

## PIC 'AFA' Type Frequency Counter V1.03

### Overview

This frequency counter was inspired by the Freq Mite by David Benson K1SWL and the Stinger Singer by Dan Tayloe, N7VE. The programming is all by me since I did not have access to the source code of either example counter.

A major feature is that the frequency is announced in Morse, so there is no display. This makes a simpler and more compact device with very low current draw, but one must be fluent in Morse to use it.

### Features

- One button control to initiate count or menu actions with options communicated in Morse
- User programming of superhet IF frequency offset
- Selectable 1 Hz or 100 Hz resolution
- Option to suppress higher digits for faster readings
- Can be built SMT or through hole
- Fits on a tiny board X by Y (later) inches
- Setup saved on EEPROM and recalled on power-up
- Three selectable Morse annunciation speeds
- Selectable audible marker sound for each 1 kHz change while tuning

### Specifications (approximate / preliminary)

- Accuracy: Depending on the time base (crystal), can be as good as 1 Hz
- Resolution: Selectable to 1 Hz or 100 Hz
- Response time: 1 s for 1 Hz resolution and 10 ms for 100 Hz resolution
- Sensitivity: See table. 35 mVpp to 180 mV through 6 meters
- Idling current draw at 12 V input is 11 mA
- My prototype SMT version will read to just over 140 MHz
- Minimum counting frequency is about 50 kHz but could be lowered with a larger input capacitor

Minimum signal required for counting:

Frequency	dBm	Vp-p
50 kHz	+15	1.2 V
100 kHz	+7	1.4 V
500 kHz	+4	317 mV
1 MHz	+4	178 mV
5 MHz	-25	36 mV
10 MHz	-26	32 mV
20 MHz	-24	40 mV
30 MHz	-18	80 mV
50 MHz	-12	160 mV
100 MHz	-1	564 mV

## Operating instructions

A single momentary (< 1 s) press of the pushbutton causes the frequency to be read out in Morse

The menu for configuration of the counter works as follows:

- Hold in the pushbutton for more than 1.0 seconds and keep it held in as these options scroll by. The Morse letters sent tell the **current status** of the counter. Releasing the button after the appropriate letters are heard (1.5 seconds allowed) causes that option to toggle to the opposite state and the change to be written to EEPROM, making it non-volatile.
- One configuration change is allowed per menu entry. Enter the menu again if additional changes are needed.
- The counter will send the following letters (one of the two listed for each option) giving the current status and prompting to change if desired:

**T:** Tick (blip) sound for each change of 1 kHz in frequency

**Q:** Quiet mode – no sound with frequency change.

**GS:** Gate Short, means 10 ms gate for 100 Hz resolution

**GL:** Gate Long, means 1 s gate for 1 Hz resolution

**IF:** Intermediate Frequency means calculations for the effects of mixing are being done

**NI:** No IF, meaning the frequency is announced as measured, with no offset calculation

**A:** All, means read out all digits of the frequency

**S:** Suppress, means the 100 MHz, 10 MHz, 1 MHz and 100 kHz digits are not announced

**M:** Morse speed selection: Fast, Medium, Slow

**X:** Cancel, release the button and no changes will be made. This option allows checking the current options without changing anything.

- Again, after hearing the letter(s) representing the desired option, release the button and the opposite state will go into effect and be written to EEPROM to make it non-volatile. Or for Morse speed, the selection process begins. After EEPROM is written, a Morse 'R' is sent to confirm it to the user.

## **Options on Start-up**

Normally on power-up, the counter sends the revision number as Vnnn and is ready to go. If the pushbutton is held closed on startup, the revision number is not sent but a BT character is sent and then a two-second pause, followed by three start-up options:

The board will send 'D' then 'IF' and then 'C' in Morse repetitively, 1.5 seconds apart. Release the button after 'D' is heard to select the *default* option. Release it after 'IF' is heard to select *program IF offset*. And release after 'C' is heard to turn cut numbers ON and OFF.

### The 'D' for *default* option:

A temporary (not written to EEPROM) change to "full featured" operation can be made by holding in the PB while powering on the counter. A "dit dit" or "EE" Morse acknowledgement is made when the switch is released. The effect is to put the counter into the mode of 1 Hz resolution, all digits announced, and no I.F. offset adjustment made. Since this change is not saved to EEPROM, powering the counter off and on again quickly restores the previous configuration. This is an easy way to quickly switch to a full featured, conventional counter mode temporarily, but return to a previous setup just by cycling power.

### The 'IF' for *program IF offset* option:

Entering 8-digits with only a single pushbutton and a Morse sounder might seem like a daunting task, but the software is designed to make it as easy as possible. And in general, you should only have to do it once if you are installing the board inside a radio. Consider setting the code speed to SLOW. After you enter the mode, the counter sends BT twice, then pauses three seconds before beginning.

To help with the process, detailed instructions are provided below:

- Eight digits are provided to allow from the tens of MHz digit through the ones digit.
- For offsets needing fewer than eight digits you should enter leading and trailing zeroes for the digits not needed. For example, 455.000 kHz would be 00455000.
- After you have selected the 'IF' power-on option, the counter will tell you the current digit to be selected, then "scroll through" in Morse, all numbers 0 through 9. You should press the button briefly immediately after the desired number is heard to select it.
- The annunciation of the digit is in the form of 'Dn', where 'n' starts with 8 and goes through 1. Digits are numbered 8 through 1, not 7 through 0.
- If you miss the desired number, no problem. The counter will continue through 9, then announce the same digit number as 'Dn' again, and begin scrolling through numbers again. So you can take a break in the middle of the process if you like. If you enter an incorrect number though, you have to power off and start over.
- I suggest getting ready by writing your IF frequency down on paper and then putting the digit numbers above what you've written. Use that as a guide during entry. For example, my current IF offset is 4,915,444 Hz. So I write:

```
8 7 6 5 4 3 2 1
0 4 9 1 5 4 4 4
```

Then I can see the number I need to select for each digit position.

Remind yourself that 0 is sent as a dash and 9 is sent as N if cut numbers is ON.

- When you finish digit 1, the software automatically saves the number you have entered in EEPROM and loads it every time it powers up.

The program treats the IF offset as being used in a low side LO radio. For example, if the IF is programmed as 4.9152 MHz and the counter reads the LO as 2.0848 MHz, the frequency announced will be the sum of the two or 7.000 MHz.

## **The 'C' for *Cut Numbers* option**

With *cut numbers* enabled, the digits 0 and 9 will be sent as T and N in the interest of speed. Releasing the button after 'C' is heard during the start-up options menu toggles the mode ON and OFF. The change is written to EEPROM and becomes permanent until start-up option 'C' is selected again.

## **Some hardware details**

The counter uses an 8-pin PIC12F683, a dollar-fifty part. The circuit requires 5 VDC and I would generally put a 5 V regulator on the board so I can use 8 to 15 volts input.

The input circuit copies the Stinger Singer closely, using a 74HC00 quad NAND gate. It provides several functions: Sensitive input signal conversion to TTL levels; a means of gating the input on and off; and a means of pulsing the input which is a requirement of the measurement scheme.

Other types of input signal processing might also be used but the gating and pulsing capability is always required.

The PIC chip's input prescaler is said to be capable of operating to 50 MHz although some say it can go considerably higher. My software allows reading to 100's of MHz if the hardware can handle that speed.

The PIC has an internal oscillator option that's fairly accurate, but for the accuracy required of a counter, a 4 MHz crystal with trimmer is used. The counter can essentially be as accurate as its time base.

A miniature speaker on the board outputs the Morse tones, which are generated by the PIC. I used a sounder from Digi-Key P/N CEM-1201(50) rated 50  $\Omega$ , 20 mA average at 1.5 V, so I put a 51  $\Omega$  resistor in series with the device.

I built a through-hole version on a Radio Shack 276-150 prototype board, one of my favorites that I found still available from B.G. Micro. I built another SMT version on a board I made using the toner transfer method. Both have the PIC ICSP programming header. If no reprogramming of the PIC is anticipated, it could be omitted.

## **Potential problems and possible solutions**

- Frequency announced is very inaccurate – far from the actual frequency
  - The counter may be in the 'IF' mode when you wanted a direct reading, or in the normal mode when you wanted an IF offset added to the frequency.
  - The counter may be in the 'S' mode which suppresses the MHz and 100 kHz digits and for faster annunciation. For example, 10,114,255 Hz will read out as 14r255. The solution is to change to the 'A' (all digits) mode.
  - There may not be enough signal amplitude at the input.
- A faint "puttering" sound is heard from the speaker.
  - This occurs when option T (audible marker sound for each 1 kHz change) is ON. The sound may be audible depending on the sensitivity of your speaker (and ears).
  - This is an artifact of the circuit design – the speaker shares a PIC pin with gating used to read the frequency. In this mode, the frequency is read about 50 times per second.
- A constant beeping is heard

- This indicates that option 'T' is ON and the counter isn't connected to a stable frequency source or to no source at all.
- Connect a frequency source or turn option 'T' OFF.
- The counter sends ERR during startup and operation is then erratic
  - This will only occur after the source code has been revised so it should never be seen unless you do edit the source code and create a new HEX file.
  - It means that the Morse character table crosses a page boundary. The source code must be edited to correct this and then re-assembled to produce a corrected HEX file.

### **Revisions & possible enhancements**

Currently, the offset for the IF feature is "hard coded" in the source code. A possible revision might make specifying that offset possible from the operating menu. DONE

Cut numbers are used to speed the Morse readout. This could be eliminated or made an option. DONE

Morse speed is about 23 WPM. It could be made slower, or an option of Fast and Slow Morse added. DONE

Add an audible pip with every 1 kHz change while tuning as an option. DONE

Add an automatic frequency update after the frequency has changed more than 1 kHz. This feature could be turned on or off. Hmm??

To get rid of the sputtering sound while in the "audible tick marks" mode, I could have a separate transistor oscillator for the time base. This would free up one pin and I could make the speaker independent of other logic. Another fix for that issue would be to go from an 8-pin PIC to one with a larger pin count.

Nick Kennedy, WA5BDU

January 25, 2017; September 18, 2017; October 5, 2017