

## GSP-810 Remote Interface Document

The following describes the interface for the dll (dynamic link library) file associated with the GSP-810 for Remote Operation. This is a preliminary description of a dll file that is expected to change and grow in the near future to accommodate additional features available in the GSP-810.

The dll file name is "gsp810.dll". It is designed to run on Windows NT 4.0, Windows 95, Windows 98, Windows 2000, or Windows XP system. It currently supports 10 commands designed to work with Visual Basic as follows:

```
extern "C" long __stdcall GetSerialNum(void);
extern "C" long __stdcall SetFreq(long freq);
extern "C" long __stdcall SetRBW(long rbw);
extern "C" long __stdcall SetReflvl(long reflvl);
extern "C" long __stdcall SetSpan(long span);
extern "C" long __stdcall ChangeComPort(long ComPort);
extern "C" long __stdcall GetAnalyzerTraceData(LPSAFEARRAY FAR *);
extern "C" long __stdcall MarkerToCenter(void);
extern "C" long __stdcall MarkerToPeak(void);
extern "C" long __stdcall GetMarkerLevel(long);
```

The following is a description of each command:

### GetSerialNum

A call to this routine will return the serial number of the GSP-810 as a long integer. This is also a good routine to call to verify proper RS-232 communications with the GSP-810. If there was no response from the GSP-810, then 999999 will be returned.

### SetFreq

A call to this routine will set the Center Frequency to the value passed in as a long integer. This value must be passed in units of kHz. For example, to set the GSP-810 to 10MHz, the value passed to the routine would be 10000.

Currently the return value should be ignored.

### SetRBW

A call to this routine will set the RBW field to the value passed in as a long integer. The following conversion can be used:

RBW	Parameter Setting
3 kHz	0
30 kHz	1
220 kHz	2
4 MHz	3

Note that the RBW can **only** be set directly when the GSP-810 is in zero span. Incorrect settings will be ignored by the GSP-810. Returned values should be ignored.

### SetReflvl

A call to this routine will set the Reference Level to the value passed in as a long integer. The following conversion can be used:

Reference Level	Parameter Setting
-30 dBm	0
-20 dBm	1
-10 dBm	2
0 dBm	3
10 dBm	4
20 dBm	5

Currently the return value should be ignored.

### SetSpan

A call to this routine will set the Span to the value passed in as a long integer. The following conversion can be used:

Span	Parameter Setting
Zero Span	0
2 kHz/div	1
5 kHz/div	2
10kHz/div	3
20 kHz/div	4
50 kHz/div	5
100 kHz/div	6
200 kHz/div	7
500 kHz/div	8
1 MHz/div	9
2 MHz/div	10
5 MHz/div	11
10 MHz/div	12
20 MHz/div	13
50 MHz/div	14
100 MHz/div	15

Currently the return value should be ignored.

### ChangeComPort

A call to this routine will change to the comm port passed in as a long integer. The expected values are 1, 2, 3, or 4 for comm ports 1-4, respectively. The other parameters normally configurable for RS-232 ports are setup automatically for the GSA-810.

### GetAnalyzerTraceData(LPSAFEARRAY FAR \*);

A call to this routine will result in one trace of data being returned. A pointer to an array of BYTE type (in Visual Basic) should be passed in. This array will be allocated, initialized, and the data from the unit will be located in it as binary data. The structure of the data is as follows:

#### Struct Data

```
{
    char freq[7];    //center frequency (ascii characters) in units of kHz
    char span;      //Decimal index indicating the current span
    char ref_level; //Decimal index indicating the current reference level
    char RBW[7];    //ascii characters indicating the current RBW
    unsigned char data[1000]; //organized as lowbyte|highbyte (500 points)
    unsigned char checksum; //checksum of the bytes
    unsigned char reserved; //unused
}
```

The total length of the data is 1018 bytes. The “data” portion of the structure is the actual trace data. NOTE: the lower 10 bits (out of 16 bits) is valid data. The Upper bits may need to be masked off (i.e. the maximum value after masking should be 0x03ffh).

To convert from a 10 bit binary sample to the equivalent level in dBm, the following algorithm can be used:

Reference Level (dBm) – (((0x03ffh – sample) \* 0.78125)/10.0) = level (dBm)

A return value of 1 indicates success. A return value of 0 indicates a failure while attempting to read from the GSA-810. If a failure occurs, the delay could be as long as 30 seconds. This normally occurs if the serial cable is not hooked up or the GSA-810 is not powered up. To verify that those items are working properly, read the serial number first since there would not be a long delay if there is an error in communications.

#### Marker Functions:

The following functions have been added to the dll. The GSP-810 software must be upgraded to version 1.20 in order to support these functions.

```
extern "C" long __stdcall MarkerToCenter(void);
```

This function performs the Marker-to-Center function in the GSP-810. The behavior of the unit is the same as when a user presses the key on the front panel. It takes no parameters and a status is returned. This value should be ignored.

```
extern "C" long __stdcall MarkerToPeak(void);
```

This function performs the Marker-to-Peak function in the GSP-810. The behavior of the unit is the same as when a user presses the key on the front panel. It takes no parameters and a status is returned. This value should be ignored.

```
extern "C" long __stdcall GetMarkerLevel(long);
```

This function will return the marker level. The long value passed in is the marker number (1 or 2). The returned value is the level reported for the specified marker in tenths of dB units. For example, if the level of the marker is –34.5 dB, the returned value will be –345.