



API_VSG6G1C_VSG2G5C

1: Preface

This document contains the API function calls to interface with the VSG6G1C/VSG2G5C Vector signal generator through Microsoft Windows.

Demo API software is written by C#. The source code is provide in this API package.

API can be still working with VC.



2: Demo API software

Open the API file package, there are three folder inside:

1: API

It is only one API drive file in this folder, the file name is TSG2019.dll, this DLL drive is include all functions for the spectrum analyzer application.

2: document

it is document file of API_VSG6G1C_VSG2G5C for vector signal generator section .

3: demo API software

open the file path:

API_VSG6G1C_VSG2G5C\Demo API software\TSG2019.Demo\bin\Debug

Click the TSG2019.Demo.exe, the demo API software is working:

TSG Api Demo

Working Mode: Single Frequency w/o Pulse Mod

Clock Mode: Internal w/o Output

IQ Mode: None

☒ Rf On ☐ LowBand On

Start Freq(Hz): 2000000000

Step Freq(Hz): 1000000

Points: 1

Main Interval(ns): 10000000

Pulse Width(ns): 1000000

Delay Width(ns): 1000000

Pulse Number: 1

Amplitude(dBm): 0

IQ Step Count: 200

Buttons: Set Timer, Set Freq, Hopping File, IQ File, Set Default

Status: dongle has connected! Serial Number:CN22800001 command successfully

Customer can modify the source code of demo API software to build special application.



3: API function

int GetDongleStatus()

para	none
return	0 : dongle is not find , 1: dongle attach but disconnect, 2: connect
description	Get the dongle's status

GetUSB DongleHandle()

para	none
return	True: success, false failure
description	Get the dongle's handle, you must use this command at first

string GetSerialNumber()

para	none
return	Serial number
description	Get the dongle's serial number

void SetDefault()

para	none
return	none
description	set the dongle's default status, then start dongle to work with default setting.

int SetRfOn(bool RfOn)

para	Bool : true on, false off
------	---------------------------



return	1: true other :false
description	Set RF On

<code>string[] GetWorkingModeDesc ()</code>	
para	none
return	0:Single Frequency w/o Pulse Mod 1:Frequency Sweeping w/o Pulse Mod 2:Frequency Hopping w/o Pulse Mod 3:Single Frequency with Pulse Mod 4:Frequency Sweeping with Pulse Mod 5:Frequency Hopping with Pulse Mod
description	Get Working Mode Description

<code>Int GetWorkingModeCurrentIndex ()</code>	
para	none
return	Working Mode List Current Index
description	Get Working Mode Current Index

<code>Int SetWorkingMode (byte WorkingMode)</code>	
para	Working Mode Index
return	1: true other :false
description	Set Working Mode

<code>string[] GetClockModeDesc ()</code>	
para	none
return	0: Internal with Output



	1: Internal w/o Output 2: External
description	Get Clock Mode Description

<code>int</code> GetClockModeCurrentIndex()	
para	none
return	Clock Mode Index
description	Get Clock Mode Current Index

<code>int</code> SetClockMode(<code>byte</code> ClockMode)	
para	Clock Mode Index
return	1: true other :false
description	Set Clock Mode

<code>string[]</code> GetIQModeDesc()	
Para	none
Return	0: None 1: Internal 2: External Fast
description	Get IQ Mode Description

<code>int</code> GetIQModeCurrentIndex()	
para	none
return	IQ Mode Index



description	Get IQ Mode Current Index
-------------	---------------------------

<code>int SetIQMode(byte IQMode)</code>	
para	IQ Mode Index
return	1: true other :false
description	Set IQ Mode

<code>int SetLowBand(bool LowBand)</code>	
para	True : Low Band On , False: Low Band Off
return	1: true other :false
description	Set Low Band

<code>int SetFrequency(ulong StartFrequency, uint StepFrequency, ushort Points)</code>	
para	Start Frequency (Hz), Step Frequency (Hz), Points (freq VSG6G1C MAX:6100000000Hz, Min: 1000Hz) (VSG2G5C frequency Max:2500000000Hz, Min: 100000Hz)
return	1: true other :false
description	Set Frequency

<code>int SetTimer(ulong MainInterval, ulong PulseInterval, ulong DelayInterval, byte PulseNumber)</code>	
para	Main interval(ns), Pulse Interval(ns), DelayInterval(ns), Pulse Number (VSG6G1C Main interval Max: 20000000000ns. Min: 40000ns) (VSG6G1C Pulse interval Max: 5000000000ns. Min: 250ns)



	(VSG6G1C Delay interval Max: 5000000000ns. Min: 710ns) (VSG2G5C Main interval Max: 20000000000ns. Min: 100000ns) (VSG2G5C Pulse interval Max: 5000000000ns. Min: 50000ns) (VSG2G5C Delay interval Max: 5000000000ns. Min: 60000ns)
return	1: true other :false
description	Set Timer

<code>int SetAmplitude(double ampl)</code>	
para	Amplitude – VSG6G1C 10dBm --100dBm VSG2G5 15dBm --70dBm (VSG6G1C 10~-100 , when freq less than 4GHz, 0~-100 when freq larger than 4GHz) (VSG2G5C 15~-70) (Low band both for VSG6G1C and VSG2G5C 0~-50) setup resolution : 0.25, it means 0.25dB
return	1: true other :false
description	Set Amplitude

<code>int SetIqStepCount(uint IqStepCount)</code>	
para	IQ Step Count (Value from 30~36000)
return	1: true other :false
description	Set IQ Step Count

<code>int SendHoppingFile(string FilePath)</code>	
para	File Path



return	1: true other :false
description	Set Hopping File

<code>int</code> SendIQRawDataFile(<code>string</code> FilePath)	
para	File Path
return	1: true other :false
description	Set IQ Raw Data File

<code>int</code> ExcuteRxPacket(<code>ref double</code> Progress)	
para	Progress: When Setting Hopping File or IQ Raw Data File, return percent progress
return	0: successful, others: error code
description	Receive Packet

<code>bool</code> LowBandIsOn()	
para	none
return	Low Band Status: on true, off false
description	Low Band Is On

<code>bool</code> RfIsOn()	
para	none
return	Rf status: on true, off false
description	Rf Is On



`ulong` GetStartFrequency()

para	none
return	Start Frequency
description	Get Start Frequency

`uint` GetStepFrequency()

para	none
return	Step Frequency
description	Get Step Frequency

`ushort` GetFrequencyPoints()

para	none
return	Point Number
description	Get Frequency Points

`void` GetTimerSetting(`ref ulong` MainInterval, `ref ulong` PulseInterval, `ref ulong` DelayInterval, `ref byte` PulseNumber)

para	Return Main interval(ns), Pulse Interval(ns), DelayInterval(ns), Pulse Number
return	none
description	Get Timer Setting

`double` GetAmplitude()

para	none
return	Amplitude (dBm)



description	Get Amplitude

uint GetIQStepCount ()	
para	none
return	IQ Step Count
description	Get IQ Step Count