The LO Power Measuring program performs a sweeping routine that gathers power readings from the Anritsu ML83A power meter when connected to the automated LO plate. The program essentially measures power versus frequency and places the date, time, power level, and frequency of each reading into an Excel spreadsheet.

On the initial input form, called UserForm1, the user enters the start frequency, frequency step size, and the stop frequency. The user then clicks the command button labeled “Sweep” which performs the sweep. Each time the program is run, a new spreadsheet is created to display the data for that particular sweep.

As shown in the attached sequence diagram, clicking the command button on the form initializes the subroutine named LOAnritsu. LOAnritsu begins by initializing the LO Driver object (oLODriv) and the GPIB object (oGPIB) with the address of the EIP Counter. After formatting the spreadsheet that will hold the data, the subroutine sends a command to the LO Driver object to set the signal to the specified frequency. The program runs in a loop either until the EIP counter locks on to the signal or until the loop has been iterated 5 times (in which case the power reading will be flagged because the specified frequency was not locked). Afterwards, the actual frequency of the signal is read and placed in the spreadsheet. Following the frequency read is the power read which is the largest portion of the program.

The power meter object oPower includes a power read function, nReadPower, which has 3 optional arguments that configure the meter. The third argument, vActiveForm, does not apply to this program and will generate an error if passed. The function nReadPower begins by initializing the Anritsu power meter and then configures the meter to a set of default conditions. Next, the meter is read once to flush the buffer of any old data. Then a series of reads are performed using a function that computes the standard error of the data and outputs the mean when the standard error falls below a specified threshold.

After the power read function nReadPower completes, all of the data for that reading is displayed on one row of the spreadsheet. The subroutine completes by calling the function bGetStatLett, which reads the status letter of the power meter. If the meter is under-ranged or over-ranged, the status letter will indicate so. Thus if the meter is in either of these two states, the program will recognize the status and flag the value generated by the power read on the spreadsheet (a comment is added to that particular cell).

Finally, the frequency of the signal is changed by the amount specified by the user on the form, and another read is taken. This loop is iterated until the frequency reaches the stop frequency.

Sequence Diagram

The following page is the sequence diagram for the program. One aspect of the program that might appear confusing to the user is the bConfigure/bInit sequence between the oPwrMtr and oAnritsu objects. The user might wonder why the bConfigure function is called if the power meter has already been initialized because the bInit subroutine calls the bConfigure subroutine itself. bConfigure is run if the meter has already been initialized just in case any of the primary meter settings that affect the program are somehow changed. Therefore it is used to prevent program errors.