# AVNA User's Manual

Bob Larkin W7PUA
Evans Wetmore K3VFA

November 25, 2020

## Contents

## Part I

# Features and Capabilities

The capabilities of Version 0.82 are the following:

- Audio Impedance Measurement
- Audio Vector Network Analyzer (Audio Transmission Measurements)
- Audio Spectrum Analyzer covering up to 40kHz with graphical display
- Vector Voltmeter with frequency selectivity and adjustable phase offset
- Four Signal Generators (added together) with calibrated output.
- Weighted and Unweighted Noise Measurement

- Envelope and Group Delay Measurement
- Screen Save to BMP file for all functions
- Calibration of input and output levels and for the touch screen
- AVNA Control via Serial Port

# 1 Power Up

When the AVNA is turned on, it starts with an "Audio Test Instrument" screen that allows selection of the function. This is, as usual, selected by the six touch screen buttons at the bottom of the screen. In addition, if an $\mu$SD card is present in the Teensy 3.6 socket, a button with "S" shows in the upper right corner for "Screen Save".

At startup, the Serial Monitor shows the status of the $\mu$SD card and a "`DIR /ls`" listing such as:

- Initializing SD card...and a card is present.
- Partition found: FAT32
- Volume size (Mbytes): 7378
- Files found on the card (name, date, time, and size in bytes):
  - AVNA1_00.BMP 2000-01-01 01:00:00 230454
  - AVNA1_01.BMP 2000-01-01 01:00:00 230454

**Note:** $\mu$SD cards have grown in GB capacity over the years. There may be problems with large capacity cards greater than 8 DB..

# 2 Audio Impedance Measurement

xxx

## 2.1 Description

xxx

## 2.2 Instructions

xxx

## 2.3 Discussion

xxx

# 3 Audio Vector Network Analyzer

xxx

## 3.1 Description

xxx

## 3.2 Instructions

xxx

## 3.3 Discussion

xxx

# 4 AVNA Vector Voltmeter

xxx

## 4.1 Description

The Vector Voltmeter (VVM) measures the input rms voltage at any frequency up to about 40 kHz. The frequency is set by the frequency of Signal Generator 1 (see below). If the test is using the left side signal generator output, the phase shift measurement is useful and can be offset by a number set on the VVM screen, using Up/Down buttons. If an external generator is used the phase will progress at a rate determined by the frequency difference. The bandwidth of

## 4.2 Instructions

xxx

## 4.3 Discussion

The operation of the Vector Voltmeter (VVM) is worthy of a few words. As a test instrument, it is closely related to the transmission measurements part of the Vector Network Analyzer AVNA). Much software is shared between the two instruments. The AVNA determines the relative gain or loss, amplitude and phase, through the transmission path. The VVM is calibrated so that the magnitude of the incoming signal is measured, along with the phase difference between Signal generator #1 and that signal.

Two mixer (multiplier) outputs are the in-phase and quadrature signal levels. The square root of the sum of the squares of these two provides the magnitude of the incoming signal. The Signal Generator #1 (SG #1) sets the frequency of the measurement. Low pass filtering after the mixers sets the requirement that the incoming signal must be within a few Hz of the SG #1 frequency. In many cases, it is easiest to just use the Signal Generator output on the Impedance measuring terminals of the AVNA which is both on frequency and without having the phase changing with time.

If the SG #1 signal is used as a signal source, the phase will not be shifting with time with time. For this the phase offset can be useful. This has up/down buttons to allow setting to 0.1 degree. The offset can be positive or negative. This sis a convenience for zeroing the displayed phase and does not change the basic measurement.

The input needs to be within the range of the ADC. The maximum input is a little more than 0.2 Vrms or 0.6 V p-p. Higher voltages require an external voltage divider. The VVM can be used with the 50-Ohm input terminator, or not for which the input impedance is 1-megohm in parallel with around 25 pF. In all cases, the VVM shows the voltage at the input terminals. The displayed voltage is the RMS value. This is the same as HP used on the (now old) 8405A VVM. If the waveform is not sinusoidal, and there are harmonics, the displayed value is for the fundamental at the frequency of SG #1.

You can see how close you are to overload by the little "ADC %p-p=xx.x" on the screen. This is the per cent of full ADC range for the input. If the level gets to 100%, the voltage display turns red.

The very narrow bandwidth of the VVM allows low level signals to be measured. With SG #1 turned off, I see a residual noise of about 10 uV. But to use that, an external source generator is needed, with SG # 1 tuned to the same frequency. Otherwise, leakage through the CAL switch U4B in the AVNA causes a signal of about 55 uV with the 50 Ohm terminator or around 1.5 mV without.

The HP 8405 VVM has two inputs and phase-locked tuning to the reference input. We only have one input channel, and so this feature cannot be supported. But wait, there actually are two inputs. If the AVNA was rewired to remove R46 to R49 and bring those leads to a switch and to a second input connector, a full 8405A style phase-locked loop could be implemented. But, not now!

Also, if you fail to see a measurement, it is most likely that SG #1 is not within a few Hz of the frequency of the input signal.

A couple of minor corrections. In the example, the comments on the SigGen settings have the wrong amplitude levels. The commands are OK. The correct comments are

`SIGGEN 1 1 1234.56 0.1 3` // SG #1 as a triangle wave 0.1 v p-p
`SIGGEN 4 1 0.0 0.003 0` // SG #4 Gaussian White Noise, $\sigma = 0.003$

# 5 Audio Spectrum Analyzer (ASA)

xxx

## 5.1 Description

xxx

## 5.2 Instructions

xxx

## 5.3 Discussion

Multiple FFT are averaged together to produce the ASA graph. The width of the display can be varied in 2:1 steps from 3 to 48 kHz with the resolution bandwidth varying proportionally. The normal scale is 80 dB which is10 dB/div, but a 5 dB/div option is available along with offsets in 5 dB steps. The peak level on the screen is indicated at the top left and the frequency of this peak is at the top right. The level may be higher than the graphics peak as it is the sum power of the peak "bin" and the two adjacent ones. The frequency indication includes interpolation between bins and has counter accuracy for stronger sine waves. The input goes to the right hand Transmission port. The calibration is in dBm into 50 Ohms and assumes that the 50 Ohm input terminator is switched in.

# 6 Audio Signal Generator(s)

xxx

## 6.1 Description

xxx

## 6.2 Instructions

### 6.2.1 Waveforms

There are three signal generators and a fourth Gaussian White Noise Generator. The sum of these waveforms appears at the left or "impedance" terminals. These are not available when the AVNA instrument is operating, that is, when impedance or transmission measurements are being made.

The signal generators can produce either sine or square waves when under direct screen control. The frequency can be specified from 10 to 40000 Hz in 1 Hz steps on-screen. The amplitude is specified by the peak-to-peak value in volts. The output impedance is 50-Ohms and the voltage displayed is that delivered to a 50-Ohm load. When open-circuited, the voltage will be double the displayed value. For convenience, there is provision to turn each generator on and off while leaving the settings alone.

For sine waves, the waveform is excellent up to about 5/12th of the sample frequency, above which the amplitude drops off. That corresponds to 40 kHz for the highest 96 kHz sample frequency. For non-sine waves harmonics of the fundamental frequency are needed to build the waveform. The missing pieces, above 5/12 of the sample frequency, cause the waveform becomes distorted. For waveforms like the square wave, this can be an important limitation and suggests keeping the frequency well below half the sample frequency. In general, it pays to look at the waveform on an oscilloscope if the details are important.

If 3 sine waves are turned on at once, there can be times when all three will add together for a maximum voltage. The sum of the three voltage settings should therefore be kept below the overload point of about 0.6 volts.

### 6.2.2 Noise

Sig Gen #4 produces Gaussian White noise. There is an amplitude setting that corresponds to the 1-sigma voltage. This is convenient since the noise power is this voltage squared and divided by 50. What is not quite as convenient is to find the voltage setting that prevents overload, as Gaussian noise theoretically has no limit and the generator used here can achieve 12 times the 1-sigma voltage. If you are looking for a high level of noise output, it is reasonable to set the 1-sigma point at 0.1 Volts overload (5-sigma = 0.50 Volts). For a 96 kHz sample rate, this will overload about every 20 seconds, on the average. This will not be important for most experiments.

The noise power has a flat spectrum up to half the sample frequency. The power density in Watts/Hz depends on the sample frequency as set by the Spectrum Analyzer. For instance, if the 1-sigma noise level is 0.1 Volts, the power delivered to a 50-Ohm load is $0.1^2/50 = 0.0002$ Watts (0.2 milliWatts). If the sample rate is 12 kHz the power is spread across 6 kHz so the power density is 0.2/6000 = 0.0000333 mW/Hz. In dBm terms, this is -44.8 dBm/Hz. As seen on the spectrum analyzer, with a noise bandwidth (for this sample rate) of 17.6 Hz, this will show about 17.6 * 0.0000333 = 0.000587 milliWatts per bin, or in dBm terms -32.3 dBm per bin. This type of arithmetic allows setting the noise generator and the signal generators for any desired situation.

### 6.2.3 Screen Control

There is an overall signal generator LCD page, accessed by the "Signal Gens" menu button on the Home screen. This gives a summary of the settings for all four generators and allows menu selection of an individual control screen for any of the four. The first three screens, for Sig Gen #1 through #3, there is frequency selection to the nearest Hertz. These use Up/Down touch buttons for control. Similarly, all four have amplitude selection to the nearest millivolt. The menu items at the bottom of the control screen allow the individual generator to be turned On or Off. Two more menu buttons allow selection of either a sine wave or a square wave for the 3 signal generators.

Note that all the control of these screens is also available from the Serial control using the USB plug. Also note that over the Serial path the frequency can be controlled to the miliHertz, amplitude to the microVolt and the selections there also include triangle and two sawtooth waveforms.

### 6.3 Discussion

The settings of the four Audio Signal Generators (ASG) are independent, but they are all added together, appearing at the left hand port. The frequencies are set in 1 Hz steps from 10 Hz to 40 kHz. The amplitude has a maximum of about 0 dBm into a 50 Ohm load, which corresponds to 0.6324 Volts peak-to-peak. The screen setting is in peak-to-peak volts in order to not be awkward with non-sine waves. The waveform is selectable from the screen as either a sine or a square wave.

## 7 Noise Measurement

xxx

### 7.1 Description

xxx

### 7.2 Instructions

xxx

### 7.3 Discussion

xxx

## 8 Delay

xxx

### 8.1 Description

xxx

### 8.2 Instructions

xxx

### 8.3 Discussion

xxx

## 9 Screen Saving

xxx

### 9.1 Description

xxx

### 9.2 Instructions

xxx

### 9.3 Discussion

The screen save to BMP allows the AVNA screen graphics to be saved. However, it is available for any screen (there are 21 of them now), except the touch-screen calibration. A BMP graphics file is saved in either of two ways. First, if an $\mu$SD card is in the holder on the Teensy 3.6, there will be a touch button in the very upper-right corner. This turns red during the screen save. Up to 100 files can be saved to the $\mu$SD card with file names "AVNA1_nn.BMP" with nn being the lowest of the hundred that is available. The date and time for the files is not set as we have no clock.

Second, as an alternative to the $\mu$SD card, the code includes the ability to send a large hex file to the Serial Monitor. Binary files don't work, so the hex version is needed.

# 10 AVNA Unit Calibration

xxx

## 10.1 Description

xxx

## 10.2 Instructions

**Touch Screen Calibration** - There is some variability to the scaling of the touch screen. Until now, a nominal value was used. The touch screen calibration allows tapping at the upper-left corner and the lower right corner, with the lowest and highest x and y values being saved. This is recorded on the calibration screen until the upper-right corner is tapped at which point the values are made permanent. In using this, be sure to tap on the screen many times, seeing if more extreme values can be found. Also, there is no true "Cancel" for this test, except that if the upper-right corner is tapped before any other, it will continue to use the old values. It should be safe to just recalibrate the screen.

**Voltage Input Calibration** - We need a basic reference to correct for variations in the gain of analog circuitry. The preset values should be close, but variations are inevitable. This calibration requires a known amplitude of sine wave set to 0.100 Volts RMS which is the same as 0.2828 Volts peak-to-peak. The frequency is not critical, but should be in the 1000 Hz range. Apply this to the right hand "Transmission" terminals. The 50-Ohm termination can be on or off as the important item is the voltage at the terminals. Follow the on-screen instructions to do the calibration. This calibrates the ASA and VVM. The AVNA uses only relative voltages and does not need this type of calibration.

**Voltage Output Calibration** - Now we need to calibrate the ASG output levels. This function assumes that the Voltage Input Calibration is correct and adjusts the ASG levels to correspond. For this to work, the non-ground left and right terminals need to be clipped together. Then follow the on-screen instructions.

**EEPROM Update** - We keep adding items to the permanent EEPROM memory in the Teensy (this is Flash memory set up to emulate EEPROM). At startup, the program checks the version that was used to write the EEPROM and adds nominal values for anything that was added after that version. This prevents overwriting tune up values or the like. If everything is up-to-date, the Serial monitor, at startup, should say, "Loading EEPROM data; Turn-on EEPROM version was 80; EEPROM Load of 536 bytes." The first time version 80 is run, there will be notes about the updating of the data.

## 10.3 Discussion

xxx

# 11 System Issues

**SIMULTANEITY** - We have limited input and output ports. Combinations of Instruments and functions is limited by this. Usually, this will be caught by the firmware if you try the impossible. For instance, the four signal generators are turned off when the Vector Network Analyzer is running. What we can do is to have multiple generators on at once. For instance, the use of two (or three) generators for intermodulation analysis is works well up to the overload point. This, in part, is because the analog output amplifier uses much negative feedback and is quite linear. Likewise combining signals and noise works well for setting specific S/N ratios, or other experiments.

**COHERENCY** - If multiple waves are added together, and they are at non-harmonic related frequencies, the relative levels change at a fast rate. In the other extreme, if the two waves are at the same frequency, they maintain a constant phase relationship. This can alter measurement results, depending on the phase. As implemented here, the phase relationship between two generators is a random value. For these reasons, you should not measure intermodulation distortion with frequencies, such as 1000 and 2000 Hz. Standard frequenciey pairs, such as 700 and 1900 Hz or 60 Hz and 7000 Hz support predictable results.

# 12 AVNA Serial Interface

xxx

## 12.1 Description

Commands are entered by the USB serial link to the Teensy 3.6 board. The Teensy appears to the Host computer as a "Serial Device". As such, it can be controlled by an external program for graphing and the like. For manual control, the "Arduino Serial Monitor" that is accessed from the Arduino IDE under the menu "Tools" is useful. This monitor (terminal) supports CTRL-C copy and data can be gathered that way.

## 12.2 General Command Syntax

The format for all commands going to the AVNA is

```
CMD param1 param2 ....
```

where `CMD` is 1 or more characters, and the number of parameters varies. Only capital letters are valid for the `CMD` field. Error checking is not done on parameters. If 5 parameters are allowed, and only the first two are to be set, the last 3 do not need to be sent. Also the delimiter is shown as a space, but commas can be used as well.

For common commands used for manual entry, there are single character short cuts. For instance, `FREQ` and `F` are equivalent. See Section 5 for a complete list.

## 12.3 Operational Commands

**`ZMEAS refR`** This command sets the measurement to Impedance and the reference resistor value to `refR`. Valid `refR` values are either 50 or 5000 ohms.

**TRANSMISSION refR** This command sets the measurement to Transmission and the reference resistor value to `refR`. Valid `refR` values are either 50 or 5000 ohms.

**FREQ f** This command sets the measurements to a single (non-sweep) frequency. The frequency is set to `f` which can be either an integer or a decimal number from 10 to 40000. The parameter `f` regards 100.0 and 100 as the same. The achieved frequency may be slightly different than `f` to provide proper averaging of the multiplier outputs.

**SWEEP** No parameters are used. The 13 frequency sweep is set up, but not run (see `RUN` command).

**LINLOG rs ts** changes the units used for outputs to the serial monitor, or to the Touch Display. The four numbers following the `LINLOG` command set:

**rs = 0** Reflection coefficient to Serial in dB, and phase in degrees

**rs = 1** Reflection coefficient to Serial in magnitude (0,1) and phase in degrees

**rs = 2** Reflection data to Serial as equivalent Series Impedance or Parallel Suseptance (see `SERPAR` command)

**ts = 0** Transmission data to Serial in dB, and phase in degrees

**ts = 1** Transmission data to Serial in magnitude and phase in degrees

Default is the original values, for backward compatibility: `LINLOG 2 1`.

Short commands work if you do not want to change the touch display. For instance, `LINLOG 0` will just make the serial output in dB and degrees for impedance.

The command `LINLOG` without any parameters will return the current settings, such as `LINLOG 0 0`.

Settings are saved in EEPROM and so survive the power shutdown.

This makes a variety of Serial outputs available. For instance, for impedance measurement of a 200 uH inductor at 10 KHz.:

With `LINLOG 0 0`: the serial monitor shows a series of lines:

`10000.000 Hz`

```
Return Loss = 0.486 dB
Phase = 150.74
```

With `LINLOG 1 0`: the serial monitor shows a series of lines:

```
10000.000 Hz
Reflection Coefficient = 0.94565
Phase = 150.74
```

With `LINLOG 2 0`: the serial monitor shows a series of lines:

```
10000.000 Hz
Series RX: R=1.494 X=13.042 L= 207.6uH Q=8.73
10000.000 Hz
Parallel GB: G=0.008667760 B=-0.075683906 R= 115.37
L= 207.6uH Q=8.73
```

With `LINLOG 2 0` and `SERPAR 1 0` the serial monitor omits the suseptance:

```
10000.000 Hz Series RX: R=1.494 X=13.042 L= 207.6uH
Q=8.73
```

If the output is intended to be read by a program, you do not want all the annotation. When this is stopped by `ANNOTATE 0`, the data fields become comma separated, which spread sheets are happy with. For instance, the inductor measurement with ”`LINLOG 1 0` looks like `10000.000,0.94565,150.74`. The second number controls the transmission format in a similar manner, but options are only 0 or 1.

**`CAL`** No parameters are required. This is an immediate calibrate of either Z or T measurements at either a single frequency or all 13 sweep frequencies. This must follow `FREQ` or `SWEEP` and `ZMEAS` or `TRANSMISSION` to do the proper calibrate. This command must precede `RUN`. Note that component connections can be left in place for `CAL` of impedance measurements, but transmission measurements need a reference path for proper calibration.

**`RUN nRun`** This command causes the selected measurement to occur, either single frequency or a full set of swept measurements. The parameter, `nRun` is the number of measurement sets made. An `nRun` value of 0 causes a continuous set of measurements to be made. That is, `RUN 1` is a single measurement set; `RUN 27` does 27 sets and stops. Any new command will break the `RUN` command, so `RUN 0` is not really forever.

**POWER** No parameters are required. This is a power sweep of transmission measurements at a single frequency. Implementation is in place, although not thoroughly tested. Documentation is coming.

**SAVE** No parameters are required. This saves the current state to EEPROM for the next power down. This is seldom needed, as it is automatic if a parameter, such as the reference impedance is changed.

**LOAD** No parameters are required. Complements `SAVE` and retrieves settings from EEPROM. Also seldom needed.

**DELAY msDelay** sets a delay between repeated runs (see `RUN` command). The parameter `msDelay` is the amount of delay in milliseconds. This applies to measurements over the serial link and not the touch display.

**CALDAT** Not used at this time.

**SERPAR ser par** sets type of output data sent back to a host PC, during Z measurements, where:

| | |
|---|---|
| ser = 0 | Do not transmit series R-X data |
| ser = 1 | Transmit series R-X data |
| par = 0 | Do not transmit parallel G-B data |
| par = 1 | Transmit parallel G-B data |

For instance, the command `SERPAR 0 1` would include data for parallel G-B representation of the measured impedance.

The normal representation for an impedance is the (`ser = 1`) series form specified by a resistance, R, and a reactance, X connected in series..

When `par = 1`, representation is the mathematically related parallel conductance, G, and susceptance, B.

Note that annotation of the general form ”R=” can be added for both impedance forms by the ”`ANNOTATE 1`” command.

Avoid having both `ser = 0` and `par = 0`, as there will be no output to the PC, and there will be no obvious reason.

Unless modified by a `SERPAR` command, the unit defaults to `SERPAR 1 1`.

**`TEST rys sws`** This command sets the three relays and switches . This is not normally used for measurements, as the proper relay settings will be made by a measurement command, such as `ZMEAS`. (See TestCommand() in the .INO)

**`BAUD`** Do not use. All USB communications on the USB port are at 12 MBits/sec

**`ANNOTATE 0 or 1`** Responses can have annotation (1) or no annotation (0), over the serial communication.

**`VERBOSE 0 or 1`** For (1), extra information is transmitted. For (0), no extra information is transmitted.

**`SCREENSAVE n`** does a single save of whatever is shown on the screen. The parameter n is

**n is 1** Send the screen BMP image over the USB Serial link using Intel Hex

**n is 2** Save the BMP image to the uSD card, exactly as is done with the touch screen command.

For example, "`SCREENSAVE 1` $< Enter >$" will cause the transmission of many lines of Intel Hex.

For n=2, there is serial feedback in the form of the new file name, only if you have set "`VERBOSE 1`". Otherwise it is silent.

Now for n=1, the next step is to convert all 14,409 lines of hex data to a BMP file! *Because there is not a Windows 10 or Mac available to test on, this procedure for transferring a screen image to a .bmp bitmap file in your PC is for Linux. If anyone has a similar procedure for Windows or the Mac, please share it with us.*

1. Open the Arduino IDE (with Teensyduino installed) and make sure that the regular serial interface is operating. This involves

selecting the menu item Tools/Board "Teensy 3.6", the menu item Tools/Port: "dev/ttyACM0 Serial (Teensy3.6)", or similar and selecting the menu item Tools/Serial Monitor to open the serial terminal.

2. Make sure the AVNA is on and connected via the USB cable. Fiddle until you have a screen that you want to save. This can be done either by touch screen operation or by serial commands.
3. On the top line of the Arduino Serial Monitor, type, "`SCREENSAVE 1` $< Enter >$" without the quotes. The Monitor will show about 14,409 lines of hex code, printed in about 5 seconds. The cursor will be at the bottom.
4. Starting with the right end of the last line, highlight that line, "`:00000001FF`". Then, being careful to not click in the text area, use the scroll bar to go to the top of the hex text. Hold down the Shift key and carefully left-click on the start of the first, "`:020000040000FA`" line. The entire hex text should now be highlighted.
5. To get the hex to the clipboard, just hold down the CTRL key and hit '`c`'. There is no feedback from this. There is no menu item for this step, only the Ctrl-c.
6. Paste this entire hex area into your text editor (I use Geany) and save it to a a file with type `.hex`. That might be `/home/me/Documents/myscreen.hex`.
7. Open a Terminal and navigate to the directory location of the file just created, such as "`cd /home/me/Documents`." Now we will create the `.bmp` bitmap file, myscreen.bmp, that was the goal. Using the nifty `objcopy` utility, the command is `objcopy -I ihex -O binary myscreen.hex myscreen.bmp`.

Any graphics program will recognize this file type. It maybe desirable to lossless compress the file to, for instance, a `.gif` file. The redundancy in the screens `.bmp` is very high and the savings from compression are quite useful.

An almost last item: I think I mentioned before that if you examine the screen saves very carefully, you will find a few pixels that are the wrong color. This is a bug in the Teensy ILI9341_t3 library that seems to be from a timing error in the DMA read of the screen. It does not effect the usefulness of the images.

A second item: The `.hex` file has three hex encoded bytes per pixel. There are 320 x 240 pixels, or a total of 230,400 bytes of data. This is more bytes than a 16-bit address can handle. Intel Hex originally was 16-bit only, but later extended to 32-bits with the "type 4 extended address." The objcopy utility reads the extended addresses, but some Intel Hex readers may not, so be aware if there are issues.

## 12.4 Setup Commands

**`PARAM1 num refR50 refR5K`** This command is used to set the "exact" values of the two reference resistors (`refR50` and `refR5K`), if known. `refR50` and `refR5K` default to the design values of 50.00 ohms and 5000.00 ohms respectively. The parameter `num` is either 0 or 99. A value 0 causes the reference resistor values to be set to the two parameter values, `refR50` and `refR5000`. A num value 99 causes ALL EEPROM values to be set to the defaults, including `PARAM2` below. *Use the 99 value with care.* An example is "`PARAM1 0 50.22 5017.3`" which sets the two reference values, `refR50` and `refR5K`, to 50.22 and 5017.3 ohms respectively. `PARAM1` with no following parameters returns the current two reference resistor values.

**`PARAM2 capInput resInput capCouple seriesR seriesL`** Serves the changing of correction factors for the impedance measurements. For instance,

```
"PARAM2 37.0  1000000.0    0.22    0.07   20.0"
[units]  pF      Ohm         uF      Ohm     nH
```

sets the input shunt capacity, the shunt resistance, the coupling cap, lead resistance and lead inductance. Note `capInput` is the most likely item to change, and it can be changed with simply "`PARAM2 34.8`" To obtain current values, type `PARAM2` with no parameters

**`TUNEUP n`** is a set of six commands (`n = 1, 2, 3, 4, 5, or 6`) used at setup time or whenever values need to be checked. These are the values that correct for circuit "stray" components to improve the accuracy of impedance measurements. This command is available only over the serial port (no touch screen). It is directed manually, since components, opens, and shorts need to be connected. These measurements determine the six values that are listed by the two commands, `PARAM1` and `PARAM2` (not including the non-critical 0.22 uF coupling capacitor). The intention is that `TUNEUP` be used in 1 to 5 order. (See the separate write-up on doing the tuneup.) The `TUNEUP` family includes:

**`TUNEUP`** with no "`n`" prints a summary of this procedure.

**`TUNEUP 1`** with a short circuit on the Z terminals measures the stray resistance and inductance (`seriesR & seriesL`).

**`TUNEUP 2 REXT50`** (1) find a resistor around 50 ohms, (2) as accurately as possible, measure the actual resistance of this resistor (in ohms), (3) then run `TUNEUP 2 REXT50`; enter the actual measured value (which will probably not be exactly 50 ohms) in place of `REXT50`.

**`TUNEUP 3 REXT5K`** (1) find a resistor around 5000 ohms, (2) as accurately as possible, measure the actual resistance of this resistor (in ohms), (3) then run `TUNEUP 3 REXT5K`; entering the actual measured value (which will probably not be exactly 5000 ohms) in place of `REXT5K`.

**`TUNEUP 4`** with the Z terminals open measures the internal 1 Megohm resistor and the input capacity (`resInput & capInput`).

**`TUNEUP 5`** causes the six values to be made permanent.

**`TUNEUP 6`** reverts to the original six values that were used before `TUNEUP` was started.

Any of the steps can be omitted, *e.g.*, if no precise resistor around 50 ohms is available. But, the ascending order from 1 to 5 is important, in that, for instance, the value of the 5000 ohm reference affects the open circuit answers.

## 12.5 Example of Commands

As an example, a swept measurement of transmission data, as commanded for non-manual use, might look like the following: This does 1 Hz steps from 950 to 1050 Hz. Obviously, the `F xxx R 1` repetition would be done by a loop in a control program.

```
T 50
SWEEP
CAL
(Connect device to be measured for transmission)
F 950
R 1
F 951
R 1
F 952
R 1
. . .
F 1049
R 1
F 1050
R 1
```

# 13 Shortened Form of Commands

| Full Command | Abbreviation |
|---|---|
| ZMEAS | Z |
| TRANSMISSION | T |
| FREQ | F |
| CAL | C |
| RUN | R |
| POWER | P |
| SAVE | S |
| LOAD | L |
| DELAY | D |
| CALDAT | none |
| SERPAR | none |
| PARAM1 | none |
| PARAM2 | none |
| TUNEUP | none |
| LINLOG | none |
| BAUD | B |
| ANNOTATE | A |
| VERBOSE | V |

## 13.1 Discussion

xxx