CREATING A NEW SDRAM "Protocols.txt" File

Presently it is imperative that there be a micro SDRam card installed in the generator's card reader. The card must have a file name "protocols.txt", in the proper format recorded on the SDRam card. Presently it is known that a micro SDRam card can be any size up to 4 gigs. However the smaller the card the better. Presently the smallest SDRAM I have found is 128 megs and it works great.

The reason a smaller card is better is because the "protocol.txt" file is extremely simple and can contain over 10,000 lines. Each line defines each of the 4 channel generators runtime parameters. It would be too cumbersome to have the code sort through too many parameter settings during a program execution, in order to run a string of lines for a particular desired protocol. A 1 gig card could easily hold over a half million parameter lines. So several smaller cards would be easier to manage and typically less expensive than the larger SDRam cards.

When the card is read or updated by the program a small file like a few hundred parameter lines is significantly quicker to read or update verse 10,000 or more parameter lines. The card can however have multiple files stored on it so you could have several protocol files with different names stored on the card. You could also have additional supporting documentation stored on the card. The important thing is to have at least 1 file called "protocols.txt".

It is advisable to print out a copy of the "protocols.txt" file for the following description of the parameters lines.

The very first line in the file simply needs to see the word "protocols", not case sensitive, followed by a space, then a number sign, and finally a "newline" character. example: "Protocols #"

All the next lines must have a line number followed by four parameters groups. Note: line 0 is different and explained further down in this document. A parameter group is simply a frequency and duty cycle for each channel with 4 parameter groups on each parameter line. A single parameter line defines the parameters for all four channels of the generator. The line is terminated with a "newline" character.
example: `1,10.00,0.20,40.00,0.50,40.00,0.50,7.83,0.50`

- Where the first number is the line number. In this example it is line number 1.
- Where the second number is a floating point frequency value for channel 1. Ranging from 0.01 to 50000.00 hertz.
- Where the third number is a floating point duty cycle number for channel 1. Ranging from 0.00 to 1.00 percent. Values 0.00 and 1.00 effectively turn a channel on or off. Great or complex protocols or turning relays or external devices on or off.
- Where the next parameter groups are for channels 2, 3, and 4 followed by a "newline".

NOTE: all parameters are separated by a comma and all lines are terminated with a "newline" character which is 0x0A in hex format. Use a word processing editor that will generate the hidden "newline" character when a return is hit. Something like Microsoft Word, or WordPad, or anyone of the Linux or Mac word processors work nicely. Microsoft's text editor does not work for this author.

There can be as many parameter lines as you would like to have however they all must be consecutively number. The range of parameters lines starts with line 1 and continues for as many lines as you would like to have. The last line of the file is called line 0 and serves as a terminating line for the all the parameter lines. It also contains the generator's general runtime parameters described below.

THE LAST LINE OF THE FILE is called line 0 - Operational Parameters

Line 0 presently has 9 parameters.

- Where parameter 1 is a number defining a parameter line in the file to open the generator with. These parameters will be displayed when first turning the generator on or hitting the RESET rotary press button function.
- Where parameter 2 is the default "Mode" to start the generator with. There are 3 modes which are "Simple" Mode, "Protocol" Range Mode, and "Sweep" Mode. (see "MODE" document for details on operation and usage).
- Where parameter 3 is the frequency integer run time value in seconds. Ranging from 0 to 9999 seconds. A value of 0 gets translated in the program to .5 seconds.
- Where parameter 4 is the program integer run time value in minutes. Ranging from 0 to 9999 minutes. A value of 0 gets translated in the program to .5 minutes or 30 seconds.
- Where parameter 5 is the starting parameter line to use if a "Protocol" range Mode of 2 is selected as is the case in this example below.
- Where parameter 6 is the stopping parameter line for the "Protocol" range Mode of operation. (see example)
- Where parameter 7 is the starting parameter line to use if a "Sweep" Mode of 3 is selected.
- Where parameter 8 is the stopping parameter line for the "Sweep" Mode of operation. (see example)
- Where parameter 9 is a frequency increment value when in the "Sweep" Mode of operation. A value of 0 means; the generator program will automatically calculate the sweep increment value base on the frequency and program run times from parameters 3 and 4.

Example: `0,1,2,5000,60000,1,5,3,4, 1.00`

- Where 0 is used to designate last line in the file.
- Where 1 is used to designate the program will open and display parameters that come from parameter line 1. This could be any valid parameter line number.
- Where 2 is used to designate the generator is to start in Mode 2, "Protocol" range mode.
- Where 5000 designates each frequency will run for 5 seconds. The value here is to be entered as a millisecond value. 5 seconds = 5000 milliseconds.

- Where 60000 designates program run time is 1 minutes. The value here is to be entered as a millisecond value. 1 minutes = 60000 milliseconds. A value of 600000 would = 10 minutes. A value of 6000000 would equal 100 minutes, etc... NOTE: this is slated to be changed so the entry reads like display screen 5.
- Where 1 is the start parameters for a "Protocol" range operation.
- Where 5 is the stopping parameters for a "Protocol" range operation. All parameter between parameter line 1 and parameter line 5 will be executed in accordance with the run time and program time timers. Hence this makes up a frequency protocol of 5 frequencies per channel run consecutively one at a time.
- Where 3 is the start parameters for a "Sweep" operation. A "Sweep" operation can be defined by any two parameter lines. They do not have to be consecutive numbered parameter lines.
- Where 4 is the stopping parameters for a "Sweep" operation. The "Sweep" mode only works with channel 1. The other channels will be set to the parameters from starting parameter line. They will stay at those frequencies and Duty Cycle levels for the duration of the "Sweep" function. The "Sweep" function will continue for the duration of the time set by the program timer. If a "Sweep" is finished before the program times out, the "Sweep" will be restarted.
- Where the sweep operation is to increment the frequency by 1 hertz every 5 seconds in accordance with the 5000 frequency run time parameter.

By far the "Sweep" is the most feature rich function of the generator. Playing with the values and parameter settings is the best way to learn this function's capabilities.

Regarding adding and deleting parameter lines

It is suggested you simply have extra parameter lines with parameter values of zero stored with every "protocols.txt" file used with the generator. You never need to delete a parameter line, you can just change it or null it with zero to bypass it. With extra Null with zero parameter lines in the file, you can always create a new protocol via the generator and bypass the need for a computer. For extensive parameter line changes, you are going to want take the SDRam card out and use the provided SDRam card USB plug to modify the file with a nice word processor program. The rule is to always use the correct tools for job. Sometimes the computer word processing program will be the right tool.