Sierra Radio Systems Controller Software and Configuration Manual

Version 1.12

June 20, 2012

Note: this manual has not been kept up to date, and badly needs editing. However, it does have some introductory and descriptive content which is not in the other manuals.

Getting Started

To get on the air with the simplest possible linked repeater configuration, just take the following steps:

1. Following the instructions in the SRS Series 200 Hardware Manual, configure the hardware
2. Optionally, connect a computer to the CPU board serial port connector, using a female/female straight-through cable (beware, most female/female cables are not straight through and swap pins 2 and 3 between the two connectors).
3. Turn on the system.
4. Transmit to the repeater input. Unless the system has already been configured with a more complex configuration, the repeater and all of the links transmitters should turn on, and the repeater port should send a CW identifier (the default is "NO ID", which is what you will get if the system has not already been configured).
5. Basic configuration can be done via DTMF transmitted to any of the receiver inputs, but it is much easier to do it via the serial port. So, if available, configure the computer's serial port for 9600 baud, 8 bits, no parity, and no handshaking.
6. Using either a terminal emulator, or the Sierra Radio Systems configuration software, define the unlock codes for executing protected commands; and define the ID string, location string, and prefix (used for commanding the controller from one of the links). See the **Basic Configuration** section below for instructions on how to do this.
7. You will probably want to adjust the input and output gain of the controller to get the deviation correct at each of the transmitters. This should be done next. See the **Level Adjustment** section below for instructions on how to do this.

Introduction

In order to configure the Sierra Radio Systems model 200 repeater controller for operation at a specific site, it is helpful to have an understanding of the overall architecture of the controller and the embedded software that makes it run.

Table 1 provides a set of definitions for the terms used in this manual.

Backplane	The circuit board into which all of the other circuit boards are plugged in. It is passive (no active silicon devices) and wires all the other circuit boards together.

CPU Board	The circuit board plugged into the backplane which contains the microprocessor which runs the main program controlling all of the operations of the controller.
Radio Control Board	One of the circuit boards plugged into the backplane, which provides the interface between the controller and a radio transmitter and receiver combination or any other audio signal source and destination, such as an IRLP node or telephone interface.
Port	Generic term for one full duplex connection to the controller. This could be a repeater, a link to another site, and IRLP node, or a remotely controller transceiver (remote base).
Physical Port Number	The Radio Control board farthest to the left in the card cage, and next to the CPU board is designated physical port number 0, the Radio Control board immediately to the right of physical port number 1, and so on.
Logical Port Number	Users need to control the system by commanding specific ports, using the port number. For example, the user may want to disconnect Link 1 at the San Francisco Bay area Black Mountain site which links to another site at Crystal Mountain. He would issue the Link 1 OFF command to do this. There may be situations where the port number that a user associates with a particular link cannot be the same as the physical port number. As a system grows, or if there is a problem with the hardware for one of the links, it is undesirable to change the logical port number for a link to some particular other site, but it may be necessary to change the physical port number attached to that link. The controller provides a logical to physical port mapping that can be used to make the logical and physical port numbers different.
Repeater or **Local Radio** or **Repeater Port**	A port which is configured to operate as a repeater. Signals coming in on its receiver will be repeated back out its transmitter.
Link or **Link Radio** or **Link Port**	A port which is configured to operate as a link. Signals coming in on its receiver will not normally be repeated back out its transmitter, but only to other ports which are connected ("linked") to this port.
IRLP Port	A port which is configured to operate as a restricted link, normally used for IRLP or other voice over IP connections into the site.
Remote Base Port	A port which is configured to operate a remotely controlled transceiver. It invokes special telemetry, and is most commonly used in a simplex mode. The v0.90 software has limited support for this type of port.
Unlock code and **Super-unlock code**	Many controller commands cannot be executed until an unlock code is entered. This prevents unauthorized users from reconfiguring the system. There are up to 20 unlock codes, which are designated unlock code number 0 to 19. There are two levels of unlock code, designated unlock code and super-unlock code. Unlock codes 0 and 1 are super-unlock codes. The rest are just unlock codes. A few commands require the super-unlock code, including the commands to change the unlock code table, commands to force reloading of default parameters, and commands to backup and restore the EEPROM parameter sets.
Group	A set of ports which are normally linked together, independently from ports which are members of a different group. Definition of groups allows the

	controller to manage several semi-independent repeaters. Group membership can be changed dynamically give great flexibility in connecting different radios together in different ways.

Figure 1 shows an overall block diagram of the controller. The basic controller functions are executed by two kinds of cards which are plugged into the controller backplane. Overall operation of the controller is managed by the "CPU Board." The interface to each radio (or other signal source such as a voice over IP node or telephone connection) is managed by a separate "Radio Control Board" for each port.

Figure 1 goes here

The Rev 1.0 CPU board contains a Microchip 18F8722 microcontroller. The Rev 1.0 Radio Control board contains a Microchip 18F4620 microcontroller. The embedded software running on these processors is programmed in a combination of C and assembler. The Microchip C18 compiler and MPLAB IDE and assembler were used to develop the code.

Hardware Details

CPU Board

In addition to the processor, the CPU board contains hardware with the following functions:

- 64kB serial eeprom for configuration storage
- 2 wire serial interface (I^2C) to communicate with the processors on the Radio Control boards and with the serial eeprom
- high speed serial interface dedicated to reading COR, PL, and DTMF status from the Radio Control boards
- high speed serial interface dedicated to controlling the PTT, fan relay, audio multiplexer, and general purpose i/o lines on the Radio Control boards
- standard serial port (RS-232) for external computer connection to provide control, configuration, and (not yet implemented) firmware download
- auxiliary serial port (TTL levels)
- 4 channel tone generator, routed to the backplane, used to generate control response telemetry
- digital pot to set maximum tone generator telemetry level
- CPU programming connector

Radio Control Board

In addition to the processor, the Radio Control boards contain hardware with the following functions:

- audio and control line (COR, PL, PTT, dtmf decode) interface to transmit and receive radios
- audio mixer/switch to select any combination of back plane audio to route to the transmitter
- buffer to drive audio from this board to the back plane

- 4 channel tone generator, used to generate ID's, key up telemetry, and busy signals
- digital pots to control receive input gain, maximum tone generator level, PL level (not currently used), and transmit level
- port selection jumpers
- dtmf decoder

Commands

The system is controlled using commands which are sent to the controller by DTMF tones transmitted to one of the receivers connected to the controller, or by the RS232 serial port on the CPU board.

Built-in Commands

There is a large set of built-in commands that can be used set up and manage the repeater system connected to the controller. These commands start with a "C", or "S", and will generally not change from one version of the controller software to the next. To prevent unauthorized users from changing everything in the controller, most of these commands require that an unlock code be entered before the command will be accepted. In the v0.90 beta software the built-in commands and command names cannot be changed by a user or site administrator.

Macros

Commands that are designed to be used for everyday control of the system are implemented as macros. Macros are just lists of built-in commands or other macros which get executed in sequence when the macro is called. The macros can be defined to require entry of an unlock code, or to be executable by anyone with knowledge of the command code which executes a particular macro.

Macros are called in one of two ways. There is a built-in command which takes the macro number to be executed as a parameter. It is possible to invoke any macro by calling this command. There are a set of predefined command names, 000 to 047, each of which force execution of a specific macro. The default contents of these macros are empty, so you are free to define them with the sequence of commands of your choice. There is also a set of built-in commands and associated default macro definitions for the most common user level commands. These are documented in a separate document, "Default Command Macros." In the v0.96 software, the contents of any macro (eg., which built-in commands the macro executes), and the set of commands which call the macros can be separately changed by the site administrator.

There is space for 250 macros in the controller, numbered 0..249. Commands "000" to "049" invoke macros 0..49. Default user commands are mapped to macros 50 and above (131 total as of srsc0096q). Default user command macros are loaded into the table of macros when the controller starts up for the first time. See the default command macro documents for details.

Essential Commands

All of the implemented commands are documented in the document, "SRSC Command List 0100a.doc". A few of the most essential commands are listed here to get started.

A (u) or (su) after the command indicates that an unlock or super unlock code must be entered in order for the command to be recognized.

For the most up to date description of these commands, see the SRSC Command List document.

Most Basic System Management

Code	Operation	Comments
C000 (u)	Reset processor	
C001	Code version	Returns the code version
C021 (u)	Radio card software verion	Returns the version string for one radio card. Syntax is C021r. r is the physical port number.
C100 (su)	Force default configuration load on next processor reset	Sets serial eeprom valid byte to 0, invalidating the eeprom. New eeprom values will be loaded from the hard coded defaults on the next reset.
C102 (u)	Configuration backup	The system configuration is stored in a serial eeprom external to the main processor on the CPU board. The serial eeprom has space for 3 copies of the configuration parameters, including the table of macros. They are designated the working set, backup set, and deep backup set. This command copies the working set to the backup set. It should be used after changes are made to the controller configuration and are verified to be working properly.
C103 (su)	Configuration restore	Restore configuration from backup set to working set. Most controller operations use values of working parameters stored in RAM on the CPU and Radio Control board processors. To make the restored parameters active, a processor reset (C000) should be done after using this command.
C104 (su)	Deep configuration backup	Copy backup configuration set to the deep backup set. This command should be used after the configuration backup command after a new configuration has been thoroughly tested.
C105 (su)	Deep configuration restore	Copy deep backup configuration set to backup set.
C110 (su)	Set unlock code	Sets unlock code for enabling locked commands. This command requires a super unlock code (code #0 or #1). The unlock code number is returned in the telemetry if successful. No checking is done for conflicts with commands, so be careful that a command is not the same as the unlock code or is

		contained in the leading characters of the unlock code. The lowest index blank unlock code is set. Syntax is *C110sssssD. Up to 6 characters can be used (recommended). The terminating D is not used if 6 characters are entered. The unlock code characters must be numeric (0..9); ABCD are not allowed.
C111 (su)	Delete unlock code	Deletes the matching unlock code. This command requires a super unlock code. Unlock code #0 cannot be deleted with this command. The unlock code index is returned if successful. Syntax is *C111sssssD.
C112 (su)	Delete unlock code index	Deletes a specific unlock code number. This command requires a super unlock code. Unlock code number 0 cannot be deleted. Confirms if an unlock code is deleted. Does not confirm if index is out of range, or no unlock code currently exists at the specified index. Syntax is *C112nnD. There is space for 20 unlock codes, with index 0..19.
C113 (u)	Check unlock code	Returns the index of a matched unlock code. Returns nothing if there is no match. Syntax is *C113sssssD.
C114	Set unlock code 0 (no radio control cards)	Sets unlock code 0. No unlock code is required, but all radio control cards must be removed from the controller first. Syntax is *C114sssssD.
C115 (su0)	Set unlock code 0	Sets unlock code 0. Requires unlock code 0 to be used to unlock the controller first. Syntax is *C115sssssD.

Basic Setup Commands

Code	Operation	Comments
C116 (u) S116	Set ID	Sets the primary ID string. Syntax is *C116c1c1c2c2c3c3c4c4…D. Each character in the string is defined by two dtmf keys. '2' is specified by "20", 'A' is specified by "21", 'B' by "22", etc. The complete character translation table is included following this command table. If the *C116 command is used from the serial port, spaces may be inserted between characters for clarity (eg. *C116 c1c1 c2c2…<CR> The S116 version is designed to be used via the RS232 port. It takes the characters directly. All spaces are included. Return terminates the string. ! escapes. For example, *S116 kj6k <CR> sets the ID to " kj6k ". This command does not download the new ID to the radio cards. Issue a

		processor reset (C000) to do this. One or two leading spaces should be included with the ID to give the transmitter enough time to come up before the CW starts; there is no delay in the program. The ID strings can be up to 19 characters long.
C117 (u)	Get ID	Returns the primary ID string
C120 (u) S120	Set prefix	Sets the prefix string. Syntax is the same as for Set ID. Beware that only 0..9, A..D, and # can be used. The prefix can be up to 4 characters long.
C121 (u)	Get prefix	Returns the prefix
C122 (u) S122	Set location string	Sets the location string. Syntax is the same as for Set ID. The location strings can be up to 19 characters long.
C123 (u)	Get location	Returns the location string

Complex Configuration Setup Commands

Code	Operation	Comments
C200 (u)	Set link map	Sets the link map which maps the logical port number to the physical port number. Syntax is *C200p0p1p2…D, where p0 is the physical port number (0..7) that will be recognized as port 0, p1 is the physical port number recognized as port 1, etc. Any ports not listed will be assigned in increasing order following the last one specified. If there is a local repeater port, the first one should be logical and physical port 0 – not absolutely required, some of the commands which treat the lower numbered physical port which is a local radio as the primary repeater port will may not send telemetry to the correct port if this is not done.
C201 (u) C202 C203 C204	Set local radios Set link radios Set remote base radios Set irlp port	Sets the port types. Syntax is *C201r1r2…D, where r1r2… is a list of physical ports from 0 to 7. For example *C2010D sets the first port (physical port 0 as the only local repeater port), and *C2021234567D would set all of the other ports as link ports. Any conflicts between these commands are resolved as the command is entered, in favor of values specified in that command. The receive signal detect qualification and unqualification delays for each radio are set to the respective normal values for the radio type. The PL/COR required state is set to the defaults for the radio type. Be very careful in using these commands remotely. The change takes place immediately

		and a mistake can make it difficult to control the system.
C210 (u)	Define group	Sets a group membership. Syntax is *C210gr1r2r3D, where g is the group number (0..3), and r1, r2, … are port numbers (linkmap applied)
C211 (u)	Define group linked	Sets the group members which are linked by default. Syntax is *C211gr1r2r3D, where g is the group number (0..3), and r1, r2, … are port numbers (linkmap applied). You do not need to define the group members first. Radios included here which are not group members will be ignored, but will be remembered if they are included in the group later.

Audio Amplitude and Deviation Calibration Commands

C219 (u)	Turn on test tone	Turns a 1 kHz tone on and off for test purposes. Syntax is *C219r, where r is the physical port of the radio (0..8, where 8 is the cpu card).
C220 (u)	Turn off test tone	Turns off the test tone from C219. Syntax is *C220r, as for C219.
C221 (u)	Set the test tone frequency	Sets the test tone frequency. Syntax is *C221ffffD, where ffff is the frequency in Hz. The value must be less than 4500 Hz. The default value is 1000 Hz, and is reset whenever the processor is reset. This value is not written to eeprom and is therefore not retained across resets.
C222 (u)	Set the test tone amplitude	Sets the test tone amplitude. Syntax is *C222aaaD, where aaa is the amplitude, from 0..255. The default value is 127, and is reset whenever the processor is reset.
C223 (u)	Set radio pot	Sets one radio pot value. Syntax is *C223rpv, where r is the radio (0..8, where 8 is the cpu card), p is the pot (0..3), and v is the value (0..255). See the schematics for pot function. The CPU telemetry pot is #3. The Radio receive, transmit, and telemetry pots are 2, 1, and 3, respectively.
C224 (u)	Set radio pot interactive	Sets the value of one pot, interactively. Syntax is *C224rp…..D, where r is the radio (0..8, where 8 is the cpu card), p is the pot (0..3), and …. Is a sequence of keystrokes to interactively set the pot value. 1 and 7 increment and decrement the pot value by 1. 2 sets the pot value to 255, 5 sets the pot value to 127, 8 sets the pot value to 0, 3 and 9 increment and decrement the pot values by 10, respectively. D saves the value, * escapes and

		returns the value to what it was before entering this command.
C298 (u)	Force transmitter on	Keys up a specific transmitter. Syntax is *C298n, where n is the physical port number. This will timeout after 5 minutes, even if it is not reset with C299.
C299	Cancel 298	Turns off force transmitter on. Syntax is *C299.

Miscellaneous

C225 (u)	Set teltolocal=0	Command telemetry goes only to source. Responsiveness to RS232 serial port commands is much faster if CW telemetry is turned off while entering commands from the serial port. This command should normally be invoked before configuration via the serial port for that reason.
C226 (u)	Set teltolocal=1	Command telemetry goes to local (and source). This is the default.

User Level Commands, normally invoked by macros

C300 (u)	Site normal	Sets PL on, remote normal, interface normal. No macro capability at this time.
C301 (u)	Link normal	Unlinks all local radios (disconnects the links from the remotes)
C302 (u)	Interface normal	Links the local radios to all linked radios in the group (inverse of C301)
C303 (u)	Remote normal	Currently implements an empty dummy function.
C310 (u)	Force ID	Set ID timer to 0 on all radios in current group, thereby forcing 1064 Hz ID on all. Default for the *808 macro. Cactus manual says 808 should only go to local radios. Palomar manual says it goes to all but doesn't key up the link transmitters unless they already are (same as this implementation)
C311 (u)	Force local ID	Same as C310, but forces only all the local radios in the current group.
C312 (u)	Send ID	Sends ID as status: 800 Hz CW. Currently goes only back to command source radio. This includes the primary local radio if teltolocal=1 (the default) If used as the *809 command, the Cactus manual says it should go everywhere.
C313 (u)	Send location string	OK
C320 (u)	440 off	Inverse of C321. Disables 440 repeat.
C321 (u)	440 on	Enables repeat (loopback) on the local from which the command comes, or the lowest logical radio in the same group as the source if it is a link, Connects the link if it was linked before.
C330 (u)	Link off	Unlinks radios in current group.

C331 (u)	Link on	Links specific radio to all other linked radios in group. Link is mapped to radio number by linkmap.
C332 (u)	Send link status	OK.
C334 (u)	Link delay off	Sets link delay off for link radios in current group.
C335 (u)	Link delay on	Sets link delay on for link radios in current group.
C340 (u)	Remote monitors off	Turns off remote monitor for the command source, with exception given above for remote monitor on. If the local port is in remmonl mode, then all remmonl ports in the group will have remmon cleared. If it is in remmon mode, remmon will be turned off for only that port.
C341 (u)	Remote monitors on	Turns on remote monitor for the command source port, if that is a local (repeater). Otherwise, turn remote monitor on for the lowest numbered logical port local in the group of which the source port is a member. If the command comes from the serial port, set remote monitor on for the lowest numbered local logical port in group 0.
C342 (u)	Remote monitor on with local link enabled	Turns on remote monitor for all of the currently linked local (repeater) ports in the same group as the command source. The local repeater ports remain fully linked to each other.
C344 (u)	Remote monitor off, port specified	Turns off remote monitor as if the command came from the specified port. Syntax is *C344n, where n is the logical port number
C345 (u)	Remote monitor on, port specified	Turns on remote monitor as if the command came from the specified port. Syntax is *C345n, where n is the logical port number.
C346 (u)	Remote monitor on with local link enabled, port specified	Turns on remote monitor with local link enabled as if the command source came from the specified port. Syntax is *C346n, where n is the logical port number.
C350 (u)	COR mode	COR only required for local repeater (all locals in group).
C351 (u)	PL mode	PL required for local repeater (all locals in group).
C352 (u)	Send PL status	Uses status of first local in group.
C354 (u)	COR mode for link	Set COR only for link. This is not sticky, it will be reset on site normal or processor reset. It is a protected command (link map is applied)
C355 (u)	PL mode for link	Set PL required for link. This is not sticky, it will be reset on site normal or processor reset. It is a protected command (link map is applied)
C356 (u)	Send PL status for link	Returns PL status for specific radio (link map is applied)
C357 (u)	Save PL mode for link	Save the PL/COR mode for link. Syntax is C357n,

		when n is the port number (link map is applied).

Overall Structure

All primary repeater control functions are executed by the processor on the CPU board. The processor on the Radio Control Board handles initialization of radio board hardware, such as gain control pots, and generation of radio specific tones (key up telemetry, CW ID, busy signal).

The software for the CPU board is compiled into a program with a name format of SRSCnnmms. The main version number is nn. The minor version number is mm. During development of any minor version, different passes of the code are given a letter, s. The current v0.98 beta code as of the date of this document is designated, SRSC0098w. The binary executable to be loaded into the CPU board processor is in the file srscnnmms.hex. The software for the Radio Control board follows a similar naming scheme, SRSRnnmms. The binary executable to be loaded into the Radio Control board processor is in the file srsrnnmms.hex. There are two versions of the Radio Control board software, one for rev. 1.0 boards that use a 40 pin 18F4620 processor, and the other for rev. 2.0 boards that use a 28 bpin 18F2620 processor. The 18F2620 version contains the letter "n" between the "srsr" and the version number.

The controller has many parameters defining its configuration and operating behavior. Most of these can be easily changed by the site manager. These parameters have hard coded default values which are chosen so that most of the parameter values will be satisfactory for site operation without any further configuration.

The CPU and Radio Control boards have several forms of memory. Flash ROM on the processor chip contains the executable program. The size of this memory is 128kB and 64kB for the 18F8722 and 18F4620 processors, respectively. Data RAM on the processor chip is used for working variables, program stack, etc. The size of this memory is 3982B for the 18F8722 and 18F4620. There is a non-volatile data EEPROM on the processor chip, which, in v1.00 of the program, is only used in the CPU board processor, to store a count and cause of any processor resets. It is 1kB on the 18F8722. Finally, there is an external serial EEPROM on the CPU board which is used to store configuration parameters. The size of this memory is 64kB, and the board has provisions for adding a second chip to double the memory.

The hardware supports changing the contents of all of these forms of memory under program control. The software version, v1.00 does not support modification of program ROM (with the exception of the table which translates command names to macro numbers and another table contains telemetry sound sequences) – an external programmer is required to do this.

Memory usage for the CPU board is as follows:

Program ROM	Executable code Library of sound sequences Command table Macro name command table

On-chip data EEPROM	Reset counters
On-chip RAM	Working values of configuration parameters Program working variables, stack, etc.
Serial EEPROM	Stored values of configuration parameters Stored values of unlock codes Stored and working table of macros Scratchpad used to modify macro name command table

In the Radio Control Boards, executable code and the library of sound sequences are stored in program ROM, as for the CPU board. Other parameters, such as digital pots settings, tone definitions and amplitudes, and ID frequency, speed, and amplitude are downloaded from the CPU board when the system is reset. Most of the parameter set functions change the CPU board stored values, but do not download the values to the Radio Control cards (digital pot setting is an exception, as it must be interactive). Therefore, if the ID is changed, the system must be reset for it to be effective. This behavior may be changed in a later version of the software.

There is space in the serial EEPROM for storing the working copy and two backups of the configuration parameters, unlock code table, and macro table. The parameter backup and restore commands are used to copy the data from one of these copies to the other.

Configuring the System

Parameter Storage

Most parameters defining operation of the controller are represented by global variables of the CPU board program. They are stored in a serial EEPROM from which they are loaded on booting the program. The EEPROM contains three copies of the controller configuration parameters, designated working copy, backup, and deep backup. The working copy is generally synchronized with the RAM copy of the parameters. Some parameters can be set with an individual command to the CPU board via the RS232 serial port or DTMF received by one of the radios. These commands check parameters and most set both the RAM and EEPROM working value. Other parameters are set by writing directly to the corresponding address in the serial EEPROM. The working copy can be copied to and from the backup copy, the backup copy can be copied to the deep backup copy and the deep backup copy can be copied to the working copy, via DTMF commands. Macros are stored only in the serial EEPROM, and not in microprocessor RAM, except as they are being executed.

The commands for changing some specific parameters do input range checking and change both RAM and eeprom, so the effect of the command may be determined immediately. Specific commands are not provided for changing many of the configuration parameters. There are commands to change the value of any address in EEPROM directly. This should not normally be invoked manually from DTMF, as a mistake can corrupt the EEPROM. When used over RS232 from a controlling program on an attached PC, these commands provide a mechanism for efficiently configuring all parameters of the system.

Some parameters are stored differently.

1. The command translation table, which translates a sequence of DTMF characters into a 16 bit internal command, is stored in program ROM. There are two such tables in the system. One contains all of the built-in, published command names. The other contains the names used to execute macros. The macro name command table can be changed in the field.
2. Macro strings (the actual macros that are executed when a macro names is entered via DTMF or the serial port) are stored in the serial eeprom. They are included in the backups.
3. A tone sequence library is stored in program ROM. Pointers to the tone sequence library for particular telemetry sequences are stored in RAM and serial eeprom.

Initialization

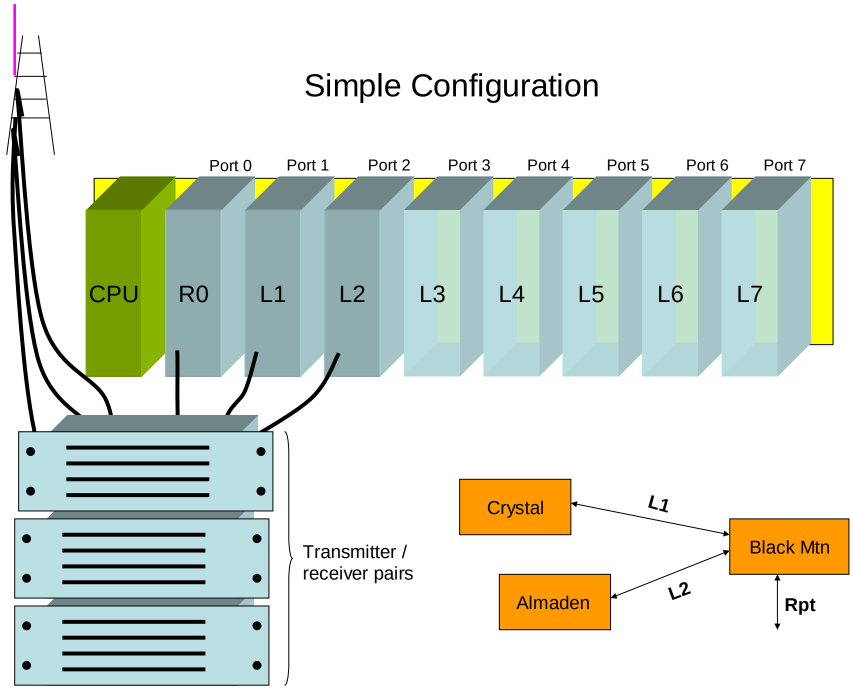
Default configuration parameter values are built into the CPU board program. The first time that the program is started after it is written to the processor chip ROM, it will load the default parameters into the serial EEPROM and into RAM. On subsequent resets of the system, the processor checks for a configuration valid byte in the EEPROM, and if it is correct, the configuration parameters are loaded from the serial EEPROM. A reset is caused by power cycling, hitting the reset button on the CPU board, by issuing the C000 command, or by the watchdog timer built into the processor detecting a program hang-up. A reload of the hard coded defaults can be forced on the next reset by resetting the EEPROM configuration valid byte. There is a command, C100, to do this.

The default parameter load takes about 12 seconds. A beep-beep-beep confirmation is sent to port 0 when this is complete. If power is interrupted during the default load, it will start over again on the next reset. Subsequent resets take less than one second.

Any new program load will overwrite the configuration parameter working copy in serial EEPROM. If the program revision is a minor one with no changes to the EEPROM parameter memory layout, then previously configured parameters can be restored by using the restore from backup command. For more significant program upgrades, this will not work. Therefore, a site manager should keep a record of changes made to the default configuration so that they can be quickly restored.

Basic Configuration

The default configuration is described in Appendix 1 of this document. It configures any system of 1 to 8 ports with the first port (physical port 0) defined as a local/repeater, and all of the other ports as links. If this is satisfactory for your site, only a few things are required to get started.



Unlock Code Definition

The initial program default defines only one unlock code, and that is the super-unlock code in unlock code number 0. The default code is 138065. The first thing that should be done to configure the system is to change the default unlock code and create some new ones.

There are two ways to set unlock code 0. The first is to use the command, C115. This command will not execute unless the current unlock code 0 is entered first, so to set it to a new value of 188767, the following could be entered by either DTMF transmitted to a receiver attached to Port 0 or by using the serial port:

```
*138065
*C115188767D
```

If a computer is attached to the serial port on the CPU card, the *'s are not necessary. The D is not required in either case, as long as the unlock code is the maximum of 6 digits long. For the serial port, the D can be replaced by a carriage return. These details are described in the command list document. The command processor is not case sensitive, so either c115 or C115 will work.

The second way to set unlock code 0 does not require any knowledge of the current code, but can only be accomplished if you are physically at the controller. Pull all Radio Control boards from the card cage, attach a computer to the serial port, and use the C114 command:

```
C114188767<CR>
```

It is recommended that the unlock codes be 6 digits long. The controller cannot distinguish between an unlock code and any other command except by the characters entered. The command processor does not use a command terminator character to separate one command from another. Therefore, no command or unlock code can be a prefix substring for any other command or unlock code. None of the commands in the default command table or default macro name table begin with 10 through 18, so unlock codes beginning with 1, except 19 are safe to use. When modifying the macro name command table, it is possible for the site administrator to create commands with any name, so caution must be exercised to avoid conflicts between commands or unlock codes.

Next, set unlock code 1, which is also a super-unlock code, and at least one additional unlock code, which will not be a super-unlock code. The command to set an unlock code is C110, and it will not execute unless a super-unlock code is entered first. This command places codes into the unlock code table in the lowest numbered available slot in the unlock code table. Since there is no unlock code in slot 1 at this point the next code will go there, and define the second super-unlock code. So, to define the second super-unlock code as 176545, and the unlock code number 2 as 123456, enter the following:

```
*198767
*C110176545D
*C110123456D
```

After this, don't use 188767 (unlock code 0) any more. It is only needed to change itself. Because anyone with a receiver and a DTMF decoder can tell what you are doing over the air, it is good practice to minimize use of the super unlock codes, particularly unlock code 0.

Note that once you enter an unlock code, you don't need to reenter it before every command that requires one. The system will remember that you entered an unlock code for 5 minutes, and that 5 minute timer gets reset every time you enter a new command. If a command comes in from a different port from which the unlock code is first entered, the unlocked state is reset and the code will need to be reentered. The site normal and remote normal commands also reset the unlock state.

ID and Location Configuration

The call sign ID should be set next. This is easy from the serial port, and cumbersome from DTMF. To set it from the serial port, use the S116 command. When the controller turns on the transmitter just to send an ID, it starts the CW at the same time as it pulls down the PTT line to the transmitter. Most transmitters take some time to come up, so this can clip the leading dit or dah of CW unless a space character or two is inserted at the beginning of the ID. A space character at the end is also recommended. So, to set the ID to " kj6k ", you would enter:

```
S116  kj6k <CR>
```

The CW generator is not case sensitive. Either kj6k or KJ6K will work.

From DTMF, a two key sequence is used to define each character in the ID, using the C116 command. A table defining the sequence for each character is included in the command list document. The same ID is entered with:

```
*C11601015251605201D
```

It is nice to define a sequence of characters describing the location. This can be returned with the C313 command (or with a macro which calls this command). This is done in the same way as for the ID, with the S122 and C122 commands. To set it to "BLK" use one of these two sequences:

```
S122blk<CR>
*C122225352D
```

Prefix Configuration

When controlling from a link radio, the command is not prefixed with *, but with a prefix code. This is necessary so that this controller can tell that the command sequence is destined for it and not for some other controller down the chain of links. The default prefix is "00". The appropriate prefix must be coordinated with the other site managers in the system of linked repeaters, otherwise nothing works. To define the site prefix, use the S120 or C120 commands in the same manner as for setting ID, above. To define the prefix as "98", enter either one of these two sequences:

```
S12098<CR>
C1209080D
```

The prefix can be up to 4 characters long. Any DTMF character can be used within the prefix except *. To use the prefix, it must be preceded with a # character – this tells the controller to start looking for a prefix.

Level Adjustment

Levels for the audio and telemetry should be set next. There are many ways that this can be accomplished. The procedure here is recommended to make sure internals levels within the controller are within the dynamic range of the circuits, and allows the default relative telemetry tone and CW levels to give the desired result.

This procedure requires a calibrated signal generator to produce a signal with 1kHz tone modulation of known deviation, and a calibrated deviation meter. If this equipment is not available, please contact Sierra Radio Systems for an alternative adjustment procedure.

This procedure will properly adjust for most component and radio gain variations. Any small channel to channel gain variations in the audio mixers on each Radio Control boards cannot be compensated for. The resistors used in the mixer are 1% resistors and the switches have very low on resistance compared to the circuit impedance, so this variation should be negligible.

All except the first step of this procedure can be accomplished remotely. The level in the first step is not critical, so if necessary, the default pot value can be used.

Set the CPU telemetry pot and backplane level

The nominal voltage for full deviation audio (most often 5 kHz) on the backplane audio bus is 1 Vp-p.

Telemetry tones generated by the processors on the CPU and Radio Control boards can have digitally defined levels of from 0 to 255. The scale is linear, with 0 giving no output. The 4 possible simultaneous tones are generated with two tones generated at each of two PWM ports on the processors. The peak sum amplitude from either of the two generators on one port must not exceed 255, so the normal maximum single tone amplitude is 127. By default, the telemetry and DTMF tone sequences are premphasized by the tone generator software. The internal tone level for a 1000 Hz tone is equal to the configured value. The relative amplitude of other frequency tones is proportional to the frequency. If you change the default level of the various telemetry tones, take care to insure that the total amplitude of multi-tone sequences (eg., DTMF) does not exceed 255 after pre-emphasis.

Turn on the CPU board test tone generator. It defaults to 1kHz at an amplitude of 127. This amplitude should produce one-half system deviation (eg., 2.5 kHz in a 5 kHz system).

```
*C2208
```

Other frequencies can be used for this calibration. The test tone generator is not pre-emphasized, so the same deviation will be achieved, independent of the test tone frequency chosen for the calibration.

The nominal backplane amplitude for all audio signals is 0.5 Vp-p for one-half of system deviation.

Adjust the CPU board telemetry pot to produce a 1kHz tone amplitude on the cpu telemetry line on the backplane of 0.5 Vp-p. Either command C223 or C224 can be used to do this. See the command list documentation for details. The pot setting should be close to 118, which is the default.

```
*C22383118D
```

Adjust the transmitter level pots

Leave the test tone on. Next adjust the transmitter level pots on each of the Radio Control boards to produce a ½ full deviation level (2.5 kHz).

```
*C223r1nnnD
```

Adjust the receive level pots

Turn off the test tone (`*C2208`). For each port, do the following. Use a service monitor or other calibrated source to deliver a ½ system deviation 1 kHz test tone to the port's receiver. Adjust the receiver gain pot on the Radio Control board to produce a ½ system deviation deviation output from the any of the previously calibrated transmitters. To be consistent, it is probably best to use the port 0 transmitter for all of these operations. Altneratively, you set the backplane amplitude for the receiver's audio to 0.5Vp-p, as was done to adjust the telemetry level.

```
*C223r2nnnD
```

Modified transmit and receive level adjustment for increase link dynamic range

Links can carry mixed audio from multiple sites. In addition, the link signal to noise should be high compared to typical mobile or handheld input to a repeater. Therefore the nominal link deviation can be set to a lower value allow for additional dynamic range. This is straightforward using the procedure above. If the link system deviation is

4 kHz, just use 2 kHz for the transmit and receive deviation when adjusting the link receivers and transmitters.

Adjust the Radio Control board tone generator pots

For each Radio Control board, turn on the transmitter and the test tone for that board with 1kHz and level 127. Adjust the tone generator level pot to deliver ½ full deviation (2.5 kHz) on that transmitter.

```
*C223r3nnnD
```

Individual telemetry tone sequences and CW levels are set digitally and will provide consistent, well defined, deviations following the above procedure. The default values (118) should be acceptable in most cases, so try them out before attempting to change the individual tone values. Please provide feedback to Sierra Radio Systems on the default tone levels after the digital pot calibration. Future version defaults will reflect the best compromise from that feedback.

Tone generator pre-emphasis

The controller CPU board and Radio Control board hardware do not do any pre-emphasis or de-emphasis in main audio paths or in the tone generator audio path. Most site configurations will interface to the radios without de-emphasis in the receivers or pre-emphasis in the transmitters. Therefore, the audio within the controller will be pre-emphasized, as it was first pre-emphasized in the transmitter that is the original source for the traffic (eg., a user's HT). The rest of this discussion assumes that pre-emphasized audio is being carried in the controller audio paths.

The audio feeding the 8870 dtmf decoder on the Radio Control boards is de-emphasized in the dtmf pre-filter provided on the board. The chip will accept twist (high tone to low tone amplitude difference) of up to 10 dB, so this should be acceptable even if previously de-emphasized dtmf is fed to the board.

The telemetry sequence generators on the CPU and Radio Control boards are optionally pre-emphasized directly in the software sound sequence generator. If the `preempflg` configuration variable is set to 1 (the default), then pre-emphasis is applied to tone sequences. If it is set to 0, then pre-emphasis is not applied. This can be changed by directly modifying its value in the serial eeprom and resetting the controller.

The test tone (commands C219 – C222), CW generator, and command parameter prompt tone directly drive the tone generators, without using the tone sequence generator, so no pre-emphasis is applied. Pre-emphasis is applied (if the pre-emphasis flag is set) to all of the other telemetry tones, which are generated by the tone sequence generators. If no pre-emphasis is applied, the tone amplitude parameter will be directly proportional to deviation out of the fm transmitters, independent of the frequency of the tone. For a given tone amplitude parameter, the deviation for pre-emphasized tones and non-pre-emphasized tones will be the same at 1 kHz. The pre-emphasized tone amplitude is proportional to frequency for all frequencies below 2441 Hz. The amplitude is undefined above 2441 Hz, so those frequencies should not be used in tone sequences. The standard 750 us time constant is not respected, so tones below 212 Hz will have an excessively small deviation. The actual amplitude sent to the tone generator for a specific tone in a tone sequence is the product of the tone sequence amplitude and the frequency divided by 1 kHz. If this would result in an individual tone amplitude exceeding 255, then 255 is used. The sum of the amplitudes of the two tones in a dtmf tone must not exceed 255, or the tone generator will overflow, producing strange sounds. This is not checked in the controller software. Therefore, the maximum tone amplitude for any dtmf tone is given by the formula, amp*(f1/1000+f2/1000)<=255. Maximum values for dtmf tones are given in the following table:

dtmf char	low f (Hz)	high f (Hz)	max amp
1	697	1209	133
2	697	1336	125
3	697	1477	117
A	697	1633	109
4	770	1209	128
5	770	1336	121
6	770	1477	113
B	770	1633	106
7	852	1209	123
8	852	1336	116
9	852	1477	109
C	852	1633	102
*	941	1209	118
0	941	1336	111
#	941	1477	105
D	941	1633	99
Dial tone	350	440	255
Busy tone	480	620	231

Remember that there will often be more than just a dual tone being transmitted at once. It can be overlayed with an additional tone, or some audio at the same time. To prevent overdeviation, the amplitude of a given dual tone should be kept below the 255 maximum, if the system is calibrated for that to be at the maximum system deviation. Also, a dual tone of a given amplitude will sound twice as loud as a single tone of the same amplitude, and will have twice the peak amplitude. Finally, pre-emphasized dual tones of the same amplitude setting will have the roughly the same audible volume level (without adjusting for the frequency dependence of human hearing).

Given the above, recommended amplitudes for most single tone key up and command telemetry sequences are 127 (1/2 full deviation), most dual tone telemetry (dial tone and busy tone, for example) are 85 (same audible amplitude).

Complex Site Configuration

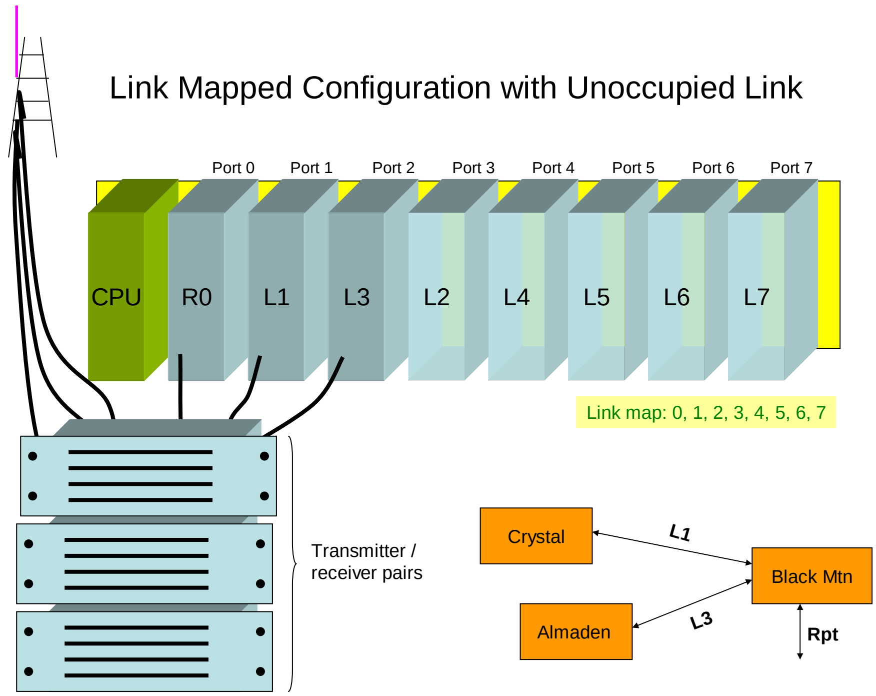
Differences between port types

The table below describes the differences in behavior between ports configured as repeaters, links, irlp nodes, and remote bases. Some of these are just the default behavior that is created when the port type is defined, and can be changed, either by port type or by individual port. Others (such as ability to accept commands) are built into the programmed behavior of the port type. See the SRS Controller Software Configuration spreadsheet for a detailed description of all of the parameters which can be changed.

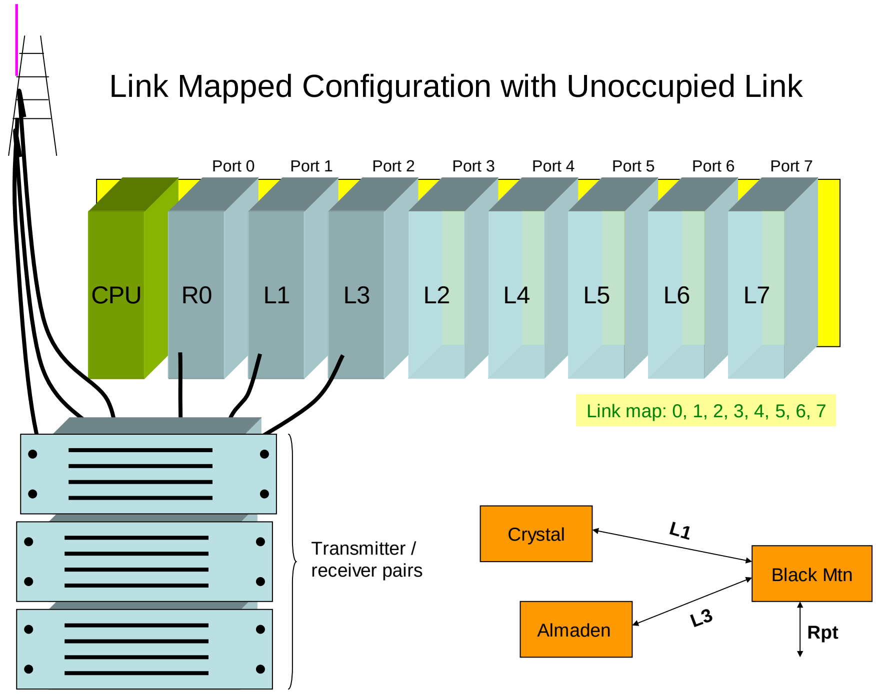
	Repeater/ local	**Link**	**VOIP link**	**IRLP**	**Remote base**
Retransmits input on output	x				
Accepts commands preceded by *	x				
Accepts commands preceded by #prefix		x	x		
Transmits ID	"polite" algorithm	"link ID 2" algorithm			
Default CW ID frequency (Hz)	1064	1064			
Default CW ID speed (WPM)	20	24			
Default ID level (0..255)	80	40			
Default transmit carrier off delay (s)	4.0	4.0 0.1 with link delay off	Same as link	0.1	0.1
Default COR/PL recognition delay (ms)	50	50	0	0	50
Default COR/PL drop recognition delay (ms)	150	5	0	0	150
Default PL status	PL on	PL on	PL off	PL off	PL off

Link map

The physical location of a Radio Control board in the card cage determines the address which the controller's software uses to direct communicate with the card. In most cases, the site should be set up so that this address corresponds to the identity by which users of the system will control a particular port. In the simple configuration setup described earlier in this document, the repeater is connected to the Radio Control board in the first slot to the right of the CPU board. The program thinks of this as port 0, and any commands directed to change something on a specific port would use 0 as the parameter to direct the command to the repeater port. The Radio Control board to the immediate right of the repeater port, is port 1. Link on/off commands would control that port by using the parameter of 1. So from left to right, the repeater is attached to the first Radio Control board, and is referred to as port 0. Link 1 is attached to the second Radio Control board, Link 2 is attached to the third Radio Control board, etc.



There will be situations in which the physical address of a Radio Control board (which is also defined by its physical position in the card cage), cannot be same as the identity that a user will want to refer to it. For example, a site may be starting out with a three port controller, with 1 repeater and 2 links. The eventual goal may be to grow this site to 4 links. In that case, it would be desirable to define which link will be link1, 2, 3, and 4 when the site is first set up, so that the designations will not change when new ports are added. If the first 2 installed links are not Link 1 and Link 2, then you need to define a link map to translate Link 1 and Link 2 to the physical port address where the links are attached. If there is at least one repeater at the site, one of them should be attached to port 0. The reason for this is that during command entry from the RS232 serial port, CW telemetry is also directed to port 0.



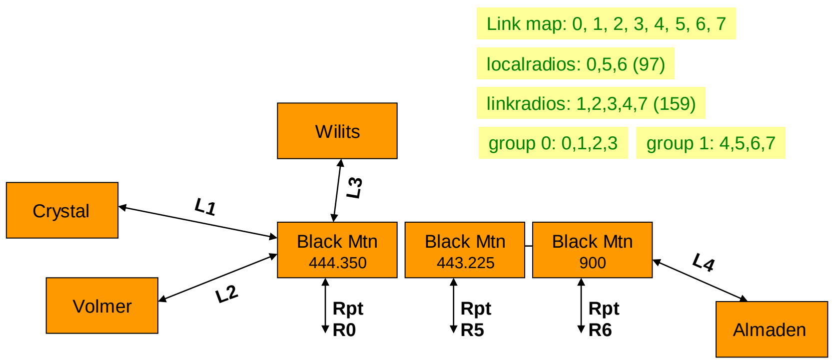
The link map describes a mapping between the logical designation for a port, and the physical location of that port in the controller card cage. The first element of the link map is the physical port for logical port 0, the second is the physical port for logical port 1, etc. So if the repeater is connected to the Radio Control board in the first slot (port 0), Link 1 is connected to the board in the second slot, Link 2 to the third slot, etc., then the link map will be 0,1,2,3,4,5,6,7. This is the default. If you start with a three port system consisting of a repeater, and links 1 and 3, with the intent of later adding link 2, then you would attach the repeater to port 0, link 1 to port 1, and link 3 to port 2. Then, you should define the link map as 0,1,2,3. Later, when you add link 2, you can put it in the 4th slot (port 3) and no additional configuration is required.

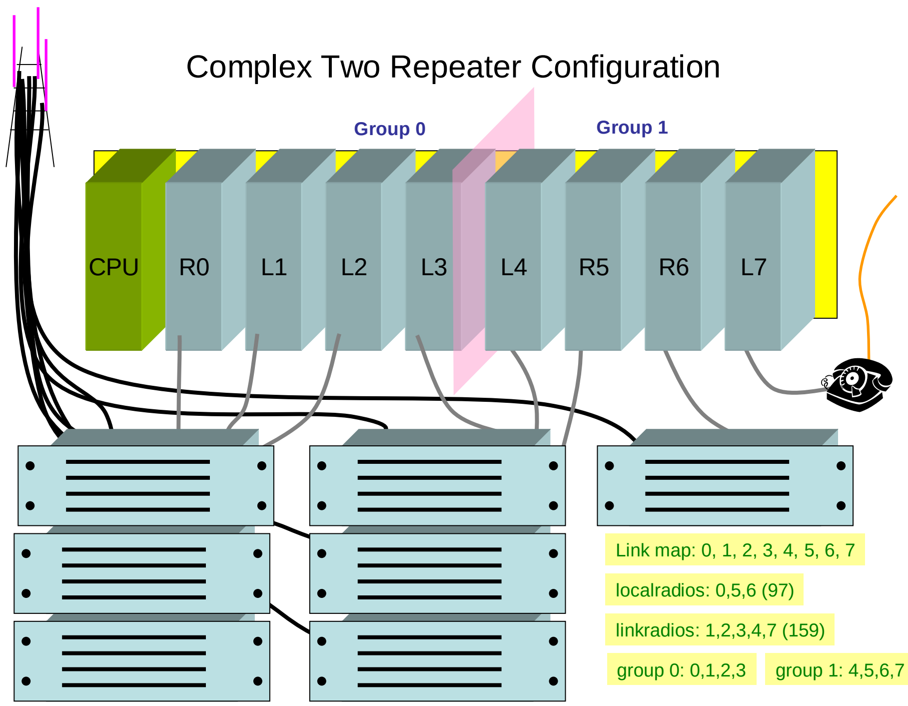
Multiple Repeater Complex Configuration

The controller is capable of managing multiple, semi-independent repeater systems. To do this, the ports for each system are grouped into a separate "group." A controller can be configured with a maximum of 5 groups, numbered 0..4. Group 4 is used for temporary, dynamic reconfiguration of the groups via the Create Linked Group (C212) and Restore Groups (C213) commands, so only groups 0..3 should be used for the site configuration.

To provide a level of command isolation between groups, most commands apply only to the ports that are in the same group as the port from which the command originates (commands from the serial port apply to group 0). There is a separate command which can be used to explicitly change the group to which commands apply, so that any command can be applied from any port.

Complex Two Repeater Configuration





Detailed Tone and Tone Level Control

The amplitude of individual telemetry tones and CW can be adjusted after calibration by setting the digital amplitude used by the tone generator. There are no specific commands to make these changes; the parameters must be changed by directly modifying the parameter value where it is stored in the serial eeprom. The addresses for all of the controller configuration parameters are listed in the SRS Controller Software Configuration spreadsheet.

As an example, lets change the amplitude of the CW ID for link ports. The default value is 30. If the pot calibration procedure described above has been used, with a system peak deviation of 5 kHz, then this tone will have a deviation of approximately 30/255*5 kHz = 588 Hz. If this is still not loud enough to be clearly heard, and a 1064 Hz notching is used at the far end link receivers, then you might want to readjust this to something like 900 Hz. To do that, the value should be 255*900/5000 = 46. The software configuration spreadsheet shows that the eeprom address for the link ID CW amplitude is 253 (version v1.00a software). To read the current value, use the read eeprom byte command, defined in the SRS Command List document as C107. After entering an unlock code, you can verify the current value with *C107253D. This will return 30, if the value has not been previously changed from the default. To change it to 46, use the set eeprom byte command (C106):

```
*C106192B46D
```

The processor will then need to be reset (C000) for the new value to be downloaded to the Radio Control cards where the ID's are generated.

Telemetry tone sequence amplitudes are slightly more complicated to adjust. Each telemetry type is configured via a 3 byte tone sequence parameter. The first (lower address) two bytes of the parameter define a 16 bit tone sequence specifier. The 3^{rd} (upper address) byte defines the tone sequence amplitude. To change the "beep-boop" telemetry provided upon key up on a repeater port when there are links active and link delay is on, the "lnkintsnd" parameter needs to be changed. There are two forms of this parameter in the eeprom, a global default, which is used to simultaneously set the values for all of the ports, and individual values for each port. Within the lnkintsnd parameter, there is a 16 bit integer which is an index into a table of sound sequences, and an 8 bit integer, which sets the tone amplitude. To change the amplitude for all ports, use the C106 command to set the amplitude part of the lnkintsnd parameter (address 282 + 2), and reset the processor to load it into RAM, then issue the set rlnkintsnd command to transfer this value to all of the individual radio specific values, in rlnkintsnd. To set this amplitude (default is 127) to 80, the command sequence is:

```
*C106284B80D
*C000
*C231
```

The amplitudes can be individually tailored for each port by writing directly to the configuration parameter for the specific ports.

Command response telemetry, such as the "beep-beep" normally sent upon command completion is generated in the CPU card, so there is not a separate instance of each telemetry type's configuration parameter for each port.

Tone sequence specification

Telemetry tone sequences are defined using a 16 bit specifier and an 8 bit amplitude. If the most significant nibble of the specifier is zero, then the lower 12 bits specifies the tone sequence library index, as shown in the table above. If the most significant bit of the specifier is 1, then the lower 12 bits specifies a single tone burst frequency in Hertz. The duration of the tone is specified by the lower 3 bits of the most significant nibble of the specifier. The single tone specifier format is pictured in the diagram below:

16 bit tone sequence specifier for direct specification of a single tone burst

15															0
1	DUR			FREQ (Hz)											

DUR	time (ms)

0	40
1	60
2	80
3	100
4	120
5	160
6	200
7	500

More complex tone sequences are defined in a tone sequence table and referenced telemetry configuration parameters by using the specifier as an index into the tone sequence table. The tone sequence table contains 32 user definable entries, and an additional 32 pre-defined entries.

If the most significant nibble of the specifier is zero, then the lower 12 bits specifies the tone sequence table index. The table entries are shown in the large table on the next two pages.

16 bit tone sequence specifier for referencing a tone sequence table entry

<table>
<tr><td>15</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>0</td></tr>
<tr><td colspan="4">0</td><td colspan="12">tone sequence table index</td></tr>
</table>

The normal key up telemetry when everything is linked is defined by the lnkintsnd configuration parameter at EEPROM address 282. By default, this parameter points to the "beep-boop" tone sequence at index 34 of the tone sequence table. To change this to the alternative "beep-bop" at index 35, a 35 needs to be written to the 16 bit word at address 282 of the configuration EEPROM. This is done with the command:

```
*C108282B35D
```

To replace the "beep-boop" with a single 100 ms long tone of 500 Hz, the other format would be used with a parameter of 32768+3*4096+400 = 45456

```
*C108282B45456D
```

As in the amplitude example above, C000 and C231 commands must also be issued for this value to be transferred to all of the port radio control boards.

Tone Sequence table

library index	**tone sequence** (freq/duration, …, in Hz and ms)	**# iter**	**comments**
0..31	user defined, defaults to no sound		
32	no sound	1	no sound or time delay
0	1633/60, 150	1	"beep", key-up linked, no link carrier delay
1	1633/60, 50, 1209/60, 150	1	"beep-boop", key-up linked, with link carrier delay
2	1633/60, 50, 697/60, 150	1	"beep-bop", key-up link w/ delay alternate
36	697/60, 150	1	"bop", key-up, remote monitor mode active
37	941/100, 150 1336/100, 150	1	DTMF0, key-up remote base xmit active
38	1209/80, 80, 1209/80, 150	1	"beep-beep", command function complete
39	350/80, 80, 350/80, 80, 350/80, 150 440/80, 80, 440/80, 80, 350/80, 150	1	"beep-beep dial tone 3x", configuration function complete (EEPROM written)
40	1209/80, 80, 1209/18, 150	2	"beep-beep" twice, site normal command complete
41	350/1000 440/1000	30	dial tone, to confirm controller is prefixed
42	480/500, 500 620/500, 500	30	busy signal, on prefixing busy controller
43	950/50, 50, 950/50, 150	76	"repeating i's", on top of dial tone for link off
44	950/50, 150	151	"repeating e's", on top of dial tone for local off
45	2100/100, 1700/100	5	"razzberry", high pitch warble, command blocked
46	1200/100, 1700,/100	8	"razzberry", alternate longer 1.5s version
47	1633/60, 50, 1633/60, 50, 1633/60, 150	1	"beep-beep-beep", receive active timeout
48	700/100, 100, 700/100, 100, 700/100, 2000	15	repeating "boop-boop-boop", pending site normal timeout (not implemented as of 0098w)
49	480/100, 100 550/100, 100	5	fast pulsed dual tone warning tone for both bad command parameters and other command errors
50	1209/60, 50, 1633/60, 150	1	"boop-beep", not used
51	967/60, 1209/60, 1633/60, 150	1	"bdleep", not used

52	reserved		
53	reserved		
54	reserved		
55	reserved		
56	reserved		
57	reserved		
58	800/39, 13, 800/13, 39, 800/39, 13, 800/39, 13, 800/39, 39	1	CW “no”, 18 WPM, 800 Hz
59	800/13, 39, 800/13, 13, 800/39, 13, 800/13, 39, 800/13, …	1	CW “err”, 18 WPM, 800 Hz
60	800/39, 13, 800/13, 39, 800/13, 13, 800/39, 39	1	CW “na”, 18 WPM, 800 Hz
61	800/39, 13, 800/13, 13, 800/39, 39	1	CW “k”, 18 WPM, 800 Hz
62	800/13, 13, 800/39, 13, 800/13, 39	1	CW “r”, 18 WPM, 800 Hz
63	800/13, 13, 800/13, 13, 800/39, 13, 800/39, 13, 800/13, …	1	CW “?”, 18 WPM, 800 Hz

Controller Timing Parameters

There are many parameters which determine the timing and delays of various operations within the controller. Most of these are described in the SRS Controller Software Configuration spreadsheet. These parameters are set by writing directly to the corresponding addresses in the serial EEPROM and then resetting the processor to make the new value active.

Transmitter on and off timing

The detailed timing involved in turning the transmitters on and off are shown in the receive qualification, squelch and transmit timing appendix. It is determined by two basic variables. Turn on is adjusted by rcvqdelay, which should normally be short, but can be lengthened to reduce the triggering of the transmitter on noise (particularly important if PL is off) or from unidentified operator kerchunking. Turn off is adjusted by two rcvunqdelay and the transmit timeout variables. Each of these has several variants depending on the type of port (repeater, link, etc.).

Starting with code version 0.98w, there is an extended long transmit timeout capability. In a linked system with many hops, there can be a substantial delay between the time a user keys up his transmitter and the time that the transmitter at the far end keys up. This is particularly true if PL is used on RF links between sites. Under these circumstances, it is desirable for the link transmitter chain to remain active for an extended time – long enough for it to still be on after the conversation returns back to the person on the side of the path that brought up the links. This would be on the order of 30 seconds. Simply extending the long transmit timeout for links to this long would cause the long link on even when there is a short burst of noise, and kerchunk, or if someone just quickly transmits there ID and listens. The extended long transmit timeout provides a long timeout which gets triggered only if the repeater input receiver driving the link output remains active for a sustained period of time, defaulting to 5 seconds. In that case, for ports in which this feature is enabled, the transmit timeout switches to a longer value with a default of 30 seconds, instead of the normal 4 second default. Parameters used to configure this feature are longxtoacts, xmittimeouts_xlong, and xlongtoports. Commands to enable and disable the feature are C3480 through C3484. In addition, the normal commands can be configured to set this long transmit timeout to its default state or to leave it alone. See the details for lnknormdefs, localnormdefs, remnormdefs, lnkrstdefs, intnormdefs. The site normal commands always restore the normal state.

ID timing

Timing of identification also uses several parameters, and is different for repeaters than for link transmitters. See the ID section of this document for a detailed description of the ID algorithms and adjustment parameters

Controller Behavior Parameters

There are many other parameters to adjust the behavior of the controller. Many are switches to select between a couple of different choices on a particular behavior. See the SRS Software Configuration spreadsheet for details.

Macros

A macro is a list of built-in commands or other macros that is executed as if it is a single command. The commands that are made available for most users of a system are implemented as macros. There is space available in the serial EEPROM for storing 400 macros. Macros are executed by entering commands which are specifically defined to call specific macros.

Defining Macros

To define a macro, just enter the sequence of characters you would use to execute the same commands from DTMF or the serial port into the macro string. This is done with the set macro commands, C130 or S130.

As an example, if you want command 003 to turn link 1 PL off, then the command C3551 should be assigned to the macro which is called by the 003 command. Referring to the command reference or Commands/Macros section, above, this command executes macro number 3. Using the S130 command from the serial port, define the macro as follows:

```
S130 3 C3551<CR>
```

From DTMF, the command is:

```
C13030012330505010D
```

You can check that this worked correctly by using the get macro command, C131:

```
C1313D
```

An unlock code must be entered prior to using these commands. You should also verify that it works by executing the macro command:

```
003
```

User Entered Parameters in Macros

Some commands take parameters to define what the commands do. When there is only a small number of valid parameter values, user level commands are best implemented by creating a macro for each of the possible command parameter values. For example, the command to turn on a specific link port, C331 takes a single digit parameter to define which link is to be turned on. To turn on link 2, you would enter C3312. The command parameter prompt tone will come on when C331 is entered and turn off when the subsequent 2 is entered. For most installations, macros should be created specifically for each port. For example, a macro 012 could be defined to turn on link 2, and a separate macro could be defined for turning on link 3.

Commands to set a remote base frequency may need to allow a wide variety of possible parameters, so that creating a macro for each value is not feasible. In this case, if the command is to be called from a macro, there needs to be a mechanism for pulling DTMF characters from outside the macro itself. This is done be inserting the % character in the macro where the parameter is to be inserted. For example, if the macro called by the 012 command is set to C331%, then entering 012 will cause the prompt tone to come on and the link port number can then be entered. A subsequent command could also be included after this command within the same macro. There is no mechanism, however, for storing the entered parameter and using it in multiple commands within one macro. The % and subsequent parameter entry would need to be used for each command in the macro.

Setting Macro Execution Permissions

You should also set the macro permission to define the conditions under which the macro will execute. This is an 8 bit mask, entered as a decimal number, with bits defined as follows:

1	Block the command if COR and a required PL is not detected
2	Block the command if the port from which the command is being entered is not currently linked
4	Block the command if no local radio (repeater) is currently linked – if you want to be able to always hear if someone is controlling the machine while monitoring from the local repeater, this should be set.
8	Block the command unless an unlock code has been entered
16	Block the command unless a super unlock code has been entered (note, that macros can be changed with an unlock code, so this is not effective security, unless the base command being called by the macro requires a super unlock code).
128	Allow the command to be executed if there is explicit code in the command to allow it when other conditions are not met. This is used primarily for link on commands directed at the same port from which the command is coming – it should be allowed even if link is off for that port and link on for other ports would be blocked.

To set the permission to block the macro defined in the previous example if the command source port is not linked or the local repeater is not linked, set the macro permission to 2+4=6, with the command sequence:

```
C1323B6D
```

Telemetry During Macros

Command confirmation telemetry is suppressed during the execution of a macro (command response telemetry, returning some value via CW, works normally). The telemetry returned by the last command in the macro is not suppressed. There are also specific commands to generate the three types of confirmation telemetry generated by other commands (confirmation tone, configuration confirmation -- tone commands that write to the configuration eeprom, and the site normal confirmation tone). These can be placed at the end of a macro to force a specific type of confirmation telemetry.

ID

Automatic identification by morse code is provided for repeater and link ports.

The repeater ports use a "polite ID" algorithm for determining when to send the ID. There are a number of parameters which can be used to configure that behavior of this algorithm. The polite ID is designed to minimize the occurrences of an ID being sent on top of an ongoing conversation, while strictly conforming to the FCC identification regulations for the amateur service. If the transmitter comes on after a long period of inactivity, then the ID will wait until either the transmitter is unkeyed, or a moderate delay elapses. This delay is configurable as the id_politeto parameter, and defaults to 30 seconds. During an ongoing conversation, an ID will be sent on top of the conversation if the transmitter is on continuously for a maximum allowed period since the last ID. This period is configurable as the id_absto parameter, and defaults to 10 minutes. In order to minimize the chances of an ID being sent over an ongoing conversation, the ID will key up the transmitter and be sent during an idle period in the conversation (transmitter not keyed) when there is still some time left before the maximum allowed timeout period. This time remaining is configurable as the id_idleto parameter, and defaults to 2 minutes. If an ID was the last thing sent (transmitter not keyed up since the last id), then no more id's will be sent.

In linked repeater systems in which most conversations take place with between people who are not on the same local repeater, this polite ID is not the best algorithm to minimize interference with an ongoing conversation. In that case, it is better to send the ID over the top of the local input, and miss sending it over someone coming in from the links. Thus the polite feature does the wrong thing. The best configuration in that case is to set the id_politeto parameter to zero, and the id_idleto parameter short (a few seconds). Keying up the transmitter to send a final ID can also be disabled by setting the nokeyup4id configuration parameter to 1. Once a repeater port ID starts, it will always complete, holding up the transmitter if required.

There are several options for the link id. The basic ground rule of these options is that an ID will not key up the transmitter or hold up the transmitter to complete an ID. This prevents ID's from keying up multiple transmitters in a linked system when there is no immediate conversation taking place. Dropping transmit immediately after releasing PTT is important in cases where a simplex remote base is being controlled from a link. The interval between ID's is set by the same id_absto parameter used for repeater port ID. If ID times out while the transmitter is active, the ID will be fired immediately. If ID times out while the transmitter is idle, ID will fire following a short delay after the transmitter next becomes active. This delay is set with the id_lnkidleto parameter. The ID will be truncated if PTT is released and any tranmit hang delay completes before the ID completes. The ID timeout timer will not be reset in this case. The linkidtype parameter can be changed to force the timer to reset even if the ID is truncated.

DTMF Muting

Default Behavior

DTMF entering via a link port is not muted on going out to a link or IRLP port. DTMF entering via a repeater port is not muted on going out to a link or IRLP port unless it starts with a DTMF *. DTMF tones are muted in all other cases. In addition to this, incoming DTMF from a remote base port is blocked from entering the command processor.

Changing Default DTMF Muting Behavior

There are a set of configuration parameters which are used to define the normal DTMF muting behavior of link and IRLP port types. Each of the bits in these 8 bit parameters defines the behavior for one port.

To allow DTMF to pass through to a remote base port as if it were a link port:

Set the bit in the muteoutaslinkr configuration parameter which corresponds to the remote base port to 1. All bits of this parameter default to 0.

To allow DTMF from a remote base port to be muted as if it were coming from a link port:

Set the bit in the muteinaslinkr configuration parameter which corresponds to the remote base port to 1. All bits of this parameter default to 0.

To allow DTMF commands from a remote base port:

Set the bit in the allowdtmfcmdr configuration parameter which corresponds to the remote base port to 1. All bits of this parameter default to 0.

To mute DTMF to passing through to an IRLP port as if it were a link port (only * from a repeater port starts muting):

Set the bit in the muteoutaslinki configuration parameter which corresponds to the IRLP port to 0. All bits of this parameter default to 1.

To allow DTMF from an IRLP port to be muted as if it were coming from a link port::

Set the bit in the muteinaslinki configuration parameter which corresponds to the IRLP port to 1. All bits of this parameter default to 0.

To allow DTMF commands from an IRLP port:

Set the bit in the allowdtmfcmdi configuration parameter which corresponds to the IRLP port to 1. All bits of this parameter default to 0.

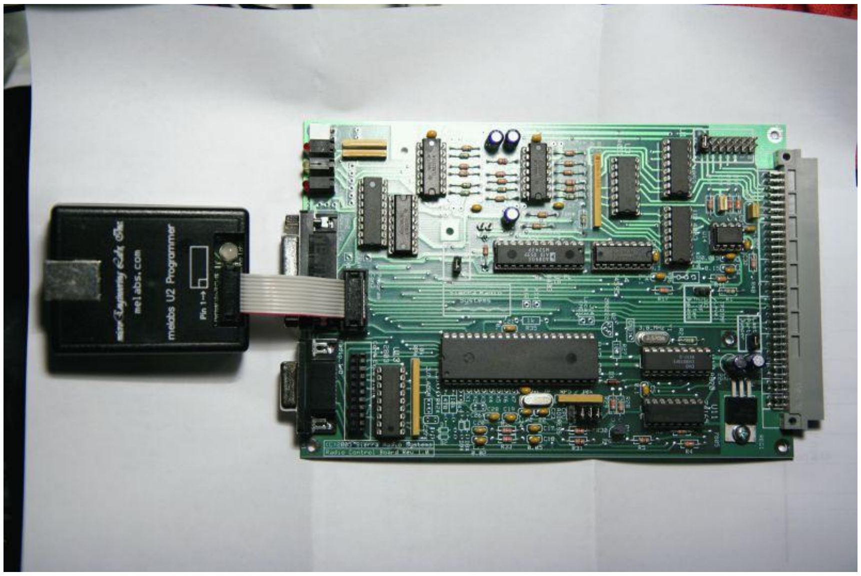
Temporarily Inhibiting DTMF Muting

There may be cases in which it is desirable to allow DTMF to pass out to a repeater port. One example would be to control a temporary IRLP/echolink node that is connected to a transceiver that is set to the repeater's frequency. This transceiver and the computer hosting the IRLP connection can be located anywhere in the coverage area for the repeater. The "disable DTMF muting" command, C253, can be used to disable DTMF muting for a particular port. This commands takes a single digit parameter that is the port for which the DTMF should be allowed. This will reset with a site normal or remote normal command, with the "cancel DTMF mute disable" command, C254, or after a 2 minute timeout. By default, muting remains enabled for a DTMF *. It can be changed to pass all DTMF by changing the nomuteopt configuration parameter from the default of 0 to 1. The 2 minute timeout value can be set to a different value by changing the nodtmfmtos configuration parameter.

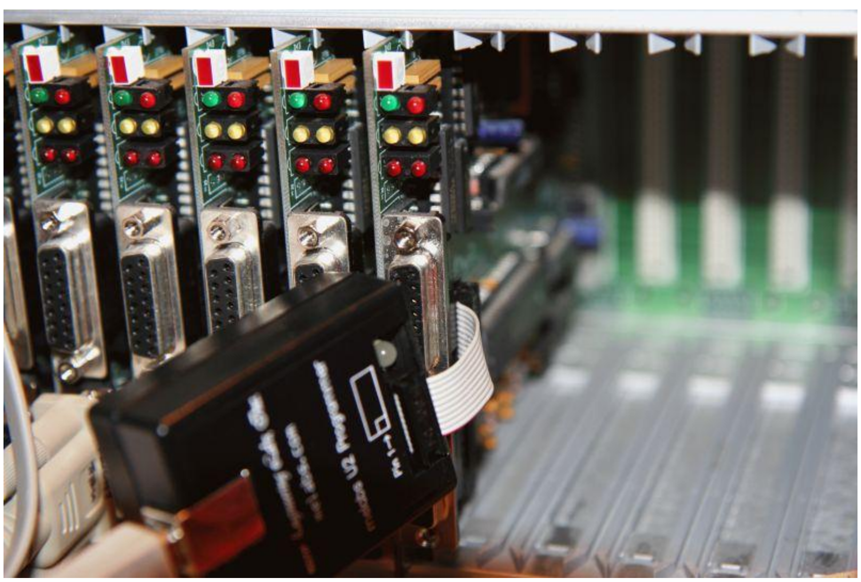
Appendices

Appendix: MELabs EPIC Programmer Setting

The ME Labs EPIC programmer should be connected to the programming port on the card to be programmed. With the connectors on the EPIC and CPU or RCB cards in the correct orientation, the board and programmer will lay out neatly side-by-side as shown in the next figure. The figure shows an RCB card. The CPU connector is higher on the board, adjacent to the reset switch.



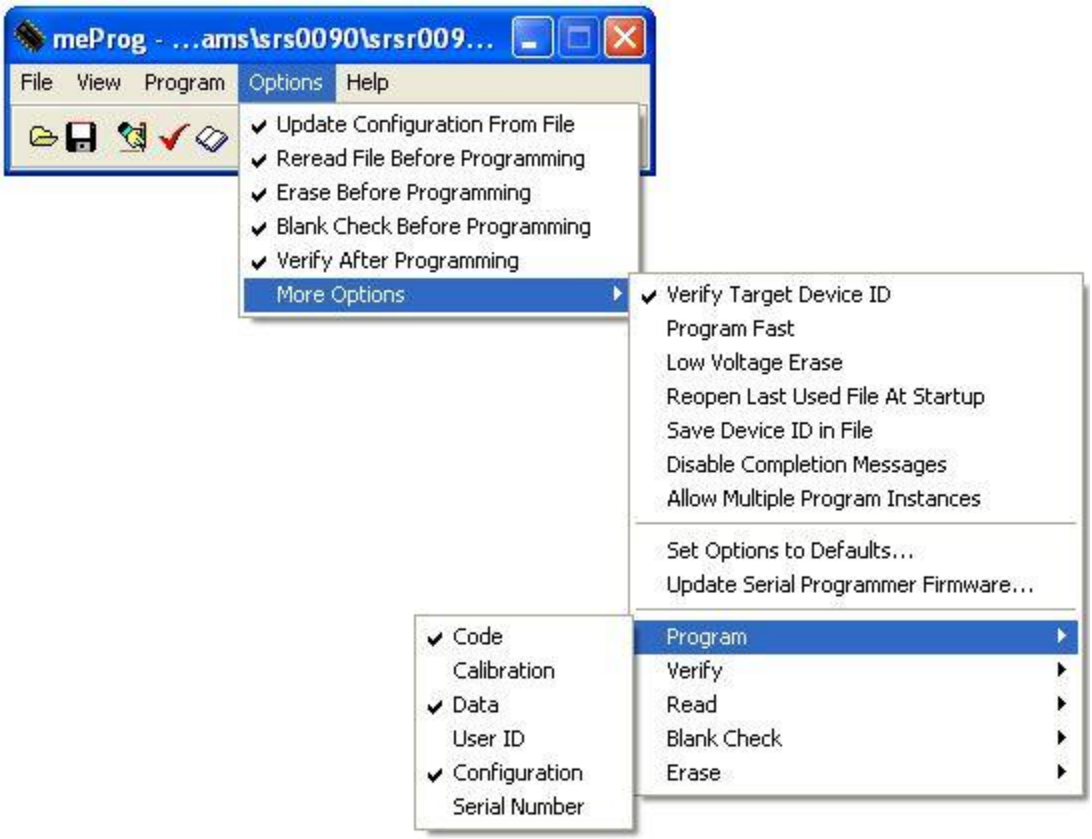
For programming, power must be applied to the card. This is most easily done with the card inserted in the controller card cage. When programming an RCB, the CPU card must be removed from the card cage.



The programmer must be configured to program the correct type of processor, and settings in the Options menu need to be defined to tell the programmer to pull configuration data from the .hex file, to erase the part before programming, and to program both the on-chip data EEPROM, and the program ROM.

The EPIC software version should be v4.20, or later. For programming the Radio Control boards with v1.0 hardware, the programmer should be set up to program the 18F4620 processor. The Melabs programmer software screen will look like this:

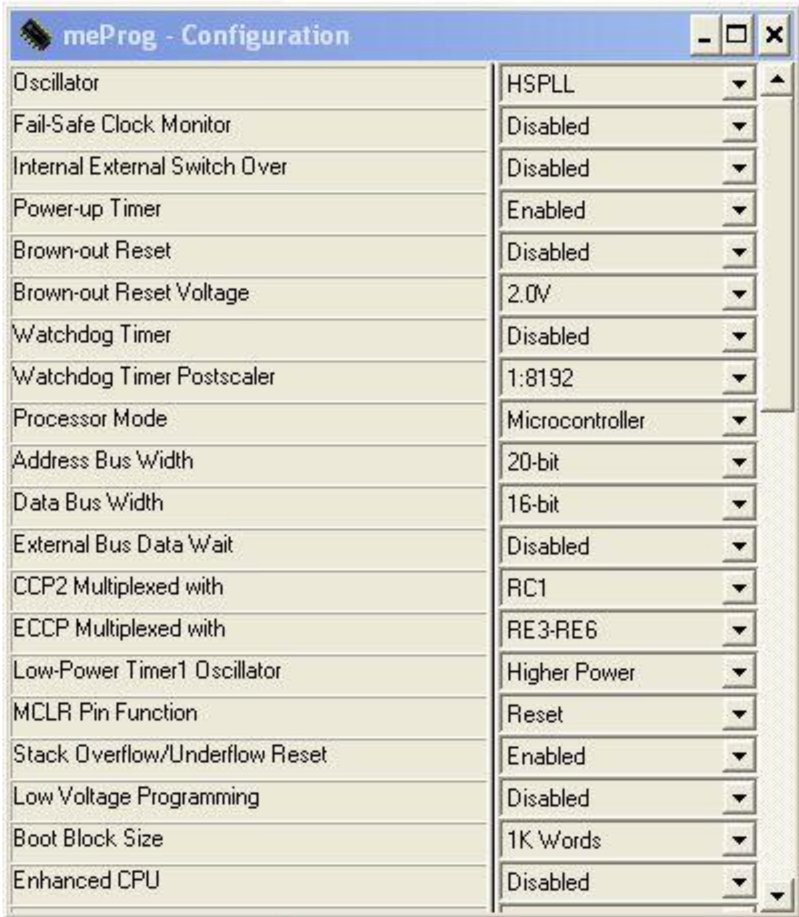
After setting the Options menu correctly, it should look like this:



After opening the Radio Control board software .hex file using the File menu, the program will show the first few items of the processor configuration as follows:

The CPU board must be pulled from the backplane while programming the Radio Control boards. This is because the CPU board can pull the the backplane reset line low to reset the CPU's on each of the Radio Control boards, thus interrupting the programming sequence for the Radio Control board. The Radio Control boards may be left in place while programming the CPU board.

To program the CPU board, the processor should be set to the 18F8722. The other options parameters should be set as for the Radio Control boards. After opening the program .hex file, the first items of the processor configuration should look like this:



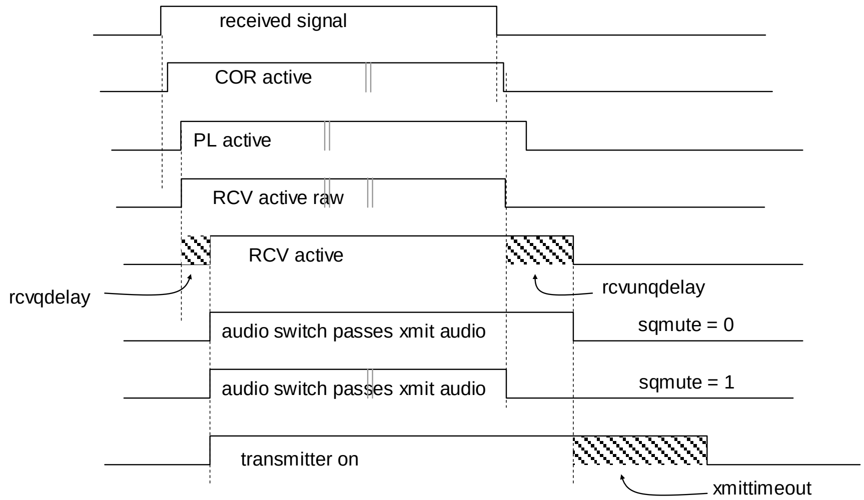
Appendix: Receive qualification, squelch and transmit timing

See the figure for a description of the timing of received signal detection, received signal squelch, and transmitter timing.

In most cases, the audio from the receiver is muted when no carrier is detected. In this case, the receiver muting via the audio switches is not critical in the controller. It's primary purpose is to avoid passing grunge or other audio which opens receiver squelch when PL is not detected. In that case, if the audio is not muted, and the transmitters are switched on due to either telemetry or input from another receiver, then the undesired audio would also be passed to the linked transmitters.

In other cases, the receiver and/or receiver interface provides COR and PL detect signals, but does not mute the receive audio. This means that the audio line from that receiver has a high level of broadband noise when there is no signal present. In this case, the switches in the controller radio card must provide the squelch function (even if only one receiver is on, the transmitter remains on for a few seconds after the incoming carrier is lost), and timing of the squelch operation is therefore critical to avoid passing high levels of audio nose.

Transmitter PTT Timing



It is desirable for the controller to have some mechanism for dealing with transient false COR or PL detect signals, which could be due to either noise, or short "kerchunking" of the repeater. This is implemented in the controller by requiring COR and PL to be present for a finite period of time before recognizing the received signal, switching on the audio and keying the transmitters. This delay is referred to as the receive qualify delay. A future direction may provide for adaptive lengthening of this delay to provide a more effective anti-kerchunk mechanism.

Several things happen when the received signal drops out. The audio switches are turned off to mute the audio from that receiver. The transmit key down delay is started, to turn off the transmitter after a delay of a few seconds. In the case of a local repeater input ("remote"), a key up telemetry delay is started to transmit key up telemetry after a delay of about a second. Incoming signals will often be marginal, due to rapid mobile flutter or weak signals from a distant handheld. It is desirable to avoid cutting off audio, transmitting key up telemetry, or cutting off the transmitters during a noisy transmission. This is handled by introducing a second delay after COR or COR and PL is lost before the controller recognizes that the received signal has dropped out. In the case of a link receiver, in which the transmitter and receiver are in a fixed location, with the link designed for high signal to noise, then this delay should be 0. In the case of a repeater input, it needs to be some fraction of a second, depending upon the behavior of the receiver COR, PL, and squelch circuits. This delay is referred to as the received unqualify delay.

Key up telemetry is sent following a delay after received signal from a local repeater drops out. An option is provided to abort the key up telemetry if the received signal comes back on before the delay timeout. This prevents momentary loss of signal during a mobile flutter from repeatedly sending key up telemetry, independent of the value of the receive unqualify delay.

Both delays may be set independently for each receiver.

A weak signal may not provide a reliable PL signal. If PL is on, this can cause the audio switches to mute the weak audio. The receive unqualify delay should be set to prevent this from happening, while still providing the protection from passing grunge or undesirable keying of the transmitter.

If the audio from the receiver is not squelched, a long squelch tail will be heard between the loss of carrier and the receiver unqualify delay timeout. There is an option to shut off the audio switch immediately on loss of carrier (squelch mute), to prevent this. Its undesireable side effect is that if the COR is not sufficiently sensitive, it may cut off the audio of a weak signal during fades. This option may be selected independently for each receiver.