To: John Effland

From: Tim Thacker

Date: June 17, 2002

Subject: Spurious Response Measurement of LO System

This document provides summaries, flowcharts, and code of the Visual Basic program written to detect spurious responses from signal sources. Detailed descriptions of the program are given section by section, along with graphical flowcharts of the logic of the program, and a file path to where to locate the code.

Hardware Setup

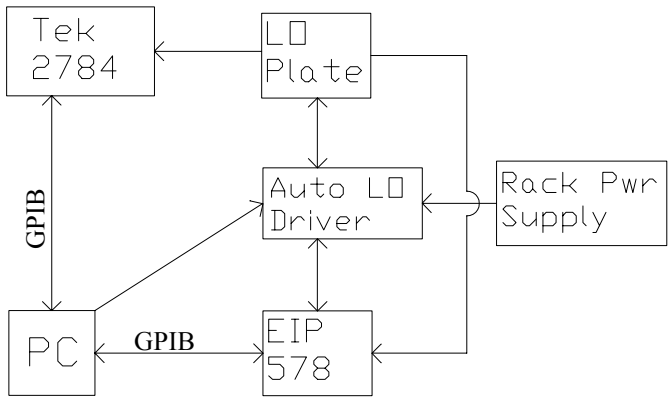


Figure 1: Hardware Block Diagram

The PC must contain a NI PCI-6025E card with the NI-DAQ v6.5 software minimum to drive the YIG. The PC must also contain a GPIB card and have all the appropriate DLL libraries loaded for programs to run. The PC must be connected to the spectrum analyzer and EIP Counter via GPIB cables and the AutoLO Driver must be connected through the 6025E card by means of a 100 line ribbon cable (comes standard with the 6025E board), which is spliced into a AMP connector cable (see **Table 1** for pin-out). The EIP Counter and AutoLO Driver Box are connected through a coax-BNC cable. This connection allows the EIP Counter to send a correction voltage that locks the YIG to a desired frequency. The rack power supply is hooked up to the AutoLO Driver Box by means of ±20V NRAO power cable. The AutoLO Driver Box is connected to the LO Plate by two cables: a power cable that gives power to the amplifiers, doubles and triplers, and a cable that connects the YIG.

AMP Wire Color	Ribbon Pin	Purpose
BLK/WHT/RED	20	DAC0OUT
GRN/WHT/BLK	33	DGND
WHT/RED/BLK	33	DGND
RED/WHT/BLK	35	+5V
BLU/WHT	51	PC7
WHT	53	PC6
GRN/WHT	55	PC5
RED/WHT	57	PC4
RED	59	PC3
GRN	61	PC2
ORG	63	PC1
BLU	65	PC0
BLU/BLK	67	PB7
BLK	75	PB3
WHT/RED	77	PB2
WHT/BLK	79	PB1
RED/BLK	81	PB0

Table 1: Pin-out of the connection between AMP and ribbon cables

Software Requirements

User Input

The system shall accept a range of source frequencies (from an absolute minimum start frequency of 198 GHz to an absolute maximum stop frequency of 312 GHz), and step a source oscillator between and including these frequencies. All frequencies will be the actual LO frequencies, and not those of the YIG oscillator. The start and stop frequencies, as well as the step size, can be changed from a dialog box (see **Figure 2**). Nominal values are presented for all inputs when the dialog box initially is displayed.

The dialog box will also have the capability of changing the start and stop frequencies, span, and reference level of a spectrum analyzer (see **Figure 2**).

Abort capability

The user can abort the measurement anytime by pressing the ESC key on the keyboard. By depressing the key, the program shall suspend and present a dialog box querying the user if they wish to continue or abort the measurement. If aborted, the software shall reset all the instruments to a pre-defined state. The dialog box also includes a "Quit/Abort" button (see **Figure 2**) for before/after the measurement of data to unload the program.

Output

The program shall generate a table showing, for a given LO, the frequencies and level in dBm and plot the results onto a chart created by the program.

Summary

Spurious signals are detected in the program by:

1. Commanding the source to the appropriate LO frequency
2. Scanning a broad range of frequencies (user-defined from dialog box) using the TEK 2784 spectrum analyzer. That instrument's "signal identify" mode is required to determine which spurious spikes are real.
3. The "signal identify" mode employs a second trace on the spectrum analyzer that provides a means of distinguishing real signals from those generated by the spectrum-analyzer. Real signals are determined by a spike on both normal and identify traces at approximately the same frequency. This contrasts with spurious signals on the normal trace generated internally by the spectrum analyzer because they don't show a corresponding spike on the identify trace.
4. The software returns both traces, detects the spikes, and compares the detected spikes from the normal and signal identify traces to determine which signals are real. Spikes are detected by first averaging all values returned for a particular trace and then flagging those values that exceed the average by an amount preset in the program.

Signals, which are observed as spikes on the spectrum analyzer traces, are detected in the software by querying the spectrum analyzer for data through a GPIB interface. The data collected is then converted to dBm values, and frequencies are assigned to each value.

The comparison of the spikes recorded in both traces of the spectrum analyzer is performed in the software by averaging all the data points (for the normal and identify traces) to find base values. These base values are then subtracted from all points to create a normalized set of data, and the program checks to see if each of the normalized data points are above specific values. If both normal and identify normalized sets are above the specific values, the program records into arrays the frequency and dBm level for that spike.

(see **Diagram 1** for overview of software flow)

Specific Software Functions

Object	Function	Purpose
CTEK2784	bInit	Initializes analyzer
	bSetAnlyzr	Sets specific parameters on analyzer
	bReadAnlyzr	Reads waveforms off analyzer
	bAvgNrmlzeData	Averages waveform data
	bFindRealSpkes	Finds the real spurious responses
	bGetPeak	Gets peak value and frequency of spike
	sGetError	Gets error strings that occur
CLODriver	bInit	Initializes YIG and EIP Counter
	bSetFreq	Sets desired frequency using YIG and EIP
CUtil	mPause	Creates pause in program x seconds long

Table 1: List of Classes used, their functions and purposes

The program begins by initializing all object variables to their respective classes (see **Flowchart 1**). The program will next display a dialog box (see **Figure 2**) with nominal values the user must verify or change to desired values for the LO Plate and spectrum analyzer. The dialog box gives the options of changing the start, stop, and step in frequency for the LO source; and the options changing the start/stop frequencies, reference level, span, and dB/DIV for the spectrum analyzer. Once the user has selected the desired values, depressing the "Read Spectrum Analyzer" button (see **Figure 2**) will cause the program to start collecting data.

The "Read Spectrum Analyzer" button leads to the "ReadSpecAnlyzr_Click()" subroutine. This subroutine begins by creating a new spreadsheet to store the spurious responses. The program then initializes the YIG, EIP Counter and spectrum analyzer, and sets the YIG and locks the EIP Counter to the users desired frequency. A while loop then cycles through the process of:

I) Setting the analyzer to the parameters selected from the dialog box
II) Reading the waveforms from the analyzer and storing them into arrays
III) Averaging data arrays taken
IV) Finding spurious responses using averaged data, and then storing the dBm values and frequencies into separate arrays
V) Changing center frequency of spectrum analyzer
(see **Flowchart 2**)

The program then records all the real data points found onto the created spreadsheet. The analyzer variable that controls the Span is then set from the users desired to 20 MHz. This effectively zooms in on the waveform. Another while loop is then invoked to examine the signals thought to be real. The loop follows the same cycle as being, but now zoomed in on the signal so it can accurately determine if the signal is real or imaginary. The loop also contains a function that retrieves the peak value and frequency of peak value if the signal is determined real.

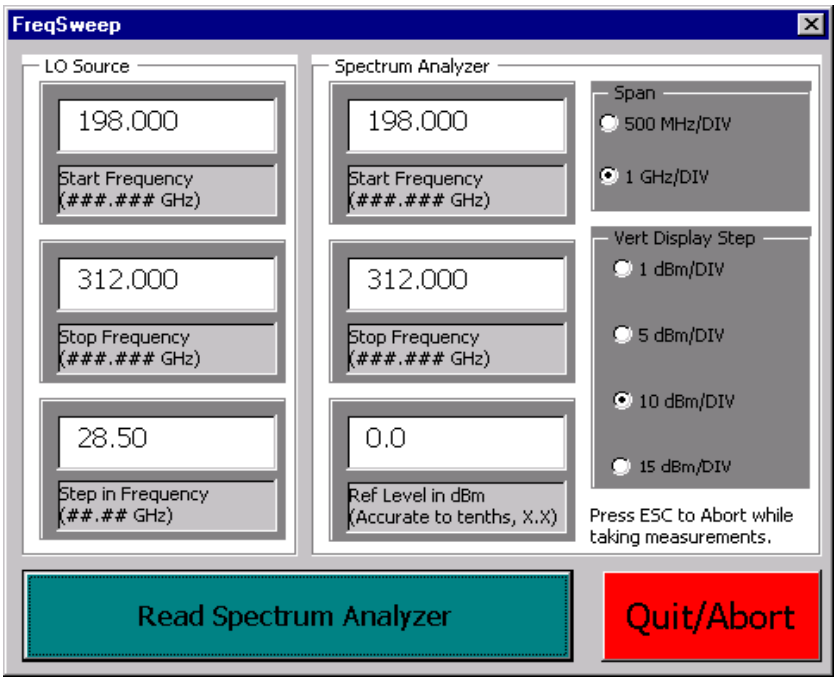


Figure 2: Image of Object Window

The program then writes all the values of the signal found to be real

The program also includes two abort features. If the user needs to quit in the middle of a measurement, they may press the ESC key on the keyboard; by pressing the ESC key, the user calls the "Quit" routine (see **Flowchart 3**). If the user is not currently running a measurement they may press the Quit/Abort button, which leads to the "Quit" routine (see **Figure 2**). This will prompt the user if they really want to quit; and if so, unload all objects and exit the program (see **Flowchart 4**).

Diagrams and Flowcharts

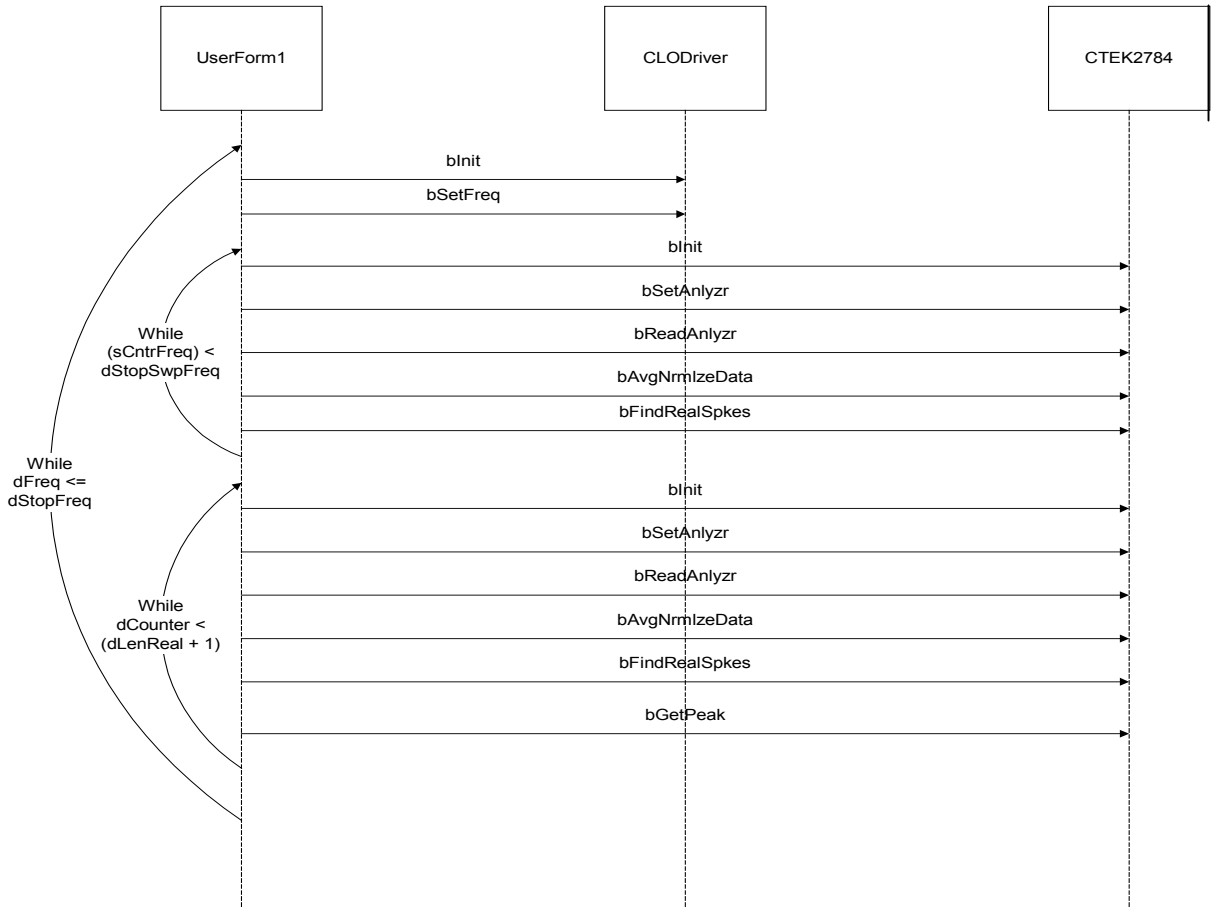
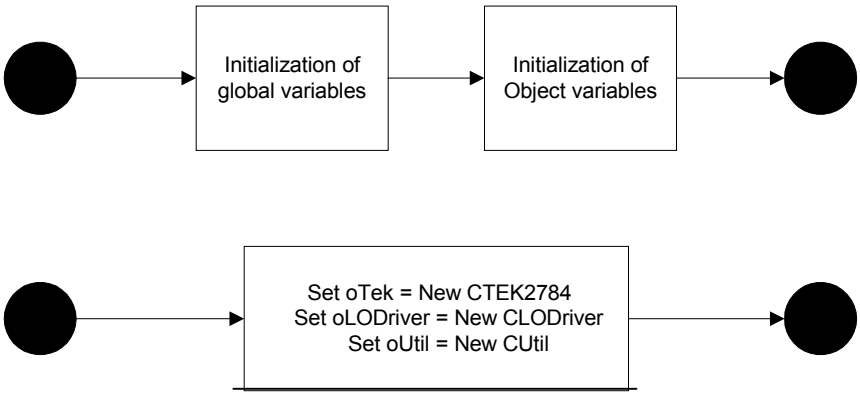
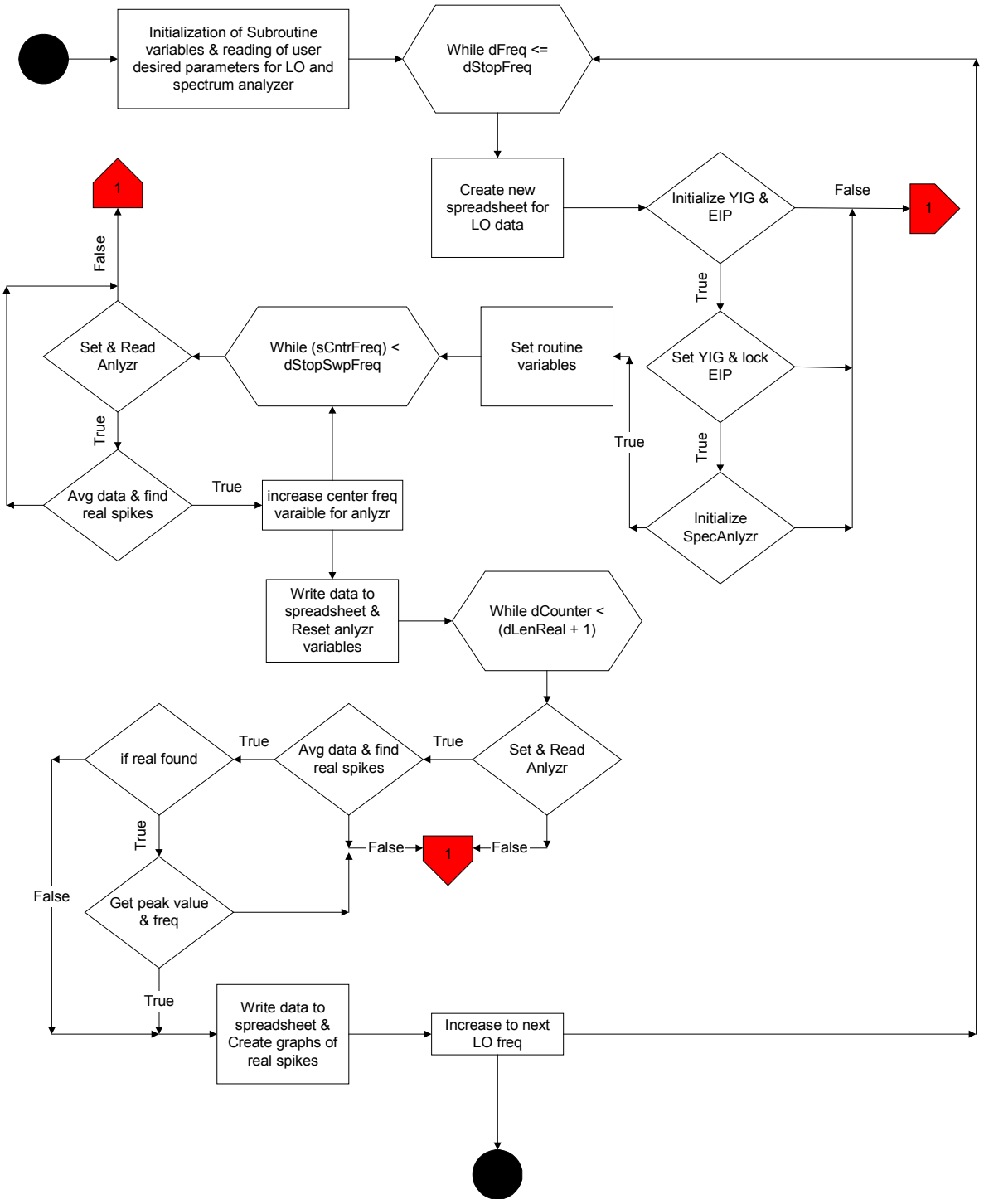


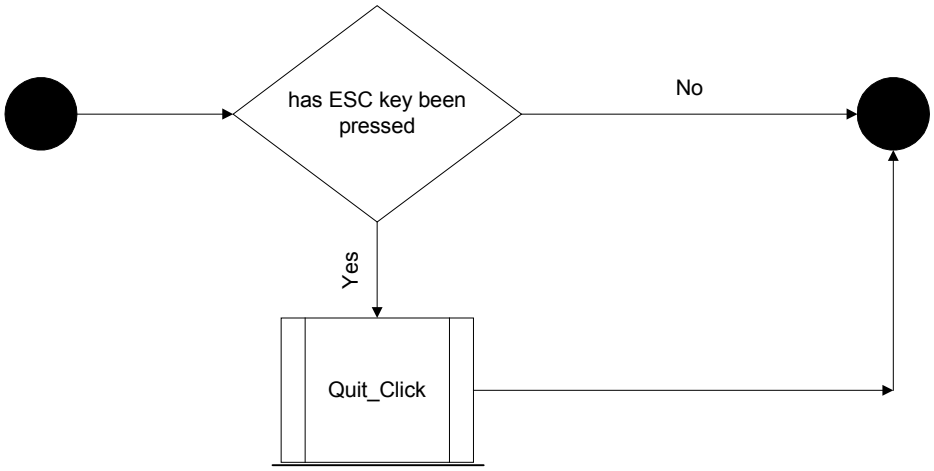
Diagram 1: overview of the flow of the software



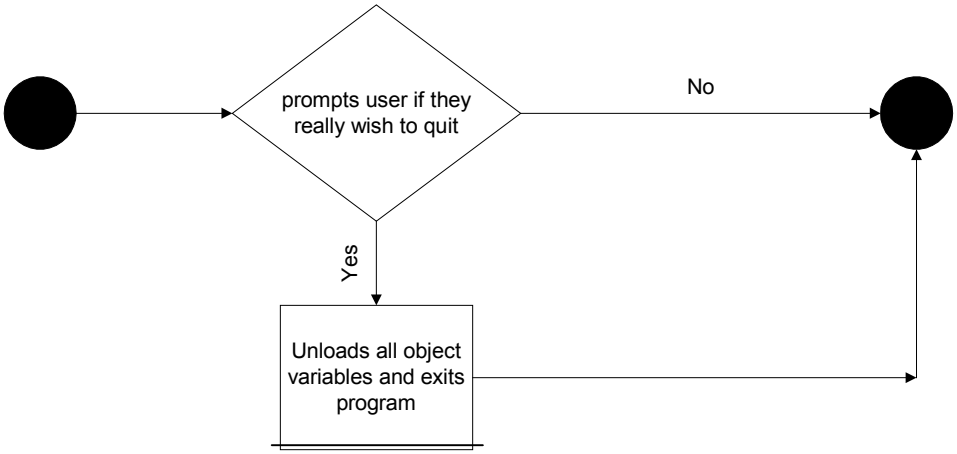
Flowchart 1: Initialization of variables and objects



Flowchart 2: ReadSpecAnlyzr Subroutine
shows how program sets and sweeps LO frequencies.



Flowchart 3: Esc key abort feature



Flowchart 4: Quit_Click routine

Code

Code can be found in spreadsheets in folder:

\\Eagle\cv-cdl-sis\Tthacker\AutoLO\LO Sweep\SpecAnlyzr Class\LOTest v3.0.xls