SDR2GO Builder's Notes May 2012

Introduction

This document has been prepared to aid the builder of the SDR2GO board. The SDR2GO board and its documentation have been prepared for the experienced builder of ham radio equipment with surface mount component installation experience. If you have successfully built a SoftRock then you are well on the way to building an SDR2GO board!

Applicability

So far several batches of kits have been prepared and sold. The original batch of 30 kits have boards marked as version V1.0. The subsequent batches kits have boards marked as V2.1. These notes address both versions of the boards.

The following changes have been made to the design of the V2.1 boards.

- Two pull up resistors (R43 & R44) have been added.
- Capacitors C25 thru C29 have been deleted.
- A missing wire for U8 has been added.
- Board mounting hole registration has been improved.
- A number of small layout tweaks have been made.

The latest version of the DSP software is SDR2GO V1.9.0.

The first batch of kits were delivered with SDR2GO V1.1 software installed.

The second batch of kits were delivered with SDR2GO V1.6 software installed. SDR2GO V1.6 provided the additional functionality of of CW operation for both transmit and receive. And, on receive SDR2GO V1.6 provided for switching between a 2100 Hz filter and a 700 Hz filter. The physical interface for these new features (CW Key and two switches) are wired via existing header J11. If you do not wish to utilize the CW mode or the variable receive filters, simply do not connect CW Key or the two switches to J11.

The third batch of kits were delivered with SDR2GO V1.7 software installed during the summer of 2011. SDR2GO V1.7 included an improved AGC algorithm. This version of the software had annoying DSP encoder operation and problems with how the user input switches behaved. These problems were corrected in SDR2GO V1.8 software.

SDR2GO V1.8 was found to have a problem with inconsistent setting of the transmit audio level. This problem was fixed in SDR2GO V1.8.1.

SDR2GO V1.9 provides an SPI interface for driving a 128 X 64 Graphic LCD. Kits shipped after April 2012 are program with SDR2GO V1.9.0.

The SDR2GO V1.9.0 has two new features. One is the addition of the SPI interface for Graphic Display of the received signal spectrum from - 6 kHz to + 6kHz of the SoftRock VFO frequency set by the SI570. To use this feature, a separate Graphics Interface Board and Graphic Display must be installed. Please see the document entitled "SDR2GO Graphics Display Add On Builder's Notes May 2012" for details of implementing this board.

The other feature is the addition of both high pass and notch filters to the TI3204 codec. These filters provide significant rejection of low frequency components which are contained in the SoftRock receive I/Q output. The high pass filter is designed to pass signals greater than 200 Hz. The notch filters are designed to reject signals at 50 , 60 , 100, 120 and 180 Hz.

NOTE: SDR2GO V1.9.0 may be used with OR without a Graphics Interface Board and Graphics Display installed!

dsPIC33 / SH32 Data Interface
SDR2GO provides for data interface between the dsPIC33 and ths SH32 processors.
This allows for the display of the following items on the LCD Display.

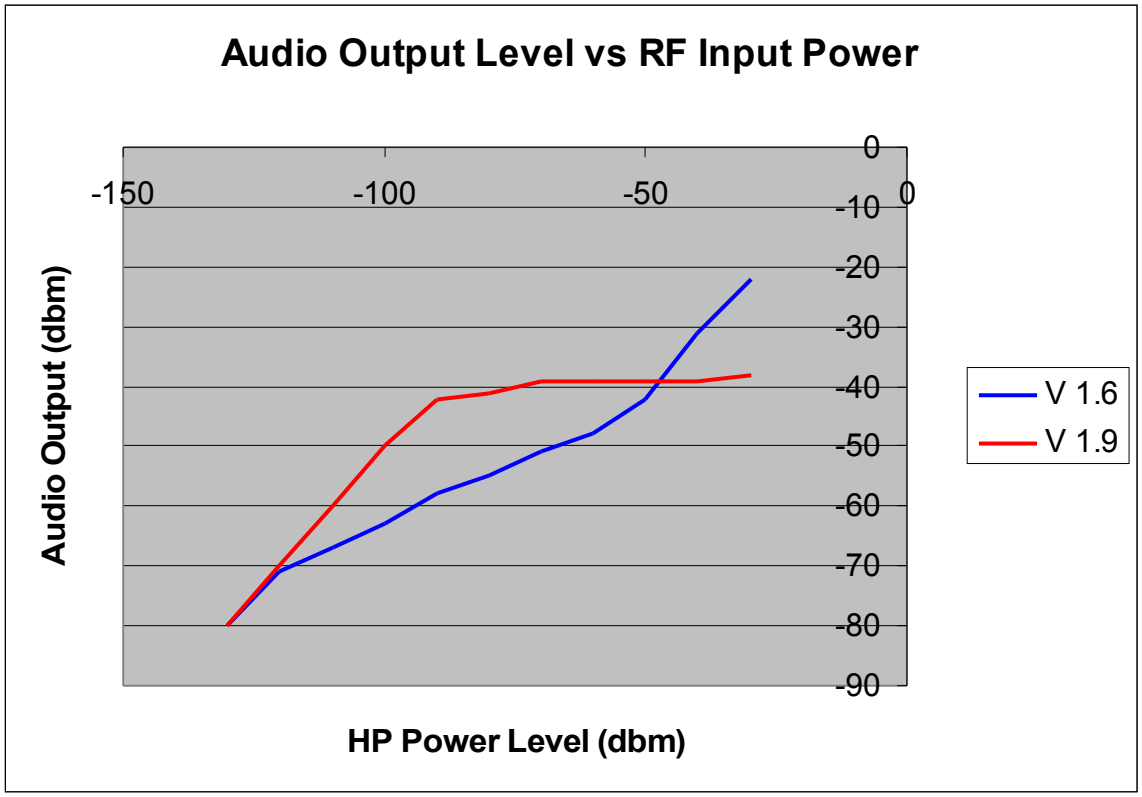
- Transmit Audio Input State
- Receive / Transmit State
- Selected Sideband
- Encoder #2 (dsPIC33 Encoder) Parameter

Setting Initial DSP Parameters
The DSP algorithm reads the stored DSP Parameters on boot up of the dsPIC. The EEPROM dedicated to the dsPIC is shipped blank. So, in order to provide the DSP algorithms with useable DSP parameters, you must first initialize the stored parameters. This is done by setting bit 0 and bit 1 high and then depressing the Encoder #2 PB. This action sets the DSP parameters listed below to default values:

Transmit IQ Amplitude
Transmit IQ Phase
Receive IQ Amplitude
Receive IQ Phase
Transmit Microphone Gain
Transmit Line Input Gain
Transmit CW Gain
Receive Audio Level

Improved AGC Scheme

After a valiant effort by Milt Cram, W8NUE, the capabilities of the TI3204 Codec have been fully exploited to significantly improve the operation of the SDR2GO AGC scheme. This AGC scheme is delivered in V1.7, V1.8, and V1.8.1. Further, the AGC scheme has been revised to work with the V1.9 code. Please see below a comparison between the V1.9 and V1.6 AGC performance.



Kit Details

1) This is basically a SDR design block and experimentation platform. As is, it will perform the functions of a SSB/CW transceiver when used with an I/Q front end such as a SoftRock RxTx Ensemble, SoftRock RxTxV6.3, or UHF SDR. It provides the capability for reprogramming and future functional expansion.

2) Many of the component footprints are altered (made longer and narrower) to enhance hand soldering. Component placement and spacing is also designed for hand soldering.

3) The CODEC chip used, a TI TLV320AIC3204IRHB, is relatively new and only available in a 32 pin QFN package.

4) There are two parts which will be provided programmed on production kits: the dsPIC33FJ128GP804 and the MC9S08SH32CWL. Programmed chips are identified with a white dot which represents the Kees Talen Certification of Programming.

5) In future there may be upgrades of the code for the SH32 or dsPIC which you may program yourself. For programming the dsPIC33 you will require a Microchip PICkit II. For programming the MC9S08SH32 you will require a BDM programmer such as the USB Multilink.

6) One of the major user advantages of the SDR2GO is the flexibility to use a Rotary Encoder and/or PS/2 Keyboard for frequency control. We haven't even figured out all the possibilities :o) .The keyboard provides many unique advantages for data entry and will be covered in another section.

The schematic shows J17 as being an interface to the NUE PSK modem. This port and the Keyboard CLK SW input allow a single user keyboard to be shared between the SDR2GO and a NUE PSK modem. No data or DSP signals are shared between the SDR2GO and a NUE PSK modem.

7) This kit is not for the inexperienced. It has one 44 LQFP, a SOT23-6, several SOICs, and a 32pin QFN. The QFN can be hand soldered with flux and "wipe technique". All Kees uses is a set of head magnifiers and a temperature controlled soldering iron (cost less than $50 on ebay). The most important step is placement and tacking the part in place first. If you solder one side, then the next, then the next, etc ….you will find that the component registration has moved due to the thermal expansion/contraction forces.

8) The 16x2 LCD interface is designed for many types of 16x2 LCDs.

- those requiring positive or negative contrast bias, various bias levels
- those with Vcc/ and Gnd as pin 1 or 2
- those with Vcc/ and Gnd as pins 2 and 1

The reason for doing this is that we supply one type of yellow-green backlit LCD with black alphanumeric characters, but you may have your favorite white on blue, black on white, etc. For the positive and negative bias levels, a charge pump made up of D2, D3, C12, C13 is provided. For contrast adjustment, R17 is provided and will supply approximately -2V to +2V.

Note that the LCD interface uses a 10pin connector and is multiplexed using inputs D4, D5, D6, and D7 (inputs D0, D1, D2, and D3 are NOT wired).

9) Si570 is the chip of choice for frequency control. You have two options a) wire the I2C interface from the SDR2GO to the I/Q RxTx board which has the Si570 installed or b) locate the Si570 chip on the SDR2GO board and wire the RF signal over to the I/Q RxTx board with a short piece of coax. We recommend option b.

10) The two LEDs provide quick confirmation of 5VDC and 3.3VD

11) J7 is for future use and need not be populated and the headers are not provided in the kit.

12) J11 is for CW operation interface.

13) J17 is only for NUE-PSK attachment and need not be populated although headers are provided in the kit.

14) For lower power consumption a DC to DC converter (80-85% efficient) may be used to go from 12VDC to 5VDC. These are available for <$10 and have much less power consumption than a linear 5V regulator. No DC to DC converters are provided with the kit.

What to do if you do not have strong SMD skills?
All is not lost. Surface mount installation and soldering is not as difficult as it may seem. The SDR2GO team members are all over 60 years of age. Each of us have built a number of SMD projects and have learned that SMD can be done successfully using soldering irons and hand soldering techniques. What is required is that you gather a good magnifying glass with light and a vise made for holding circuit boards along with a few basic hand tools and soldering supplies to include small diameter solder, solder flux formulated for SMD and solder wick for removing unwanted solder. Also, a solvent such as 91% isopropyl alcohol is required to remove the solder flux. The flux available today is not corrosive and is much, much easier to remove than the old rosin based solder fluxes.

There are a number of websites which have instructions, guidance and videos on SMD installation. Here are a few:

http://store.curiousinventor.com/

http://www.youtube.com/watch?v=3NN7UGWYmBY

http://www.youtube.com/watch?v=wQXhny3R7lk

http://www.metacafe.com/watch/2536889/tangent_tutorial_3_surface_mount_soldering_techniques/

Reference Documents
At the back of this document you will find an overall block diagram and two sheets of schematics which define the SDR2GO board.

The SDR2GO board has been designed to provide a great deal of flexibility in how the board is interfaced with the SoftRock of you choice or other SDR RF section such as the UHF SDR board. For example, an SI570 may be installed on the SDR2GO board or the SoftRock board. And, you may use a CMOS or LVDS SI570 with or with out direct output, transformer output or broadband buffer output.

Other interface options are provided as well. To best make use of these interfaces we recommend that you spend some considerable time reviewing the schematics in detail and reading and then re-reading this document before you start construction.

Parts List

You will find below a detailed parts list that you may use to check off each part as it is installed. With SMD components we strongly recommend that you install all components of the same type and value at one go before you start on another type or value. This discipline has helped the author to build five out of five SoftRock transceivers which worked on the first power up.

SDR2GO Parts List as of April 2011

Capacitors			Resisitors			Miscellaneous		
C5	0.1uF		**R4 ***	**0**		D1	Not Supplied	
C6	0.1uF		**R5 ***	**0**		D2	1N4148	
C11	0.1uF		**R7 ***	**0**		D3	1N4148	
C17	0.1uF		**R8 ***	**0**		D4	LED	
C18	0.1uF		R36	100		D5	LED	
C30	0.1uF		R17	10K		L1	47uH	
C34	0.1uF		R16	2.7K		Q1	2N7000	
C36	0.1uF		R40	2.7K		Q2	2N7000	
C38	0.1uF		R41	2K		Q3	2N7000	
C40	0.1uF		R42	2K		**T1 ***	**Not Supplied**	
C41	0.1uF		R11	3.3K		J1	IQ - IN from SoftRock	
C43	0.1uF		R14	3.3K		J2	IQ - OUT to SoftRock	
C48	0.1uF		R10	4.7K		J3	Mike / Line Input + PTT Input	
C49	0.1uF		R13	4.7K		J4	SI570 RF Output	
C50	0.1uF		R15	4.7K		J5	I2C IO for SoftRock	
C7 *	0.1uF		R18	4.7K		J6	Encoders IO	
C8 *	0.1uF		R19	4.7K		J7	dsPIC GPIO	
C44	47uF		R21	4.7K		J8	Keyboard Input	
C45	47uF		R30	4.7K		J9	LCD IO	
C9	10uF		R31	4.7K		J10	BDM Programmer	
C10	**10uF**		R32	4.7K		J11	dsPIC GPIO Access	
C12	**10uf**		R33	4.7K		J12	dsPIC SPI	
C13	10uf		R34	4.7K		J13	dsPIC Switch IO	
C14	10uF		R20	8.2K		J14	Headphone Outputs	
C15	10uF		R35	8.2K		J15	dsPIC ICSP Programmer	
C16	10uF		R37	8.2K		J16	BPF IO	
C31	10uF		R38	8.2K		J17	NUE Keyboard Input	
C32	10uF		R39	8.6K		X1	12MHz Crystal	
C35	10uF		R43+	4.7K		U1	TLV320AIC3204	
C37	10uF		R44+	4.7K		U2	dsPIC33JF128GP804I/PT	
C39	10uF					U3	MC9S08SH32CWL	
C42	10uF					U5	24AA128	
C46	20pF					U6	24(AA/LC)32AT	
C47	20pF					U7	Si570	
						U8	SN74LVCIG3157DBVR	
						Reset	Reset Pushbutton	
						3 V Reg	MCP1703-3302E	

Notes:

1) Items marked with " * " are optional depending on SI570 Configuration.
Read Builder's Notes and research SI570 output options before proceeding.

2) Items marked with " + " are on V2.1 board.

Build Sequence
We recommend that the parts be installed in the order listed below to provide maximum protection to the semiconductors while providing maximum physical access for hand soldering of SMD devices.

- Install all SMD Capacitors
- Install all SMD Resistors
- Install Semiconductors U1 thru U7
- Install Semiconductors Q1 thru Q3
- Install electrolytic capacitors C44 and C45
- Install Diodes D1 thru D5
- Install L1
- Install T1 if transformer output option is selected
- Install header pins J1 thru J17 as required to support the design options you choose
- Install Reset Push button

Design Option Discussions
For basic operation the following items are deemed essential and will not be discussed in detail in this section.

- IQ Input
- IQ Output
- Microphone / Line Input
- Power Connections
- Frequency Tune Encoder
- DSP Parameter Encoder
- dsPIC Switches
- LCD Connections
- PTT In Signal Input
- PTT Out Signal Output

The following items may be configured in several ways or may be ignored. Each item will be discussed in more detail below.

- Optional CW Key, Mode Switch and Receive Filter Switch
- SI570 Installation and Interface
- dsPIC GPIO
- MC9S08SH32CWL Programming Interface
- dsPIC33JF128GP804I/PT Programming Interface
- NUE PSK Modem Keyboard Interface
- SPI Interface
- Board Reset

CW Operation Interface
The SDR2GO software provides for CW operation.

The software now incorporates a switch selectable receive band pass filter which is approximately 600 Hz wide (300 - 900 Hz). The selection of filter width is controlled by GPIO bit 3, pin 4 of J11. When this input is high, the receive bandwidth is approximately 2.1 kHZ. When this input is low (connected to ground, pin 1 of J11, the receive bandwidth is 600 Hz.

SSB / CW operation is controlled by GPIO bit 4, pin 5 of J11. When this input is high, SSB operation may be used wherein the audio applied to J3 is fed into the software on transmit. When this input is low (connected to pin 1 of J11) a CW transmit a tone controlled by the CW Key input is fed into the transmit I/Q chain which has a frequency of 600 Hz. The PTT_IN signal, pin 3 of J3 places the software into transmit for both SSB and CW operation.

The CW tone is switched on an off by connecting a CW Key to GPIO bit 5, pin 6 of J11. When this input is high no tone is fed into the transmit I/Q chain. When this input is low (connected to pin 1 of J11) a tone is fed into the transmit I/Q chain. Filtering is provided on both the beginning and end of the CW tone to eliminate key clicks. The level of the CW I/Q audio output signals to the SoftRock, J2, may be independently adjusted by the user. This means that for transmit three separate audio level adjustments may be made by the user: microphone input audio level and line input audio level for SSB operation plus CW output audio level for CW operation.

Further, the CW tone is fed out the headphone output on J14 to provide the user with a keying side tone when the SDR2GO is in the CW mode. The audio level of the CW side tone is not user adjustable.

SI570 Installation and Interface
The SI570 may be installed on the SDR2GO board or the SoftRock board.

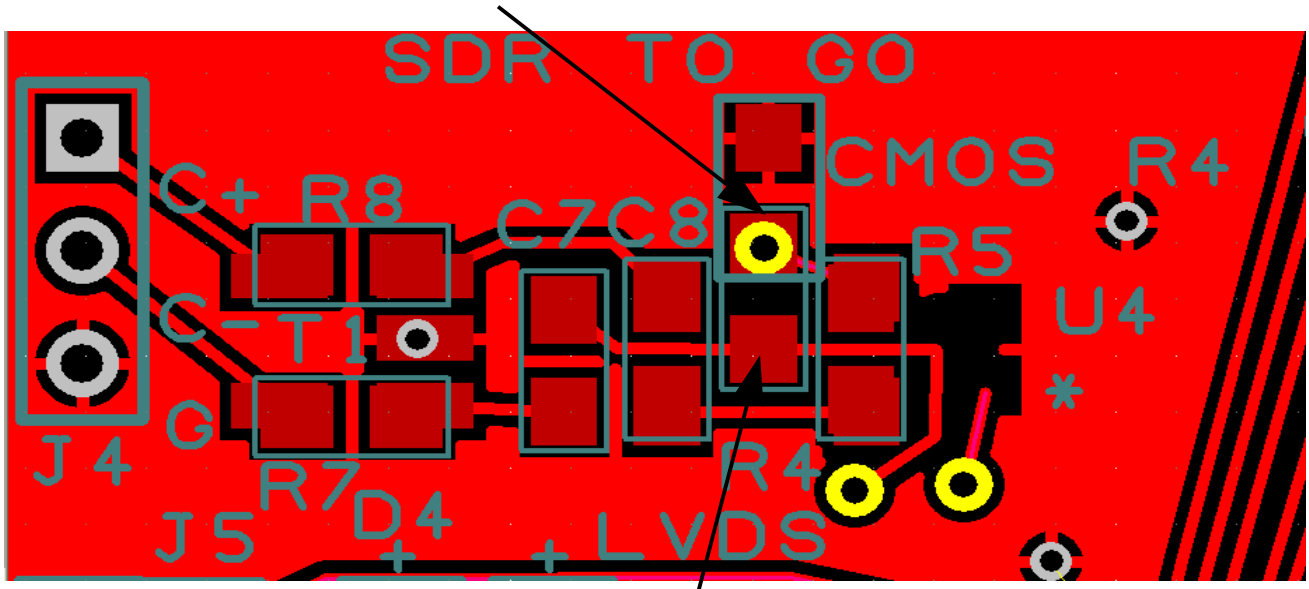
SI570 Installed on SoftRock
In the case of a SI570 installed on a SoftRock board, the items listed below are not installed on the SDR2GO board. And, header J5 is used to interface the 3.3V I2C signals to the SoftRock. It is recommended in this case pull up resistors not be installed on the SoftRock board since pull ups are provided on the SDR2GO board (See R11 & R14).

- U4 & U7
- R4,R5,R7,R8
- C5, C6, C7,C8
- T1 & J4 Header

SI570 Installed on SDR2GO
In the case of a SI570 installed on the SDR2GO board, U7 plus C5 and C6 should be installed regardless of SI570 output interface design. Three main SI570 output options are supported on the SDR2GO board: direct coupling, coupling using FIN1002 buffer, or transformer coupling.

Please note that R4 may be installed in one of two locations.

R4 installed for CMOS SI570 with one end grounded.



R4 installed for use with FIN1002 On Transformer Interface

On board SI570 Using CMOS Direct Interface
For an interface using a CMOS SI570 the follow items should be populated with the values listed below:

- R4 50 or 100 ohm to ground (Upper Position)
- R5 0 ohm
- C8 0.1 uf
- R8 0 ohm

On-board SI570 Using FIN1002
For an interface using a LVDS SI570 with the FIN1002 the following items should be populated with the values listed below:

- R4 100 ohm (Lower Position)
- C8 0.1 uf
- R8 0 ohm

On-board SI570 Using Transformer Interface
For an SI570 interfaced with a trifilar transformer the following items should be populated with the values listed below:

- R5 0 ohm
- C7 0.1 uf
- C8 0.1 uf
- T1 Trifilar Transformer (Not Supplied with kit)

dsPIC GPIO

Five of the dsPIC inputs are connected to the SH32 for future interface between the two processors (see J7 development details). The dsPIC inputs that are connected to J7 are 5 volt tolerant. Please note that not all of the dsPIC IO is 5 volt tolerant. Please see sheet 2 of the schematics for 5 volt tolerant pin identification.

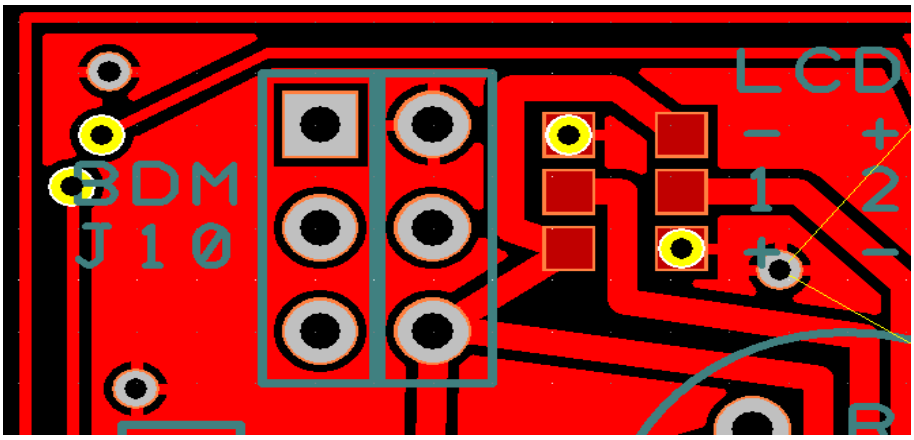
SPI Interface
The four signals of the dsPIC SPI interface plus ground may be accessed via J12. This interface is used to drive the Graphics Interface Board described in the document entitled "SDR2GO Graphics Display Add On Builder's Notes May 2012".

Board Reset
If the user wishes, the push button may be installed as labeled on the board. This input only resets the dsPIC. This reset is normally not required since the dsPIC starts up nicely when power is applied.

MC9S08SH32CWL Programming Interface
The SH32 will be provided pre-programmed in the production run of kits.

Please note that programming of the SH32 should not be attempted until the Initial Board Start Up procedure outlined below has been completed.

If the user wishes to upgrade SH32 firmware in future the programming interface may be accessed by installing the header pins for J10 which is designed to accommodate the mating plug from a BDM programmer. Please see the figure below which shows the J10 location. The mating BDM programmer cable mates with J10. The mating connector is connected to a USB Multilink programmer via a six conductor ribbon cable. The connecting ribbon cable should extend to the left of J10 with the red edge (wire 1) at the top of the cable.



For details on the USB Multilink programmer please see this link:
http://www.pemicro.com/products/product_viewDetails.cfm?product_id=33

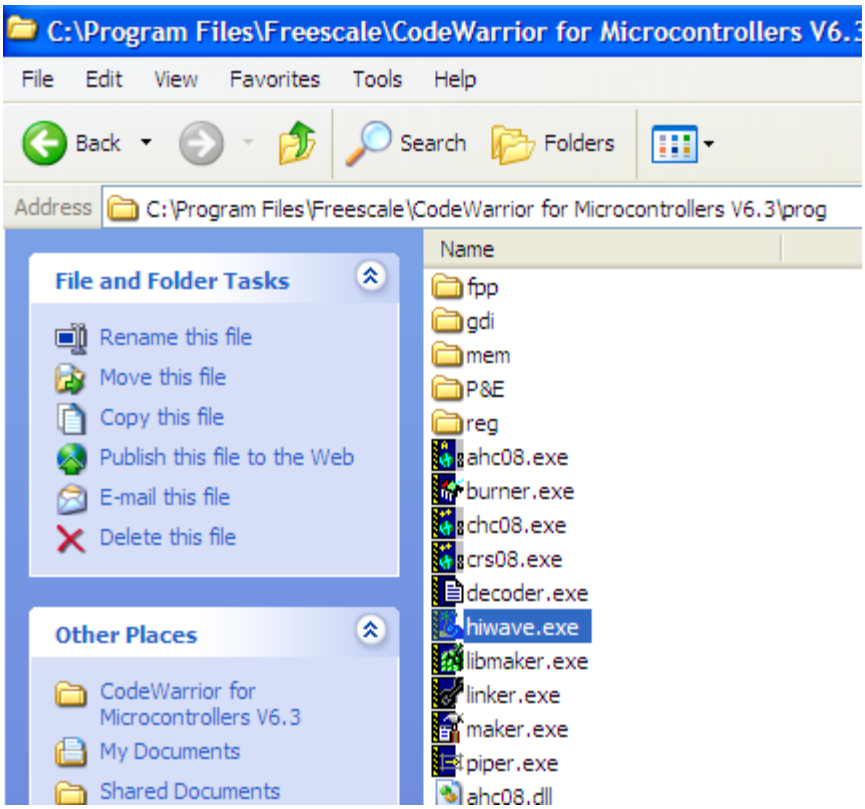
In order to use the Multilink programmer FreeScale's CodeWarrior must be downloaded and installed on your machine. FreeScale CodeWarriorSpecial Edition Download

Next, download the SDR2GO program files from the AQRP site. The archive should be extracted so that the folder is C:\SDR2GO_Programming folder.

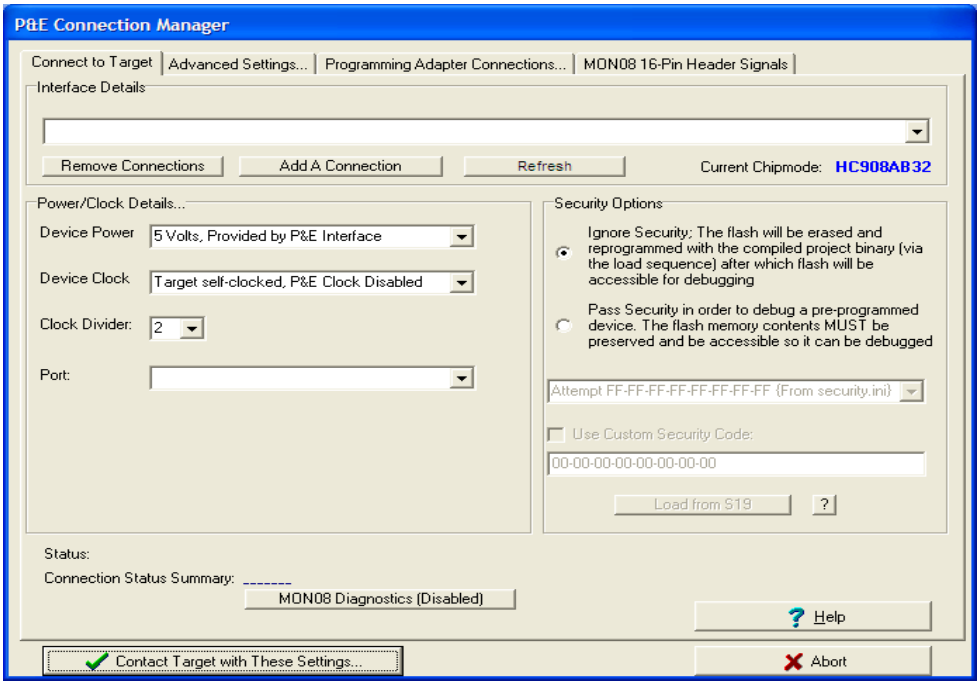
At this point, attach the Multilink programmer and power up the SDR2GO board. Both the blue and yellow LEDs on the Multilink Programmer should be lit.

The CodeWarrior application is not required for programming the SH32. Instead an application called True-Time Simulator & Real- Time Debugger which is part of the CodeWarrior IDE is used.

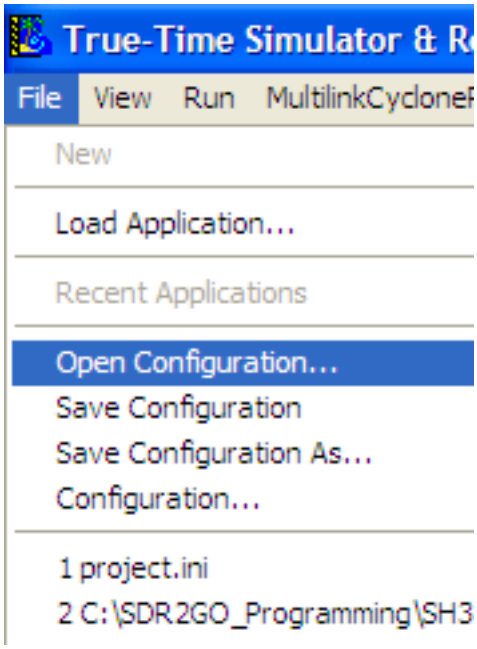
To start this application browse down to the folder: C:\Program Files\Freescale\CodeWarrior for Microcontrollers V6.3\prog and look for the application called **hiwave.exe** as shown below and double click the icon.



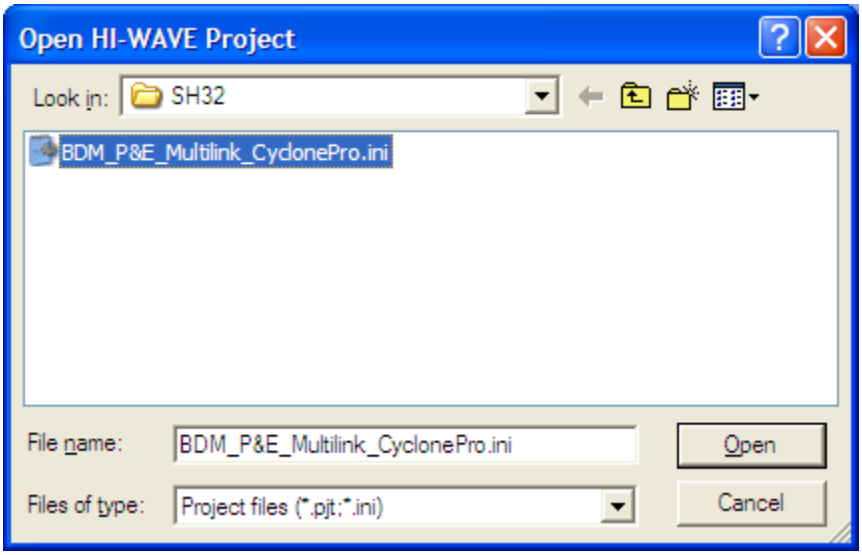
When the simulator opens you will probably see the dialogue box shown below. Click Abort.



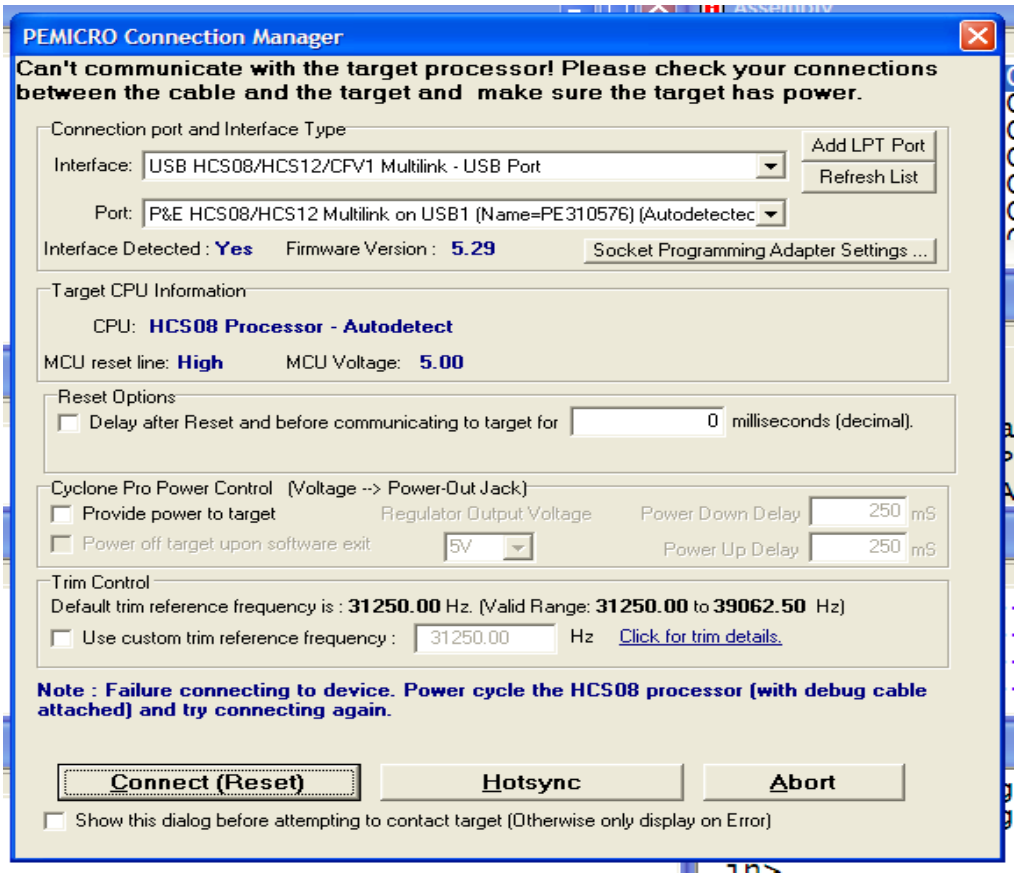
After clicking Abort, go to the File Tab and click on the Open Configuration item as shown below:



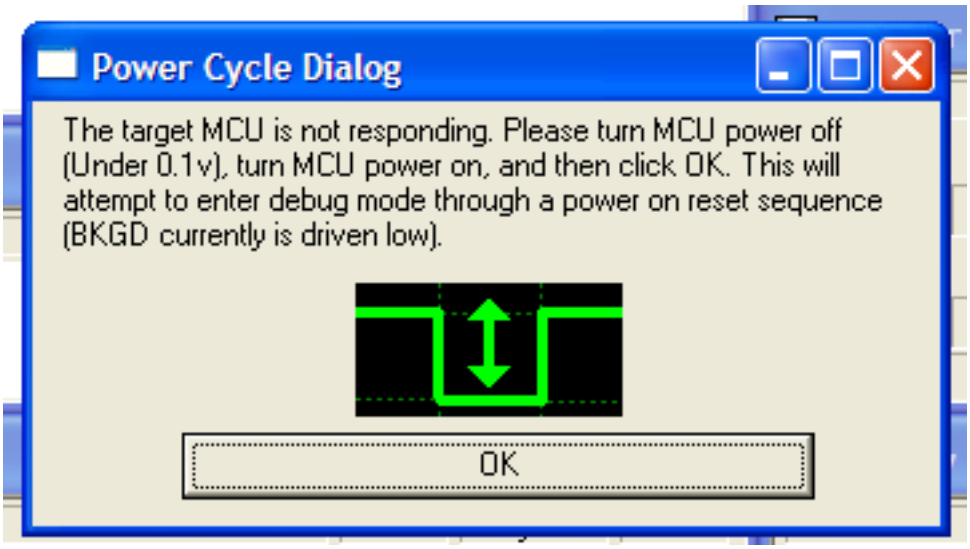
Next, use the dialogue box as shown below to locate the Multilink ini file as shown below. This file is in C:\SDR2GO_Programming\SH32 folder. Click Open.



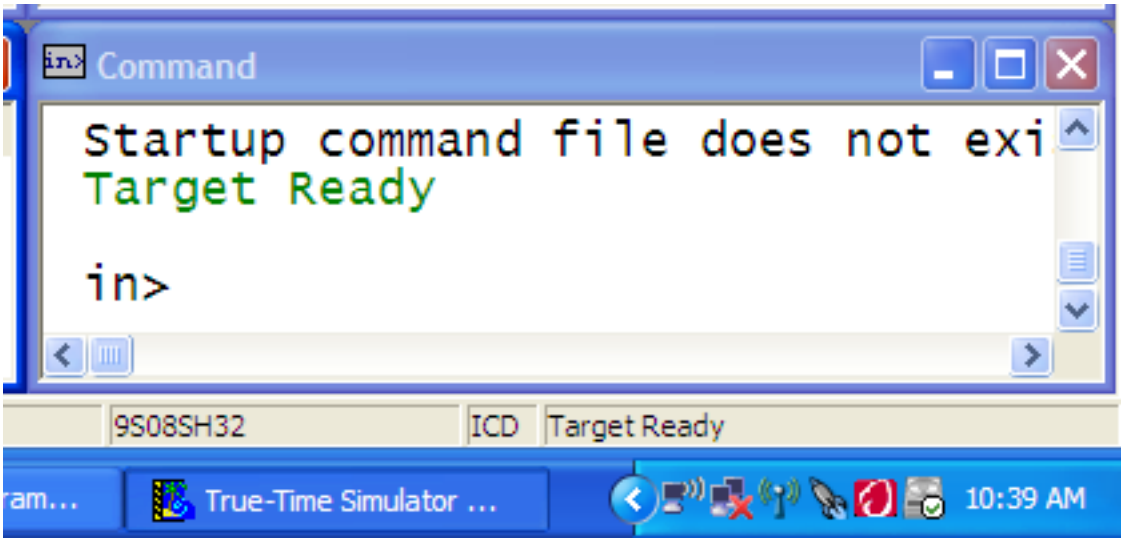
You will then see a connection dialogue box as shown below. Click Connection (Reset).



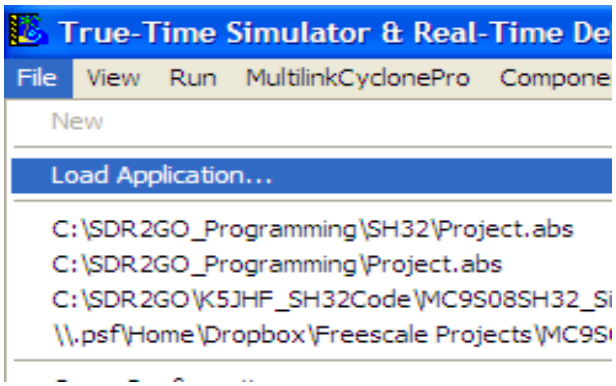
You will then probably get the nasty note shown below:



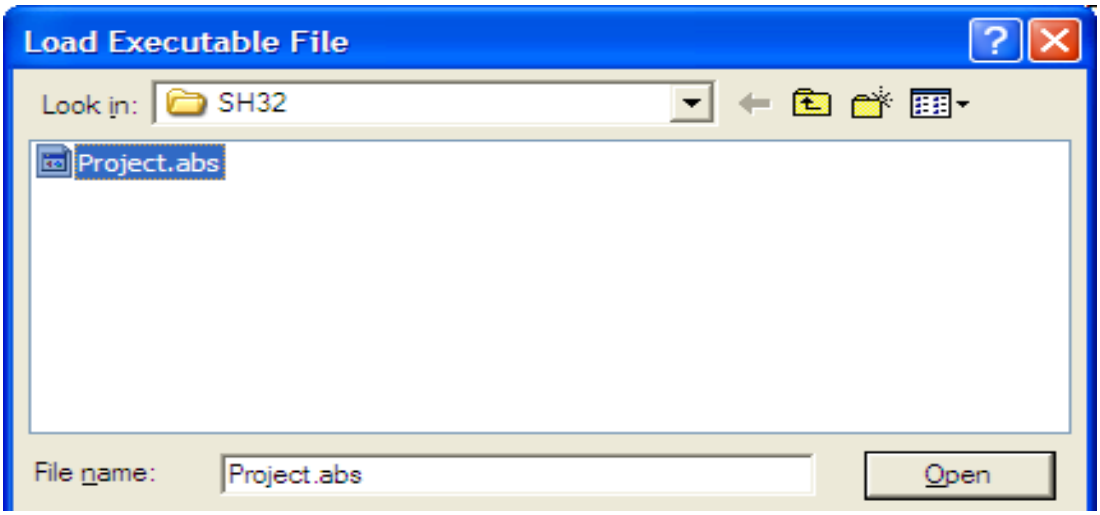
Follow the instructions. You then should get the message shown below in the extreme lower right corner of your display as shown below which says Target Ready:



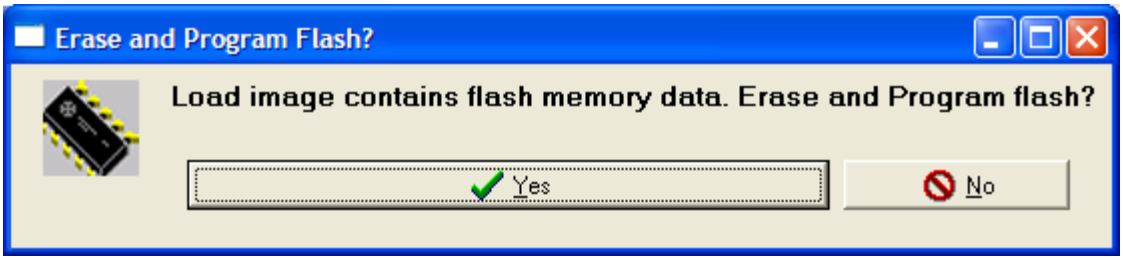
Then, got back to the File Tab and click on Load Application.



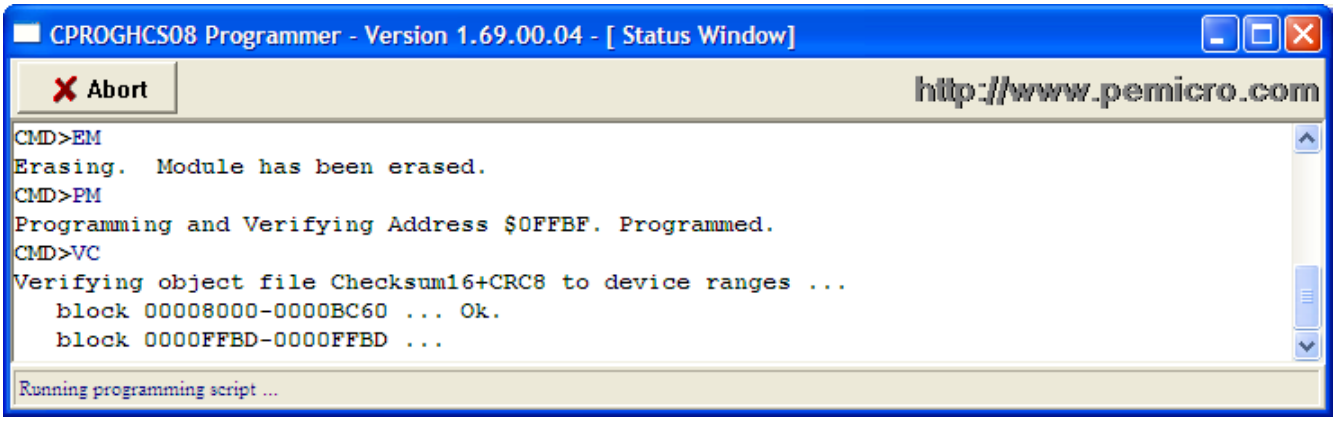
Finally, drill down to the C:\SDR2GO_Programming\SH32 folder then select and open the Project.abs file.



You will then be presented with the dialogue box below, click Yes.



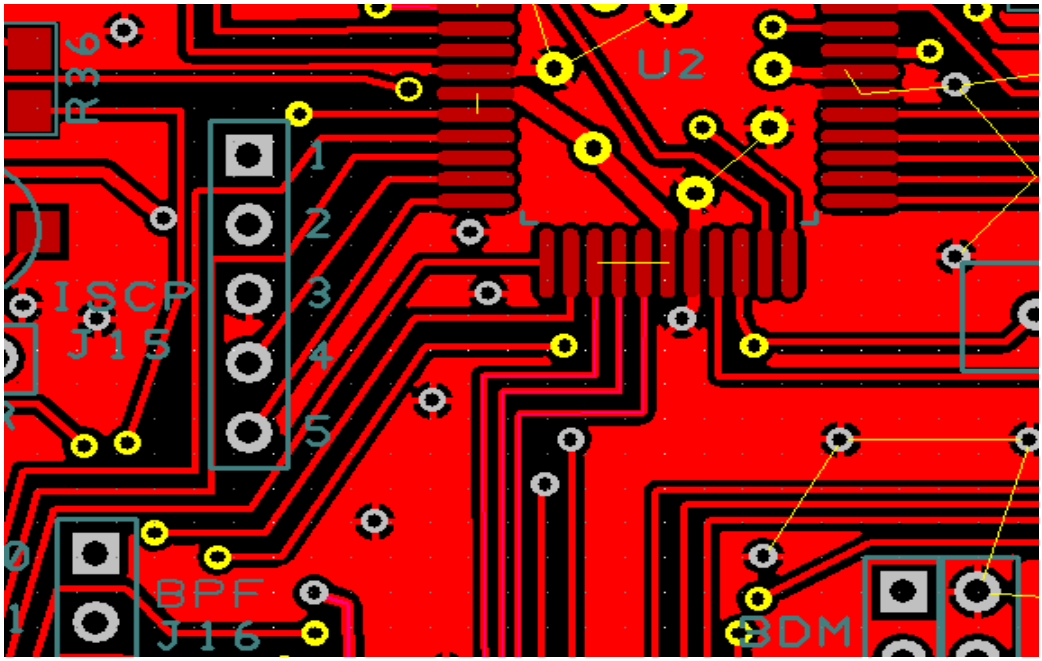
You will then see a Status Window open which displays the operations taking place. When this window closes, the programming of the SH32 is completed.


CPROGHCS08 Programmer - Version 1.69.00.04 - [Status Window]
Abort
http://www.pemicro.com
CMD>EM
Erasing. Module has been erased.
CMD>PM
Programming and Verifying Address $0FFBF. Programmed.
CMD>VC
Verifying object file Checksum16+CRC8 to device ranges ...
block 00008000-0000BC60 ... Ok.
block 0000FFBD-0000FFBD ...
Running programming script ...

dsPIC33JF128GP804I/PT Programming Interface
The dsPIC will be provided pre-programmed in the production run of kits.

Please note that programming of the dsPIC should not be attempted until the Initial Board Start Up procedure outlined below has been completed.

If the user wishes to upgrade the dsPIC firmware in future the programming interface may be accessed by installing the header pins for J15 which is designed to accommodate the mating plug from a PICkit 2 programmer. Please see the figure below which shows the J15 location.



The PICkit 2 plugs directly onto J15 with the PICkit Label and arrow on PICkit2 facing to the right and the arrow directly above Pin 1 of J15.

For details on the PICkit 2 programmer please use this link:
http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en023805

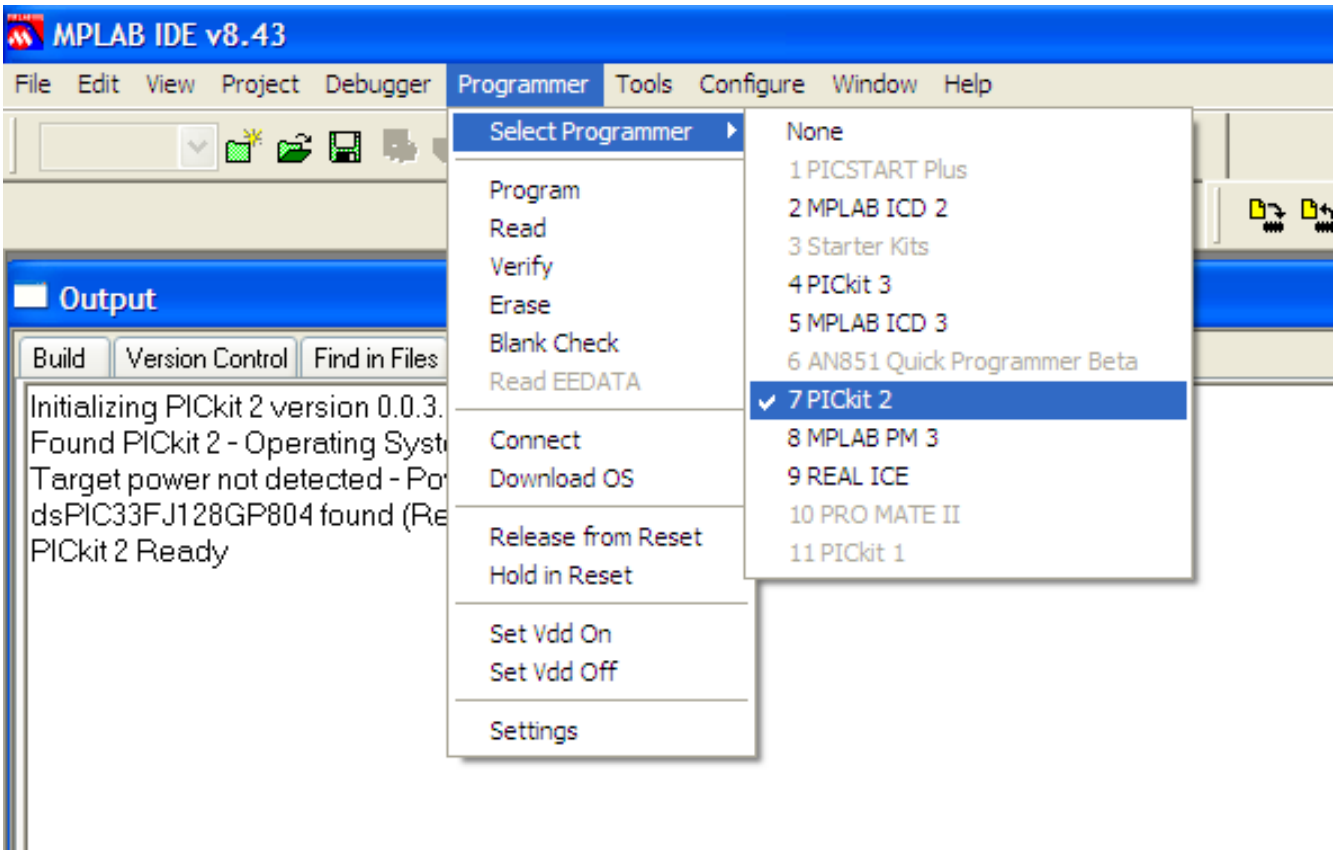
In order to use the PICkit2 programmer Microchip's MPLAB IDE must be downloaded and installed on your machine. Please use this link by depressing Ctrl and clicking this link.

http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en019469&part=SW007002#P175_5158

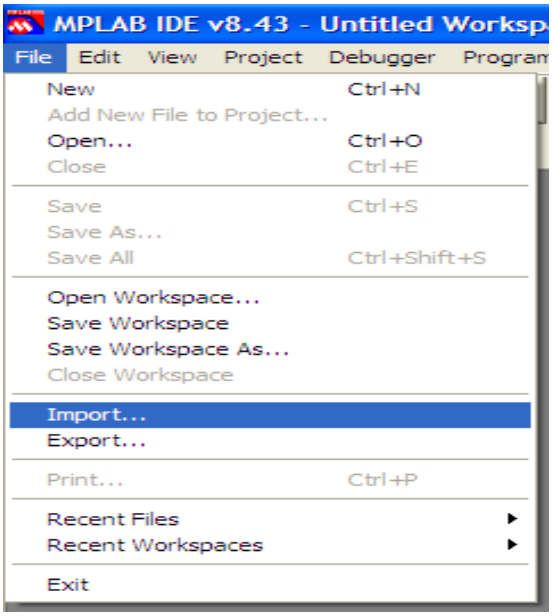
The HEX file required for programming the dsPIC is included in the C:\SDR2GO_Programming folder referenced above.

After you have completed the installation of the MPLAB IDE connect the PICkit 2 to the SDR2GO board. If the SDR2GO board is not powered, the PICkit 2 will supply power via the USB connection. Now, start the MPLAB IDE. When it opens, go to the Programmer Tab and select the PICkit 2 from the drop down list as shown below. You will see the result of the PICkit 2 successful connection to the

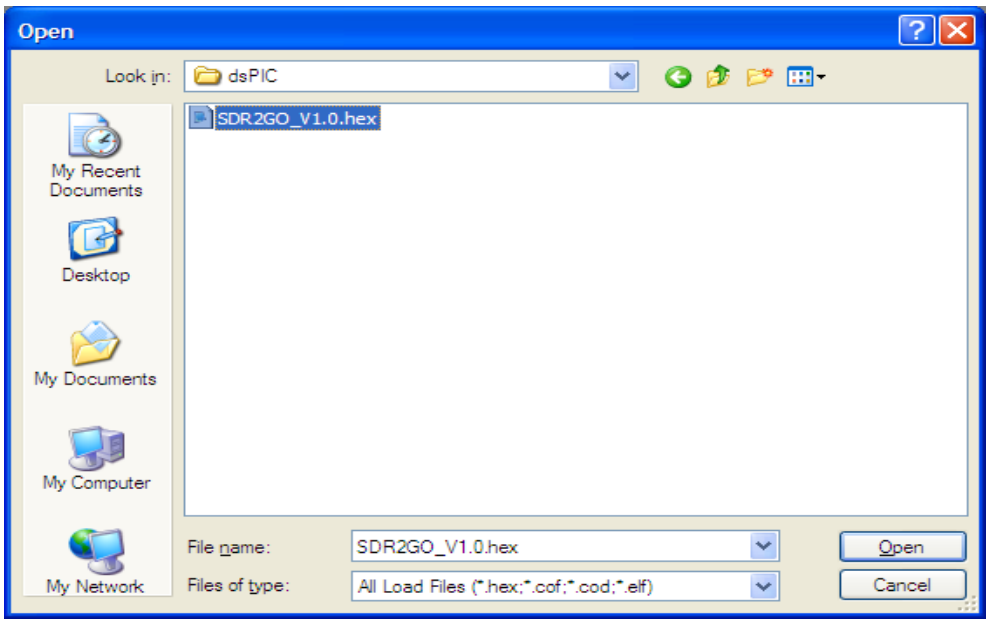
SDR2GO board in the Output Window.



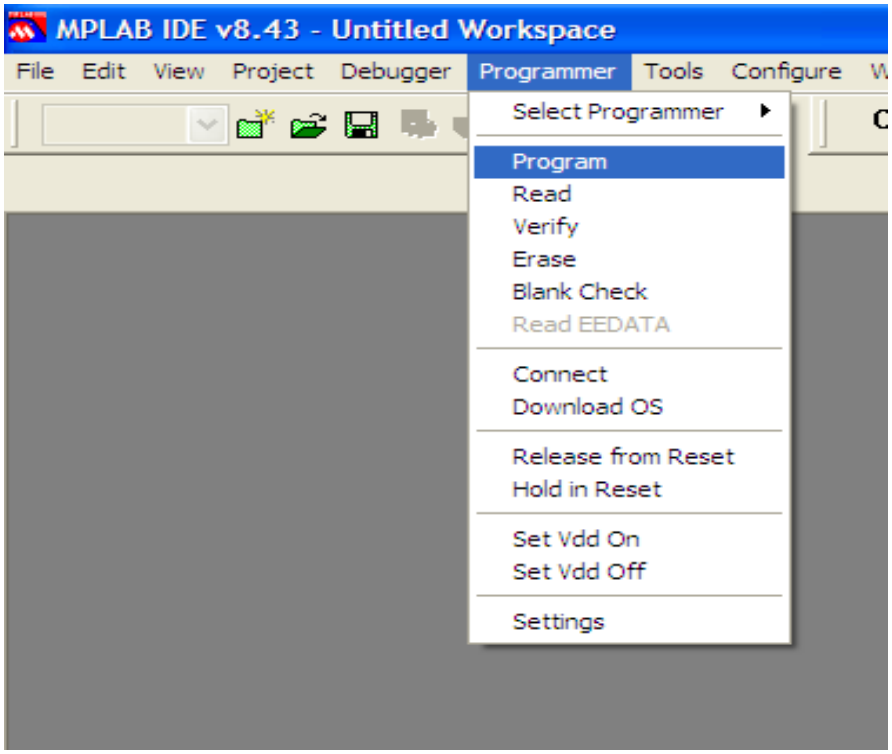
Next, go to the File Tab and select and click on Import as shown below.



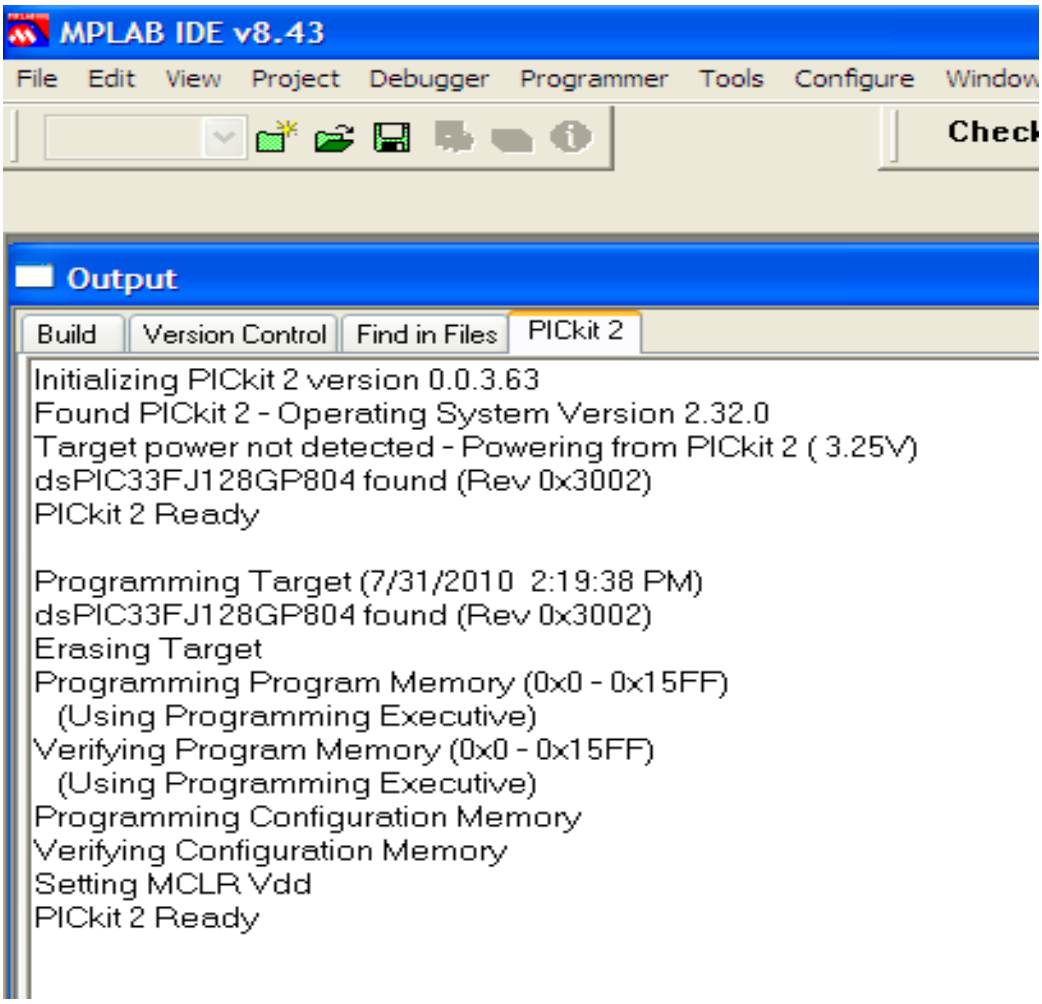
Then, drill down to the C:\SDR2GO_Programming\dsPIC folder and open the SDR2GO_V1.0.hex file.



Finally, go to the Programmer Tab and click on Program as shown below.



Upon successful programming you will then see the message shown below.


MPLAB IDE v8.43
File Edit View Project Debugger Programmer Tools Configure Window
Check
Output
Build Version Control Find in Files PICkit 2
Initializing PICkit 2 version 0.0.3.63
Found PICkit 2 - Operating System Version 2.32.0
Target power not detected - Powering from PICkit 2 (3.25V)
dsPIC33FJ128GP804 found (Rev 0x3002)
PICkit 2 Ready
Programming Target (7/31/2010 2:19:38 PM)
dsPIC33FJ128GP804 found (Rev 0x3002)
Erasing Target
Programming Program Memory (0x0 - 0x15FF)
(Using Programming Executive)
Verifying Program Memory (0x0 - 0x15FF)
(Using Programming Executive)
Programming Configuration Memory
Verifying Configuration Memory
Setting MCLR Vdd
PICkit 2 Ready

NUE PSK Modem Keyboard Interface
The SH32 may be interfaced with a keyboard via J8 so that the SI570 frequency may be adjusted and the associated control parameters configured. In addition, the same keyboard may be shared with the NUE PSK modem by using header J17. The selection of either the SH32 or the NUE PSK to interface with a connected keyboard is controlled by setting the Keyboard CLK SW input either high or low. This signal input is accessed via pin 6 of J6.

Initial Board Start Up
As with any other board project, take the time to do a thorough clean up of the board and make a careful visual inspection of all solder connections before applying power. The SH32 runs at 5 VDC while the dsPIC33 and the two EEPROMs run at 3.3 VDC. Two voltage translators, Q1 and Q2, provide a 5 volt to 3.3 volt interface for the I2C bus that is controlled by the SH32. The I2C bus controlled by the dsPIC does not require voltage translation.

The inputs to the dsPIC use internal pull up resistors and should be around 2.3 volt when the connected input circuit is open.

The inputs to the SH32 use internal pull up resistors and should be around 4.8 volt when the connected input circuit is open.

Listed below are the voltages that are present on the codec, U1, when it is running properly. Please note that the data signals (Pins 4 &5) will not be present until the dsPIC is programmed and running.

Pin 1:	0.2 to 2.9 volt clock	Pin 18	0.9 volt
Pin 2:	0 to 3.1 volt clock	Pin 19	1.2 volt
Pin 3:	0 to to 3.1 volt clock	Pin 20	0.9 volt
Pin 4:	0 to 3.1 volt data	Pin 21	0.9 volt
Pin 5:	0 to 3.1 volt data	Pin 22	0.9 volt
Pin 6:	3.2 volt	Pin 23	0.9 volt
Pin 7:	0 volt	Pin 24	1.8 volt
Pin 8:	0 volt	Pin 25	1.9 volt
Pin 9:	0 - 3.2 volt	Pin 26	3.2 volt
Pin 10:	0 - 3.2 volt	Pin 27	0.9 volt
Pin 12:	0 volt	Pin 28	0 volt
Pin 15:	0.9 volt	Pin 29	1.8 volt
Pin 16:	0.9 volt	Pin 30	3.3 volt
Pin 17	0 volt	Pin 31	3.3 volt

After you have verified that there are no bolted on faults and the processor plus codec pins are at the expected voltages you may proceed with programming or board start up.

PTT Interface and Operation
The dsPIC is provided with an input for PTT IN. When this input is high, the DSP algorithm will be in the receive mode. When the PTT IN input is set low, the algorithm will be in the transmit mode. The dsPIC controls an output signal called PTT Drive which is fed to Q3 so that a 0 to 5 volt signal may be fed into the SoftRock PTT IN input.

DSP Switch Operation and Initial Testing

To properly exercise the DSP functions the switch inputs to the dsPIC which control the DSP functions must be set properly. These inputs are labeled as bit 0 thru bit 3 and are accessed via J13. These inputs are pulled high by internal pull up resistors. So, you may either wire up switches to J13 or simply use jumpers to pull the desired inputs low.

In addition, the PTT IN signal on J3 must be set during testing.

And, the Encoder #2 inputs (A, B, & PB) should be connected to J6.

Here is an explanation of how all of these inputs play together.

Bit 0 and bit 1 control how the Encoder #2 inputs are used as listed below:

bit 0 & bit 1 low:	Encoder #2 sets Audio Level Adjustments on Receive & Transmit
bit 0 high & bit 1 low:	Encoder #2 sets IQ Amplitude Adjustment Parameters
bit 0 low & bit 1 high:	Encoder #2 sets IQ Phase Adjustment Parameters
bit 0 & bit 1 high:	Default IQ parameters are stored when Encoder #2 PB depressed

The PTT_IN signal puts the DSP algorithm into the receive mode when set high and the transmit mode when set low.

When the DSP is in the receive mode, Encoder #2 will adjust the Receive IQ Amplitude and IQ Phase Parameters plus the receive headphone audio output level available on J14.

When the DSP is in the transmit mode, Encoder #2 will adjust the Transmit IQ Amplitude and IQ Phase Parameters plus transmit audio output level available on J2.

These two sets of IQ Parameters (Amplitude & Phase) plus transmit audio level adjustments (microphone input gain, line input gain and CW output level) are stored separately in the EEPROM.

When the Encoder #2 PB is depressed, the parameter under the control of Encoder #2 will be stored in the EEPROM and used in the DSP algorithm.

Bit 2 controls the sideband selection. When bit 2 is low, the lower sideband is selected. When bit 2 is high, the upper sideband is selected. The selected sideband will be used in both the receive and transmit mode.

Bit 3 controls the audio input to the codec for transmit operation. When bit 3 is low, the microphone input gain is low and no bias is applied to the microphone input terminal. This mode of operation is termed “Line Input Mode”. When bit 3 is high, bias is applied to the microphone input terminal and the input gain is high. This mode of operation is termed “Microphone Input Mode”.

Setting Initial IQ Parameters

The DSP algorithm reads the stored IQ Parameters on boot up of the dsPIC. The EEPROM dedicated to the dsPIC is shipped blank. So, in order to provide the DSP algorithms with useable IQ parameters, you must first initialize the stored IQ parameters. This is done by setting bit 0 and bit 1 high and then depressing the Encoder #2 PB. This action sets the transmit and receive IQ Parameters to Amplitude 1.0 and Phase = 0.0.

Setting Receive IQ Parameters
After the IQ Parameters have been initialized, the Receive IQ Parameters may be adjusted to minimize the unwanted signal. This may be done as follows:

First, place the DSP function in the receive mode and select the upper sideband operation. Next, feed in a RF signal of known frequency into the SoftRock with an amplitude of 1 to 50 microvolts. Then tune the SoftRock to a receive frequency that is 1500 Hz below the RF signal frequency and then observe the signal in the Head Phone output, J14. Please note that the demodulated signal is fed to both the left and right Head Phone outputs. You should see a strong audio signal at 1500 Hz. Finally, place the DSP functions into the lower sideband mode by setting bit 2 low. You should see a reduced amplitude audio signal at 1500 Hz. Now using bit 0 and bit 1 in conjunction with Encoder #2 adjust the Receive Amplitude and Phase Parameters to minimize the lower sideband audio signal. You will need to alternate between Amplitude and Phase adjustments to get a good minimum. The ultimate unwanted signal rejection should be on the order of 50 to 70 dB. At this point, depress the Encoder #2 PB to store the Receive IQ Parameters in the EEPROM.

Setting Transmit IQ Parameters
After the IQ Parameters have been initialized, the Transmit IQ Parameters may be adjusted to minimize the unwanted signal. This may be done as follows.

First, place the DSP function in the transmit mode and select the upper sideband operation. Next, set bit 3 low to place the codec audio input into the Line Input Mode. Then, feed in a 1500 Hz audio signal into the codec audio input, J3. At this point you can observe the SoftRock output signal in the frequency region of where the SoftRock frequency is set. You should see a strong upper side band signal and a weaker lower sideband signal. At this point, adjust the amplitude of the audio input signal so that no 1500 Hz harmonics are present in the RF output of the SoftRock. Finally, use bit 0 and bit 1 in conjunction with Encoder #2 to adjust the Transmit Amplitude and Phase Parameters to minimize the observed lower sideband signal. At this point, depress the Encoder #2 PB to store the Transmit IQ Parameters in the EEPROM.

The Transmit I/Q Parameters stored in the EEPROM are used for all modes of transmission: SSB Voice, Digital PSK, and CW.

Setting Receive Audio Level
Receive audio level to the headphone output, J14 may be adjusted when the DSP algorithm is in the receive mode. Set bit 0 and bit 1 to low and use Encoder #2 to adjust the output audio level. This adjustment is stored in the EEPROM when Encoder #2 is depressed.

Setting Transmit Audio Levels
In the transmit mode three separate output audio levels may be adjusted depending on the state of Bit 3 input and the state of the GPIO_bit 4 (SSB /CW) input.

When the GPIO_bit 4 input is low the DSP algorithm is in the CW mode. In this mode Encoder #2 may be used to adjust the audio output to the SoftRock on J2 by setting Bit 0 and Bit 1 low. After setting the CW audio level, use Encoder #2 PB to store the adjusted CW audio level in the EEPROM.

When the GPIO_bit 4 input is high the DSP algorithm is in the SSB/Digital mode. In this mode Encoder #2 may be used to adjust the audio output to the SoftRock on J2 by setting Bit 0 and Bit 1 low. The adjustment of the output audio level in this mode is actually done by adjusting the microphone

input gain or the line input audio gain. As explained above, the audio input available on J3 is controlled by the state of Bit 3. When Bit 3 is high, the software is in the Microphone Input mode. When Bit 3 is low, the software is in the Line Input mode. After setting either the Microphone Input audio gain or the Line Input audio gain, use Encoder #2 PB to store the audio gain setting in the EEPROM.

SI570 Control Set Up and Operating Instructions
To complete the testing of the SDR2GO board the SI570 control functions are set up and exercised. At this point Encoder #1 should be connected to header J6 and the LCD connected to header J9. Please note the two alternate ways that your LCD may be connected to J9 as explained above and shown on sheet 3 of the schematics.

At this point you are ready to use:

Secret Decoder Ring for the New Si570 Controller II

NEW ADDITIONS:
1) Backlit 16x2 LCD
2) Keyboard Interface in addition to a Rotary Encoder
3) Scanning capability
4) Intuitive operation with additional feedback to user
5) Offsets selectable on a memory location basis instead of one for the whole controller
6) 12 characters of your custom alphanumeric text may be displayed
7) The frequency is shown in KHz with 1Hz resolution

Many of the features and operations are as implemented on the current Si570 Controller and Frequency Generator Kit #2.

Initialization

Initialization is done by holding in the Rotary Encoder Push Button (PB) while powering up the SDR2GO board. Keep holding it “in” until the start up memory location is displayed. It will show the code date, “John H. Fisher”, count down, and reset all EEPROM locations

The cursor is an underscore and will appear the first time anything on the screen changes. The cursor can be controlled by holding the PB “in” while rotating the knob --or-- the arrow keys on the keyboard.

Data is saved by holding the PB “in” for a few seconds --or-- with the keyboard “Enter” key. Feedback will be seen on the LCD.

Data entry can be locked/unlocked by momentarily pushing the PB --or--with the keyboard “L” key. A “L” will appear in the lower left hand corner of the LCD.

Low memory (location “000” to “040”) is divided into three types, “P” prefix for parameters which effect the whole controller, “T” prefix which shows text you have selected, and “M” prefix for memory specific data.

PARAMETERS:

P000 Default Si570 Frequency (set for the part you are using)
P001 Start up Memory Location (set for where you want to start)
P002 Start up Cursor Position (set for where you want to start)
P003 Fout Mult (Frequency out multiplier for SoftRock …"4")
P004 Fout Div (Frequency out divisor for SoftRock)
P005 Disp Mult (display multiplier when using an external frequency multiplier and you want the output frequency from the multiplier shown correctly on the LCD)
P006 Disp Div (display divisor when using an external frequency divider and you want the output frequency from the divider shown correctly on the LCD …handy when using the Si570 circuit at xxx Khz which is well below the Si570 cutoff spec)
P007 Tune Rate (Rotary Encoder pulses per step)
P008 Tune Dir (Clockwise or Counterclockwise is "increase")

TEXT:

T009 – T040 The Si570 Controller II has five bits of input (32 possibilities) to define 32 possible "12 alphanumeric character" combinations which will appear on the lower line of the LCD. These five bits are labeled as GPIO bit 0 Rx thru GPIO bit 4 and may be accessed via header J7 on the SDR2GO board.

The text is independent of Frequency, Band, Offset, etc. It's only dependent on how those 5 bits are set.

The SDR2GO V1.6 software for the dsPIC33 provides for dsPIC33 pins 25, 26, 27, 13 and 12 to be configured as outputs. These output pins are mapped to follow the state of five disPIC33 inputs as follows:

Pin 25, GPIO bit 0, mirrors bit 0 input

Pin 26, GPIO bit 1, mirrors bit 1 input

Pin 27, GPIO bit 2, mirrors Sideband Select input

Pin 13, GPIO bit 7, mirrors Microphone Power input

Pin 12, GPIO bit 10, mirrors PTT_IN input

Please find below a suggested set of messages which may be programmed by the user for SH32 memory locations T009 thru T040.

Suggested SDR2GO Message Karnaugh Map

MI = Mic. Input	RX = Receive	USB = Upper Sideband	HP = Headphones
LI = Line Input	TX = Transmit	LSB = Lower Sideband	QA = Quadrature Amplitude
			QP = Quadrature Phase
			DF = Default Parameters

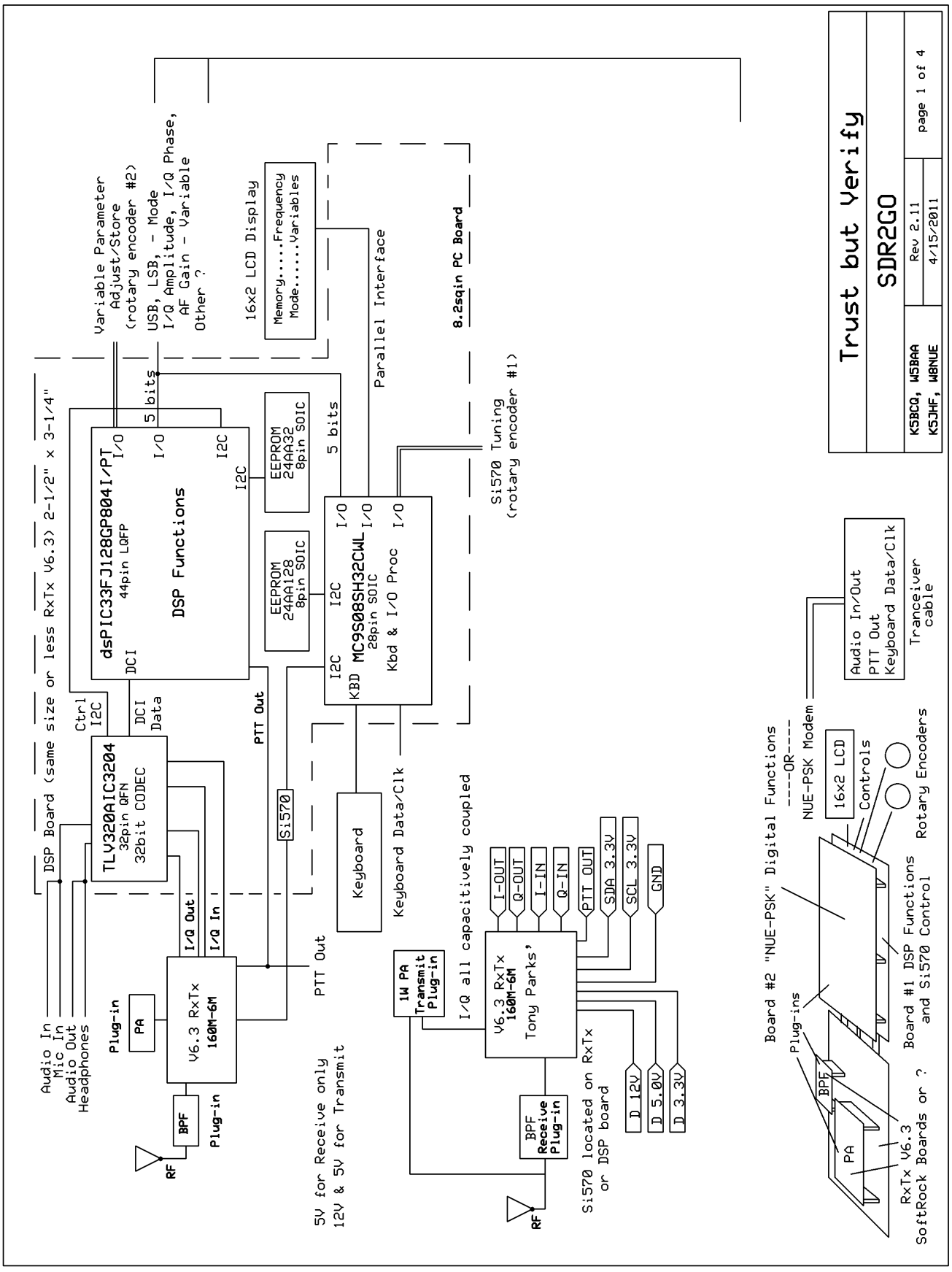
SH32 Loc.	GPIO 4 State	GPIO 3 State	GPIO 2 State	GPIO 1 State	GPIO 0 State	Message
T09	Lo	Lo	Lo	Lo	Lo	TX LSB HP LI
T10	Lo	Lo	Lo	Lo	Hi	Tx LSB QA LI
T11	Lo	Lo	Lo	Hi	Lo	TX LSB QP LI
T12	Lo	Lo	Lo	Hi	Hi	TX LSB DF LI
T13	Lo	Lo	Hi	Lo	Lo	TX USB HP LI
T14	Lo	Lo	Hi	Lo	Hi	TX USB QA LI
T15	Lo	Lo	Hi	Hi	Lo	TX USB QP LI
T16	Lo	Lo	Hi	Hi	Hi	TX USB DF LI
T17	Lo	Hi	Lo	Lo	Lo	RX LSB HP LI
T18	Lo	Hi	Lo	Lo	Hi	RX LSB QA LI
T19	Lo	Hi	Lo	Hi	Lo	RX LSB QP LI
T20	Lo	Hi	Lo	Hi	Hi	RX LSB DF LI
T21	Lo	Hi	Hi	Lo	Lo	RX USB HP LI
T22	Lo	Hi	Hi	Lo	Hi	RX USB QA LI
T23	Lo	Hi	Hi	Hi	Lo	RX USB QP LI
T24	Lo	Hi	Hi	Hi	Hi	RX USB DF LI
T25	Hi	Lo	Lo	Lo	Lo	TX LSB HP MI
T26	Hi	Lo	Lo	Lo	Hi	TX LSB QA MI
T27	Hi	Lo	Lo	Hi	Lo	TX LSB QP MI
T28	Hi	Lo	Lo	Hi	Hi	TX LSB DF MI
T29	Hi	Lo	Hi	Lo	Lo	TX USB HP MI
T30	Hi	Lo	Hi	Lo	Hi	TX USB QA MI
T31	Hi	Lo	Hi	Hi	Lo	TX USB QP MI
T32	Hi	Lo	Hi	Hi	Hi	TX USB DF MI
T33	Hi	Hi	Lo	Lo	Lo	RX LSB HP MI
T34	Hi	Hi	Lo	Lo	Hi	RX LSB QA MI
T35	Hi	Hi	Lo	Hi	Lo	RX LSB QP MI
T36	Hi	Hi	Lo	Hi	Hi	RX LSB DF MI
T37	Hi	Hi	Hi	Lo	Lo	RX USB HP MI
T38	Hi	Hi	Hi	Lo	Hi	RX USB QA MI
T39	Hi	Hi	Hi	Hi	Lo	RX USB QP MI
T40	Hi	Hi	Hi	Hi	Hi	RX USB DF MI

MEMORY:
M041 to M999 This is the main display you will see showing the Frequency in KHz with a 1Hz resolution on the top line and Mxxx and whatever text you have on the bottom line. You can move the cursor with the Rotary Encoder or the Keyboard to change frequency. If you move the cursor to the "M" position by rotating the Rotary Encoder or by pushing an arrow key on the keyboard the Memory Setup screen will be toggled. This shows the Frequency, the Offset (+ or -) desired for that memory location, and the Band selection (0-7). All of the data can be changed using the Rotary Encoder or Keyboard. When saved, the Controller will go to the next available memory location carrying those data (but not store it). This is for your convenience setting up many memories.

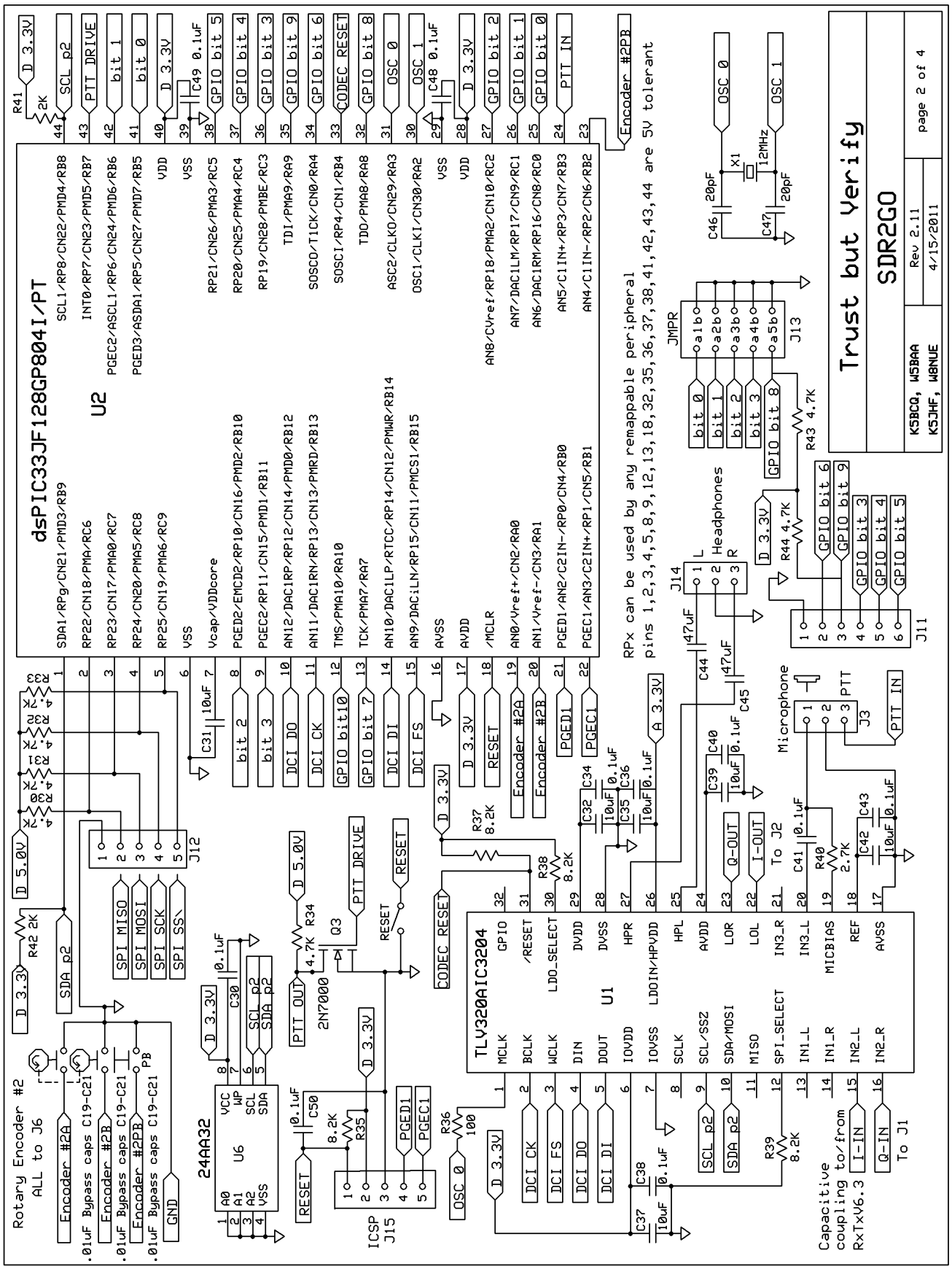
SCANNING:
This is only used with a keyboard. The scan will go from one memory location frequency to the next memory location frequency. After selecting the start memory location, push the "S" key and the LCD will show a "S" where the "M" was, indicating Scan Setup data and a "D" will show the delta step size in Hz. Enter the desired step size and save it. Pushing the "space" bar will initiate the scan and stop the scan. This will also be available to a microcontroller to "stop on receipt of signal".

KEYBOARD SUMMARY:
"l" ….toggles Lock on/off
"r" ….toggles between Radio mode and Data mode
"m" ….selects entry of Memory location
"f" ….selects entry of Frequency data
"o" ….selects entry of Offset data
"b" ….selects entry of Band selection
"d" ….selects entry of Delta Frequency data (step size)
"c" ….copies current data to next available memory location
"s" ….toggles sweep mode on/off
"spacebar" ….sweep stop/start
"Delete" ….deletes/zeros current memory location data
"Enter" ….saves all displayed Data to current Memory location
"Home" ….goes to initial Memory location
Left Arrow ….moves Cursor left
Right Arrow ….moves Cursor right
Up Arrow ….increments Data
Down Arrow ….decrements Data

SDR2GO Block Diagram – Sheet 1



SDR2GO DSP Schematic – Sheet 2



SDR2GO SI570 Control – Sheet 3