Radio Repeater Controller

User Guide

Version 1.9

Luc Romain, VE2LRS

ve2lrs@gmail.com

July 2011

Table of Contents

Introduction

A radio repeater is a combination of a radio receiver and a radio transmitter that receives a low-level signal and re-transmits it at a higher level or power, so that the signal can cover long distances without degradation. A repeater controller interfaces with all the radios and is used to monitor and control the radios. When a radio receives a signal, its audio redirected to the transmitting radio through the repeater controller circuit and the controller activates the PTT[1] switch. This repeater controller can be used to link up to three radios and has a large number of features that enables it to adapt to various configurations.

The repeater controller has a DTMF[2] receiver, an audio amplifier, a temperature sensor and auxiliary ports. The main processing unit is a Microchip PIC16F690 micro-controller. The DTMF receiver listens for control sequences on the audio signal. This enables all functions in the repeater controller to be remotely controllable.

This document provides a functional overview of the the repeater controller. Usage information for all the functions is also thoroughly described in this manual.

1 PTT stands for Push To Talk.
2 DTMF stands for Dual Tone Multi Frequency

Controller Overview

A typical repeater installation is shown in Illustration 1. This setup shows a repeater controller connected to three radios. Radio A is the link radio and is connected to a Yagi directional antenna. The purpose of Radio A is to provide a radio link to a remote station. Radios B and C are connected to omni-directional antennas that broadcast the radio signals on the VHF and 6 meter bands. When Radio A receives a signal, its RX audio is sent to the TX audio of Radios B and C. When Radio B receives a signal, its RX audio is sent to the TX audio of Radios A and C. The same behavior occurs when Radio C receives a signal.

This repeater installation is an example of how the repeater controller can be used. This repeater controller is very flexible and can be configured to operate differently for each radio when a signal is received. Basically any radio can be connected to the repeater controller and used in the repeater station as long as the following signals are available from the radio:

Signal	Description
Transmission	
TX audio	Radio input signal to which the repeated audio is sent.
PTT	Radio input signal that activates the radio wave transmission
Reception	
RX audio	Radio output signal of the received audio
COR[3]	Radio output signal that indicates the reception on the tuned frequency

3 COR stands for Carrier Operated Relay (http://www.rac.ca/opsinfo/defi1.htm)

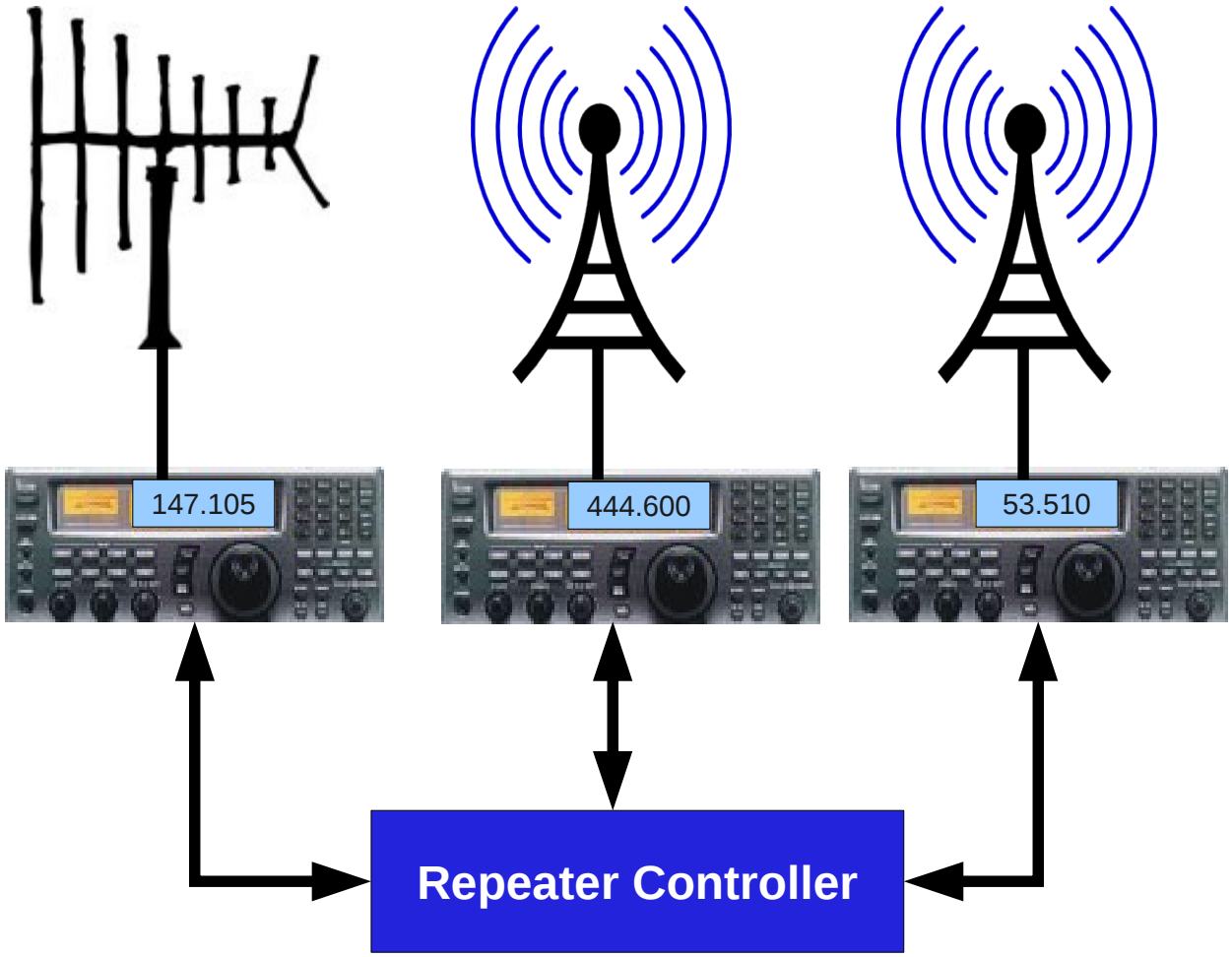


Illustration 1: Typical repeater installation

The repeater installation shown in Illustration 1 is made of a VHF link radio, a UHF and a 6 meter band radio. The UHF and 6 meter band radios are connected to omni-directional antennas. Depending on the signals that are connected between the radios and the controller, they can be configured as transmitting only, receiving only or used in reception and transmission. When one radio receives a signal on its tuned frequency, the repeater controller sends its RX audio into the TX audio of the two other radios (if they are connected) and ties the PTT signals of the radios to ground. Tying the PTT signals to ground activates the radio transmission and the audio received by one radio is re-transmitted into two different bands.

Schematic and PCB layout

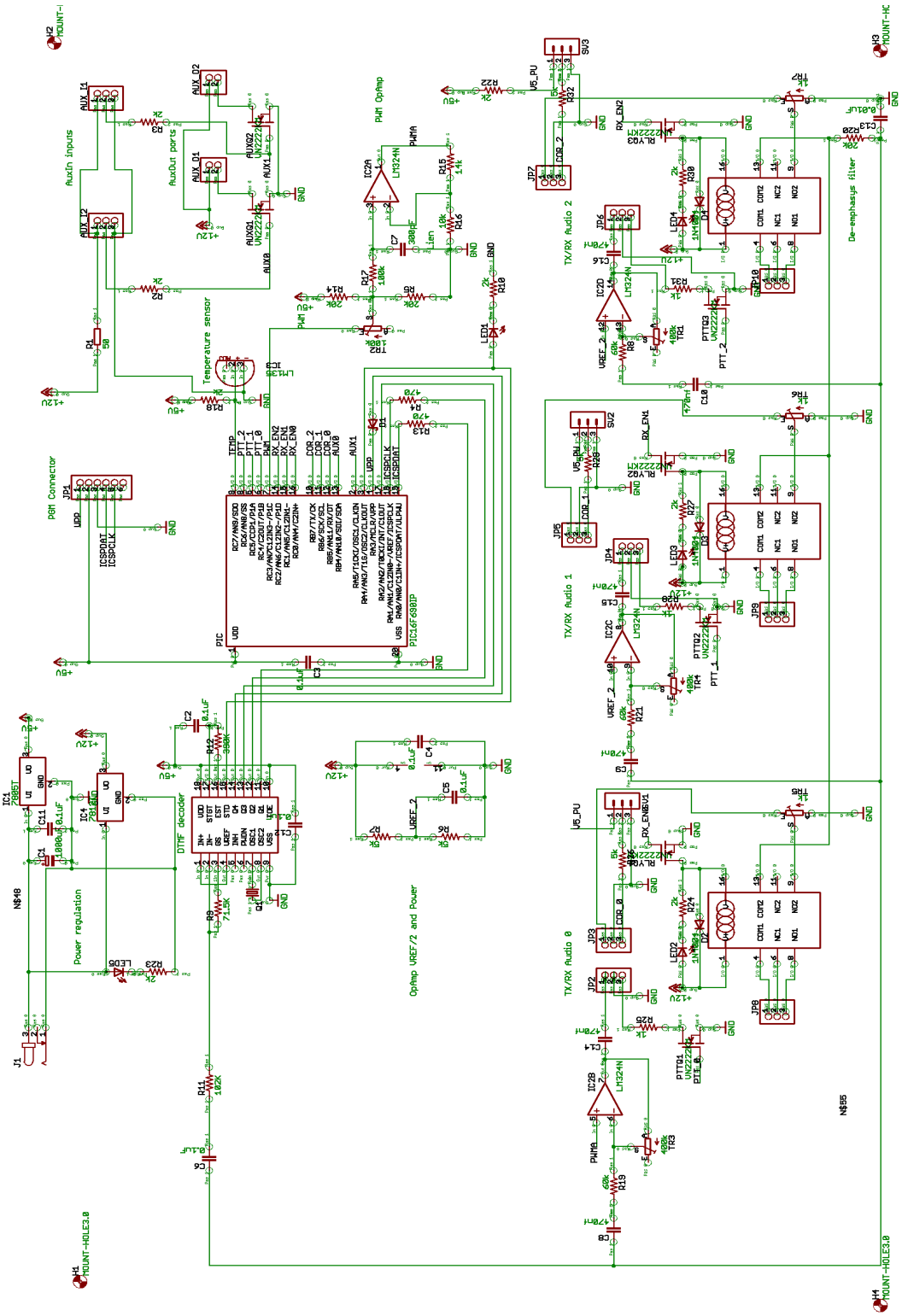


Illustration 2: Repeater controller schematic

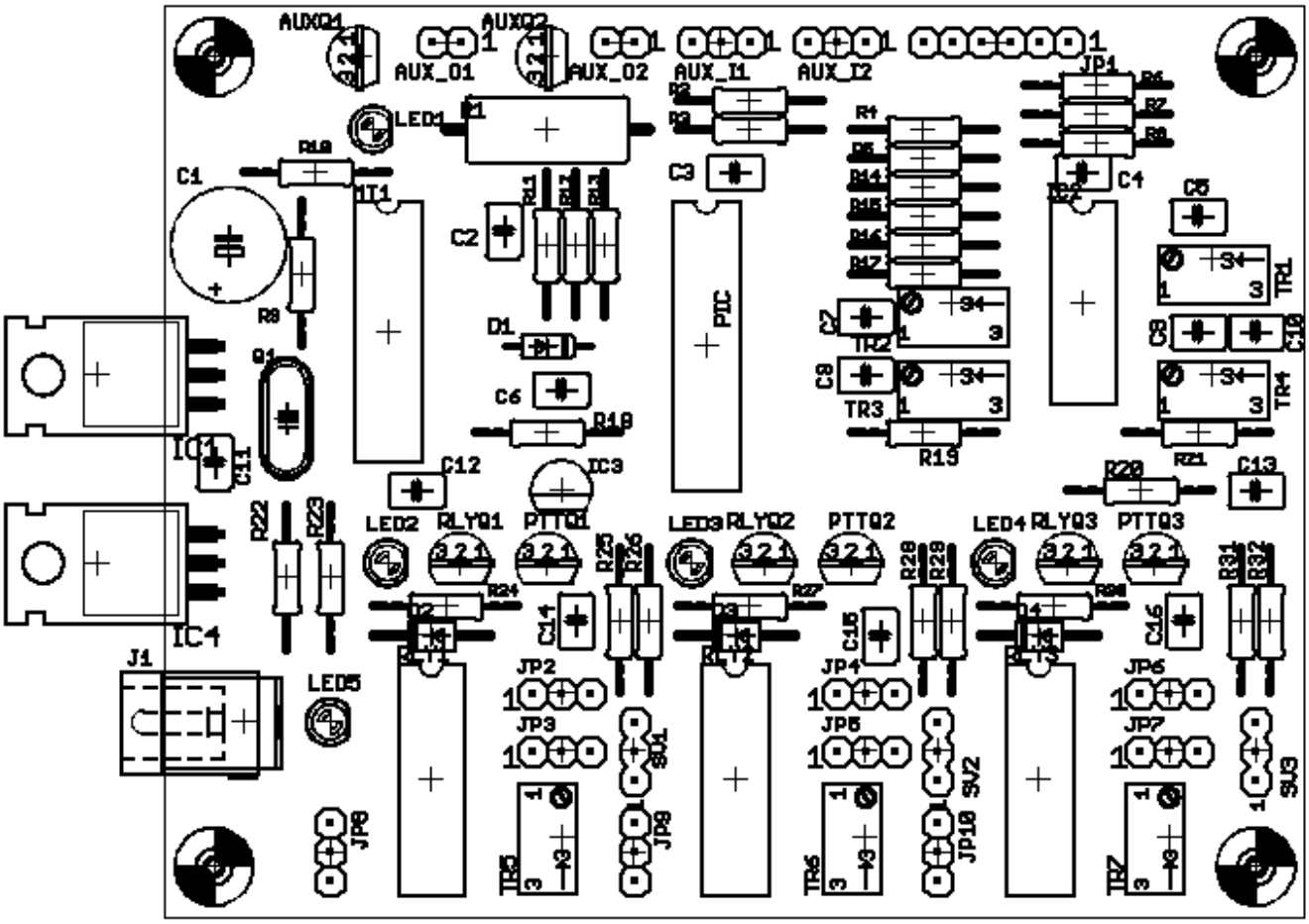


Illustration 3: Controller PCB layout

Power connector

The circuit board uses a standard 2.1mm circular connector to provide the electrical power to the system. The circuit must be powered with a 12.5VDC source. The circuit draws approximately 50mA in standby mode and approximately 200mA when a relay is activated. It is recommended to use a 3Amp fuse on the +12.5VDC power line near the connector.

Caution : The center connector must be connected to +12.5VDC and the outer connector is the Ground. Failure to observe this configuration will result in permanent damage to the components on the PCB.

Functional overview

Radio connections

This section describes the radio interface used to connect the radios to the controller board. There are three identical radio interfaces which are referred in this document as *Radio 1*, *Radio 2* and *Radio 3*. The pin out for the radio interface is shown in Illustration 4.

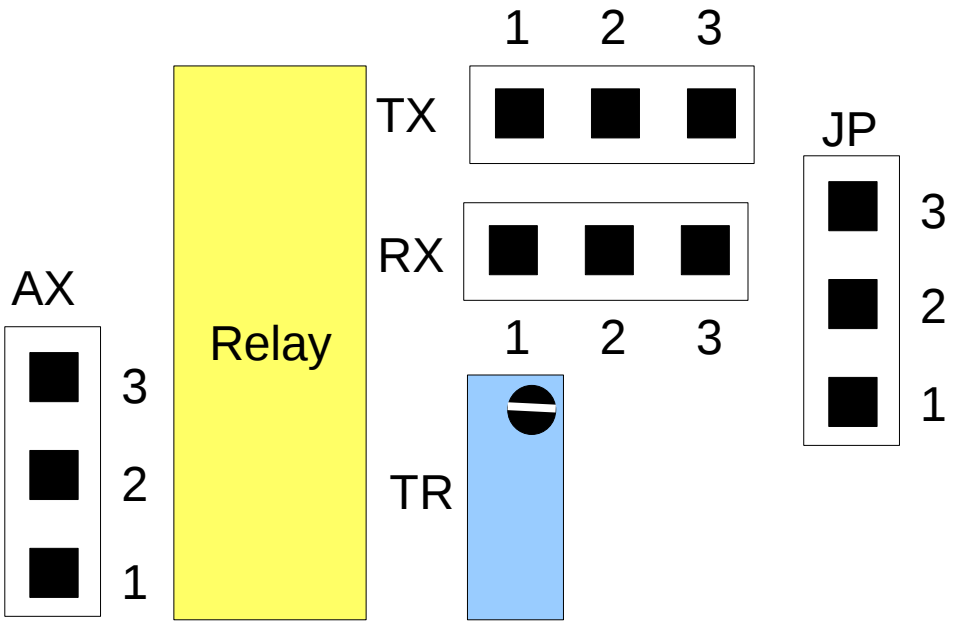


Illustration 4: Controller radio connections pinout

The next sections describe the pinout for the connectors TX, RX, AX and JP.

TX Connector

The TX connector connects to the radio microphone or TX, signal ground and PTT pin. The *AudioTX* pin must be connected when the radio is used for transmitting. The *AudioTX* pin signal is AC coupled and its amplitude can be adjusted up to 10Vp-p. If a radio is used in reception only, this connector may be left open or unconnected. The PTT signal is grounded by the controller when the radio must transmit.

Pin	Signal	Direction	Radio Connection
1	Audio TX	Output	Microphone input
2	Ground	--	Signal ground
3	PTT	Output	PTT

Table 1: TX Pinout

RX Connector

The RX connector connects to the radio *AudioOut* or *Speaker Out* signal ground and *COR (Carrier Operated Relay)* pin. The *AudioRX* pin must be connected when the radio is used in reception. The *AudioRX* pin signal is AC coupled and its input impedance is 10KΩ. If a radio is used in transmission only, this connector may be left open or unconnected. Radios with active high and active low *COR* signals are supported by the controller. The radio *COR* polarity must be configured in the controller in order for the controller to correctly detect the *COR* signal. The *COR* signal is active when a radio signal is received.

Pin	Signal	Direction	Radio Connection
1	Audio RX	Input	Speaker/Audio out
2	Ground	--	Signal ground
3	COR	Input	COR

Table 2: RX Pinout

AX Connector

The repeater controller circuit uses three DPDT (*Double Pole Double Throw*) relays to link the receiving radio's *Audio RX* signals to the *Audio TX* signals. The *Audio RX* signal uses one pole on the relay. The relay's secondary pole can be used for auxiliary or custom purposes. The micro-controller controls the relays and guarantees that only one relay is active at any time. A red LED above the relays indicates which relay is active.

Pin	Relay Signal	RX Disabled	RX Enabled
1	NO	Open	Connected to Pin 3 (COM)
2	NC	Connected to Pin 3 (Com)	Open
3	COM	Connected to Pin 2	Connected to Pin 1

Table 3: AX User relay pin header

Radio COR Polarity jumpers (JP)

The controller supports radios with ActiveHigh and ActiveLow *COR* simultaneously. The *COR* polarity for each radio must be configured in the controller using a DTMF sequence. The programming of the radios *COR* polarity is described later in this document.

It is also important to set the *COR* polarity jumpers. The *COR* polarity jumper must be tied to the radio's inactive *COR* state. If the jumper is incorrectly set, the controller will cause the other radios to transmit when a radio physically is disconnected from the controller. Refer to the following table for the *COR* jumper polarity pinout.

COR polarity	Shunt
Active High	2 and 3
Active Low	1 and 2

Table 4: COR polarity jumper connections for JP header

Connector identification on the circuit PCB

The connectors described above can be located in the controller PCB[4]. All the parts and connectors are identified on the top side of the controller PCB. The radio controller has connections for three radios. Refer to Table 5 for the location of the corresponding radio connectors on the PCB.

	TX	RX	AX	JP
Radio 1	JP2	JP3	JP8	SV1
Radio 2	JP4	JP5	JP9	SV2
Radio 3	JP6	JP7	JP10	SV3

Table 5: PCB connector identification

4 PCB stands for Printed Circuit Board

Audio adjustments

It is important that the audio levels are properly balanced. This guarantees that the audio signal from any of the receiving radio are re-transmitted with the same amplitude. The controller circuit board has adjustable resistors and audio amplifiers that enables the adjustment of the audio levels. There are two level adjustments for each radio. There is one potentiometer used to calibrate the *RX Audio* and the second potentiometer is used to calibrate the *TX Audio*. Refer to Error: Reference source not found for a functional overview of the audio signal paths and calibration components.

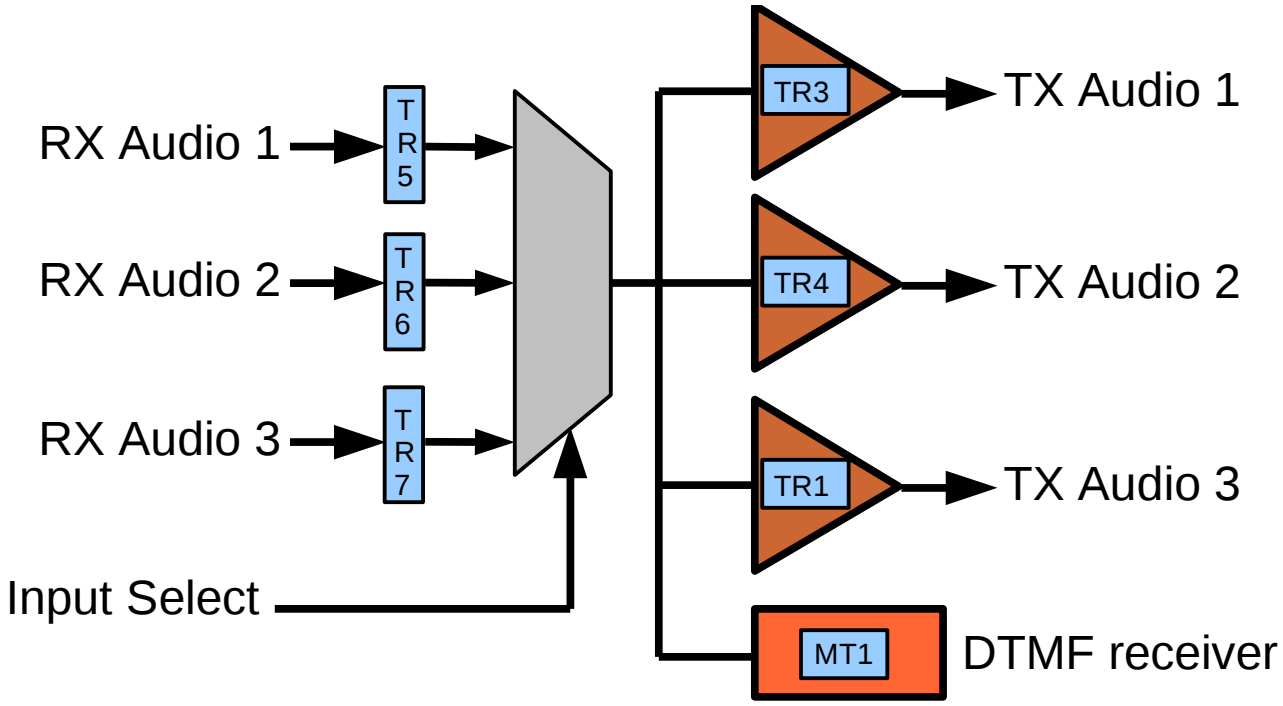


Illustration 5: Audio paths and gain adjustment potentiometers

While adjusting the TR5, TR6 and TR7 potentiometers, make sure that the signal level going to the DTMF receiver is sufficiently high to be decoded. The yellow LED1 turns on when a DTMF signal is received and decoded. After tuning the TR5, TR6 and TR7 potentiometers, send a DTMF signal into each radios and make sure that the LED1 turns on, indicating that a DTMF signal is correctly decoded. If the LED1 does not turn on when sending a DTMF signal, you will need to tune the potentiometer in order to increase the signal level of the radio from which the audio level is too low for DTMF detection. Once the input level potentiometer have been correctly tuned, you can proceed to the Audio level balancing procedure that is documented on page 15.

The *Input Select* signal shown on Illustration 5 is driven by the micro-controller. A single *RX Audio* signal can be selected at any time. The TR5, TR6 and TR7 resistors are used to attenuate the *RX Audio* signals. The *RX Audio* signals can be attenuated between -1dB and -10dB when the receiving radio's audio output is too high. The selected *RX Audio* signal is sent to the input stage of the three *TX Audio* amplifiers. Each individual *TX Audio* amplifier has a variable resistor that is used to adjust the gain. The *TX Audio* gain ranges can be adjusted from 1dB to 6dB and is used to calibrate the signal strength sent to the radio microphone or audio input.

Audio level balancing

The following procedure should be executed in order to balance the audio levels. A proper balancing of the audio levels will ensure that the sound level received from any radio will be transmitted with the same level on any radio. An oscilloscope or service monitor is recommended to perform this procedure. A common source signal should be used to inject a signal in the radios. The *COR* polarity values must be properly set inside the micro-controller before performing this procedure.

1. Measure the *Speaker Out* or *Audio Out* signal of the radios that will be connected to the repeater controller
2. Connect the radios to the repeater controller
3. Start with the adjustments of *RX Audio 1 (near the TR5 potentiometer)*. Inject an audio signal into this radio and make sure its relay is active (the red LED adjacent to its relay is lit).
4. Increase (turn clockwise) its *RX Audio* variable resistor TR5 to its maximal value; this enables the full amplitude of the radio's audio to be sent to the TX amplifiers.
5. Observe or measure the transmitted audio output on Radio 1, Radio 2 and Radio 3. The gain of the transmitted audio can be individually adjusted for each radio by setting the TR3, TR4 and TR1 potentiometers. Turning the variable resistors clockwise increases the gain. Adjust the gain such that the received audio level is identical for all radios. Do not overdrive the *TX Audio* gain. Excessive gain on the audio signals may cause the amplifiers to saturate and the audio quality will degrade significantly. Once the TR3, TR4 and TR1 potentiometers have been tuned, you should not need to change their value from this point.
6. Once the *RX Audio 1* has been tuned, the attenuation must be set for the *RX Radios* 2 and 3. Inject the audio signal into *RX Audio 2* and make sure that its relay is active (the red LED adjacent to its relay is lit). Adjust its *RX Audio* attenuation level using the TR6 potentiometer such that the transmitted audio level is the same as the one sent by the first radio.
7. Proceed to the *RX Audio* attenuation must be set for Radio 3. Inject the audio signal into *RX Audio 3* and make sure that its relay is active (the red LED adjacent to its relay is lit). Adjust its *RX Audio* attenuation level using the TR7 potentiometer such that the transmitted audio level is the same as the one sent by the two other radios.

Once this procedure is completed, the audio levels should be properly balanced. You may now proceed to functional tests by verifying that transmissions to and from any radios is equally balanced and that the sound quality is good.

I/O ports

Two auxiliary I/O ports are provided for custom use. The direction of each auxiliary port is programmable in the software by DTMF sequence. Refer to Table 6 for the AuxConfig register values used when programming the auxiliary I/O port directions.

The output ports can be driven by temperature limit operators, input ports or manually using the DTMF user functions.

AuxPort code	AuxO1	AuxO2	AuxI1	AuxI2
0	Unused	Unused	Input	Input
1	Output	Unused	Input	Unused
2	Unused	Output	Unused	Input
3	Output	Output	Unused	Unused

Table 6: Auxiliary port usage codes

The internal *Auxiliary Output Port* register is used to store the values that are sent to the auxiliary output ports. The table below lists the register values and the corresponding result on both auxiliary output ports.

Auxiliary Output Port register value	Aux Out 2	Aux Out 1
0	Off	Off
1	Off	On
2	On	Off
3	On	On

Table 7 : Auxiliary Output register values

Note that if the Auxiliary Output 1 or Auxiliary Output 2 port is configured as unused as per Table 6, the value stored in the Auxiliary Output control register is ignored for that auxiliary output.

The following limitations apply to these ports:

- Maximum input voltage : 12 Volts

- Maximum output current : 500mA

Exceeding these values will result in permanent damage to the circuit.

Temperature sensor

A temperature sensor is mounted on the circuit board and can be used to monitor the temperature. A maximum and minimum temperature set points can be programmed in software using a DTMF sequence. When the temperature crosses a set point, custom operations can be programmed. The following operations can be programmed when the temperature crosses the low or high temperature set points:

- Set auxiliary output port 1
- Clear auxiliary output port 1
- Send a Morse character on the tail of each transmission

Refer to Function (09,10,11) and Function (23,24) on pages 48 and 60 respectively for details on how to program the temperature operators and set points.

Programming port

The main controller used on the circuit board is a Microchip® PIC16F690 micro-controller. This controller can be programmed in circuit using a PicKit2® programmer. The 6 pin programming interface header is located on the top right-hand side of the circuit and is identified by JP1. The circuit board must be powered off when programming new firmware in in the micro-controller. New firmware will be provided when enhancements will be implemented.

Morse output

The micro-controller has the capability to generate Morse codes using a 1KHz sinusoidal signal. The audio level of the Morse output can be adjusted using the variable resistor TR2. When the corresponding functions are enabled, Morse characters are transmitted in the following conditions:

- Temperature set point exceeded
- Command mode confirmation
- Send or diagnose software register values
- Auxiliary input port monitoring on each transmission tail
- SiteID sent on each transmission tail
- Morse identification transmission every 30 minutes

There are two ways to control the amplitude of the Morse sinusoidal signal. The first method is by adjusting the TR2 potentiometer on the PCB. When the potentiometer is turned clockwise, the amplitude of the Morse signal increases and when the potentiometer is turned anti-clockwise, the amplitude decreases.

The second method to control the Morse amplitude is by accessing the *Morse Amplitude* (28) register. This method provides a way to control the amplitude of the Morse signal from a remote location. The

default *Morse Amplitude* register value is 5. The legal values can range from 0 to 10 where 0 is the minimal value and 10 is the maximum value. Refer to the *Morse Amplitude Function (28)* section for details on how to set this register.

The Table 8 shows the Morse characters that can be used for any of the functions listed above:

Code	Character	Code	Character
0	**0**	18	**i**
1	**1**	19	**j**
2	**2**	20	**k**
3	**3**	21	**l**
4	**4**	22	**m**
5	**5**	23	**n**
6	**6**	24	**o**
7	**7**	25	**p**
8	**8**	26	**q**
9	**9**	27	**r**
10	**a**	28	**s**
11	**b**	29	**t**
12	**c**	30	**u**
13	**d**	31	**v**
14	**e**	32	**w**
15	**f**	33	**x**
16	**g**	34	**y**
17	**h**	35	**z**

Table 8: Software codes for Morse characters

When programming Morse characters in the software, the code shown in the table below must be entered in the DTMF sequence. For example, if you want to program the sequence **VE2REH** as the site identifier, the following codes must be used:

Character	V	E	2	R	E	H
Value	31	14	02	28	14	17

DTMF control overview

There are a large number of software functions that can be programmed in the micro controller. In order for the functions to be programmed, at least one radio must be properly connected to the repeater controller circuit. DTMF sequences can be sent using a radio with a DTMF keypad. Make sure the DTMF tones are detected properly before attempting to configure software functions. Using a transmitter with a DTMF keypad, send several DTMF codes and make sure the yellow LED is lit; this confirms that the repeater controller is able to decode and process the DTMF codes. If the yellow LED does not go on when DTMF tones are sent, proceed to the *DTMF troubleshooting* section.

The DTMF receiver is located in the top left hand side of the circuit board, next to the two voltage regulators. The yellow LED is located just above the DTMF receiver.

DTMF Troubleshooting

Send several DTMF codes using the keypad on your radio. If the yellow LED does not go on when DTMF tones are sent, then one of the following items should be verified:

- The audio level from the DTMF radio receiver is too low (too much attenuation? Check TR5, TR6, TR7; Check radio volume adjustment)
- The level of the transmitted DTMF tones is too low
- There is too much noise with the DTMF tones
- The DTMF tones are invalid or outside the standard frequencies

If none of the bullets above helps, try using a different radio to send the DTMF codes.

Radio 1 timeout

The controller can be programmed to disable the Radio1 (also known as the link radio) to be disabled automatically after a pre-programmed timeout period when this feature is enabled. There are different ways to enable Radio 1:

- Temperature operators
- Send User Functions
- Write to the Enable (01) register

The Radio1Timeout register can be programmed with the timeout period in minutes. Valid values range from 0 (disabled) to 99. When the Radio1Timeout register contains the value 0, the timeout

function is disabled and the Radio 1 remains active.

Operational Mode

When the controller is in operational mode, the audio signal received by a radio are sent to the other radios so they can re-transmit the audio on different frequencies or bands. The user may modify the controller settings in order to specify the repeating radios for each individual receiving radio. Several more functions may be programmed in the controller.

During normal operation, the controller also listens for DTMF sequences in the transmitted audio signal. Various controller functions can be programmed remotely by sending DTMF tones to any radio connected to the controller. If the command mode DTMF sequence (or password) is sent, the system enters the command mode. The circuit must be in command mode in order to change the circuit functions. Once a circuit function has been completed, the system returns in normal operational mode. When the circuit is in operational mode, DTMF sequences that do not match the command mode sequence are ignored.

The command mode password can be programmed remotely. It contains up to 8 digits. It is highly recommended to use a long password. A shorter sequence increases the risk of a random DTMF sequences to gain control of the repeater controller functions and modify its functions or behavior.

Each DTMF sequence must start with a '*' and end with a '#' digit. The '*' digit clears the DTMF sequence buffer in the controller. The '#' digit instructs the controller to process the sequence sent after the '*' character.

Each repeater controller has a unique site identification number (SID) and a group identification number (GID). The SID and the GID are two digit identifiers. When a control sequence is sent to the controller, the first two digits must match the site ID or the group ID. If the site ID or group ID don't match, the entire DTMF sequence is ignored.

The site ID is used when sending instructions to a single repeater controller. The group ID is used when the same instruction must be broadcast to several repeater controllers on a network.

In order to enter command mode, the following sequence must be sent to the repeater controller:

```
* [SITE ID] 0 [PASSWORD] #
```

Example

Assuming that the site ID is 04 and the password is 21578, the following sequence must be sent to the controller:

```
*04 0 21578#
```

Once the sequence is sent, the controller will remain in command mode and wait for a command to be

sent. The side ID and password used in this example will be used in this document. Note that the site ID and password can be changed remotely by accessing these two functions using DTMF sequences. Refer to the Controller Functions section for information on how to change these registers.

User Functions

When the controller is operational, short DTMF sequences can be sent to execute predefined instructions. These instructions have the following syntax:

```
[DIGIT 1][DIGIT 2][* OR #]
```

For example:

```
39*
```

Four user functions can be programmed inside the controller. Each user function has a number and an operator. In the example shown previously, the user function number is 39. The last characters provides the argument to the user function's operator. The * character enables the operator while the # character disables the operator.

In order to disable a user function, its operator should be set to *NoOperation (0)*.

The same number can be used for multiple user functions. For example, userFunction1 and userFunction2 can be set to 39. When this user function is triggered (by 39* or 39#), the userFunctionOperators for both functions will be executed, starting with userFunction1Operator followed by userFunction2Operator. Refer to these register settings later in this manual.

Refer to Table 9 for a description of the operators that can be used with the user functions. These operators can also be used by the Temperature operators and the AuxiliaryInputOperators.

Value	Name	Description
0	NoOperation	The user function is not used.
1	Morse Echo	Sends predefined characters on each transmission tail when the functions are enabled (*): 'L' : Below Low temperature 'N': Within Normal temperature limit; 'H' : Above High temperature limit 'A' : Auxiliary input 1 set 'B' : Auxiliary input 2 set No morse is sent when the function is disabled (#)
2	Control AuxOut1	Sets the auxiliary port 1 (AUX_O1) On when the function is enabled (*) and turns Off the auxiliary output when the function is disabled (#).
3	Control AuxOut2	Sets the auxiliary port 2 (AUX_O2) On when the function is enabled (*) and turns Off the auxiliary output when the function is disabled (#).
4	Master Repeater Control	Sets the Enable register with its default value when the function is enabled (*). Else, it disables all radios by writing 0 in the Enable register when the function is disabled (#).
5	Control Radio1	Enables Radio 1 in the Enable register when the function is enabled (*). It disables only Radio 1 from the Enable register when the function is disabled (#).
6	Control Radio2	Enables Radio 2 in the Enable register when the function is enabled (*). It disables only Radio 2 from the Enable register when the function is disabled (#).
7	Control Radio3	Enables Radio 3 in the Enable register when the function is enabled (*). It disables only Radio 13from the Enable register when the function is disabled (#).

Table 9: User defined controller instructions

The *Morse Echo* operator is not typically used as an operator for the UserFunctions. It is usually used with the *Auxiliary Port Operator* or *Temperature Operator*. If the *Morse Echo* operator is used with as a *UserFunction* operator, it sends a Morse 1 when the function is enabled (*) and a Morse 0 when the function is disabled (#).

Command Mode

When the controller is in command mode, it is expecting a DTMF sequence with the following structure:

```
* [SITE/GROUP_ID] [OPERATOR] [REGISTER] [ARGUMENTS] #
```

The table below explains the structure of the command mode sequence.

Name	Number of digits	Comment
*	1	Clears the input buffer
Site/Group_ID	2	Corresponds to the Site ID or Group ID. A DTMF sequence that starts with a "*" and matches the site ID or group ID will be processed. Otherwise, the DTMF sequence is ignored and the controller remains in Command mode.
Operator	1	Specifies the operation to be performed on the register. Refer to the Command mode operators for information on the operators.
Register	2	Specifies the function register on which the operation is applied.
Arguments	Depends on the operator and selected function register	Arguments that are applied or written into the function register. The number of argument digits depends on the operator and function register. Refer to the Controller Functions sections for information on the register functions and their argument size.
#	1	Instructs the controller to process the command.

Table 10: Command mode sequence description

Command Mode operators

Various operations can be performed on the selected function registers. The operator is specified in the fourth field of the command mode sequence. The operator is used to modify the selected function register value. When a function register value is modified, its updated value is copied into a flash memory which makes the register value persistent even after a loss of power. Refer to Table 11 for a list of operators and their description. The micro controller uses 8 bit registers to store all of its function settings. Some function registers use 4 bit nibbles or less out of the 8 bits in the register. A nibble is the computing term for a four-bit aggregation, or half an octet (an octet being an 8-bit byte)[5]. Some operators modify the lower register nibble (bits 4 down to 0) and some operators modify the register octet (bits 8 down to 0). A nibble is used to set a value that ranges from 0 to 15 while an octet is used to set a value that ranges from 0 to 255.

Digit	Name	Description	Argument size
0	Command Mode entry/exit	Enters (with password) or exits the command mode.	0
1	Reboot MCU	Reboots the master controller unit	0
2	Set register nibble	The the function register is replaced by argument nibble value.	1
3	Set register bits	Sets bits in the function register to 1. The argument nibble bits that are set are copied into the function register.	1
5	Set register word	Sets the function register to the 8 bit argument value. A 2 digit decimal value is specified as the argument.	2
6	Set nibble array	Copies the array of argument nibbles into the array of function registers. This operator is generally used to program the Command Mode password.	1 to 8
7	Send Morse register value	The value of the function register is sent as Morse code to the radios.	2
8	Master reset	Forces the micro controller flash memory to reset to default values upon the next power-up.	0

Table 11: Command Mode operators

5 Definition of Nibble taken from http://en.wikipedia.org/wiki/Nibble

Examples

This section provides examples for each of the command mode operators. The DTMF sequences shown in these examples must be sent to the controller when it is in Command Mode. Refer to the Operational Mode section on page 20 for the procedure to enter the Command Mode. The controller Site/Group ID used in these examples is 01. The appropriate Site/Group identifier must be used with all the operators. The controller ignores all commands that don't match its Site/Group identifier.

Operator 0 (Exit)

Use the Exit operator to exit from the Command Mode and return to the Operational Mode by sending the DTMF sequence below:

```
* 01 0 #
```

Digit	Description
*****	Clears the input buffer
01	Site identifier
0	Exit operator
#	Instructs the controller to process the command

Table 12: Exit operator sequence details

Operator 1 (Reboot MCU)

The Reboot operator is used to restart the controller. It emulates a power-cycle sequence and performs the re-initialization of the controller:

```
* 01 1 #
```

Digit	Description
*****	Clears the input buffer
01	Site identifier
1	Reboot operator
#	Instructs the controller to process the command

Table 13: Reboot operator sequence details

Operator 2 (Set Register Nibble)

Use the *Set Register Nibble* operator to write a 4 bit nibble into the selected register. For example, the value 7 (binary 111) is written to the Polarity register (02). The nibble argument is a single DTMF character that ranges from 0 to 9. The following DTMF sequence must be sent when the controller is in *command mode* :

```
* 01 2 02 7 #
```

Digit	Description
*****	Clears the input buffer
01	Site identifier
2	Set Register Nibble operator
02	Select the Polarity register
7	Value to be written in the Polarity register
#	Instructs the controller to process the command

Table 14: Set Register Nibble operator sequence details

Operator 5 (Set Register Word)

Use the *Set Register Word* operator to set an 8 bit word value into the selected register. The argument is a two-digit DTMF decimal value that ranges from 00 to 99.

For example, the temperature low limit value must be set to -10°C, the following DTMF sequence must be sent:

```
* 01 5 09 30 #
```

Digit	Description
*****	Clears the input buffer
01	Site identifier
5	Set Register Word operator
09	Select the *Low Temp* register
30	Set the *Low Temperature* limit to -10°C (-10 + 40)
#	Instructs the controller to process the command

Table 15: Set Register Word operator sequence details

Operator 6 (Set Nibble Array)

Use the *Set Nibble Array* operator to set an array of 4 bit values into the selected register. The argument is a sequence of DTMF digits that range from 0 to 9.

This operator is generally used to set the *Command Mode Password* (0) registers.

For example, the DTMF sequence below must be sent in order to change the *Command Mode Password* to “19223”

```
* 01 6 00 1 9 2 3 3 #
```

Digit	Description
*****	Clears the input buffer
01	Site identifier
6	Set Nibble Array operator
00	Select the *Command Mode Password* register
19233	Change the Command Mode Password to 19233
#	Instructs the controller to process the command

Table 16: Set Nibble Array operator sequence details

Operator 7 (Send Morse Register Value)

The *Send Morse Register Value* operator is used to transmit the register value as Morse characters. This operator can be used to monitor various registers such as: verify the controller status, monitor the temperature, monitor the status of the auxiliary ports and verify the controller register configuration.

For example, the following DTMF sequence must be transmitted in order to monitor the temperature of system:

```
* 01 7 64 #
```

Digit	Description
*****	Clears the input buffer
01	Site identifier
7	*Send Morse Register Value* operator
64	Select the Auxiliary registers
#	Instructs the controller to process the command

Table 17: Send Morse Register Value operator sequence details

Operator 8 (Master Hardware Reset)

The *Master Hardware Reset* operator is used to restore all the values stored in the non-volatile EEPROM memory to their factory default values. This operator also resets the Command Mode password to the value “1 2”. The register argument for this operator must be '00'. Adding this requirement reduces the risk of accidentally resetting the controller.

For example, the following DTMF sequence must be transmitted in order reset the controller to the factory default settings:

```
* 01 8 00 #
```

Digit	Description
*****	Clears the input buffer
01	Site identifier
8	*Master Hardware Reset* operator
00	
#	Instructs the controller to process the command

Table 18: Send Morse Register Value operator sequence details

Controller Functions

The following section gives a detailed description of all the controller registers. Detailed examples are also provided that show how to modify the registers. The list of operators that can be used on each register is also shown in a table for each register.

The registers 00 to 36 are non-volatile registers. The values programmed inside these registers are stored in a non-volatile EEPROM memory. This makes the register values persistent even after an extended power loss.

The registers 64 and 65 are stored in volatile RAM memory; their register values are not preserved after a power loss.

The examples that are shown in the following pages will assume that the controller's siteID is 01. Make sure to specify the correct siteID for your controller.

Function (00) : Command Mode Password

Description:

The *Command Mode Password* registers holds the password sequence that enables the access to the controller's command mode. This register may contain 1 to 8 digits. The *Command Mode Password* sequence can be modified by first entering the command mode using the original password. Then, use the *Set Nibble Array (6)* operator to specify a new *Command Mode Password* sequence. The "#" character must be entered at the end of the new sequence in order to complete the command.

Operator	Code
Set Nibble Array	6

Table 19 : Valid operator for the Command Mode Password register

Example:

Assuming that the site ID is 01 and the initial *Command Mode Password* sequence is 12. The desired *Command Mode Password* will be changed to 8247. The following sequence must be entered:

```
* 01 0 12 #
* 01 6 00 8 2 4 7 #
```

The *Command Mode Password* register value can be modified in the controller command mode. The following operators can be used to modify or verify the Enable register:

Each digit in the sequence is explained in the table below:

Digit	Description
*****	Clears the input buffer
01	Site identifier
0	Command mode entry operator
12	Initial command mode password
#	Process the command
The controller is now in command mode	
*****	Clears the input buffer
01	Site identifier
6	Set Nibble Array operator
00	Select the Command Mode password register
8247	New Command Mode Password
#	Process the command
The new password is now effective	
*****	Clears the input buffer
01	Site identifier
0	Exit command code operator
#	Process the command
The controller exits command mode and returns to operational mode	

Table 20: Changing the Command Mode password sequence details

Function (01) : Enable register

Description:

The *Enable* register is used to activate or disable the radios that are connected to the repeater controller. Each radio can be individually controlled. When a radio is enabled, the controller will select the radio's audio RX signal and sent to the other radios audio TX pin and the corresponding PTT signals are activated. When a radio is disabled, its audio RX signal is ignored and no PTT is activated. Note that although all radios have been disabled, the controller is always listening and responding to DTMF sequences on any of the radio's audio signal. The table below shows the values that can be programmed into the *Enable* register in order to enable or disable each radio.

Enable value	Radio 3	Radio 2	Radio 1
7	On	On	On
6	On	On	Off
5	On	Off	On
4	On	Off	Off
3	Off	On	On
2	Off	On	Off
1	Off	Off	On
0	Off	Off	Off

Table 21 : Enable register values

The *Enable* register value can be modified in the controller command mode. The following operators can be used to modify or verify the Enable register:

Operator	Code
Set Nibble	2
Set Word	5
Send Morse register value	7

Table 22 : Valid operators for the Enable register

Example:

Assuming that the site ID is 01 and the initial Command Mode Password sequence is 12. We want to disable the radio transmissions. The following sequence must be entered:

```
* 01 0 12 #
* 01 1 01 #
```

Each digit in the sequence is explained in the table below:

Digit	Description
*	Clears the input buffer
01	Site identifier
0	Command mode entry operator
12	Initial command mode password
#	Process the command
The controller is now in command mode	
*	Clears the input buffer
01	Site identifier
1	Use the Clear Word operator
01	Select the Enable register
#	Process the command
The Enable register is now set to 0	
*	Clears the input buffer
01	Site identifier
0	Exit command code operator
#	Process the command
The controller exits command mode and returns to operational mode	

Table 23: Setting the Enable register off sequence details

Function (02) : Radio COR Polarity register

Description:

The *Polarity* register specifies the polarity of the COR signal for the three radios. The radio COR signal is active when a radio is receiving an RF signal. Some radios have an active high COR signal and some other radios have an active low COR signal. The correct COR polarity must be specified for each radio in order for the controller to select the appropriate receiving radio. If the COR signal is incorrectly specified, the controller will listen to a radio when it is idle and will ignore that radio when it is receiving an RF signal. The controller can accommodate up to three radios. The first three bits of the *Polarity* register specify the polarity of each radios. Refer to the table below do determine the appropriate value that needs to be specified in the *Polarity* register.

Polarity Value	Radio 3 COR	Radio 2 COR	Radio 1 COR
7	Active Low	Active Low	Active Low
6	Active Low	Active Low	Active High
5	Active Low	Active High	Active Low
4	Active Low	Active High	Active High
3	Active High	Active Low	Active Low
2	Active High	Active Low	Active High
1	Active High	Active High	Active Low
0	Active High	Active High	Active High

Table 24 Polarity register value based on individual radio COR signal polarity

Operator	Code
Set Nibble	2
Set Word	5
Send Morse register value	7

Table 25 : Valid operators for the Polarity register

Example:

Assuming that the site ID is 01 and the initial Command Mode Password sequence is 12. The polarity of the radios 1, 2 and 3 is Active High, Active Low and Active High respectively. The following

sequence must be entered:

```
* 01 0 12 #
* 01 2 02 2 #
```

Each digit in the sequence is explained in the table below:

Digit	Description
*	Clears the input buffer
01	Site identifier
0	Command mode entry operator
12	Initial command mode password
#	Process the command
The controller is now in command mode	
*	Clears the input buffer
01	Site identifier
2	Use the Set Nibble operator
02	Select the Polarity register
2	Value to be saved in the Polarity register
#	Process the command
The Polarity register is now set to 2	
*	Clears the input buffer
01	Site identifier
0	Exit command code operator
#	Process the command
The controller exits command mode and returns to operational mode	

Table 26: Setting the Polarity register sequence details

Function (03, 04, 05) : RX1PTT (03), RX2PTT (04), RX3PTT (05) registers

Description:

The controller can be configured such that each receiving radio's audio signal can be redirected to specific transmitting radio using the RXnPTT registers. Each RXnPTT registers specify the transmitting radios to which an incoming audio signal is to be retransmitted. Each transmitting radio has a PTT (Push To Talk) signal that activates the transmission of the microphone audio signal to an RF signal.

RXnPTT	Radio 3	Radio 2	Radio 1
0	Off	Off	Off
1	Off	Off	TX Enabled
2	Off	TX Enabled	Off
3	Off	TX Enabled	TX Enabled
4	TX Enabled	Off	Off
5	TX Enabled	Off	TX Enabled
6	TX Enabled	TX Enabled	Off
7	TX Enabled	TX Enabled	TX Enabled

Table 27 RX[1-3]PTT register values

As an example, let's assume that an incoming signal on Radio 1 is to be retransmitted on Radio 1 only, then RX1PTT should be set to 1. If an incoming signal on Radio 2 is to be retransmitted to Radios 3 and 1, then RX2PTT should be set to 5. Finally, if an incoming signal on Radio 3 is to be retransmitted on all three radios, then RX3PTT should be set to 7.

Operator	Code
Set Nibble	2
Set Word	5
Send Morse register value	7

Table 28 : Valid operators for the RXnPTT registers

Example:

Assuming that the site ID is 01 and the initial Command Mode Password sequence is 12. The audio

signal received from Radio 2 must be retransmitted to Radios 3 and 1. The following sequence must be entered:

```
* 01 0 12 #
* 01 2 04 5 #
```

Each digit in the sequence is explained in the table below:

Digit	Description
*	Clears the input buffer
01	Site identifier
0	Command mode entry operator
12	Initial command mode password
#	Process the command
The controller is now in command mode	
*	Clears the input buffer
01	Site identifier
2	Use the Set Nibble operator
04	Select the RX2PTT register
5	Value to be written in the RX2PTT register
#	Process the command
The Enable register is now set to 0	
*	Clears the input buffer
01	Site identifier
0	Exit command code operator
#	Process the command
The controller exits command mode and returns to operational mode	

Table 29: Set the RXnPTT register sequence details

Function (06) : MorsePTT register

Description:

The controller has the capability to send Morse characters on all three radios. The Morse characters can be used to monitor the controller temperature, auxiliary input values and register values. The MorsePTT register controls the destination of the Morse characters when the controller is in the normal operation mode. When the controller is in command mode, the Morse characters are sent to all radios. The table below lists the *MorsePTT* values and the corresponding radios to which the Morse characters are transmitted.

MorsePTT	Radio 3	Radio 2	Radio 1
0	Off	Off	Off
1	Off	Off	TX Enabled
2	Off	TX Enabled	Off
3	Off	TX Enabled	TX Enabled
4	TX Enabled	Off	Off
5	TX Enabled	Off	TX Enabled
6	TX Enabled	TX Enabled	Off
7	TX Enabled	TX Enabled	TX Enabled

Table 30 RX[1-3]PTT register values

The operators specified in the table below can be used to modify the MorsePTT register:

Operator	Code
Set Nibble	2
Set Word	5
Send Morse register value	7

Table 31 : Valid operators for the RXnPTT registers

Example:

Assuming that the site ID is 01 and the initial Command Mode Password sequence is 12. All Morse audio must be sent to radios 1 and 2. The following sequence must be entered:

```
* 01 0 12 #
* 01 2 06 3 #
```

Each digit in the sequence is explained in the table below:

Digit	Description
*	Clears the input buffer
01	Site identifier
0	Command mode entry operator
12	Initial command mode password
#	Process the command
The controller is now in command mode	
*	Clears the input buffer
01	Site identifier
2	Use the Set Nibble operator
04	Select the MorsePTT register
3	Value to be written in the MorsePTT register
#	Process the command
The Enable register is now set to 0	
*	Clears the input buffer
01	Site identifier
0	Exit command code operator
#	Process the command
The controller exits command mode and returns to operational mode	

Table 32: Set the MorsePTT register sequence details

Function (07) : Morse Dit Duration (07) register

Description:

The speed of the Morse characters that are sent by the controller can be controlled using the *Morse Dit Duration* registers. By convention, a Dah duration is three times the length of a dit.

Each duration value corresponds to a 10ms duration (or 0.01 second duration). The default Dit duration is set to 60ms.

The operators specified in the table below can be used to modify the Dit duration register:

Operator	Code
Clear Word	1
Set Nibble	2
Set Bits	3
Clear Bits	4
Set Word	5
Send Morse register value	7

Table 33 : Valid operators for the Morse Dit Duration registers

Example:

Assuming that the site ID is 01 and the initial Command Mode Password sequence is 12. The desired Dit duration needs to be set to 70ms:

```
* 01 0 12 #
* 01 2 07 7 #
```

Each digit in the sequence is explained in the following table:

Digit	Description
*****	Clears the input buffer
01	Site identifier
0	Command mode entry operator
12	Initial command mode password
#	Process the command
The controller is now in command mode	
*****	Clears the input buffer
01	Site identifier
2	Use the Set Nibble operator
07	Select the Dit duration register
7	Set the Dit duration to 70ms
#	Process the command
The Dit duration register is now set to 70ms	
*****	Clears the input buffer
01	Site identifier
0	Exit command code operator
#	Process the command
The controller exits command mode and returns to operational mode	

Table 34: Set the Dit duration sequence details

The Dah duration is automatically set to three times the Dit duration. For the example above, the Dah duration is automatically set to 210ms.

Function (08) : Radio 1 Timeout (08) register

Description:

A timeout can be specified on Radio 1 when it is enabled. The timeout value specifies the number of minutes that the Radio 1 stays enabled. After the timeout has expired, the controller disables Radio 1 only. This does not affect the settings of Radios 2 and 3. If the value 0 is stored inside the *Radio 1 Timeout* register, then this feature is disabled and the Radio 1 will remain active.

The operators specified in the table below can be used to modify the Dit duration register:

Operator	Code
Set Nibble	2
Set Word	5
Send Morse register value	7

Table 35 : Valid operators for the Radio 1 Timeout register

Example:

Assuming that the site ID is 01 and the initial Command Mode Password sequence is 12. The desired Radio 1 Timeout is 10 minutes:

```
* 01 0 12 #
* 01 5 08 10 #
```

Each digit in the sequence is explained in the following table:

Digit	Description
*	Clears the input buffer
01	Site identifier
0	Command mode entry operator
12	Initial command mode password
#	Process the command
The controller is now in command mode	
*	Clears the input buffer
01	Site identifier
5	Use the Set Word operator
08	Select the Radio 1 Timeout register
10	Set the timeout to 10 minutes
#	Process the command
The Dit duration register is now set to 70ms	
*	Clears the input buffer
01	Site identifier
0	Exit command code operator
#	Process the command
The controller exits command mode and returns to operational mode	

Table 36: Set the Dit duration sequence details

Function (09, 10, 11) : Low Temperature Operation (09), High Temperature Operation (10), Normal Temperature Operation (11) registers

Description:

The *Low Temperature Operation, High Temperature Operation* and *Normal Temperature Operation* are used to execute specific instructions when the repeater controller temperature is too low, too high or in the normal operating range respectively. The temperature inside the repeater controller is measured every 2 seconds. When the temperature crosses a set point, the corresponding instruction for this new temperature is executed. When the temperature is between the low and high set points, the controller executes the *Normal Temperature Operation*.

The low and high temperature set points can be programmed in the *Low Temperature (17), High Temperature (18)* registers.

Refer to Table 9: User defined controller instructions on page 22 for details on the custom instructions that can be used for the temperature operators.

The operators specified in the table below can be used to modify the *Low, High and Normal Temperature Operation* register:

Operator	Code
Set Nibble	2
Set Word	5
Send Morse register value	7

Table 37 : Valid operators for the Low, High and Normal Temperature Operation registers

Example:

Assuming that the site ID is 01 and the initial Command Mode Password sequence is 12. The repeater function must be disabled when the temperature exceeds the upper limit. The repeater function is also enabled when the temperature is within the temperature limits.

```
* 01 0 12 #
* 01 2 10 5 #
* 01 2 11 4 #
```

Each digit in the sequence is explained in the table below:

Digit	Description
*****	Clears the input buffer
01	Site identifier
0	Command mode entry operator
12	Initial command mode password
#	Process the command
The controller is now in command mode	
*****	Clears the input buffer
01	Site identifier
2	Use the Set Nibble operator
10	Select the *High Temperature Operation* register
5	Set the *Repeater Disable* operation
#	Process the command
The controller is now programmed to disable the repeater function when the temperature exceeds the upper limit	
*****	Clears the input buffer
01	Site identifier
2	Use the Set Nibble operator
11	Select the *Normal Temperature Operation* register
4	Set the *Repeater Enable* operation
#	Process the command
The controller is now programmed to enable the repeater function when the temperature is within the specified limits	
*****	Clears the input buffer
01	Site identifier
0	Exit command code operator
#	Process the command
The controller exits command mode and returns to operational mode	

Function (12, 13) : Aux Input 1 Operation (12), Aux Input 2 Operation (13) registers

Description:

There are two auxiliary ports available on the controller. The auxiliary ports can individually configured as inputs or outputs. When a port is configured as an input port, the *Aux Input x Operation* register is used to store an operation to be executed when the input port is active (5V). Each port has its own operation register. There are several types of operations that can be executed.

The first type of operation is to send a Morse character when the input is active.

The second type of operation is to set the other auxiliary port to follow or invert the other auxiliary port when it is configured as an input port. This operation requires that the other auxiliary port is configured as an output port in order for the value to be driven on the auxiliary port; otherwise, the *Aux Input x Operation* stores the value in memory and the value is applied on the auxiliary port when its direction is changed to output.

The last type of operation is to control the *Enable* register. The controller can be enabled or disabled based on the value driven on one of the auxiliary input ports.

The direction of the auxiliary ports can be changed using the *Auxiliary Port Configuration* register.

Refer to Table 9: User defined controller instructions on page 22 for details on the custom instructions that can be used for the *Aux Input # Operators*.

The operators specified in the table below can be used to modify the *Auxiliary Input 1 Operation* and *Auxiliary Input 2 Operation* register:

Operator	Code
Set Nibble	2
Set Word	5
Send Morse register value	7

Table 38 : Valid operators for the Aux Input 1 Operation and Aux Input 2 Operation registers

Example:

Assuming that the site ID is 01 and the initial Command Mode Password sequence is 12. The controller Enable register must be controlled using the Auxiliary Input 2:

```
* 01 0 12 #
* 01 2 13 4 #
```

Each digit in the sequence is explained in the table below:

Digit	Description
*****	Clears the input buffer
01	Site identifier
0	Command mode entry operator
12	Initial command mode password
#	Process the command
The controller is now in command mode	
*****	Clears the input buffer
01	Site identifier
2	Use the Set Nibble operator
13	Select the *Aux Input 2 Operation* register
4	Set the *Repeater Enable* operation
#	Process the command
The controller Enable register is now controlled by the Auxiliary input 2 port	
*****	Clears the input buffer
01	Site identifier
0	Exit command code operator
#	Process the command
The controller exits command mode and returns to operational mode	

Table 39: Set the Repeater Enable operation sequence details

Function (14) : Auxiliary Output Power-Up Values registers

Description:

When the controller is powered up, the auxiliary output ports are reset to the values that are stored inside this register. Note that the value on the auxiliary output ports may be modified by the user or by any other operation after the controller has been powered on.

If an auxiliary port is configured in input mode, the value of this register is stored in memory but it is not applied on the port. If the auxiliary port is later configured in output mode, then the value stored in memory will be applied on the output port.

The table below shows the register value and the associated power up values for each auxiliary output port.

Register value	Aut Out 2	Aux Out 1
0	Off	Off
1	Off	On
2	On	Off
3	On	On

Table 40 : Auxiliary Ouput Power-On values

The operators specified in the table below can be used to modify the *Auxiliary Output Power-On* register:

Operator	Code
Set Nibble	2
Set Word	5
Send Morse register value	7

Table 41 : Valid operators for the Auxiliary Output Power-On register

Example:

Assuming that the site ID is 01 and the initial Command Mode Password sequence is 12. The desired power-on values is Off for the aux output port 2 and On for the aux output port 1. The following DTMF sequence must be sent:

```
* 01 0 12 #
* 01 2 14 1 #
```

Each digit in the sequence is explained in the table below:

Digit	Description
*****	Clears the input buffer
01	Site identifier
0	Command mode entry operator
12	Initial command mode password
#	Process the command
The controller is now in command mode	
*****	Clears the input buffer
01	Site identifier
2	Use the Set Nibble operator
14	Select the *Auxiliary Output Power-On* register
1	Write the value 1 inside the register
#	Process the command
The power-on values for aux output 2 is “Off” and for aux output 1 is “On”.	
*****	Clears the input buffer
01	Site identifier
0	Exit command code operator
#	Process the command
The controller exits command mode and returns to operational mode	

Table 42: Set the Auxiliary Output Power-On values sequence details

Function (15) : Site ID register

Description:

Each controller has a unique site identifier. A controller responds to a DTMF sequence when its site identifier matches the one sent in the DTMF sequence. If the site identifier sent in the DTMF sequence does not match the controller's site identifier, then the DTMF sequence is ignored.

The site identifier is used to address a single controller. This enables a user to send commands to a single controller on a network where multiple controllers are installed and monitoring for DTMF sequences.

Similarly to the site identifier, each controller has a group identifier. The group identifier is not programmable and is used when the SiteID value has been lost or when the controller does not repond to its SiteID. As for the site identifier, the controller will also respond to DTMF sequences if the group identifier sent in the DTMF sequence matches its own group identifier. The group identifier also enables a user to simultaneously change settings of several controllers by broadcasting a DTMF sequence to a group of controllers with a common group identifier. Different group ID's may be programmed. The group ID for your board should be provided when you purchase the board.

The network administrator is in charge of assigning the site ID for each controller on the network. It is important that each controller has a unique site ID and that they do not collide with the group ID of other controllers on the network. Each identifier is made of two digits that range from 01 to 98.

Please refer to the DTMF control overview section on page 19 for further details.

The operators specified in the table below can be used to modify the *Site ID* register:

Operator	Code
Set Nibble	2
Set Word	5
Send Morse register value	7

Table 43 : Valid operators for the Site ID register

Example:

Assuming that the site ID is 09 and the initial Command Mode Password sequence is 12. The site identifier for the controller must be changed to 50. The following DTMF sequence must be sent:

```
* 09 0 12 #
* 09 5 15 50 #
* 50 0 #
```

Each digit in the sequence is explained in the table below:

Digit	Description
*****	Clears the input buffer
09	Site identifier
0	Command mode entry operator
12	Initial command mode password
#	Process the command
The controller is now in command mode	
*****	Clears the input buffer
09	Site identifier
5	Use the Set Word operator
15	Select the *Site Identifier* register
50	Change the site identifier to 50
#	Process the command
The site identifier for the controller is now set to 50	
*****	Clears the input buffer
50	** New Site identifier **
0	Exit command code operator
#	Process the command
The controller exits command mode and returns to operational mode	

Table 44: Set the Site Identifier sequence details

Function (16) : Max COR Per Minute register

Description:

The controller has the ability to disable itself autonomously in the presence of a receiver malfunction. When a receiver is activated repeatedly in the presence of noise, its COR signal can toggle several times per minute and it transmits noise on the network. The *MaxCorPerMinute* register specifies an uppper limit for the number of COR signals that are tolerated per minute. If the number of COR signals exceeds this limit, the controller sets the *Enable (01)* register to 0. This disables all PTT signals during one minute and the RX1 relay remains activated. After one minute, the controller resets the *Enable* register to its original value and resumes normal operation.

During this period, the controller will still listen to DTMF commands on Radio 1 but no Morse confirmation is sent during the timeout period.

The default value stored inside the *MaxCORPerMinute* register is 15.

The operators specified in the table below can be used to modify the MaxCORPerMinute register:

Operator	Code
Set Nibble	2
Set Word	5
Send Morse register value	7

Table 45 : Valid operators for the MaxCORPerMinute

Example:

Assuming that the site ID is 01 and the initial Command Mode Password sequence is 12. A new *MaxCORPerMinute* of 12 is desired. The following DTMF sequence must be sent:

```
* 01 0 12 #
* 01 5 16 12 #
```

Each digit in the sequence is explained in the table below:

Digit	Description
*****	Clears the input buffer
1	Site identifier
0	Command mode entry operator
12	Initial command mode password
#	Process the command
The controller is now in command mode	
*****	Clears the input buffer
1	Site identifier
5	Use the Set Word operator
16	Select the *MaxCORPerMinute* register
12	Change the Max COR per minute limit to 12
#	Process the command
The *MaxCorPerMinute* register is now set to 12	
*****	Clears the input buffer
01	Site identifier
0	Exit command code operator
#	Process the command
The controller exits command mode and returns to operational mode	

Table 46: Set the MaxCORPerMinute sequence details

Function (17 - 22) : Morse Site Identifier character registers

Description:

The controller has the capability to transmit a 6 character Morse identifier every 30 minutes. The transmission of the Morse site identifier characters is controlled by the *Transmit Morse Identifier* register (26). The Morse identifier characters can be programmed inside the *Morse Site Identifier* registers. There are a total of 6 registers in which each Morse character used to identify the station is stored. All 6 Morse characters must be programmed individually.

Refer to Table 8 on page 18 the list of Morse characters and their corresponding register values. The values from this table can be programmed in any order. The operators specified in the table below can be used to modify the *Morse Site Identifier* register:

Operator	Code
Set Word	5
Send Morse register value	7

Table 47 : Valid operators for the Morse Site Identifier register

When using the *Send Morse Register Value* operator (7) on the first register (17), the controller sends all 6 characters to the radios. This can be used to validate that the Morse station identifier characters have been programmed correctly. When using the *Send Morse Register Value* operator (7) on any following registers (18 to 22), only one character will be send to the radios.

Example:

Assuming that the site ID is 01 and the initial Command Mode Password sequence is 12. The desired Morse site identifier is “VE2ABC”, the following DTMF sequence must be sent:

```
* 01 0 12 #
* 01 5 17 31  #      -- V
* 01 5 18 14  #      -- E
* 01 5 19 02  #      -- 2
* 01 5 20 10  #      -- A
* 01 5 21 11  #      -- B
* 01 5 22 12  #      -- C
* 01 0 #             -- EXIT
```

Each digit in the sequence is explained in the table below:

Digit	Description
*	Clears the input buffer
01	Site identifier
0	Command mode entry operator
12	Initial command mode password
#	Process the command
The controller is now in command mode	
*	Clears the input buffer
01	Site identifier
5	Use the Set Word operator
17	Select the *Morse Site Identifier* register
31	Write 31 (for “V” in the first character register)
#	Process the command
The first Morse Site Identifier character has been set to “V”	
*	Clears the input buffer
01	Site identifier
0	Exit command code operator
#	Process the command
The controller exits command mode and returns to operational mode	

Table 48: Set the Morse Site Identifier sequence details

The commands in the yellow section above must be repeated for all registers 18 to 22 as shown in the example. Once all characters have been entered, the exit operator can be entered and the system returns to normal operation mode. If the *Transmit Morse Identifier* register (26) is enabled, the Morse site identifier will be transmitted every 30 minutes.

Function (23, 24) : Low temp (23), High temp (24) registers

Description:

The circuit board is equipped with a temperature sensor. The temperature is sampled every 2 seconds and compared against two limit values. If the temperature exceeds the low or high temperature limit, special instructions can be executed. The AUX1 auxiliary output port can be programmed to be set or cleared when the temperature exceeds any of the limits. The controller can also send Morse characters to the radios if the limits are exceeded.

The *Low Temp* and *High Temp* registers are used to specify the minimum and maximum temperature limits respectively. If the temperature is lower than the *Low Temp* register value, then the *Low Temp Operation* is executed. If the temperature is higher than the *High Temp* register value, then the *High Temp Operation* is executed. Otherwise, if the temperature is between the low and high set points, then the *Normal Temp Operation* is executed. The *Low Temp Operation (09), High Temp Operation (10)* and *Normal Temp Operation (11)* registers are documented earlier in this document.

The values specified in the registers are in degrees Celsius with an offset of -40°C. The value 40 must be added to the desired temperature limit. For example, if the desired temperature limit is 10°C, then the register should be set to 50. The lowest temperature limit that can be set is -40°C; this can be achieved by writing the value 0 in the *Low Temp* register.

The operators specified in the table below can be used to modify the *Low Temp* and *High Temp* register:

Operator	Code
Set Nibble	2
Set Word	5
Send Morse register value	7

Table 49 : Valid operators for the Low Temp and High Temp registers

Example:

Assuming that the site ID is 01 and the initial Command Mode Password sequence is 12. The desired high temperature limit is 25°C, the following sequence must be sent:

```
* 01 0 12 #
* 01 5 24 65 #
```

Each digit in the sequence is explained in the table below:

Digit	Description
*****	Clears the input buffer
01	Site identifier
0	Command mode entry operator
12	Initial command mode password
#	Process the command
The controller is now in command mode	
*****	Clears the input buffer
01	Site identifier
5	Use the Set Word operator
24	Select the *Set High Temp* register
65	Set the high temperature limit to 25°C (25 + 40)
#	Process the command
The High Temperature limit is now set to 25°C	
*****	Clears the input buffer
01	Site identifier
0	Exit command code operator
#	Process the command
The controller exits command mode and returns to operational mode	

Table 50: Set the High Temperature limit sequence details

Function (25) : Auxiliary Port Configuration register

Description:

There are two auxiliary ports on the controller. The ports can be used as auxiliary inputs or outputs. The value of the *Auxiliary Port Configuration* register is used to set the mode for each auxiliary port. Refer to Table 6 on page 16 for the register values that can be programmed in this register.

The operators specified in the table below can be used to modify the *Auxiliary Port Configuration* register:

Operator	Code
Clear Word	1
Set Nibble	2
Set Bits	3
Clear Bits	4
Set Word	5
Send Morse register value	7

Table 51 : Valid operators for the Auxiliary Port Configuration register

Example:

Assuming that the site ID is 01 and the initial Command Mode Password sequence is 12. The following auxiliary port configuration is required:

Auxiliary port 1 : Output port controlled by the temperature sensor (Aux Out 1)

Auxiliary port 2 : Input port (Aux In 1)

According to Table 6, the register value '5' must be programmed in the *Auxiliary Port Configuration* register*:*

```
* 01 0 12 #
* 01 2 25 5 #
```

Each digit in the sequence is explained in the table below:

Digit	Description
*****	Clears the input buffer
01	Site identifier
0	Command mode entry operator
12	Initial command mode password
#	Process the command
The controller is now in command mode	
*****	Clears the input buffer
01	Site identifier
2	Use the Set Nibble operator
25	Select the *Auxiliary Port Configuration* register
5	Use port configuration 5 (Refer to Table 6)
#	Process the command
The Auxiliary ports are configured as described in the example above	
*****	Clears the input buffer
01	Site identifier
0	Exit command code operator
#	Process the command
The controller exits command mode and returns to operational mode	

Table 52: Set the Auxiliary Port Configuration sequence details

Function (26) : Transmit Morse Identifier register

Description:

The *Transmit Morse Identifier* register is used to transmit the site identifier characters every 30 minutes. The site identifier Morse characters are programmed in registers (17 – 22). Some network configurations have a central transmission of the Morse identifier that would be received by Radio 1 and the repeater controller will re-transmit the Morse identifier to the two other radios. However, if the link radio is disabled, the repeater controller can generate the Morse identifier and send it to the radios 2 and 3 every 30 minutes.

The following table shows the various modes for the Morse site identifier transmission.

TransmitMorseID register value	Description
0	TX Disabled
1	Site ID Morse ID is sent every 30 minutes
2	Site ID Morse ID is sent every 30 minutes when Radio 1 is disabled. Refer to Enable (01) register values for details.

Table 53 : TransmitMorseID register values

The operators specified in the table below can be used to modify the *Transmit Morse Identifier* register:

Operator	Code
Clear Word	1
Set Nibble	2
Send Morse register value	7

Table 54 : Valid operators for the Transmit Morse Identifier register

Example:

Assuming that the site ID is 01 and the initial Command Mode Password sequence is 12. The following DTMF must be sent in order to enable the Morse site identifier transmission:

```
* 01 0 12 #
* 01 2 26 1 #
```

Each digit in the sequence is explained in the table below:

Digit	Description
*****	Clears the input buffer
01	Site identifier
0	Command mode entry operator
12	Initial command mode password
#	Process the command
The controller is now in command mode	
*****	Clears the input buffer
01	Site identifier
2	Use the Set Nibble operator
26	Select the *Transmit Morse Identifier* register
1	Write value 1 (Enable) into the register
#	Process the command
The Morse site ID will be transmitted every 30 minutes	
*****	Clears the input buffer
01	Site identifier
0	Exit command code operator
#	Process the command
The controller exits command mode and returns to operational mode	

Table 55: Enable Morse side identifier transmission sequence details

Function (27) : Transmit Site Identifier register

Description:

The transmission of the site identifier can be used to diagnose the source of interference and also monitor the use of the repeaters on a network with multiple controllers. The site identifier is defined in the *Site Identifier* register (15). The Morse code that is transmitted corresponds to the value in Table 8. For example, if the site identifier register contains the value 11, a Morse “B” character is transmitted.

The controller can be programmed to transmit its Morse site identifier on each tail (when all PTT's are released). This mode can be individually enabled per radio. When the *Transmit Site Identifier* function is enabled for a given radio, the controller will send the SiteID in Morse when the COR from this radio falls. When this occurs, the controller activates the PTT to transmitters specified by the MorsePTT register (06). Refer to Table 56 for the *TransmitSiteID* register values for each radio. Multiple radios can be enabled by adding the register value of each radio.

TransmitSiteID register value	Description
0	Disabled
1	Enable SiteID transmission for Radio 1
2	Enable SiteID transmission for Radio 2
4	Enable SiteID transmission for Radio 3

Table 56 : TransmitMorseID register values

The operators specified in the table below can be used to modify the *Transmit Site Identifier* register:

Operator	Code
Clear Word	1
Set Nibble	2
Send Morse register value	7

Table 57 : Valid operators for the Transmit Site Identifier register

Example:

Assuming that the site ID is 01 and the initial Command Mode Password sequence is 12. The following DTMF must be sent in order to enable the Morse site identifier transmission for local radios (Radio 2

and Radio 3):

```
* 01 0 12 #
* 01 2 27 6 #
```

Each digit in the sequence is explained in the table below:

Digit	Description
*****	Clears the input buffer
01	Site identifier
0	Command mode entry operator
12	Initial command mode password
#	Process the command
The controller is now in command mode	
*****	Clears the input buffer
01	Site identifier
2	Use the Set Nibble operator
27	Select the *Transmit Site Identifier* register
6	Write value 6 (Radio2 + Radio3) into the register
#	Process the command
The Morse site ID will be transmitted every 30 minutes	
*****	Clears the input buffer
01	Site identifier
0	Exit command code operator
#	Process the command
The controller exits command mode and returns to operational mode	

Table 58: Enable Morse side identifier transmission sequence details

Once the controller returns to normal operation mode, the site identifier will be transmitted on each tail when CORs from Radios 2 and 3 fall. The Site identifier will be sent in Morse on all radios specified by the MorsePTT register value.

Function (28) : Morse Amplitude register

Description:

The amplitude of the Morse sinusoidal signal can be controlled by the *Morse Amplitude* register. The register value can range from 0 to 4 where 0 corresponds to the minimal amplitude and 4 corresponds to the maximum amplitude. The default value is 3. This register enables a remote user from adjusting the amplitude of the Morse signal when the access to the controller is not available. Note that when the access to the controller is available, the potentiometer TR2 on the PCB can also be used to control the amplitude of the Morse signal. The *Morse Amplitude* register can be programmed with the values specified in the following table:

Register Value	Amplitude level
0	0.00%
1	12.50%
2	25.00%
3	50.00%
4	100.00%

Table 59: Morse Amplitude level register values

When the amplitude level is set to 100%, the signal amplitude is determined by the TR2 potentiometer. If the TR2 potentiometer is set to a low value, the effect of the Morse Amplitude register will be limited. It is recommended that the level of the TR2 potentiometer is set higher than the desired level and the Morse Amplitude register set to 3. This provides the capability to increase the Morse audio level remotely at a later time.

When the register value is increased by 1, the amplitude level doubles and when the register value is decreased by 1, the amplitude level is reduced by half.

The operators specified in the table below can be used to modify the *Morse Amplitude* register:

Operator	Code
Clear Word	1
Set Nibble	2
Send Morse register value	7

Table 60 : Valid operators for the Morse Amplitude register

Example:

Assuming that the site ID is 01 and the initial Command Mode Password sequence is 12. The following DTMF must be sent in order to set the Morse Amplitude to 2:

```
* 01 0 12 #
* 01 2 28 2 #
```

Each digit in the sequence is explained in the table below:

Digit	Description
*****	Clears the input buffer
01	Site identifier
0	Command mode entry operator
12	Initial command mode password
#	Process the command
The controller is now in command mode	
*****	Clears the input buffer
01	Site identifier
2	Use the Set Nibble operator
28	Select the *Morse Amplitude* register
2	Write value 2 into the register
#	Process the command
The Morse amplitude level is set to 2	
*****	Clears the input buffer
01	Site identifier
0	Exit command code operator
#	Process the command
The controller exits command mode and returns to operational mode	

Table 61: Change Morse amplitude sequence details

Once the controller returns to normal operation mode, the new Morse amplitude level is set to 2.

Function (29,30,31,32,33,34,35,36) : UserFunction# (29,31,33,35) and UserFunction#Operator (30,32,34,36) registers

Description:

The controller can be programmed to run user defined instructions (called UserFunctions) when a specific DTMF sequence is entered. Up to four UserFunctions can be defined. Each user function is executed by sending the two-digit UserFunction number followed by the '*' or '#' digit. When the '*' digit is sent, the UserFunction operator argument is enabled. When the '#' digit is sent, the UserFunction operator is disabled. Refer to the User Functions section on page 21 for usage instructions.

Each user function has an associated operator. Refer to the User defined controller instructions table on page 22 for details on the operators.

Refer to the following table for the UserFunction and the UserFunctionOperator register addresses. The UserFunction address register hold the binary value of the desired UserFunction number (0 to 99). The UserFunctionOperator address register hold the binary value corresponding to the Operator that will be executed by this UserFunction.

Register	Name
29	UserFunction1
30	UserFunction1Operator
31	UserFunction2
32	UserFunction2Operator
33	UserFunction3
34	UserFunction3Operator
35	UserFunction4
36	UserFunction4Operator

Table 62: UserFunction and UserFunctionOperator register values

The operators specified in the table below can be used to modify the *UserFunction* and *UserFunctionOperator* registers:

Operator	Code
Clear Word	1
Set Nibble	2
Send Morse register value	7

Table 63 : Valid operators for the UserFunction and UserFunctionOperator register

Example:

Assuming that the site ID is 01 and the initial Command Mode Password sequence is 12. A UserFunction is needed to control Radio1 using the following DTMF entries:

- 39* (Enable Radio1)
- 39# (Disable Radio1)

The UserFunction1 value needs to be set to 39 and the UserFunction1Operator must be set to 5 (Control Radio1)

```
* 01 0 12 #
* 01 5 29 39 #
* 01 2 30 5 #
```

Each digit in the sequence is explained in the table below:

Digit	Description
*****	Clears the input buffer
01	Site identifier
0	Command mode entry operator
12	Initial command mode password
#	Process the command
The controller is now in command mode	
*****	Clears the input buffer
01	Site identifier
5	Use the Set Word operator
29	Select the *UserFunction1Operator* register
39	Write value 2 into the register
#	Process the command
The *UserFunction1* is now 39	
*****	Clears the input buffer
01	Site identifier
2	Use the Set Nibble operator
30	Select the *Morse Amplitude* register
5	Write value 5 (Control Radio1 Operator) into the register
#	Process the command
The *UserFunction1Operator* is now "Control Radio1"	
*****	Clears the input buffer
01	Site identifier
0	Exit command code operator
#	Process the command
The controller exits command mode and returns to operational mode	

Table 64: Setting UserFunction1 and UserFunction1Operator sequence details

Once the controller returns to normal operation mode, the controller is now responding to the 39* and 39# commands and will execute the "Control Radio1" operator.

Function (37) : PTT Time-Out register

Description:

A transmission time out value can be programmed inside the controller to prevent a radio from transmitting indefinitely. When a radio begins to transmit, a timer counts the transmission time. If the transmission time exceeds the *PTT Time-Out* register value, all PTT's will be released and all radio transmissions will stop. Once the COR from the radio that timed-out is released, then the time-out timer is reset and the transmissions can resume. By default, a time out timer of 8 minutes is programmed in the controller.

The value of the time out timer limit can be set using the *PTT Time-Out* register. The value 0 is used to disabled the time-out feature. The value programmed inside the *PTT Time-Out* register is specified in minutes.

The operators specified in the table below can be used to modify the *PTT Time-Out* register:

Operator	Code
Clear Word	1
Set Nibble	2
Set Word	5
Send Morse register value	7

Table 65 : Valid operators for the PTT Time-Out register

Example:

Assuming that the site ID is 01 and the initial Command Mode Password sequence is 12. The following DTMF must be sent in order to set the PTT time-out to 10 minutes:

```
* 01 0 12 #
* 01 5 37 10 #
```

Each digit in the sequence is explained in the table below:

Digit	Description
*	Clears the input buffer
01	Site identifier
0	Command mode entry operator
12	Initial command mode password
#	Process the command
The controller is now in command mode	
*	Clears the input buffer
01	Site identifier
5	Use the Set Word operator
37	Select the *PTT Time-Out* register
10	Write value 2 into the register
#	Process the command
The PTT time-out is set to 10 minutes	
*	Clears the input buffer
01	Site identifier
0	Exit command code operator
#	Process the command
The controller exits command mode and returns to operational mode	

Table 66: Change PTT time-out sequence details

Once the controller returns to normal operation mode, the PTT time-out is set to 10 minutes..

Function (64) : Auxiliary Output Port register

Description:

When the auxiliary ports are configured as output ports, their values can be driven by the following instructions:

- Normal, Low & High Temperature Operator (registers 9, 10 and 11)
- Auxiliary Input Operator (registers 12 and 13)
- Manually using the Auxiliary Port register (64)

The *Auxiliary Port* register can be modified even when the *Auxiliary Input* or *Temperature* operators are used. However, note that the value that is entered in the register may be overwritten shortly after by one of the operators mentioned above. If manual control of the auxiliary output ports is desired, it is recommended to set the NOOP operator inside the *Temperature* and *Auxiliary Input* operator registers (9 to 13).

Refer to Table 7 on page 16 for the auxiliary output port register values and their corresponding values on the auxiliary output 1 and auxiliary output 2 ports.

The operators specified in the table below can be used to modify the *Auxiliary Output Port* register:

Operator	Code
Clear Word	1
Set Nibble	2
Set Bits	3
Clear Bits	4
Send Morse register value	7

Table 67 : Valid operators for the Auxiliary Output Port register

Example:

Assuming that the site ID is 01 and the initial Command Mode Password sequence is 12. The following DTMF must be sent in order to set the auxiliary output 1 On and the auxiliary output 2 Off:

```
* 01 0 12 #
* 01 2 64 1 #
```

Each digit in the sequence is explained in the table below:

Digit	Description
*****	Clears the input buffer
01	Site identifier
0	Command mode entry operator
12	Initial command mode password
#	Process the command
The controller is now in command mode	
*****	Clears the input buffer
01	Site identifier
2	Use the Set Nibble operator
64	Select the *Auxiliary Output Port* register
1	Write 1 in the register
#	Process the command
The aux output 1 is turned On and aux output 2 is turned Off	
*****	Clears the input buffer
01	Site identifier
0	Exit command code operator
#	Process the command
The controller exits command mode and returns to operational mode	

Table 68: Modify the Auxiliary Output Port register sequence details

The outputs on the controller now reflect the state of the *Auxiliary Output Register* value.

Function (65) : Temperature Sensor register

Description:

A temperature sensor is mounted on the circuit board and monitored by the micro-controller. The temperature is sampled every 2 seconds and stored inside the *Temperature Sensor* register. This register is a read-only register. Thus, a single operator can be used on this register:

Operator	Code
Send Morse register value	7

Table 69 : Valid operators for the Temperature Sensor register

The *Send Morse Register Value* operator is used, the temperature is sent as one or two Morse digits. However, when the temperature drops below 0°C, the Morse character "N" for **n**egative is sent before the temperature digits.

Example:

Assuming that the site ID is 01 and the initial Command Mode Password sequence is 12. The following DTMF sequence is transmitted in order to monitor the temperature:

```
* 01 0 12 #
* 01 7 65 #
```

Each digit in the sequence is explained in the table below:

Digit	Description
*****	Clears the input buffer
01	Site identifier
0	Command mode entry operator
12	Initial command mode password
#	Process the command
The controller is now in command mode	
*****	Clears the input buffer
01	Site identifier
7	Use the Set Nibble operator
65	Select the *Auxiliary Output Port* register
#	Process the command
The temperature is transmitted as Morse digits	
*****	Clears the input buffer
01	Site identifier
0	Exit command code operator
#	Process the command
The controller exits command mode and returns to operational mode	

Table 70: Transmit Morse temperature sequence details

Once the PTT is released, the temperature is transmitted as Morse characters. For example, if the temperature is 12°C, the following Morse characters are transmitted:

```
   1        2
•————   ••———
```

Where • corresponds to a Morse “dit” and — corresponds to a Morse “dah”.

If the temperature is -14°C, the following Morse characters are transmitted:

```
N        1       4
—•     •————  ••••—
```

The preceding “N” indicates that the temperature is below 0°C (negative).

Controller Return Codes

When entering DTMF sequences in the controller command mode, the controller may respond with a confirmation codes or error codes. When an error occurs, the error code is transmitted on the tail of the DTMF transmission. The table below lists all the controller return codes and their meaning.

Return Code	Morse	Description
A	• –	The controller is in command mode
E	•	The controller exits command mode and returns to normal operation mode
E1	• • – – – –	An invalid command mode operator was specified
E2	• • • – – –	An invalid auxiliary register operator was specified
E3	• • • • – –	The *Send Morse Register Value* operator was selected on the *Command Mode Password* register. For security reasons, the *Command Mode Password* register value cannot be transmitted using the *Send Morse Register Value* operator (7)

Table 1: Controller return code description

When using the *Set Nibble (2)* or the *Set Word (5)* operators in the command mode, the controller will send a Morse signal that corresponds to the value that has been written to the register. This Morse signal is a confirmation that the correct value has been saved in the register.

Appendix A

This section contains detailed information for the radio connections to the circuit board. Three radios can be connected to the circuit. The connectors for each radio is highlighted in different colors. The TX/RX/Aux and Polarity jumper connections are shown in Illustration 6:

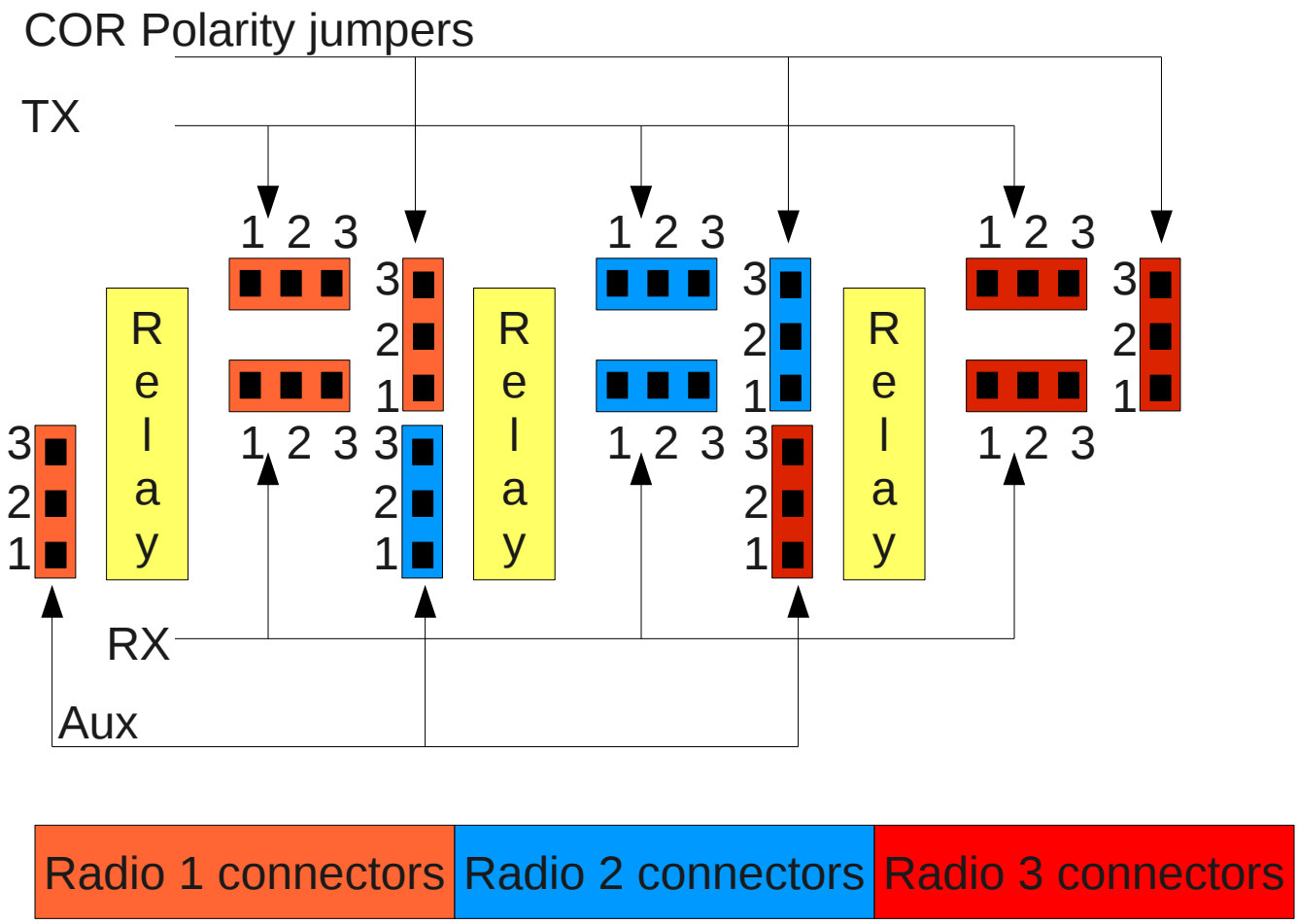


Illustration 6: Connector layout

Refer to the following tables for the connector pinouts:

Pin	Signal	Direction	Radio Connection
1	Audio TX	Output	Microphone input
2	Ground	--	Signal ground
3	PTT	Output	PTT

Table 1: TX connector pinout

Pin	Signal	Direction	Radio Connection
1	Audio RX	Input	Speaker/Audio out
2	Ground	--	Signal ground
3	COR	Input	COR

Table 2: RX connector pinout

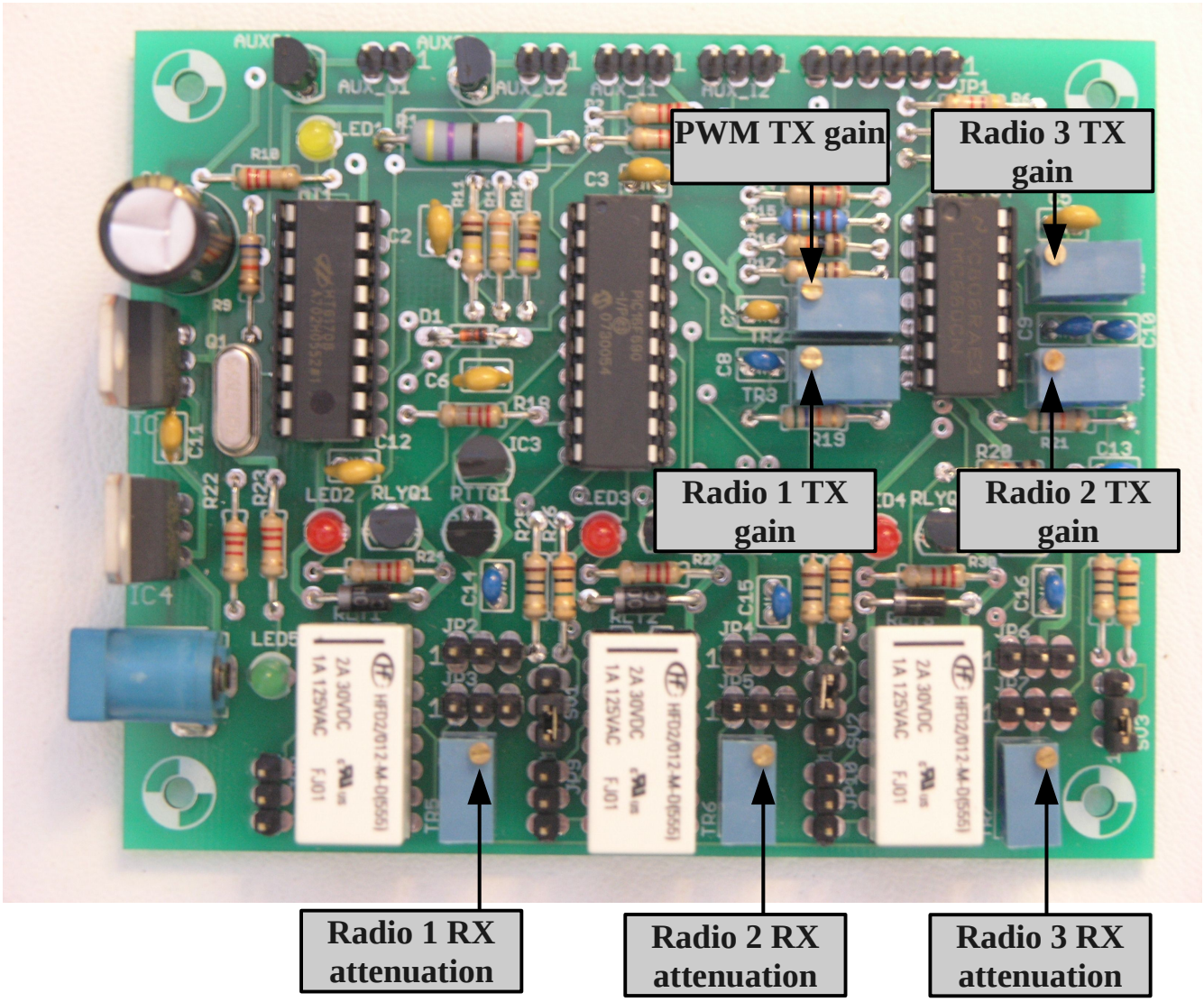
Pin	Relay Signal	RX Disabled	RX Enabled
1	NO	Open	Connected to Pin 3 (COM)
2	NC	Connected to Pin 3 (Com)	Open
3	COM	Connected to Pin 2	Connected to Pin 1

Table 3: Auxiliary relay pin header

COR polarity	Shunt
Active High	2 and 3
Active Low	1 and 2

Table 4: COR polarity jumper connections

Audio Level Potentiometers



Appendix B – Functions summary

Register	Function	Notes	See page
00	Command Mode Password	Use Set Array Operator (6)	34
01	Enable	0 – Disable 1 - Enable	36
02	COR polarity	0 – Active-High 1 - Active-Low	38
03	RX1PTT		40
04	RX2PTT		40
05	RX3PTT		40
06	MorsePTT		42
07	DitDuration		44
08	Radio1 Timeout	Enter value in minutes	46
09	Low Temp Operation		48
10	Normal Temp Operation		48
11	High Temp Operation		48
12	Aux Input 1 Operation		50
13	Aux Input 2 Operation		50
14	Aux Power-up values		52
15	SiteID		54
16	Max COR per Minute		56
17 – 22	Morse Site ID characters		58
23	Low Temp limit	Add +40 Ex: -10˚C = 30	60
24	High Temp limit	Add +40 Ex: -10˚C = 30	60
25	Aux Port Configuration		62
26	Transmit Morse ID		64
27	Transmit SiteID		66
28	Morse Amplitude		68
29/30	UserFunction1Reg/Operator		70

31/32	UserFunction2Reg/Operator		70
33/34	UserFunction3Reg/Operator		70
35/36	UserFunction4Reg/Operator		70
37	PTT Time-Out	Enter value in minutes	73
64	Auxiliary Output Port (RAM)		75
65	Temperature sensor (RAM)	Read-only register (Operator 7)	77

Technical support

For inquiries or technical assistance, please send an email to ve2lrs@gmail.com.