation of intensity or ratio(index 3)

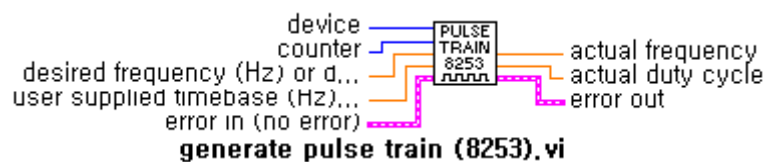
For example, if you save the data in wavelength mode and transmittance mode, This array includes lowest wavelength, highest wavelength, 0, 100(or above) because generally, you see the transmittance between 0% ~ 100% (or above)

### 3-9. Smoothing\_Array. vi



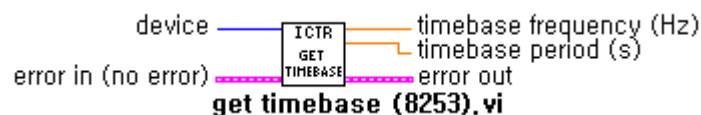
This vi is a kind of binning average vi. If you connect intensity data to Input Array and any value for pixel numbers for binning average, the binning-averaged data are calculated as Smoothed Array.

### 3-10. generate pulse train (8253). vi



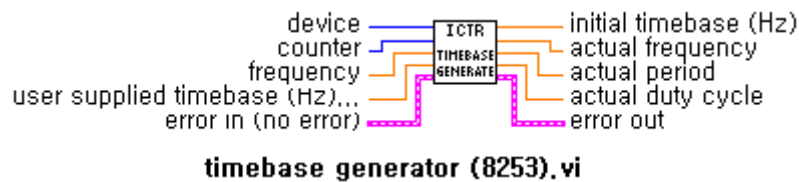
This vi is used for Counter\_Setup(1200).vi only. When you first used the Counter\_Setup(1200).vi and if the LabVIEW tries to find this library, you can select this library in out LLB. Later you don't have to worry about this library. (LabVIEW automatically refer this vi)

### 3-11. get timebase (8253) .vi



Similar with above vi.

### 3-12. timebase generator (8253) .vi



Similar with above vi.

### 3-12. AI Start\_for\_PCI6023E .vi



This vi is used for Spectrum\_Acquisition(6023).vi only. When you first used the Spectrum\_Acquisition(6023).vi and if the LabVIEW tries to find this library, you can select this library in out LLB. Later you don't have to worry about this library. (LabVIEW automatically refer this vi)

## 4. Training Course

### Acquiring Simple Spectral Data

#### 4-1. Overview

This example is made for acquiring simple spectral data continuously using basic Vis. By this example you can understand how to wire the basic Vis to operate SM Series spectrometers.

#### 4-2. Training Procedure

(1) In the front panel, first make the STOP Boolean button. Then make XY Graph as Fig 4-1. (You can change background color and Line color, shape of XY graph as you wish)



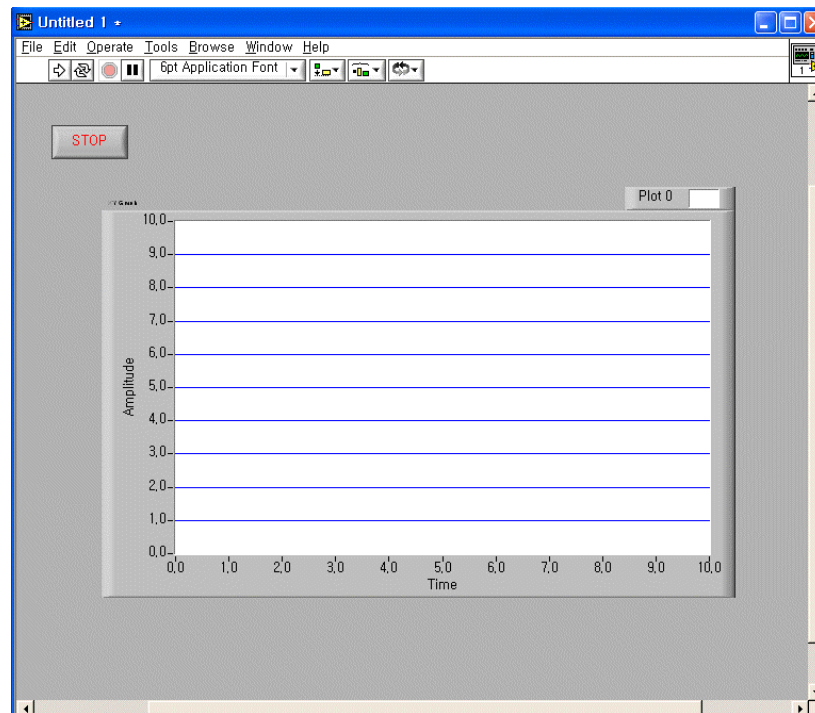


Fig 4 – 1

(2) In Diagram, make the Sequence and and insert Counter\_Setup(6023).vi inside the Sequence. (If you are using PCI1200 Card, you should insert Counter\_Setup(1200).vi. And then create constant of int time and Device terminal. Default values are 35 and 1 each.

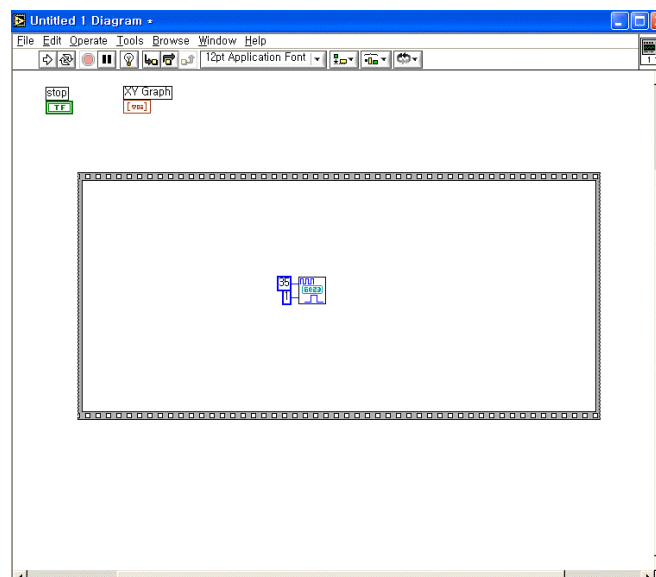


Fig 4 – 2

(3) Add Frame and insert Calibration.vi inside 2<sup>nd</sup> frame. Then, create indicator of Pixel-WL array. Then make the local variable of Pixel-WL array and then change this local variable to Read property. (to use this local variable as input later)

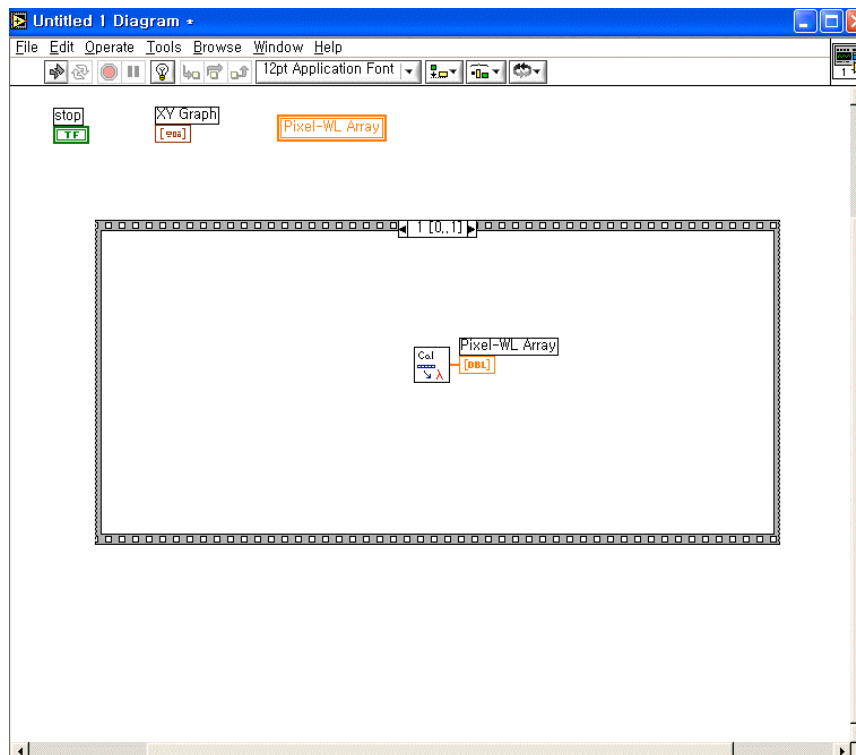


Fig 4 – 3

(4) Add 3<sup>rd</sup> frame and make while loop inside 3<sup>rd</sup> frame. And then make the while loop stop when stop button is clicked as Fig 4-4 and Fig 4.5.

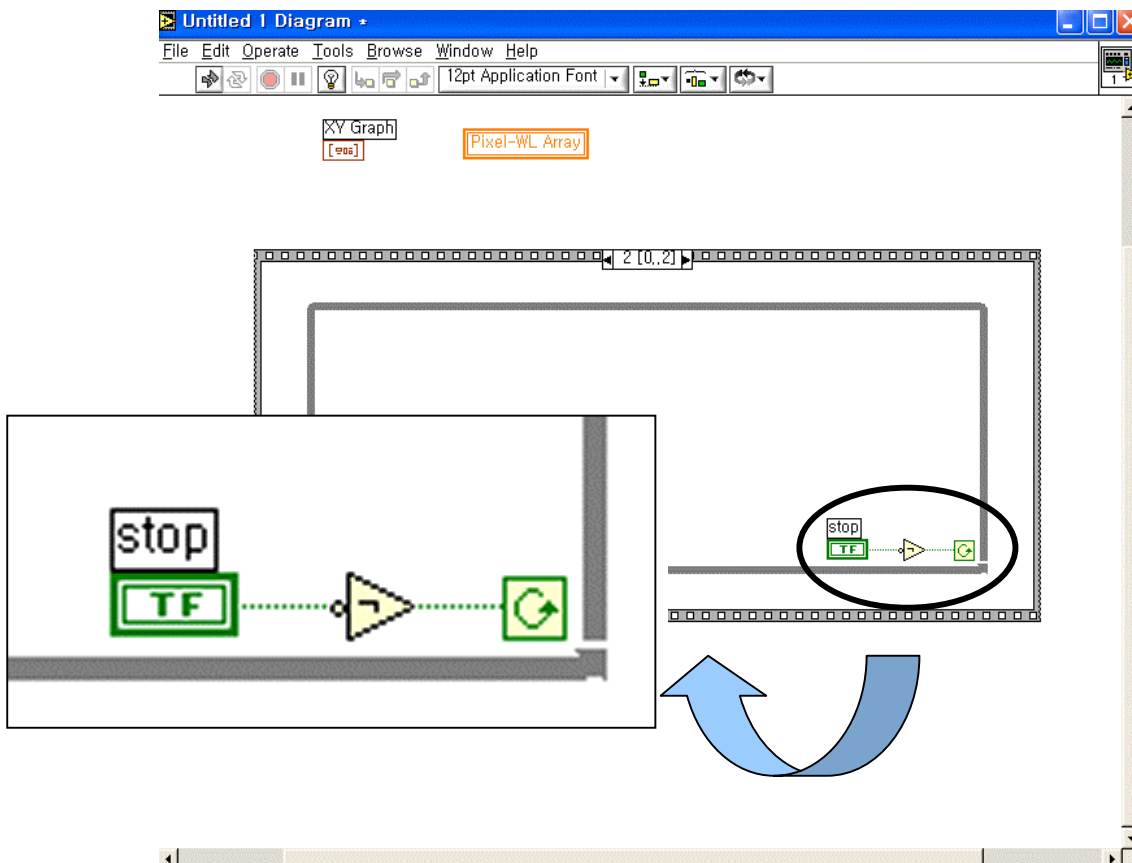


Fig 4 – 4

[5] Insert Spectrum\_Acquisition(6023).vi inside 3<sup>rd</sup> frame. And create Device and TimeAverage constant. (Default values are 1 and 1) And insert Spectrum\_Graph.vi inside the 3<sup>rd</sup> frame. And then, connect Intensity Data of output of Spectrum\_Acquisition(6023).vi to Data of input of Spectrum\_Graph.vi.

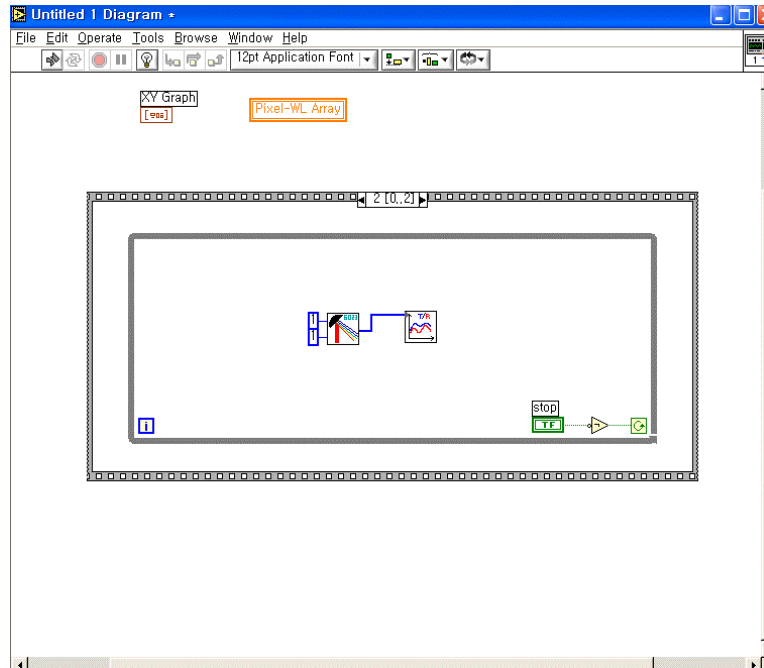


Fig 4 – 5

[6] Connect the Pixel-WL Array (Local Variable) to the P-W input terminal of Spectrum\_Graph.vi. Then connect X-Y data terminal of Spectrum\_Graph.vi to XY Graph.

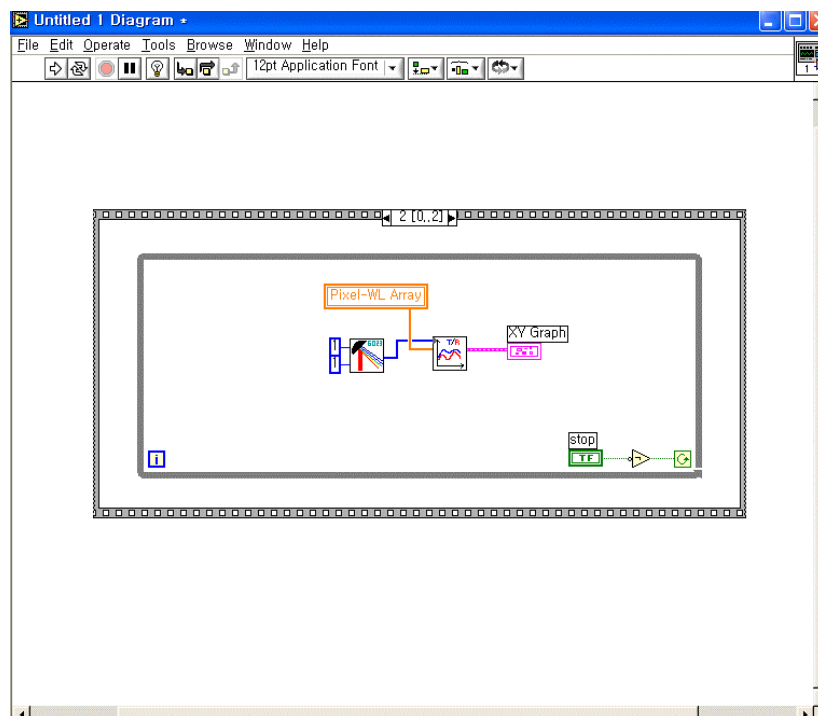


Fig 4 – 6

(7) In front panel, change color of plot to any color(Here, we change to red plot) And change the maximum value of Y-axis to 4096, unlock Autoscale Y of XY Graph. (For the time being, we selected Autoscale X for convenience. Later you can change the minimum and maximum value of X and Y scale) And run this program. Then you'll see the open dialog box as follows.

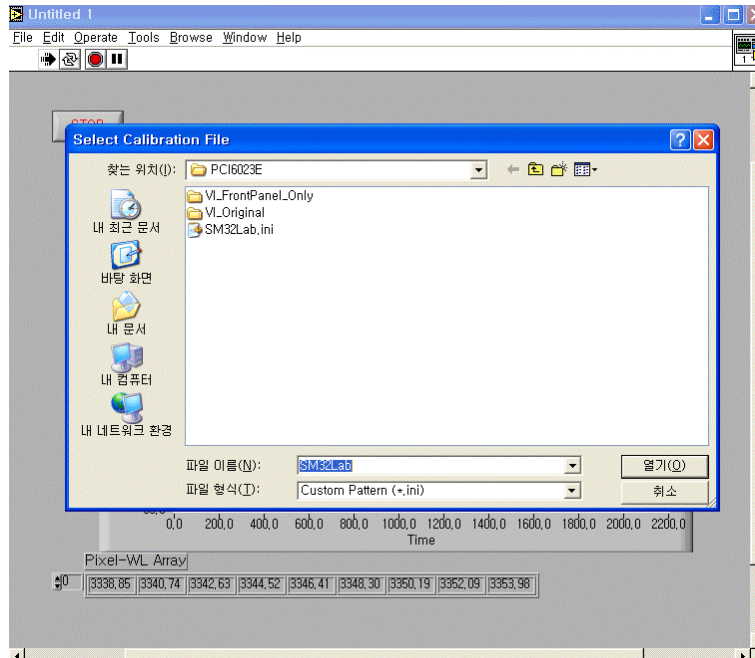


Fig 4 - 7

(8) Then select the ini file. (You can make any kind of name for initial file) Then you can now see the real-time spectrum data through the XY Graph ! as Fig 4-8.

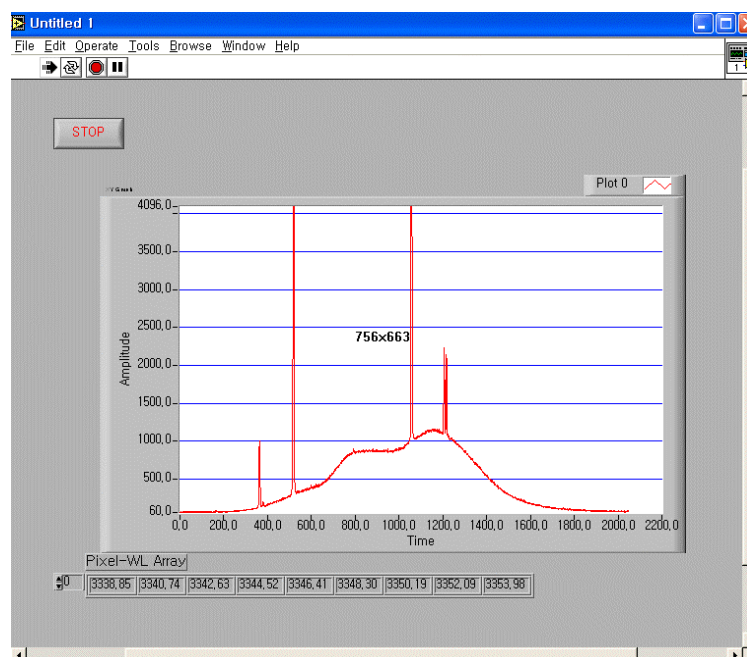


Fig 4 - 8

(9) If you want to see the spectrum data as wavelength unit, in diagram create Boolean constant of W/P terminal of Spectrum\_Graph.vi. And change this Boolean to True value. And then run this program again. Then you can see the spectrum data as wavelength unit as follows.

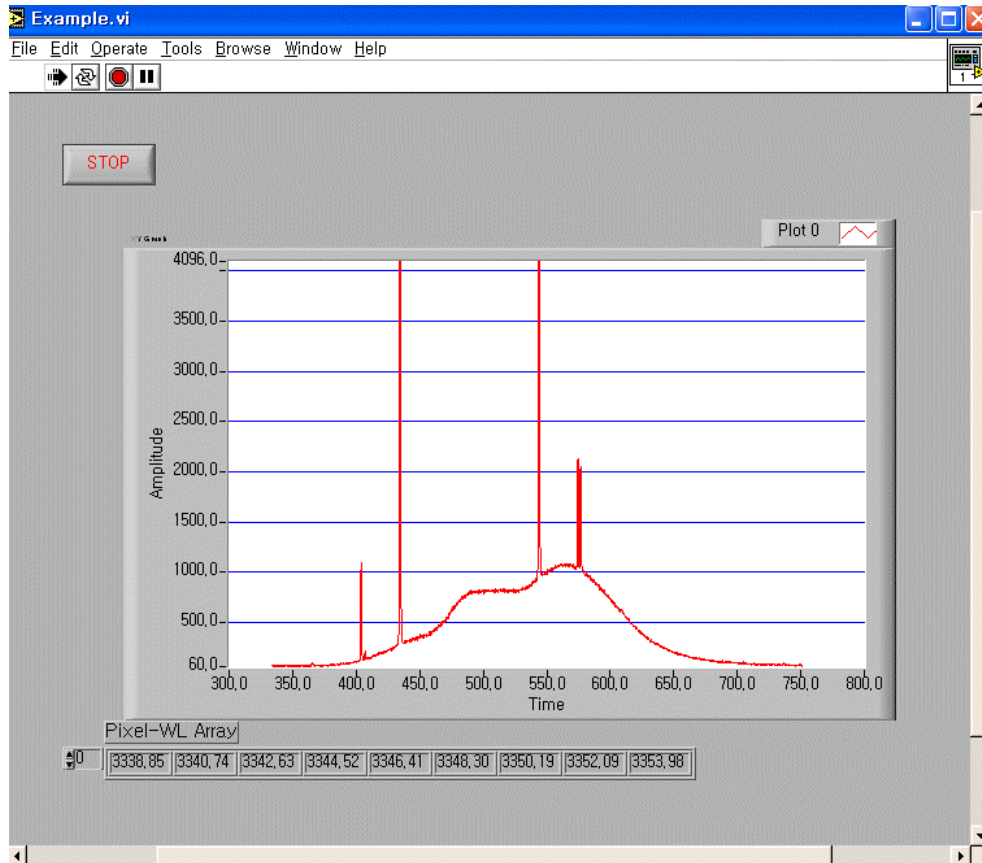


Fig 4 – 9