

Accurate Frequency Measurements with your WSPR Setup

Joe Taylor, K1JT

Last updated: November 4, 2010

Introduction

The popular program **WSPR**, “Weak Signal Propagation Reporter,” has drawn new attention to the importance of frequency accuracy and stability in weak-signal amateur radio communications. Appendix C of the [WSPR 2.0 User's Guide](#) explains how to calibrate your modern transceiver so that you can know your own frequency and measure the frequency of other over-the-air signals with accuracies better than one Hertz. Using **FMT** — a package of new software tools described below — you can adapt these techniques to make measurements for other purposes, including the ARRL's Frequency Measuring Test. The next running of this friendly competition is November 10-11, 2010 — less than two weeks away, as I write. Measurements made with your standard **WSPR** setup can be good enough to put you in the top tier of contestants in the Frequency Measuring Test. You will need a transceiver set up for using **WSPR**, including CAT control. With this technique you won't need a GPS receiver, rubidium oscillator, high-accuracy frequency counter, or any other expensive laboratory-grade equipment.

The **FMT** tools go a long way toward automating the calibration of your radio and making frequency measurements with it. However, obtaining good results still requires you to think carefully about what you are doing, and why. These instructions should help to set you off on the right path.

Software Installation

The **FMT** software is packaged with the latest **WSPR** release, which can be downloaded from the [WSJT Home Page](#). Click on **WSPR** at the left margin, and then on the appropriate link under Downloads. Install the program in the usual way for your operating system. If you are using Vista or Windows 7, I recommend installing to a directory such as C:\Radio\WSPR rather than the default location C:\Program Files\WSPR.

Calibrating Your Radio

The **FMT** software tools are driven from the command line. (If you're not comfortable working from the command line, you'll have a bit more to learn. If you must have a GUI-driven program, you might need to look elsewhere.) Verify that **WSPR 2.11** works correctly and is properly configured for your soundcard and for CAT control of your radio. Execute at least one band-change command within WSPR, to be sure that your radio obeys CAT commands. **WSPR 2.11** will create a file `fmt.ini` containing parameters that `fmt` needs for CAT and soundcard control.

Open a command-prompt window and `cd` to the directory where you installed **WSPR 2.11**. Verify that the executable programs `fmt`, `fmtave`, `fcad` and `fmeasure` are present by entering each program name at the command prompt. Messages in response to these commands give simple guides to their required command syntax:

```
C:\Radio\wspr> fmt
Usage:  fmt <kHz> <0|1> <offset> <range> <tsec> <call>
Example:  fmt 10000 1 1500 100 30 WWV
```

```
C:\Radio\wspr> fmtave
Usage:  fmtave <infile>
Example:  fmtave fmt.all
```

```
C:\Radio\wspr> fcal
Usage:  fcal <infile>
Example:  fcal fmtave.out
```

```
C:\Radio\wspr> fmeasure
Usage:  fmeasure <infile>
Example:  fmeasure fmtave.out
```

Program `fmt` requires six parameters after the program name — a nominal frequency in kHz, a flag (0 or 1) to indicate whether this is a calibration measurement, the desired offset in Hz, search range in Hz above and below the nominal offset, specified duration of measurement in seconds, and callsign. For example, to make a 30-second calibration measurement of the WWV 10 MHz signal, offsetting your receiver so that the WWV carrier provides an expected beat note of 1500 Hz and searching over a range ± 100 Hz from that audio frequency, use the command:

```
C:\Radio\wspr> fmt 10000 1 1500 100 30 WWV
```

In response to this command, program `fmt` will set the radio's dial to 9.998500 MHz and acquire audio for 30 seconds. The audio signal is then analyzed in overlapping segments about 6 s long, spaced by about 3 s. For each segment the program prints a line similar to those in the following table:

UTC	Freq (kHz)	CAL ?	Offset (Hz)	fMeas (Hz)	DF (Hz)	Level (dB)	S/N (dB)	Call
15:34:44	10000	1	1500	1516.339	16.339	5.3	44.5	WWV
15:34:47	10000	1	1500	1516.491	16.491	9.1	46.3	WWV
15:34:49	10000	1	1500	1516.493	16.493	8.4	48.0	WWV
15:34:52	10000	1	1500	1516.317	16.317	6.7	42.3	WWV
15:34:55	10000	1	1500	1516.302	16.302	6.7	44.5	WWV
15:34:58	10000	1	1500	1516.500	16.500	7.2	40.5	WWV
15:35:00	10000	1	1500	1515.917	15.917	4.4	41.3	WWV
15:35:03	10000	1	1500	1515.745	15.745	5.2	40.8	WWV
15:35:06	10000	1	1500	1516.523	16.523	6.6	41.0	WWV
15:35:09	10000	1	1500	1516.423	16.423	5.4	43.2	WWV

Here `fMeas` is the observed frequency of the beat note and `DF = fMeas - Offset` is the measured dial error. Output from `fmt` also appears (without column headings) in the file

fmt.out — which is started afresh at each program invocation — and in fmt.all, where it accumulates until you explicitly delete the file.

As described in Appendix C of the [WSPR 2.0 User's Guide](#), to calibrate your radio you should make a number of measurements like this example, over a range of frequencies. This process can be automated by executing a Windows batch file or Linux shell script similar to the following one, which I call gocal.bat:

```
C:\Radio\wspr> type gocal.bat
fmt 660 1 1500 100 30 WFAN
fmt 880 1 1500 100 30 WCBS
fmt 1210 1 1500 100 30 WPHT
fmt 2500 1 1500 100 30 WWV
fmt 3330 1 1500 100 30 CHU
fmt 5000 1 1500 100 30 WWV
fmt 7850 1 1500 100 30 CHU
fmt 10000 1 1500 100 30 WWV
fmt 14670 1 1500 100 30 CHU
fmt 15000 1 1500 100 30 WWV
fmt 20000 1 1500 100 30 WWV
```

In Linux the name of the script is simply gocal (without the extension), but the contents are the same. Executing the batch file causes calibration sequences to be run for each of a number of stations whose frequency calibration you have good reason to trust. For this example I used local AM broadcast stations WFAN (660), WCBS (880), WPHT (1210), and the HF standard-frequency transmissions of WWV and CHU. Use gocal as a template and make your own batch file using stations you can receive well at the relevant time of day. You don't need as many calibration stations as shown in the example, but be sure to use enough to cover a range of frequencies from the AM broadcast band up through at least 10 MHz.

After typing gocal at the command prompt you should see the output from fmt for each line in the file, and the measurements will accumulate in fmt.all. Here's an abbreviated copy of fmt.all obtained by executing gocal.bat at my station one afternoon (the column headings were added by hand):

UTC	Freq (kHz)	CAL ?	Offset (Hz)	fMeas (Hz)	DF (Hz)	Level (dB)	S/N (dB)	Call
15:41:51	660	1	1500	1503.103	3.103	1.5	54.4	WFAN
15:41:54	660	1	1500	1503.103	3.103	1.7	54.1	WFAN
15:41:56	660	1	1500	1503.103	3.103	2.1	53.6	WFAN
...								
15:42:22	880	1	1500	1503.459	3.459	1.8	56.4	WCBS
15:42:25	880	1	1500	1503.460	3.460	2.6	55.5	WCBS
15:42:27	880	1	1500	1503.459	3.459	1.5	56.6	WCBS
...								
15:42:53	1210	1	1500	1503.850	3.850	1.0	56.6	WPHT
15:42:56	1210	1	1500	1503.851	3.851	-0.8	58.4	WPHT

15:42:58	1210	1	1500	1503.849	3.849	0.5	57.2	WPHT	
...									
15:43:23	2500	1	1500	1559.336	59.336	-4.1	9.7	WWV	*
15:43:26	2500	1	1500	1403.312	-96.688	-3.9	7.8	WWV	*
15:43:28	2500	1	1500	1418.310	-81.690	-4.1	8.1	WWV	*
...									
15:43:54	3330	1	1500	1506.960	6.960	1.4	65.2	CHU	
15:43:57	3330	1	1500	1506.961	6.961	0.5	65.7	CHU	
15:43:59	3330	1	1500	1506.960	6.960	0.1	65.7	CHU	
...									
15:44:25	5000	1	1500	1509.350	9.350	0.7	27.5	WWV	
15:44:28	5000	1	1500	1509.345	9.345	0.5	27.4	WWV	
15:44:30	5000	1	1500	1509.342	9.342	0.9	28.5	WWV	
...									
15:44:55	7850	1	1500	1513.547	13.547	17.1	58.6	CHU	
15:44:58	7850	1	1500	1513.550	13.550	15.5	60.0	CHU	
15:45:00	7850	1	1500	1513.556	13.556	17.1	58.4	CHU	
...									
15:45:26	10000	1	1500	1516.681	16.681	6.7	46.2	WWV	
15:45:29	10000	1	1500	1516.656	16.656	4.8	42.8	WWV	
15:45:31	10000	1	1500	1516.789	16.789	6.6	49.0	WWV	
...									
15:45:57	14670	1	1500	1523.601	23.601	-3.4	28.6	CHU	
15:46:00	14670	1	1500	1523.066	23.066	-3.4	31.0	CHU	
15:46:02	14670	1	1500	1523.660	23.660	-3.5	30.3	CHU	
...									
15:46:27	15000	1	1500	1523.940	23.940	5.1	49.2	WWV	
15:46:30	15000	1	1500	1523.818	23.818	4.5	53.1	WWV	
15:46:32	15000	1	1500	1523.832	23.832	1.7	50.5	WWV	
...									
15:46:58	20000	1	1500	1530.897	30.897	22.3	35.6	WWV	
15:47:01	20000	1	1500	1530.793	30.793	22.1	33.4	WWV	
15:47:03	20000	1	1500	1530.779	30.779	22.1	36.9	WWV	
...									

Notice that the DF measurements for a particular station usually agree to within a few tenths of a Hz. In addition, they generally increase (or decrease) in proportion to the station frequency. In mid-afternoon I can't hear WWV at 2.5 MHz, so those entries in this particular example are garbage. They should be deleted from the file before further processing. A rough guide to potentially questionable entries is provided by the S/N measurement, and an end-of-line asterisk flags any measurement with S/N < 20 dB. You must learn to recognize and remove spurious measurements from the table.

Use Windows Notepad (or your favorite text editor) to delete any bad lines from `fmt.all`. After this is done, run program `fmtave` on it to average all valid measurements for each calibration station. Your screen should look something like the output reproduced on the next page. The same output (without column headings) will appear in file `fmtave.out`. In this table `N` is the number of valid measurements at each frequency, and `rms` is the root-mean-square scatter of individual measurements of DF about the quoted average. Any entry in the

rms column larger than a few tenths of a Hz is a likely indicator of problems with the measurement.

```
C:\Radio\wspr> fmtave fmt.all
```

Freq (kHz)	DF (Hz)	CAL ?	N	rms (Hz)	UTC	Call
0.660	3.103	1	10	0.00	15:41:51	WFAN
0.880	3.459	1	10	0.00	15:42:22	WCBS
1.210	3.850	1	10	0.00	15:42:53	WPHT
3.330	6.960	1	10	0.00	15:43:54	CHU
5.000	9.343	1	10	0.00	15:44:25	WWV
7.850	13.559	1	10	0.02	15:44:55	CHU
10.000	16.772	1	10	0.11	15:45:26	WWV
14.670	23.245	1	10	0.25	15:45:57	CHU
15.000	23.824	1	10	0.05	15:46:27	WWV
20.000	30.834	1	10	0.07	15:46:58	WWV

Your next step is to fit a straight-line calibration function to the sequence of measurement pairs (Freq, DF) appearing in the first two columns of `fmtave.out`. As described in Appendix C of the User's Guide, you can do this by executing the program `fcalf`, passing it the name of the file containing your averaged calibration measurements:

```
C:\Radio\wspr> fcalf fmtave.out
```

Freq (MHz)	DF (Hz)	Meas Freq (MHz)	Resid (Hz)	Call
0.660	3.103	0.660003103	-0.036	WFAN
0.880	3.459	0.880003459	0.004	WCBS
1.210	3.850	1.210003850	-0.080	WPHT
3.330	6.960	3.330006960	-0.019	CHU
5.000	9.343	5.000009343	-0.038	WWV
7.850	13.559	7.850013559	0.078	CHU
10.000	16.772	10.000016772	0.199	WWV
14.670	23.245	14.670023245	-0.045	CHU
15.000	23.824	15.000023824	0.059	WWV
20.000	30.834	20.000030834	-0.123	WWV

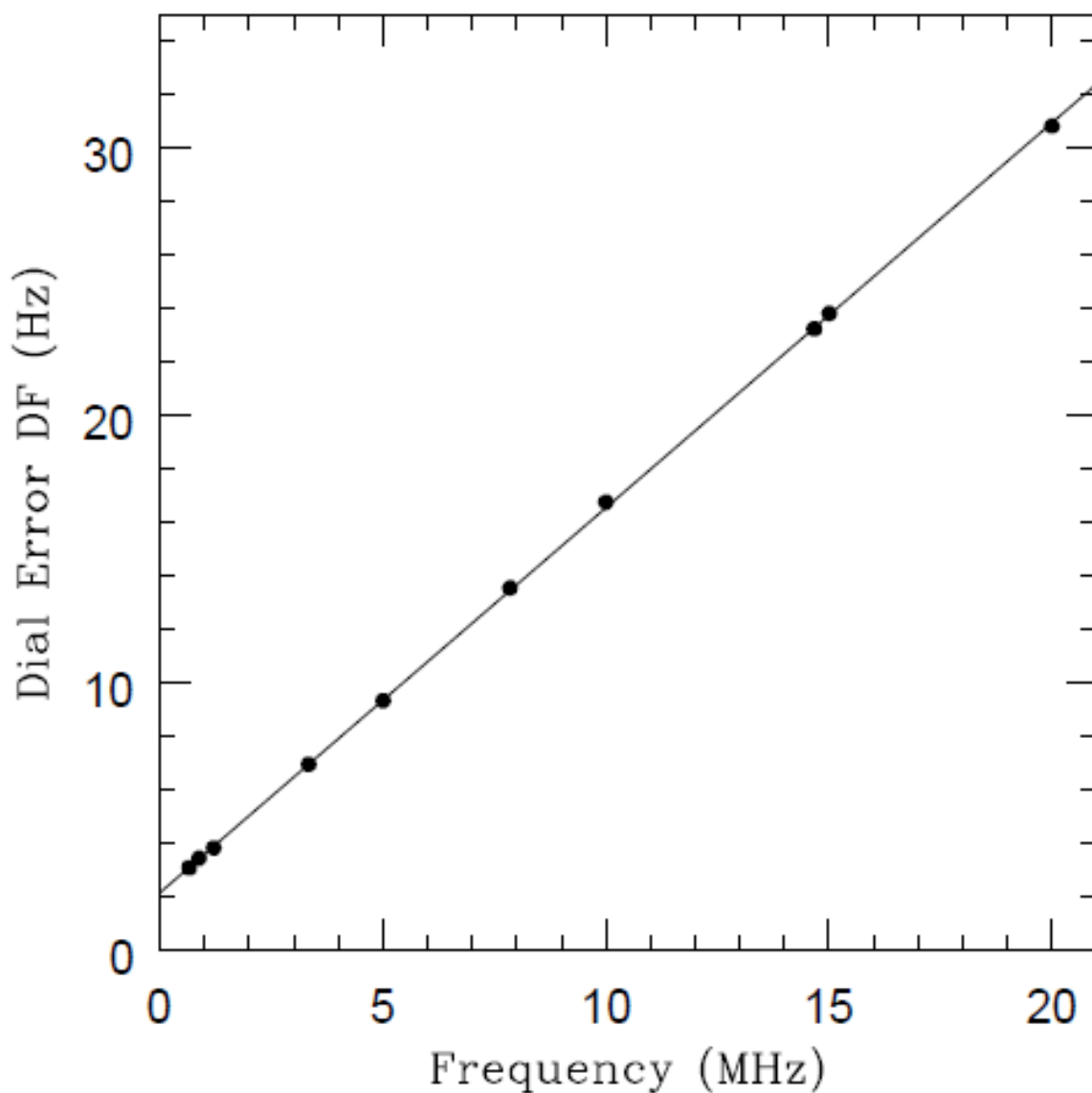
```
A: 2.19 Hz      B: 1.4384 ppm      StdDev: 0.097 Hz
err: 0.05      0.0047
```

Any lines that have 0 (zero) in column 3 of file `fmtave.out` are ignored when calculating the calibration parameters.

Column 3 of the on-screen output from program `fcalf` gives the frequency of each calibrating station, according to your uncalibrated radio. Column 4 is the difference between your *corrected* measurement and the nominal station frequency — in other words, the residual dial

error, in Hz, after applying your radio's calibration function. The last two lines of output give the intercept A and slope B of the straight-line calibration function, and the estimated uncertainties of A and B . (These are the constants that should be entered on the Setup | Advanced screen of **WSPR**. They are written into file `fcal.out`, and you can enter them automatically, the next time you start **WSPR**, by clicking on "Read A and B from fcal.out" on the Setup | Advanced screen.) The calibration constants will also be used in the final step of making **FMT** measurements.

The graph plotted below illustrates the excellent fit of the straight line $DF = A + Bf$ to the averaged measurements in the example file `fmtave.out`. The scatter of measurements about the fitted line is approximately 0.1 Hz — just as it should be, according to the calculated standard deviation 0.097 Hz, shown above.



Suggested Procedure for the ARRL Frequency Measuring Test

The November 2010 Frequency Measuring test extends over seven 15-minute intervals, each providing a different test signal. W8SKE will transmit an unmodulated carrier on about 7055 and 3575 kHz, K5CM on 3578 and 1844, WA6ZTY on 7097, and W6OQI on 7067 and 3567. See <http://www.arrl.org/files/file/W1AW/Other%20FMT%20files/FMT.pdf> for additional details about the Frequency Measuring Test.

To avoid the possibility of typing errors at crucial times, I recommend preparing batch files with names equal to the starting UTC of each test segment. For example, my first file for the November 2010 FMT will be named `0230.bat` (or simply `0230` in Linux) and will contain this single command line:

```
fmt 7055 0 1500 1000 180 W8SKE
```

Typing `0230` at the command prompt will run `fmt` for three minutes, producing 64 frequency measurements of the strongest carrier found between 7054 and 7056 kHz. Similar batch files should be prepared in advance for each of the other test signals.

My overall recipe for measurements in the FMT thus involves the following steps:

1. Delete or rename any existing file `fmt.all`.
2. Run the batch file `gocal`.
3. Use batch files to run `fmt` for each station, during its three key-down minutes.
4. Run `gocal` again.
5. Make a backup copy of `fmt.all`, naming it something like `fmt.all.nov2010`.

Running `gocal` both before and after your test measurements lets you test your radio's calibration stability. Note: you should turn it on and leave it on for at least 24 hours before the test, to minimize any possible thermal drift.

After making your backup file copy — step 5 above — your measurement tasks are complete!

Analyzing your Data

So... now it's time to analyze your data. To provide an example for this last step, let's imagine that the WWV signal at 5 MHz was a station of unknown frequency, rather than a calibrator. To effect this change I edited one line in the file `fmtave.out` described above, changing

```
5.000    9.343    1    10    0.00    15:44:25    WWV
```

to

```
5.000    9.343    0    10    0.00    15:44:25    WWV
```

(Your own measurements made during the Frequency Measuring Test will already have created lines in file `fmtave.out` with zeros in column 3, indicating that they are to be treated as unknown frequencies.) Your final step is now a simple one: run the program `fmeasure` on your averaged measurements, as illustrated on the next page.

```
C:\Radio\wspr> fmeasure fmtave.out
```

Freq (MHz)	DF (Hz)	A+B*f (Hz)	Corrected (MHz)
5.000	9.343	9.381	4.999999962

The output will also be saved in file `fmeasure.out`. Column 3 is the interpolated value of your radio's dial error at the frequency of the test signal. Column 4 is what you've been waiting for, your best estimate of the true frequency of the test signal. In this case, the one "unknown" frequency was measured to be 4.999999962 MHz — just 38 milliHertz below the correct value 5.000000000 MHz.

Individual Program Functions

To summarize: each of the utility programs and batch files in the **FMT** package performs a single function. Here they are again, in brief:

1. **fmt** – Sets the dial frequency of a CAT-controlled radio and performs a sequence of measurements of the strongest resulting audio tone near a specified offset frequency. Input parameters are taken from the command line, and output goes to files `fmt.out` and `fmt.all`. The latter file is cumulative.
2. **gocal** – Batch file, must be edited to your specific needs. Executes program `fmt` for a number of specified frequency-calibration stations. Output will accumulate in file `fmt.all`.
3. **0230, 0245, ...** – Optional batch files that you prepare in advance to execute program `fmt` for each key-down measurement interval in the ARRL Frequency Measuring Test. Each invocation of `fmt` will append more lines to `fmt.all`.
4. **fmtave** – Averages data found in a specified file having the format of `fmt.all`. Output goes to file `fmtave.out`.
5. **fcal** – Calculates a best-fit straight line for a data saved in `fmtave.out`. Results are saved in file `fcal.out`.
6. **fmeasure** – Calculates the properly calibrated frequency of each test signal found in file `fmtave.out`. Results are saved in file `fmeasure.out`, and these are the numbers you should report if you are entering the Frequency Measuring Test.