67-15233 34th Ave.
Surrey, BC V3S 2T7
604-637-2167
info@triarchytech.com

# TSA6G1,TSA4G1 and TSA8G1

# USB mini Spectrum Analyzer

# Application Programming Interface (API)

Triarchy Technologies CORP.

67-15233 34th Avenue, Surrey, V3S 2T7
British Columbia, CANADA

Tel: +1 604 637 2167

Email: info@triarchytech.com

Aug 11, 2014

Version 1.6

67-15233 34th Ave.
Surrey, BC V3S 2T7
604-637-2167
info@triarchytech.com

Company Proprietary Information

67-15233 34th Ave.
Surrey, BC V3S 2T7
604-637-2167
info@triarchytech.com

# Table of Contents

67-15233 34th Ave.
Surrey, BC V3S 2T7
604-637-2167
info@triarchytech.com

# 1. Introduction

The purpose of this document is to specify the application program interface for TSA6G1,TSA4G1 and TSA8G1.

The application programming Interface is a tool for software engineers to design custom program to control TSA6G/TSA4G1/TSA8G1. The API is used to send commands to, and receive data from, TSA USB mini spectrum analyzer. The API will provide to you low level data interface. You can control the Dongle device by sending commands, and receiving the measured data from device, and then store and process the data in any format in your application.

API is unlike TSA PC program which is Graphical User interface (GUI) for TSA USB mini spectrum analyzer. TSA will process the receive the data and show it on the display windows with current view, Average view and Max view. But TSA (GUI) function is fixed, and difficult to integrate TSA (GUI) into customer application SW. API will be easy to integrated into custom program.

DISCLAIMER---Customer can use API into their own application with free of charge, but no warranty or support will be provided.

# 2. Commands

1: **Get_Hid_Handle**

This is **Setup command.** It will setup hid link for USB product, then get the Handle data. This Handle data will be used for the next command. You must sent this command first, it will initiate TSA USB spectrum analyzer system.

If the returned value is not zero, it means the hanlde is got.

Parameter: N/A

VC declare:

```
HANDLE __stdcall Get_Hid_Handle();
```

VB declare:

```
Declare Function Get_Hid_Handle Lib "TSA.dll" () As Int32
```

C# declare:

```
[DllImport("TSA.dll", CharSet = CharSet.Auto, CallingConvention = CallingConvention.StdCall)]
public extern static IntPtr Get_Hid_Handle();
```

67-15233 34th Ave.
Surrey, BC V3S 2T7
604-637-2167
info@triarchytech.com

2: **Output_Serial_Number**

This is **SN command**. It will read the product series number (SN) from calibration file.

If everything is ok, the return value is True(1) or False (0).

Parameter: N/A

VC declare:

```
BOOLEAN __stdcall Output_Serial_Number(CHAR* DIR_PATH,CHAR* SN);
```

VB declare:

```
Declare Function Output_Serial_Number Lib "TSA.dll" (ByRef bytDIR_PATH As Byte, ByRef bytSN As Byte) As
    Byte
```

C# declare:

```
[DllImport("TSA.dll", CharSet = CharSet.Auto, CallingConvention = CallingConvention.StdCall)]

public extern static Byte Output_Serial_Number(ref Byte dir_path, ref Byte sn);
```

3: **Start_Dongle**

This is **Start command,** it will launch the TSA USB mini spectrum analyzer to execute measurement task.

Parameter:

**Handle data**--- get from Setup command.

**Center frequency**—Range from 1 to 6150 for TSA6G1, 1 to 4150 for TSA4G1, 1 to 8150 for TSA8G1, unit is MHz.

**Frequency step**-- Range from will be:

When RBW = 50KHz
fast scan step = 24000 ~ 320000 HZ
normal scan step = 400 ~ 200000 HZ

When RBW = 100k
fast scan step = 24000 ~ 640000 HZ
normal scan step = 400 ~ 400000 HZ

67-15233 34th Ave.
Surrey, BC V3S 2T7
604-637-2167
info@triarchytech.com

When RBW = 200k

fast scan step = 24000 ~ 1280000 HZ

normal scan step = 400 ~ 800000 HZ

When RBW = 500k

fast scan step = 24000 ~ 3200000 HZ

normal scan step = 400 ~ 2000000 HZ

Span =( Frequency step)* (Frequency points-1)

if Frequency step=20000 (20KHz), and Frequecny point is 501, Span will be 10MHz.

if Frequency step=20000 (20KHz), and Frequecny point is 101, Span will be 2MHz.

if Frequency step=200000 (200KHz), and Frequecny point is 501, Span will be 100MHz.

if Frequency step=2000000 (2MHz), and Frequecny point is 501, Span will be 1000MHz.

**Resolution bandwidth**-- 50MHz (1),

100MHz (2),

200KHz (3),

500KHz (4)

**Frequency points**-- Range from 101 to 501 for **normal scan** （number shall be odd value)

Range from 100 to 250 for **fast scan** （number shall be even value)

Please note: the fast scan and normal scan be will selected by Odd or Even value of frequency points.

**Amplitude level**-- TSA4G1

-60dBm (0)

-50dBm (1)

-40dBm (2)

-30dBm (3)

67-15233 34th Ave.
Surrey, BC V3S 2T7
604-637-2167
info@triarchytech.com

-20dBm (4)

-10dBm (5)

0dBm (6)

TSA6G1

TSA6G1 band 1 (1MHz ~5000MHz)

-60dBm (0)

-50dBm (1)

-40dBm (2)

-30dBm (3)

-20dBm (4)

-10dBm (5)

0dBm (6)

TSA6G1 band 2 (5000MHz ~6150MHz)

-40dBm (0)

-30dBm (1)

-20dBm (2)

-10dBm (3)

0dBm (4)

TSA8G1

TSA8G1 band 1 (1MHz ~5000MHz)

-60dBm (0)

-50dBm (1)

-40dBm (2)

-30dBm (3)

67-15233 34th Ave.
Surrey, BC V3S 2T7
604-637-2167
info@triarchytech.com

-20dBm (4)

-10dBm (5)

0dBm (6)

TSA8G1 band 2 (5000MHz ~8150MHz)

-40dBm (0)

-30dBm (1)

-20dBm (2)

-10dBm (3)

0dBm (4)

**Sweep time**-- SWP_TIME_1_CW (0) // CW mode 1

SWP_TIME_1_5_BM (1) // Burst Mode 1.5

SWP_TIME_2_BM (2) // Burst Mode 2

SWP_TIME_4_BM (3) // Burst Mode 4

SWP_TIME_8_BM (4) // Burst Mode 8

SWP_TIME_16_BM (5) // Burst Mode 16

SWP_TIME_32_BM (6) // Burst Mode 32

**External attenuator**-- ture(1): selected, false(0): not selected , add extra 30dB attenuation if it selected, and Amplitude level will move 30dB.

The return value:

#define EVERYTHING_OK (0)

#define SEND_COMMAND_FAIL (1)

#define FILE_UNFOUND (2)

#define FILE_FORMAT_ERROR (3)

67-15233 34th Ave.
Surrey, BC V3S 2T7
604-637-2167
info@triarchytech.com

#define CONFIGURED_FREQ_ERROR (4)

#define CONFIGURED_STEP_ERROR (5)

#define CONFIGURED_RBW_ERROR (6)

#define CONFIGURED_SCANING_POINTS_ERROR (7)

#define CONFIGURED_AMP_ERROR (8)

#define CONFIGURED_SWEEP_TIME_ERROR (9)

#define CONFIGURED_EXT_ATT_ERROR (10)

#define CONFIGURED_FREQ_BEYOND_BAR (11)

#define CONFIGURED_FREQ_850_RULES_ERROR (12)

VC declare:

```
Declare Function Start_Dongle Lib "TSA.dll" (IhDongle As Int32, C_FREQ As Double, FSTEP As UInt32, iRBW As Byte, POINTS As UInt16, AMP As Byte, SWEEP_TIME As Byte, EXT_ATT As Byte, ByRef bytDIR_PATH As Byte) As Byte
```

VB declare:

```
Declare Function Start_Dongle Lib "TSA.dll" (IhDongle As Int32, C_FREQ As Double, FSTEP As UInt32, iRBW As Byte, POINTS As UInt16, AMP As Byte, SWEEP_TIME As Byte, EXT_ATT As Byte, ByRef bytDIR_PATH As Byte) As Byte
```

C# declare:

```
[DllImport("TSA.dll", CharSet = CharSet.Auto, CallingConvention = CallingConvention.StdCall)]

public extern static Byte Start_Dongle(IntPtr hDongle, Double C_FREQ, UInt32 STEP, UInt16 iRBW, Byte POINTS, Byte AMP, Byte SWEEP_TIME, Byte EXT_ATT, ref Byte dir_path);
```

4: **Stop_Dongle**

This is **Stop commad**, you can send it to stop measurement.

The return value:

#define EVERYTHING_OK (0)

#define SEND_COMMAND_FAIL (1)

Parameter:

67-15233 34th Ave.
Surrey, BC V3S 2T7
604-637-2167
info@triarchytech.com

**Handle data**--- get from Setup command.

VC declare:

```
BYTE __stdcall Stop_Dongle(HANDLE hDongle);
```

VB declare:

```
Declare Function Stop_Dongle Lib "TSA.dll" (hDongle As Int32) As Byte
```

C# declare:

```
[DllImport("TSA.dll", CharSet = CharSet.Auto, CallingConvention = CallingConvention.StdCall)]
public extern static Byte Stop_Dongle(IntPtr hDongle);
```

5: **Receive_Data_From_Dongle**

This is **Data command;** API gets measurement data from dongle, inputs the Handle, and get return ID to indicated status of Dongle.

Parameter:

**Handle data**--- get from Setup command.

Return ID—0 61 122 183 ….The maximum is 488. If the points number is less than 501.

The maximum will be less than 488.

Data_Length: The value is less than 61. Ex: the point number is 501. The last data length is 13.

**Received Data**-- the receive dated will be in groups of 61. If frequency point is 501 points, the received data will have a total of 9 groups. First 8 group will have 61 data, and last group will have 13 data.

If everything is ok, the return value is True(1) or False (0).

VC declare:

```
BOOLEAN __stdcall Receive_Data_From_Dongle(HANDLE hDongle,INT &ID,DOUBLE* rev_data,INT &Data_Length);
```

VB declare:

```
Declare Function Receive_Data_From_Dongle Lib "TSA.dll" (hDongle As Int32, ByRef ID As Int32, ByRef
  rev_data As Double, ByRef Data_Length As Int32) As Byte
```

C# declare:

```
[DllImport("TSA.dll", CharSet = CharSet.Auto, CallingConvention = CallingConvention.StdCall)]
```

67-15233 34th Ave.
Surrey, BC V3S 2T7
604-637-2167
info@triarchytech.com

```
public extern static Byte Receive_Data_From_Dongle(IntPtr hDongle, ref Int32 ID, ref Double rev_data, ref
    Int32 Data_Length);
```

## 3. Demo program

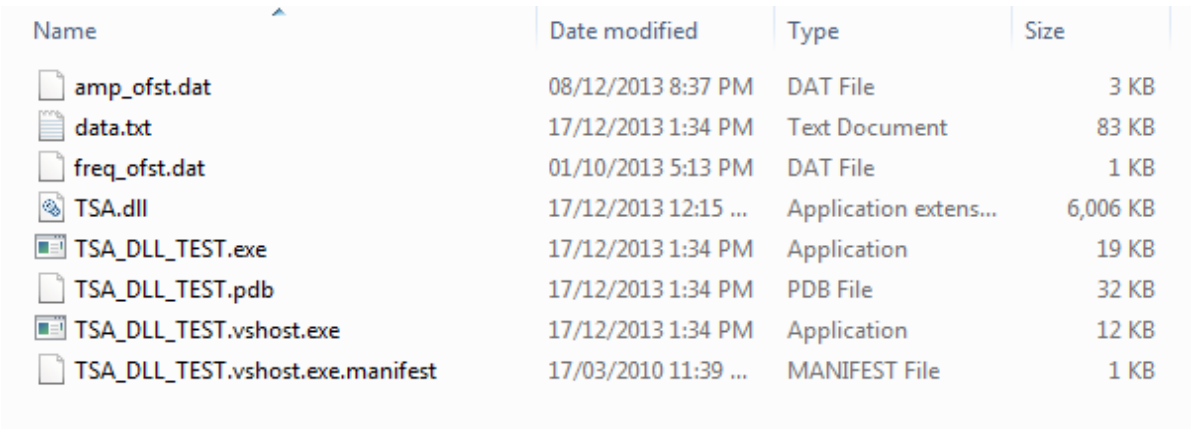
In the API package, We provide TSA.DLL file for API. We also provide three demo program which will work on the VC, VB and C #. When you order the TSA series USB mini spectrum analyzer, we will have hardware dongle device and calibration file; This API is still need calibration file to work with.

You can see the follow file in the folder: TSA.dll is API file, amp_ofst,dat and freq_ofst.dat. They are the calibration of your device. TSA_DLL_TEST will be your application sofware, you can build it with VC, VB or C#.

Please note: TSA6G1 license file was lic.dat and freq_ofst.dat, please change to name of lic.dat to amp_ofst.dat. So that TSA6G1 calibration fill will be amp_ofst.dat and freq_ofst.dat.

TSA.dll don't recognize the lic.dat, please change the lic.dat name to amp_ofst.dat, when product is TSA6G1.

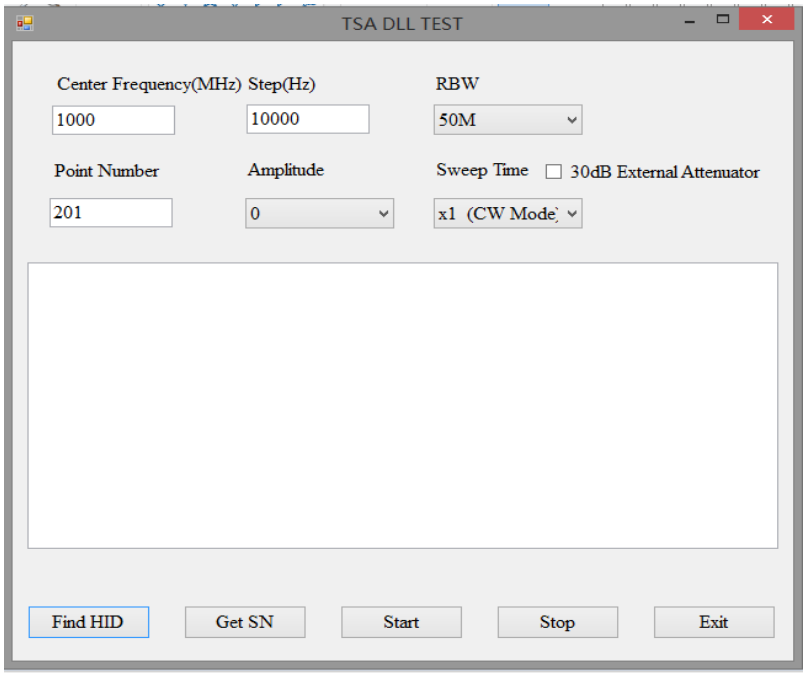
（*data.txt file is only in the C# programe）



| Name | Date modified | Type | Size |
|---|---|---|---|
| amp_ofst.dat | 08/12/2013 8:37 PM | DAT File | 3 KB |
| data.txt | 17/12/2013 1:34 PM | Text Document | 83 KB |
| freq_ofst.dat | 01/10/2013 5:13 PM | DAT File | 1 KB |
| TSA.dll | 17/12/2013 12:15 ... | Application extens... | 6,006 KB |
| TSA_DLL_TEST.exe | 17/12/2013 1:34 PM | Application | 19 KB |
| TSA_DLL_TEST.pdb | 17/12/2013 1:34 PM | PDB File | 32 KB |
| TSA_DLL_TEST.vshost.exe | 17/12/2013 1:34 PM | Application | 12 KB |
| TSA_DLL_TEST.vshost.exe.manifest | 17/03/2010 11:39 ... | MANIFEST File | 1 KB |

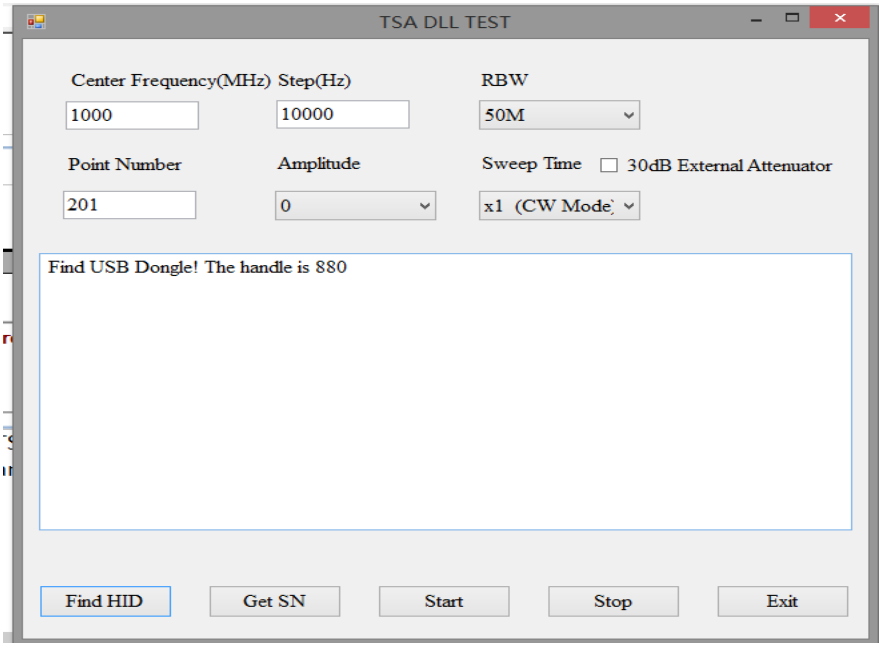
Click the TSA_DLL_TEST.exe to run the Demo program. It will show the follow window:

67-15233 34th Ave.
Surrey, BC V3S 2T7
604-637-2167
info@triarchytech.com

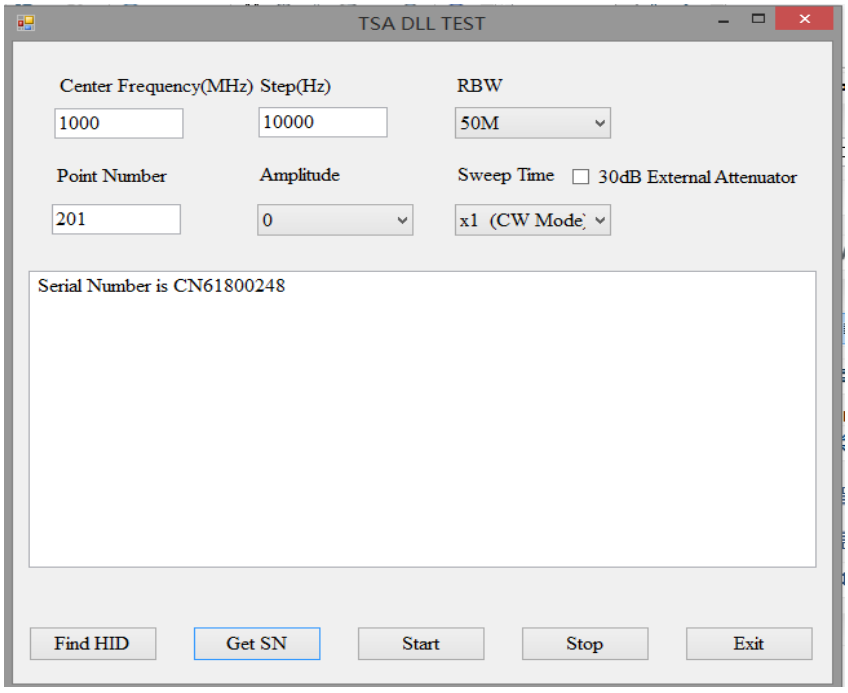



Please plug the TSA series Dongle into the PC, and click the Find HID to send setup command. The USB HID link will setup and handle data will show up:

67-15233 34th Ave.
Surrey, BC V3S 2T7
604-637-2167
info@triarchytech.com

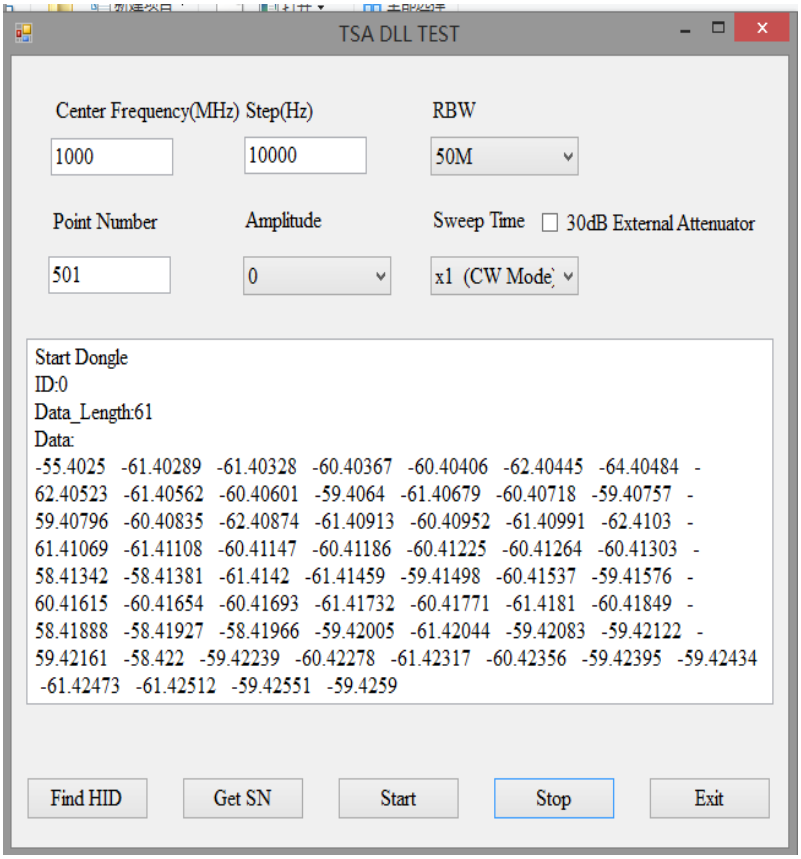



Click Get SN key, the SN command will send to Dongle and SN number will show on the window. please note, the calibation file must match with your dongle, otherwise, the program will not work.



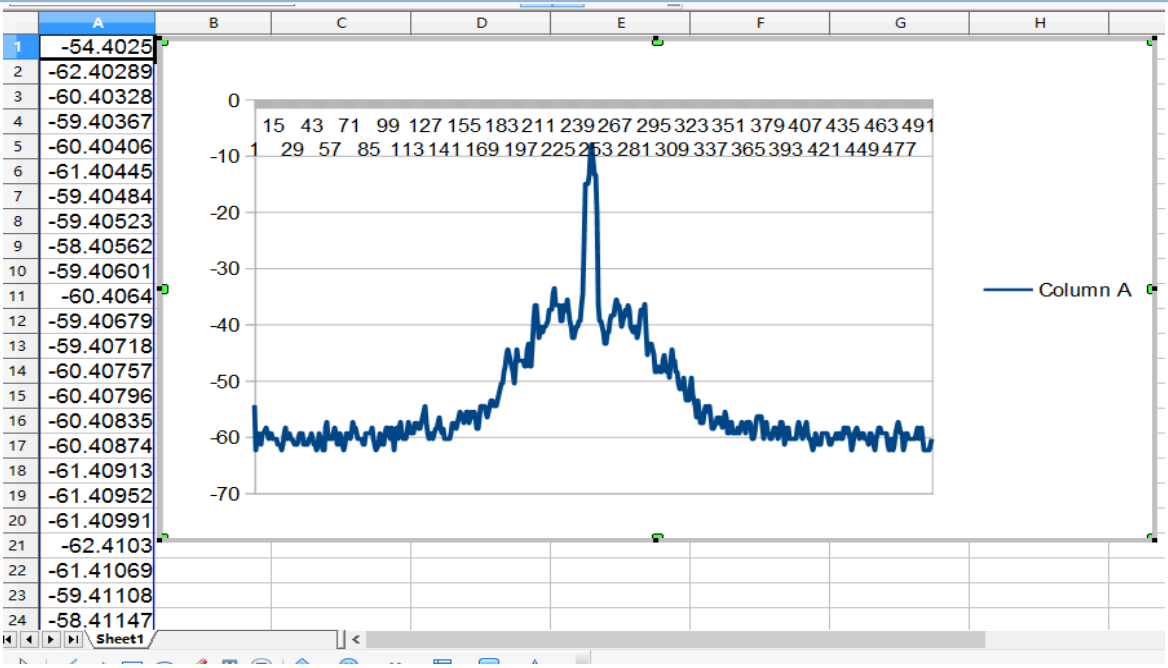
67-15233 34th Ave.
Surrey, BC V3S 2T7
604-637-2167
info@triarchytech.com

Connect TSA USB mini spectrum analyzer dongle with RF signal generator (1000MHz), then click the Start key to send Start command to Dongle. We will see a lot data shown on the received windows, wait a while, then click stop to send Stop command to the Dongle.




You can find data.txt file, change this log file name into data.csv, and open it by excel. Just keep one frame data, for this case the 501 point data, then show date with image. You can find signal waveform which will be exactly same as TSA program.

67-15233 34th Ave.
Surrey, BC V3S 2T7
604-637-2167
info@triarchytech.com




If you have Visual Studio VS10, or VS12, you can open the project of demo program. Just click TSA_DLL_TEST.sln file, you can opent the project, and click the F5 you can run the project.