AN_2901_061 FEBRUARY 2009

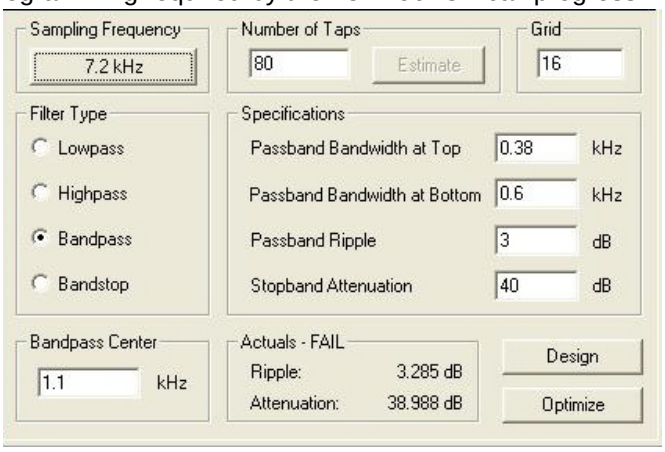
Programming the 73M2901CE Call Progress Monitor Filter

The Teridian 73M2901CE integrated circuit modem provides all the microprocessor control, modulation, and demodulation functions required to implement a V.22 *bis* 2400 bps modem. There are some ways to make the 73M2901CE do functions that they were not originally intended to do, for instance detecting and generating tones that are not normally used for normal modem functions.

Although all the 73M2901CE modem is preprogrammed for most normal dial-up low speed modem applications, in some cases it may be desirable to detect frequencies that are not part of its default set. The 73M2901CE modem can be programmed to detect and generate other frequencies beside the ones normally used. This application note describes the procedure for modifying the detect frequency band of the imprecise call progress detector of the 73M2901CE. Another app note covers the programming of the precise Call Progress Detectors (CPD), there are also detectors that can be selected that cover a wider range that can also be modified so they can be used for other detection purposes. This application note provides some insight only into how the imprecise detector can be programmed to detect frequencies other than those normally provided.

A number of registers are involved in the detection of call progress tone. These tones fall into one of two categories, pre-dial and post-dial call progress. Pre-dial tone are tones that would be detected before a number is dialed, specifically dial tone. Post dial tones would be tones such as ringback (the tone you hear when the called telephone is ringing) or busy tones. Both categories can be detected precisely by the specific frequencies they use or imprecisely by the band of frequencies they fall into. Pre and post dialing is controlled by register S20. The specific frequencies or bandwidth of the precise and imprecise detectors can also be selected. S19 selects the precise tones that are detected and S88 selects the imprecise bandwidth that is used. All of these detector results can be read back in read only S register 63 (Bit 7 in the imprecise detector monitor). S88 also has a setting called "USER DEFINED" and this is the setting that would be used if a user programmed bandwidth is required.

It would be nice if all that was needed was to select two corner frequencies and that was the end of it, but unfortunately there is a little more to it than that. First, the filter used for the imprecise detector is an 80-tap FIR type filter, and all the taps must be calculated and input into the appropriate call progress filter registers in the 73M2901CE. Fortunately there are third party programs available that can do these calculations for you. The one I have used is available for about $200 on line from www.iowegian.com called ScopeFIR. I recommend this one because it is very user friendly as well as relatively inexpensive. It is free to download and try out, but the feature-set is more limited and not quite enough to do the filter programming required by the 73M2901CE call progress filter. There may be others that are less expensive or even free, but the interface may not be as easy to work with. The results I got agreed closely with those obtained by the proprietary methods used by our software engineers. That does not mean I got the same coefficients, but the performance was in close agreement.



Using the program was simple and fast. I used the "Simple Parks-McClellan" filter template. The sample rate used in the 73M2901CE is 7.2 kHz, and this and the units (kHz,ms) must be input into the appropriate places in the pop-up. Enter 80 into the Number of Taps, and filter type (Bandpass), the desired

© 2007 Teridian Semiconductor Corporation

Passband at the top and bottom, ripple, stopband attenuation, and the center frequency in the appropriate places and press the “Design” button to calculate the tap data. When you do the “design”, you will see there are 80 numbers, as well as the filter shape, and impulse response plot now displayed. You will notice that there is a number of dots on the impulse response plot and these correspond to the taps you are generating (80 dots, 80 taps; the scale on the left indicates the same values as the taps that are created). If you change the input parameters you will see a corresponding change to the filter and impulse response.

Refer to the 73M2901CE user manuals for information on writing to the 73M2901CE memory. The data for the filter is most easily entered into the memory in 2’s compliment decimal notation but 2’s compliment decimal is not one of the data formats that is available for output, Excel can be used to do the conversion (it can also be entered in HEX directly, but I already have scripts that use decimal),. There is a math function available in Excel called Bin2Dec that can do this for you, so in ScopeFIR convert the output to binary (actually it is 2’s compliment binary); right click in the data field and select “Select Filter and data format”, “Text:Binary”, and for options, 8 bit data. All of the numbers for the taps are going to range from approximately +/- 20, so the numbers are going to be small (negative numbers have the MSB set to one, so they will look a little odd, e.g., 11111111 is actually -1). Select all the data in the coefficient editor pane and copy it to Excel in the first column, remembering it is 80 rows. Now use the Excel function to select bin2dec (this will be found different places in different versions of Excel. Some versions may require loading this function from the installation disk). The output will now be in decimal with negative numbers looking large again as with the binary (see figure 1). This is exactly what you want to load into the 72M2901CE. If you do happen to use Hex, make sure all the data has a ”$” preceding it, e.g., “$FE”. Having to add the “$” to each value is a little tedious, so I find the decimal input easier to work with, especially if you need to try a few different filters.

TEXT:bin, 8-bit	800-1400 Hz detector filter
BIN output from ScopeFIR	bin2dec
0	0
0	0
0	0
0	0
0	0
0	0
11111111	255
11111111	255
11111111	255
1	1
10	2
10	2
11111111	255
11111101	253
11111101	253

Figure 1: Portion of the ScopeFIR Output Copied to Excel and Converted to Decimal

Once you have the tap data in the proper format it is simple to copy and paste the taps into a text editor and format it into the command data. The following are sample commands to write 16 taps at a time into the 73M2901CE memory. Since there are 16 taps per line, it takes 5 memory writes to load all 80 taps. This example also shows where the data must be written, at memory addresses A2.$30 to A2.$70+15. When a number of successive locations are written, they can be strung together with a period in between as shown below. Also note that the first 40 values are mirrored in the last forty values.

```
AT@A2.$30=1.1.1.254.255.0.0.1.0.0.0.0.0.255.255.0
AT@A2.$40=2.2.1.254.253.255.1.1.0.0.1.2.0.252.250.253
AT@A2.$50=6.11.7.251.242.245.2.14.14.2.245.242.251.7.11.6
AT@A2.$60=253.250.252.0.2.1.0.0.1.1.255.253.254.1.2.2.0
AT@A2.$70=0.255.255.0.0.0.0.0.1.0.0.255.254.1.1.1
```

One more memory write may be necessary if the detector will need to operate for frequencies above 1 kHz. Normally there is a filter to prevent energy outside the normal call progress band from interfering with the detectors, and this must be disabled to detect frequencies above 1 kHz. The command to do this is:

```
AT@A2.$2C=$00.$7F.$00
```

Note that this command uses 8-bit Hex notation.

There a few more items that needs to be noted when programming the registers. The registers should be written to after the 73M2901CE has been taken off hook. The S88 register can be written to at any time before going off hook and remember there is a pre-dial and post dial mode that can be selected. The user defined filter is selected by setting the value for the S88 register to $F for the mode required. I select them both for convenience, so S88=$FF.

You may find that the filter's range is not exactly what you specified using the filter program. I have found that they tend to be slightly narrower than what was specified, so test the detector and make adjustments to get exactly what you need. I have tried various settings, but set the stopband to 40 or 50 dB and the ripple from 1 to 3 dB for starters. Also remember the stopband frequency range must be wider than the passband range.

To actually use the detector, it is necessary to poll the S63 register to see if the detector bit is on or not. If it is detecting energy in the band, it will read 128 (sometimes a precise detector could also come on well if its tone is present, but the MSB is the one we are interested in). The fastest way to poll the register is to give it the "ATS63?" command the first time, and then "A/" to repeat the last command. Clearing S73 bit 5 will speed up execution (disable delay between commands) as will sending commands at 9600 bps. If the results seem to not be like you expected, make sure you are doing the register programming in the exact order outlined above. I occasionally forgot a step and got results that were not what I wanted.

I have not tried other programs to generate the taps, so I cannot say specifically how to use them. As long as you can get a result that can be used to get a text output in either Hex or 2's compliment Binary or decimal, you should be able to use it to generate the necessary tap data.

Revision History

Revision	Date	Description
Rev. 1.0	2/4/09	First publication.

Maxim cannot assume responsibility for use of any circuitry other than circuitry entirely embodied in a Maxim product. No circuit patent licenses are implied. Maxim reserves the right to change the circuitry and specifications without notice at any time.

Maxim Integrated Products, 120 San Gabriel Drive, Sunnyvale, CA 94086 408-737-7600

© 2011 Maxim Integrated Products Maxim is a registered trademark of Maxim Integrated Products.