Guide to the N3ZI DDS2 Computer Control interface. Aug, 2012

The DDS2 board includes a simple low cost serial interface with RS232 levels, no interface electronics is necessary for control from most computers serial port. Control via a USB port is also possible by purchasing a simple USB to RS232 converter cable. These are available on Ebay for a few dollars.

Make a cable from your PC's serial port There is a 3 pin connector near the upper right hand corner of the PCB. Pin1 is ground, Pin2 is DDS Output data, Pin3 is DDS input data.
Pin 1 (Ground) goes to Computer DB9 connector pin 5 Ground
Pin 2 (DDS TxData) goes to Computer DB9 connector pin 2 Computer Rx Data
Pin 3 (DDS RxData) goes to Computer DB9 connector pin 3 Computer Tx Data

You should set your serial port to: 4800 baud, 8bits, no parity, 2 stop bits, no flow control. Once you are in you can change the baud rate. But if you set the wrong number you won't be get it to change, but the EEPROM reset function will set it back to 4800 baud.

The common **Windows** HyperTerminal program works fine. Under **linix** Minicom works. Many others will work too. If you are using a USB to RS-232 converter, then you have to figure out which serial port it is, on Windows it's usually just the next com port. On linix the device is /dev/ttyUSB0, for a regular serial port it's /dev/ttyS0 I have been told that W7AY's Serial Tools will work with a **MAC.**

On power up you will see a greeting similar to: N3ZI SUPER DDS V1.x
If you send anything, you will get this response: 'ON' to activate
Type ON to activate serial communications.

Command	**Action**
ON	Turns on Comm's
<ENT>	Disp Frequency and Mem #
Q	QSY, change operating frequency
I	Change IF
M	Incr DDS Mem#
F	Turn off USART
+	Incr Freq (By Default step)
-	decr Freq (By Default step)
Advanced Commands	
R	Readout EEPROM contents.
W	Edit EEPROM <space> increment to next location <ent> exit "=" enter new value (Zero must be entered with decimal point)
D	Debug mode (provides more verbose display)
L	Load parameters from EEPROM
S	Save all to EEPROM

Note that the Serial control and normal front panel controls operate at the same time.
So if the tuning dial is rotated, or memory # is changed, the change will be sent out the RS-232 port. But some RS-232 command will delay acceptance of front panel commands. For example it takes a few seconds to dump out the EEPROM contents, if the dial is turned while this is happening it will be ignored. Similarly while you are in the EEPROM editor, front panel controls are locked out. If you primarily use the front panel controls, you should turn off the serial comm's using the F command, when they are not in use. With the comm's on tuning will lag a bit, more so in debug mode.

Advanced commands: In general changes to EEPROM will not actually be used until you reboot, or use the "L" command. Double check your changes by using the "R" command before you reboot or use L. Note: use of this is for experienced users only. Parameters are not range checked, and bad values can cause malfunction.

Loc	Default	Description	Normal Range	Units	Description
0	305,419.896	Indicates Valid EEPROM	Do not change	None	Do not change
1	0.259	Baud Rate	0.699 to 0.014 (1200 to 57600Baud)		=(1250/BAUD)-0.001 (When using 20MHz Xtal)
2	900,719.925	timebase factor	900,000 to 901,000		=2^56/1000/Mosc(in Hz) Used for calibration
3	0.000	Saved Memory Number	0.00 to 0.11	Integer (ignore decimal point)	Saved Memory Number
4	0.025	Default Step	0.001 to 1.000	1Hz	Also can be set by tuning knob
5	0.001	Encoder Trigger	0.001 or 0.003		0.001 causes both edges of encoder pulse to be used, 0.003 only one edge is used. 0.001 is faster but less precise tuning
6	0.100	Main loop delay	0.050 to 5.000	100mSec	How often frequency is actually updated. Tied into #5 and #15
7	1.025	Tuning Acceleration	0.999 to 2.100	Binary	The fractional part is the number of encoder ticks per acceleration step, the integer part is the rate of accel. Use 0.999 to turn off acceleration.
8	0.010	Rotary encoder Debounce	0.000 to 0.5000	100mSec	Rotary encoder Debounce Higher values cause slower but more predictable control. Use 0.000 with an Optical Encoder
9	0.002	Tick Divider	-0.010 to 0.001 to 0.010	+/- Integer (ignore decimal point)	Number of encoder ticks it takes to cause a freq increment. Can be negative which reverses direction.
10	2.048	Phase	0.000 to 4.095	87.89Degrees	The phase offset selected by the PH test point, 0 to 359.9 degrees (Not used V1.0)
11	100.000	Acceleration Cap		1KHz	Selects the speed limit (per encoder tick) during extended acceleration
12	0.110	SLCD Delay	0.101 to 9.999	100uSec	Delay time for SLCD xxyy xx is final delay, yy is initial delay, in tenths of ms. Use long delays if a long cable or system noise causes bad readings. Use short delays for direct connect. Disregard decimal point.
13	13,500.000	Band Frequency		KHz	Frequency when bandswitch output changes
14	0.500	RIT	+/-9999.000	KHz	RIT frequency
15	0.050	Encoder Idle	0.020 to 0.100	Integer (ignore decimal point)	How many counts that encoder is idle before acceleration is reset.
16	0.001	LCD Update	0.001 to 0.010	Khz	Degree of frequency change that causes the LCD to update.
17-31		Not Used			
32-44		Saved DDS Frequencies for each memory			You can changes values here but it's much easier to use the "QSY command
45-63		Not used			
64-76		Saved IF Frequencies for each memory			You can changes values here but it's much easier to use the "I" command
77-127		Not Used			