Minima Transciever Construction Guide, Release 7

Introduction

This document is a construction guide for the Minima all-band HF transceiver using the PCB (printed circuit board) designs developed by Adrian Preda YO4HHP. This is a work in progress, with updates being release sporadically and with evolving rights restrictions. The intent is to retain control until completion and then provide a final released under an appropriate rights mechanism (e.g. Creative Commons or GPL).

The build instructions will be performed in sections, according to PCB board. The first board constructed is the Digital Board, which contains the Arduino microcontroller and the Si570 VFO (variable frequency oscillator). The second board constructed will be the Main Board, which contains the RF, IF, and audio circuitry. The Digital Board will be required during the testing of some sections of the Main Board. The Serial Board will not be used in this build of the Minima transceiver. Instead, a USB-to-TTL serial adapter will be used.

The design of the Minima transceiver is evolving and it is the intent that this document keep track with that evolution in order to avoid divergence and maintain relevancy. That being said, new PCB designs for the Minima are on hold after the pioneering work accomplished by Adrian YO4HHP. To support this evolution under this constraint, it may be necessary to adapt the PCB to support the current state of the Minima design.

Minima Digital Board Assembly and Testing

This section presents the instructions for the assembly and test of the Minima Digital Board. Assembly and test of this board is relatively straightforward, but requires the mounting of a fewSMDs (surface mount devices). In addition, both hardware and software testing are required to assure that the Digital Board is functioning properly.

Minima Digital Board Theory of Operation

The Digital Board provides the controling 'brains' of the Minima transciever as well as the VFO used to convert radio frequency signals to and from the IF (intermdiate frequency) of 20 MHz. A schematic of the Digital Board is show in Illustration 49 in Appendix A. The Digital Board uses an Atmel ATMEGA328P microcontroller, programmed with an Arduino-compatible boot loader. The hardware configuration of the microcontroller is compatible with the Arduino Uno specification. The digital and analog control lines are assigned to support PTT (push-to-talk), keyline, function select, and frequency tuning inputs and filter, BFO, and CW tone control outputs. An interface to a sixteen-character by two-line LCD (liquid crystal display) is provided to visualize the user interface functions (frequency selection, sideband mode, transmit/recieve mode). An I2C (Inter-Integrated Circuit) bus provides interface to the VFO. Voltage regulators are provided to allow operation with a 12 volt power supply.

The VFO is a Silicon Labs Si570 Any-Rate I2C Programmable XO (crystal oscillator). This device generates square waves with accurate frequency (below 100 ppm) and very low phase jitter (below 1 ps). The CMOS version used on the Digital Board operates from below 10 MHz to above 160 MHz.

The Digital Board utilizes a potentiometer to control the tuning interface. By adjusting the pot, a voltage is varied, which is measured by an ADC (analog-to-digital converter). The voltage indicates the direction and step size that the VFO frequency should be changed. A central position of the potentiometer results in the frequency of the VFO remaining unchanged.

Minima Digital Board Parts List

The parts list for the Digital Board is presented in Table 1, as well as in Appendix B. It is based on the components list created by Sandeep Lohia VU3SXT. The component numbers listed will be used to reference the parts during construction. Note that component numbers for some headers on the Digital Board have been consolidated and assigned alternate component numbers. This was done in order to simplify construction and maintain contiguous pin headers where possible.

Part	Value	Device
C1	10 uF, 50 V	CPOL-EUE2.5-5
C2	22 pF SMD	C-EUC1206
C3	22 pF SMD	C-EUC1206
C4	0.1 uF	C-EU050-024X044
C5	50 uF 6.3 V	CPOL-EUE2.5-5
C6	0.1 uF	C-EU050-024X044
C7	0.1 uF	C-EU050-024X044
C8	0.1 uF	C-EU050-024X044
DIS1	LCD	16X2 TUXGR_16X2_R2
IC1	Arduino Uno	ATMEGA328-P
IC2	Si570	SI570
IC3	LM1117-3.3V	LM1117TO252
IC4	LM7805	78XXL
P1	12V	PINHD-1X2
P2	PTT, FN, TUNE, KEY	PINHD-1X9
P3	BFO, T/R, CW, BAND	PINHD-1X7
P4	5V, RST, SERIAL	PINHD-1X5
Q1	16 Mhz	CRYSTALHC49S
R1	10k	TRIM_EU-CA6V
R2	33R	R-EU_0207/15
R4	1k	R-EU_0204/5
R5	4k7	R-EU_0204/5
R6	4k7	R-EU_0204/5
R7	10k	R-EU_0204/5
X2	VFO	BU-SMA-V

Table 1: Parts list for the Minima Digital Board.

Minima Digital Board Assembly Instructions

Assembly of the Digital Board will be performed in several sessions. Each session will be followed by testing of that section. It is important to follow this assemble-then-test approach in order to assure successful construction of the Digital Board and eventual success in using the Minima transceiver.

Jumper Installation

The PCB for the Digital Board has circuit traces on only one side. In order to complete the circuit routing, it is necessary to add some wire jumpers. The placement of these jumpers is shown in

Illustration 1. Wires of different colors were used for the power buss (red) and various signals (orange).

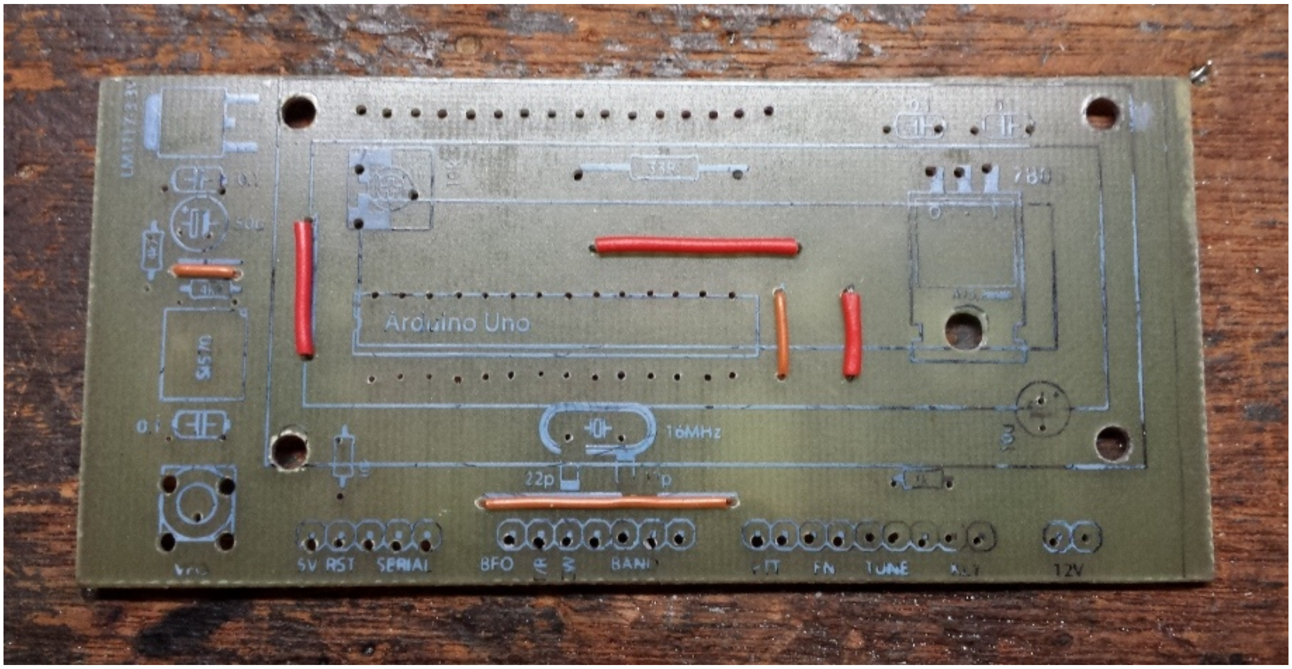

Illustration 1: Jumper placement for the Digital Board.

Power Buss Assembly

The following steps describe the installation of the components for the power buss of the Digital Board.

1. Install the LM1117 3.3 volt regulator on the circuit side of the Digital Board. The LM1117 is a surface mount device, but it is easy to solder to the board. First, solder the input power pin, indicated by the red '1' in Illustration 2, to hold the regulator in place on the board. Align the regulator so that the ground pin (red '2') rests on the ground pad and the output tab (red '3') rests on the large output pad. It is important to make sure that the tab only touch the output pad and does not short to the ground plane around it. If necessary, re-melt the solder on the input pin and shift the regulator for proper positioning. Once this is done, solder the ground pin and the output tab, as shown in Illustration 2.

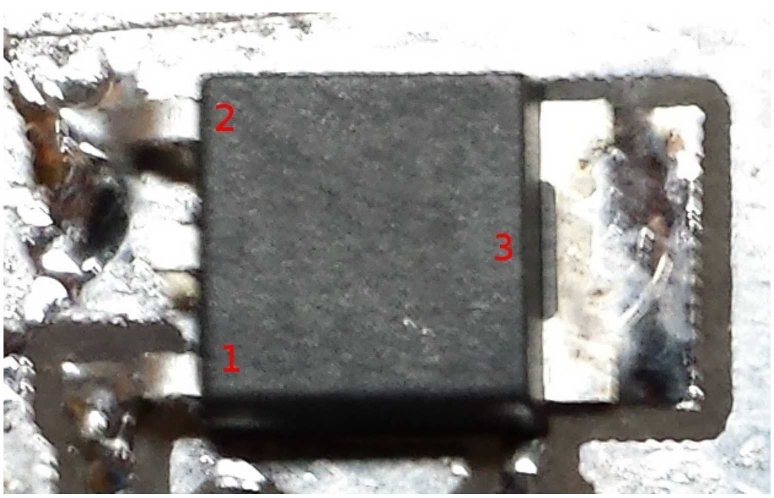

Illustration 2: Detail of the installation of the 3.3 volt voltage regulator.

2. Solder the 0.1 uF capacitors (C4, C6, C7, and C8) on to the Digital Board. These are the gold, rectangular capacitors shown in Illustration 3.
3. Solder the 10 uF, 50 volt electrolytic capacitor C1 on to the Digital Board. This capacitor is shown on the center right side of the Digital Board in Illustration 3.
4. Solder the 50 uF, 6.3 volt electrolytic capacitor C5 on to the Digital Board. This capacitor is shown on the upper left side of the Digital Board in Illustration 3.
5. Solder the 10 Kohm trimmer potentiometer R1 on to the Digital Board. This potentiometer is shown on the upper left side of the Digital Board in Illustration 3.
6. Solder P1, the two-pin, right-angle header, to the pin pads labeled '12V'. This is located on the lower right side of the Digital Board in Illustration 3.
7. The leads for the LM7805 5 volt regulator IC4 need to be bent to allow the regulator to lie flat on the heat sink and circuit board. First, test fit the regulator and heat sink to the Digital Board using a 6-32 screw to determine where to bend the leads. Assure that the heatsink does not contact the 10 uF capacitor. Trimming of the heatsink material may be necessary. Remove the regulator and bend the leads to form a right angle and to allow there insertion into the three circuit pad holes.
8. Apply some heatsink compound to the regulator IC4 and press it against the heat sink. Use the a 6-32 screw to maintain aligment of the regulator and heatsink and insert the entire assembly into the circuit board. Mechanically secure the regulator and heatsink to the circuit board with a #6 flat washer, #6 lock washer, and 6-32 hex nut.
9. Solder the three pins of the LM7805 regulator IC4 and remove the excess leads.

The completed assembly of the power buss components is shown in Illustration 3 and Illustration 4.

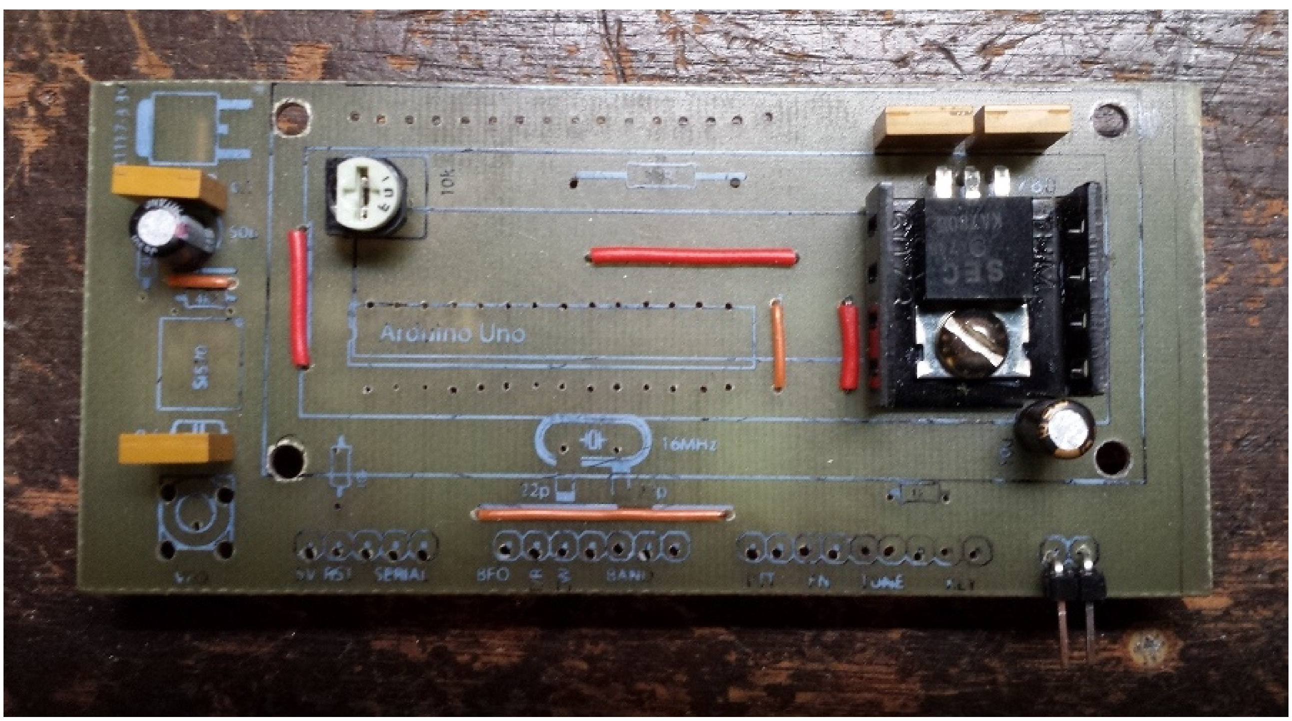

Illustration 3: Component side of the Digital Board with all power buss components installed.

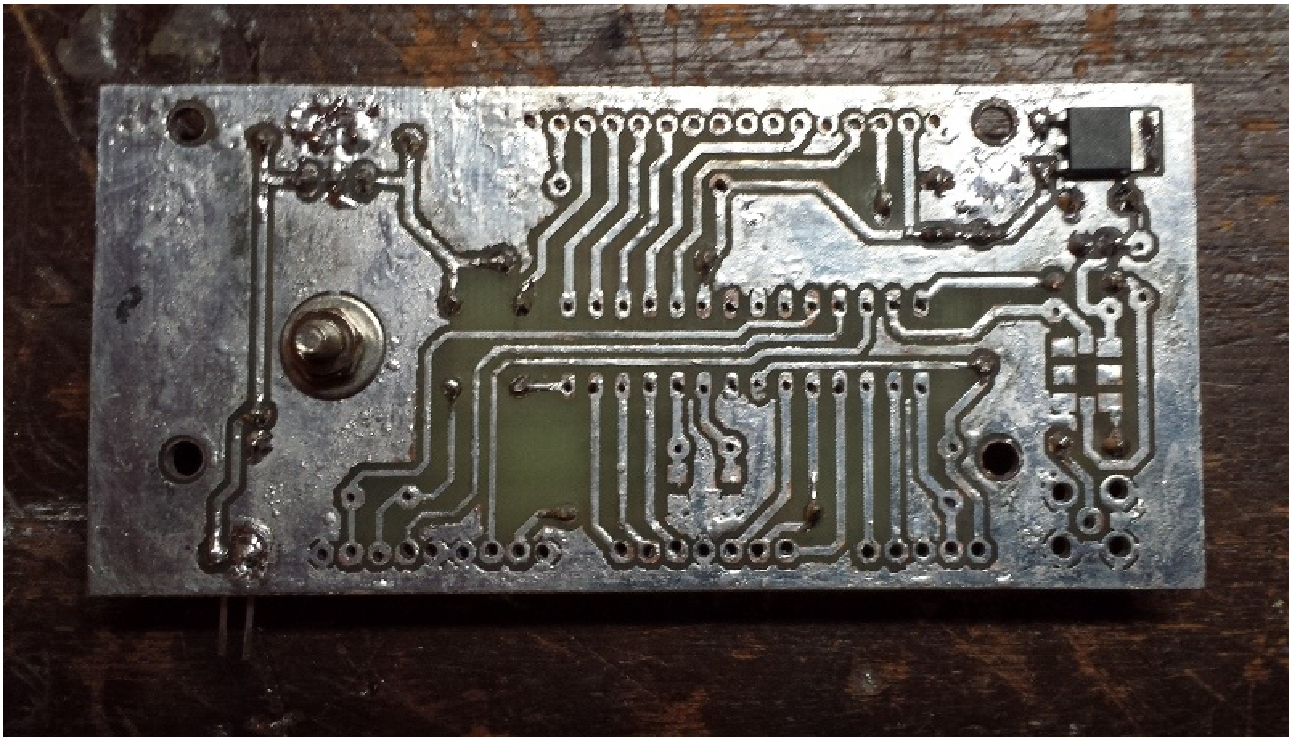

Illustration 4: Circuit side of the Digital Board with all power buss components installed.

Power Buss Testing

The following steps describe the testing of the power buss of the Digital Board.

1. Connect a 12 volt power supply to P1 with the power supply turned off. Use a multimeter to measure the voltage between the +12 volt and ground pins of P1. Turn on the power supply and note whether the multimeter displays approximately 12 volts. If the multimeter indicates 12 volts, proceed on with the next testing step. If it does not, immediately turn off the power supply and check for shorts on the Digital Board. It is also possible that one of the regulators or capacitors may be faulty.
2. Measure the output of the LM7805 voltage regulator. It should indicate approximately 5 volts. If the multimeter indicates 5 volts, proceed on with the next testing step. If it does not, immediately turn off the power supply and again check for shorts or damaged components on the Digital Board. If the multimeter does indicate 5 volts, also check for 5 volts on the input pin for the LM1117 and pins 7 and 20 of the ATMEGA328 microcontroller.
3. Measure the voltage on the tab of the LM1117 voltage regulator. It should indicate approximately 3.3 volts. If it does not, immediately turn off the power supply and again check for shorts or damaged components on the Digital Board. If the multimeter does indicate 3.3 volts, also measure the voltage on pin (TBD) where the Si570 will be soldered. If all voltages measure correctly, power buss testing has been successfully completed.

Arduino Microcontroller Assembly

The following steps describe the installation of the components for the Arduino microcontroller ofn the Digital Board. **Please note: DO NOT install the ATMEGA328 microcontroller at this time.**

1. On the circuit side of the Digital Board, solder two 22 pF, 1206 surface mount capacitors C2 and C3, as show in Illustration 5. Soldering of these components must be done carefully in order to prevent shorting to the surrounding ground traces or cracking the capacitors. One technique is to first apply a small amount of solder to one pad. Using a pair of tweezers, position one end of an SMD capacitor to the pad with the solder and use a soldering iron to remelt the solder. The solder should wick to the end of the SMD capacitor held against the solder. After allowing the solder to cool are resolidify, solder the SMD capacitor to the other pad.

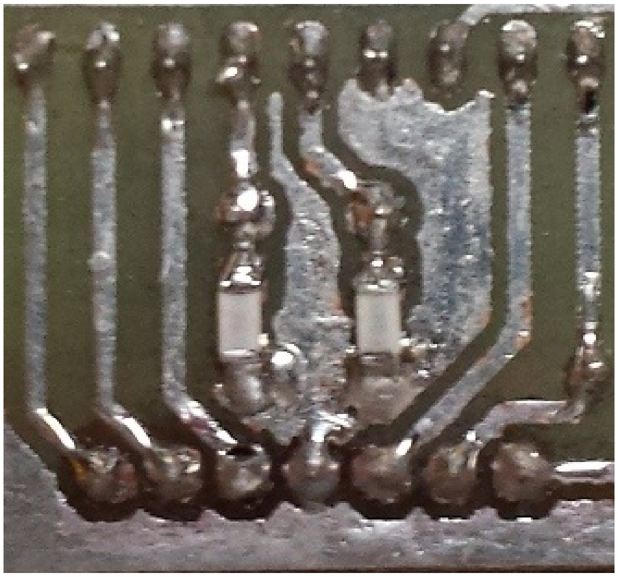

Illustration 5: Installation of 22 pF capacitors.

2. Solder the 1 Kohm pull-up resistor R4 on to the Digital Board. This resistor is shown on the lower right-hand side of Illustration 6. If a ¼ Watt resistor is used, it will need to be install in a 'hair pin' fashion.
3. Solder the 10 Kohm pull-up resistor R7 on to the Digital Board. This resistor is shown on the lower left-hand side of Illustration 6. If a ¼ Watt resistor is used, it will need to be install in a 'hair pin' fashion.
4. Solder the 16 MHz crystal X2 on to the Digital Board. The crystal is shown on the lower center of Illustration 6.
5. Solder the 28-pin DIP socket on to the Digital Board. The socket is shown on the center of Illustration 6.
6. Solder the 9-pin right-angle header on to the Digital Board. The socket is shown on the lower right of Illustration 6.
7. Solder the 9-pin right-angle header P2 on to the Digital Board. The socket is shown on the lower right of Illustration 6.
8. Solder the 7-pin right-angle header P3 on to the Digital Board. The socket is shown on the lower center of Illustration 6.
9. Solder the 5-pin right-angle header P4 on to the Digital Board. The socket is shown on the lower center of Illustration 6.

10. Solder the 16-pin SIP socket into the display connection pads labeled DIS1 on to the Digital Board. The socket is shown on the upper center of Illustration 6.

The completed assembly of the Auduino microcontroller components is shown in Illustration 6 and Illustration 7.

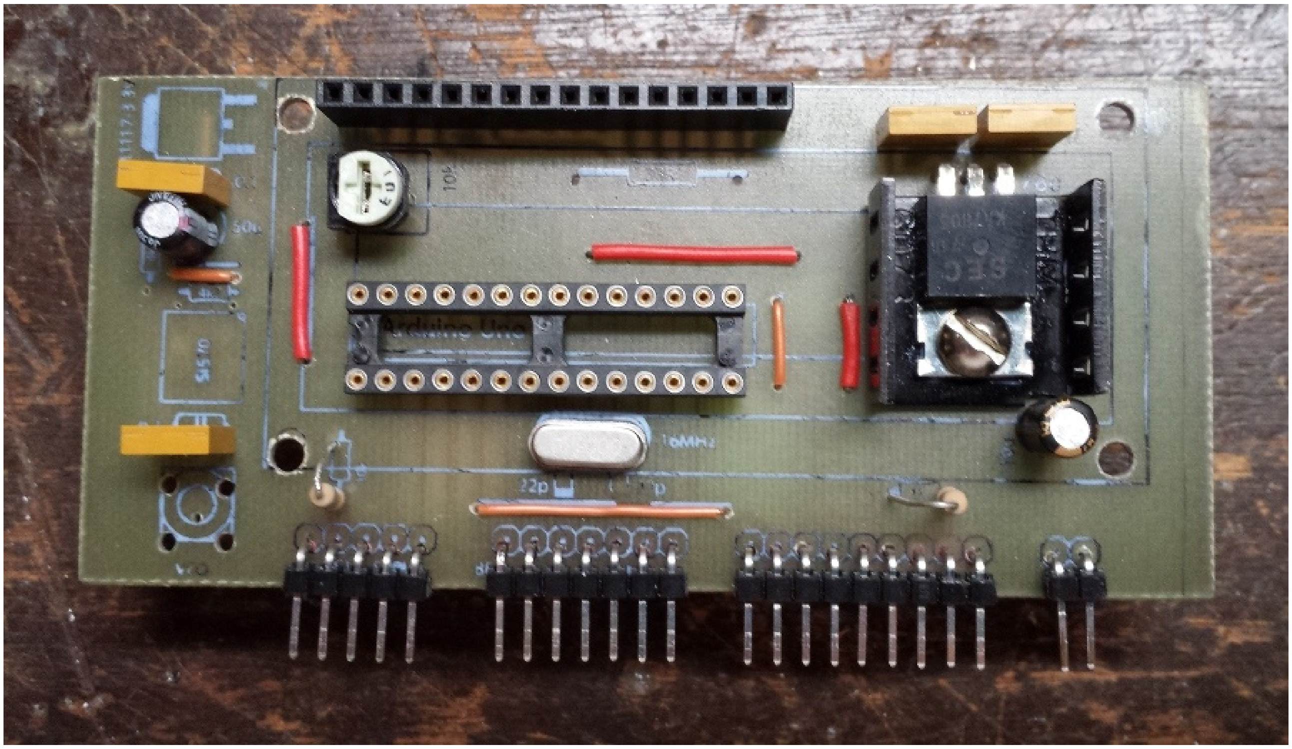

Illustration 6: Component-side installation of Arduino components on the Digital Board.

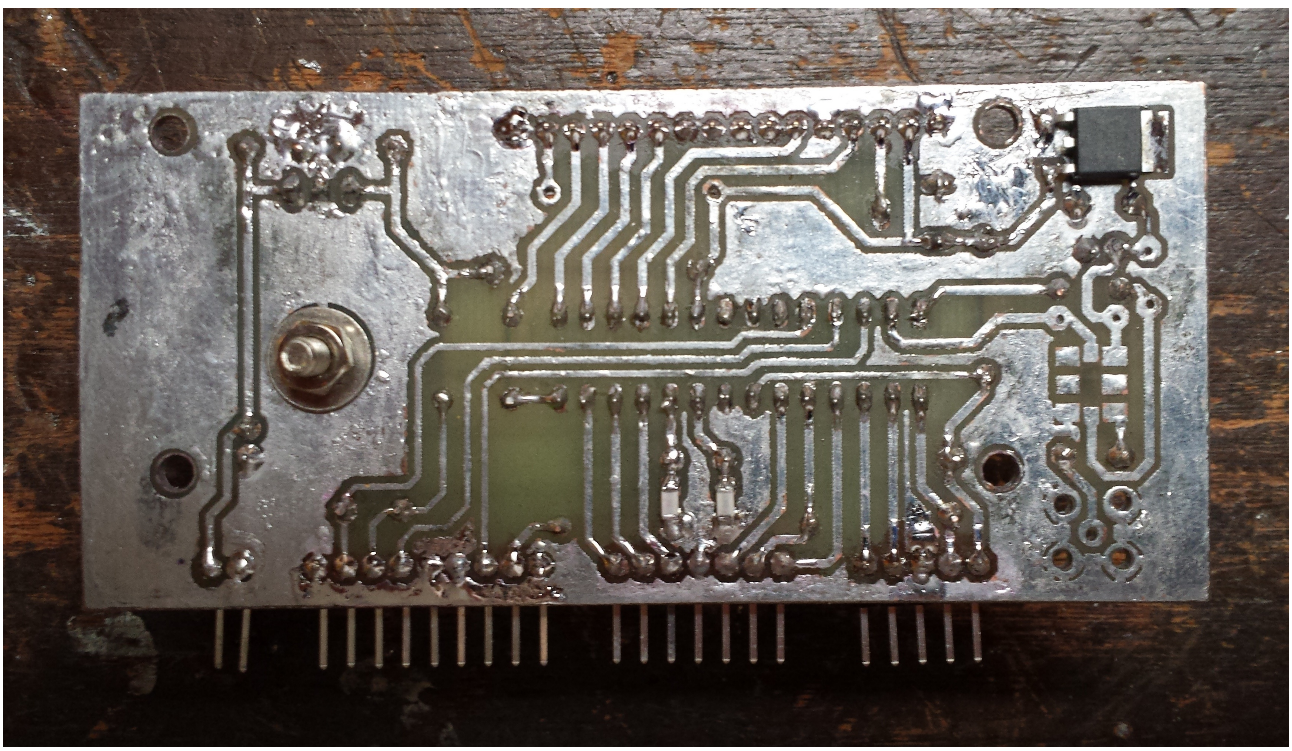

Illustration 7: Circuit-side installation of Arduino components on the Digital Board.

Static Power Test of the Arduino Microcontroller Components

The following steps describe the static power test of the Arduino of the Digital Board.

1. Repeat the Power Buss testing described above to ensure that no shorts or damage has occured to the Digital Board.
2. Use a multimeter to measure 5 volts on the PTT pin and adjacent TUNE voltage pin on P2.
3. Use a multimeter to measure 5 volts on the 5V pin and RST pin on P4.
4. If all voltages measure correctly, then testing has been successfully completed.

Installation of the Arduino Microcontroller

The following steps describe the installation of the Arduino microcontroller into the Digital Board. It is assumed that the ATMEGA328 microcontroller has already had an Arduino bootloader programmed into it. This must be done prior to insertion of the microcontroller into the Digital Board. There are no provisions made for in-circuit serial programming (ICSP) in the Digital Board.

1. Prior to pluging the Arduino microcontroller (ATMEGA328) into the Digital Board, gently bend its leads so that they are parallel with the sides of the DIP package and aligned with the socket pins. This will reduce the possibility of bending a lead during insertion into the socket.
2. Align the notch on one end of the Arduino microcontroller with the notch on the 28-pin DIP socket. Proper orientation must be ensured to avoid distroying the microcontroller when power is applied to the Digital Board.
3. Gently insert the Arduino microcontroller into the 28-pin DIP socket. Take care to not bend any of the microcontroller's leads, as they are fragile and can be bent or broken, rendering the microcontroller unusable.

The installation of the Arduino microcontroller is shown in Illustration 8. In this example, a Sparkfun labeled ATMEGA328 with Arduino Optiboot (Uno) pre-programmed into it.

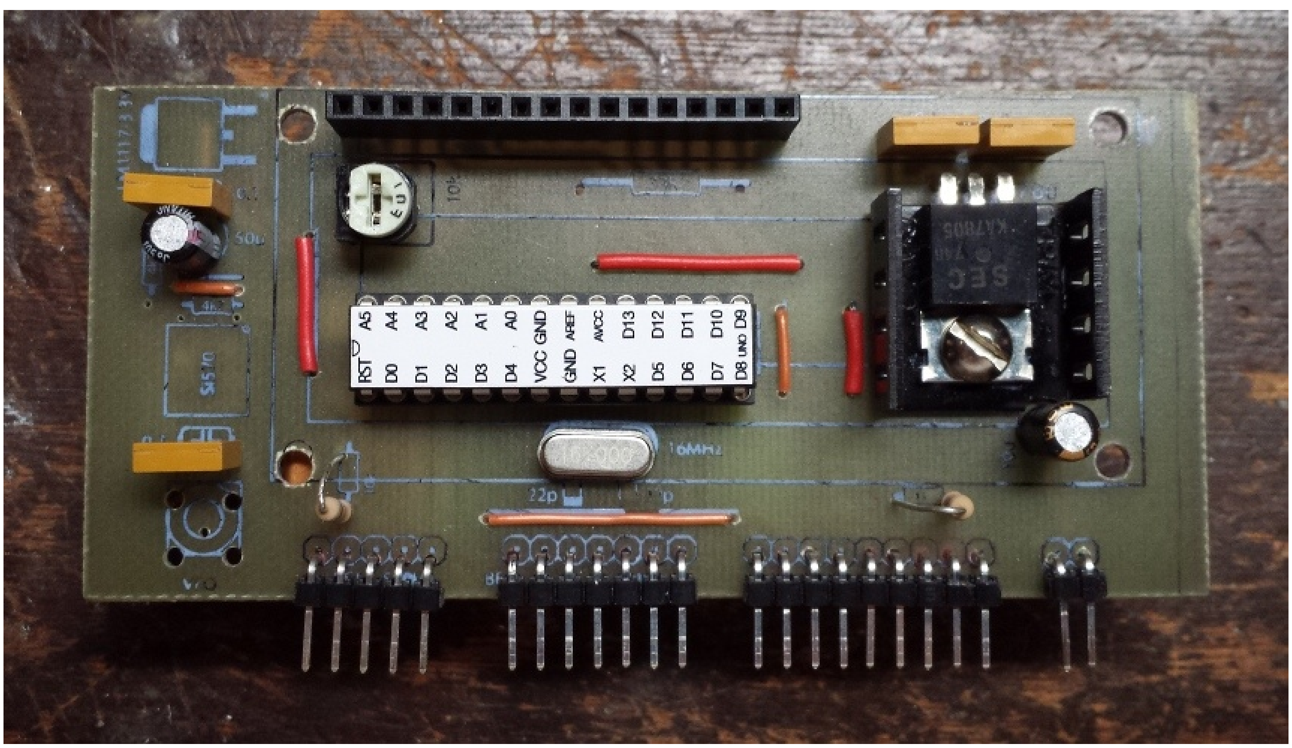

Illustration 8: Arduino microcontroller installed in the Digital Board.

Functional Test of the Arduino Microcontroller

Testing of Arduino microcontroller involves programming it with several Arduino sketches and seeing that the desired behavior is exhibited. There are two ways to program the Digital Board. The first method is to connected the Minima Serial Board to the Digital Board and to a RS232 serial port on a computer running the Arduino IDE. The second method is to connect a USB-to-TTL serial adapter to the Digital Board and to a USB port on a computer running the Arduino IDE. Each method has its advantages and disadvantages. Some of these trade-offs are listed in Table 2.

Table 2: Some trade-offs for RS-232 versus USB programming interfaces.

Method	RS-232 Serial Board	USB-to-TTL Serial Adapter
Pros	Requires only an RS-232 interface cable and a PC with an RS-232 port to program the Arduino Controller.	Auto-reset feature simplifies programming. Construction cost for the Minima is less due to no Serial Board and its components.
Cons	Newer computers may lack an RS-232 port. Proper timing of the reset signal manually may make programming of the Arduino microcontroller difficult. A Serial Board must be constructed for each Minima built.	Requires a computer with a USB port. Requires a USB-to-TTL serial adapter and a USB cable. Require construction of an interface cable between the USB-to-TTL serial adapter and the Digital Board.

For this assembly guide, the Arduino microcontroller will be programmed with a USB-to-TTL serial adapter. Consequently, it is necessary to construct an appropriate interface cable. Given that the pinout of the TTL serial interface may vary for different USB-to-TTL serial adapters, the cable will be described using signal names only. The specification of the programming cable is provided in Table 3.

Table 3: Pin specifications for the Arduino microcontroller programming cable.

USB/TTL Serial Adapter Pin	Five-Pin Connector on Digital Board
Vcc	5V (left-most pin)
DTR	RST (pin between center and left-most pin) – Place a 0.1 uF (approximately) capacitor in series between DTR and RST pins
Ground	Ground (center pin)
TX	RX (pin between center and right-most pin)
RX	TX (right-most pin)

A programming cable was constructed using crimp-on pins and connector shells taken from discarded PC wiring harnesses. A capacitor with a value of approximately 0.1 uF was placed between the DTR and the reset (RST) pin as part of the auto-reset capability. The completed programming cable is shown in Illustration 9.

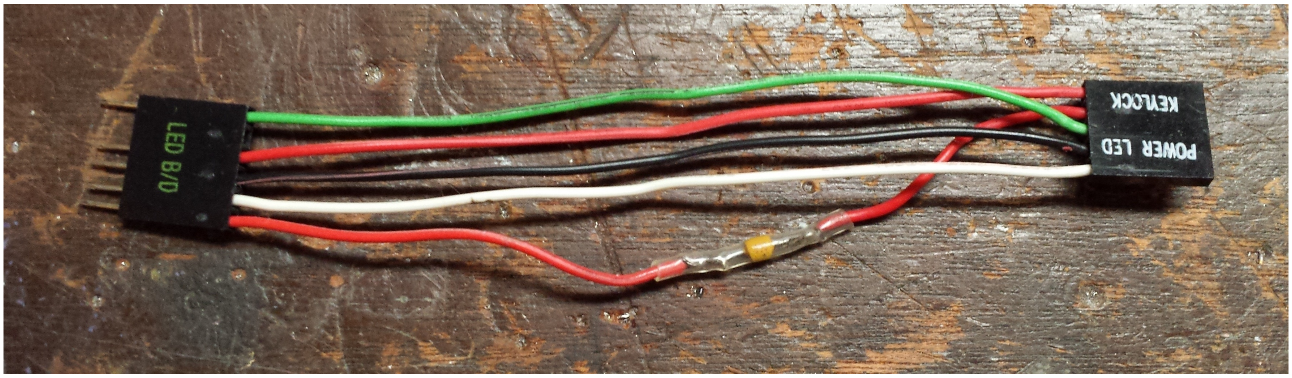

Illustration 9: Interface cable from the USB-to-TTL serial adapter to the Digital Board.

The connection of the USB-to-TTL serial adapter to the Digital Board via the programming cable is shown in Illustration 10.

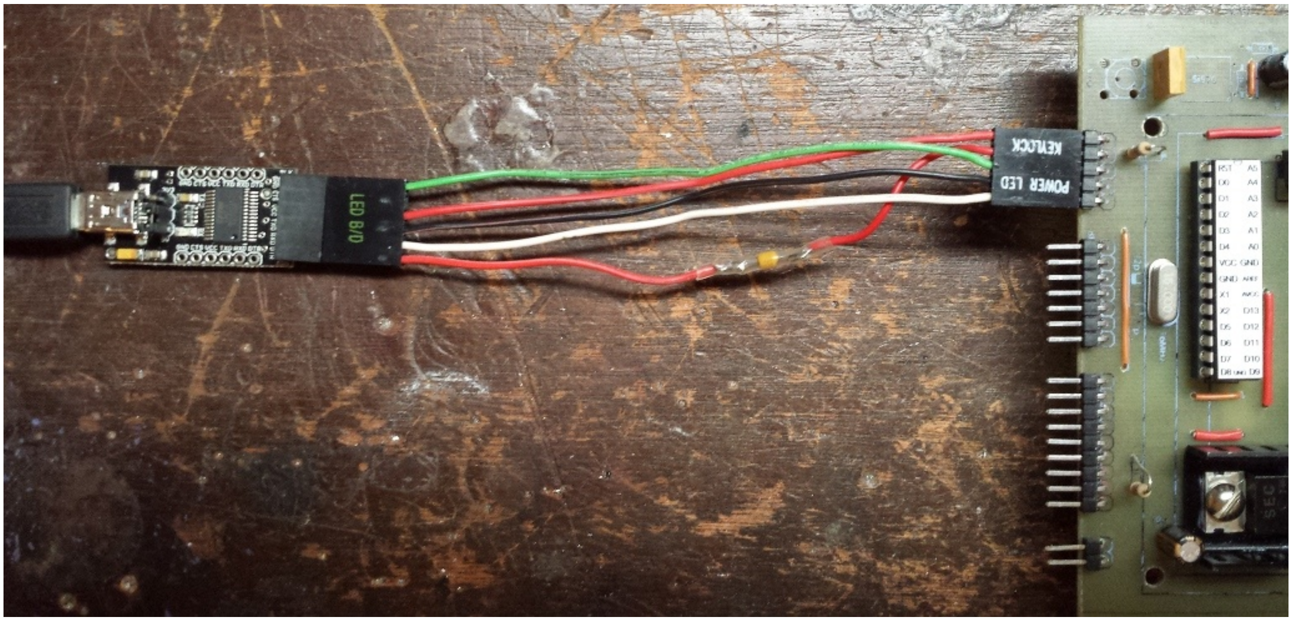

Illustration 10: Connection of USB-to-TTL serial interface to the Digital Board.

Arduino Microcontroller 'Blink' Test

The following steps describe the initial test of Arduino microcontroller using the example 'Blink' sketch. This test is crucial, as it demonstrates basic, but essential board functionality.

1. Temporarily connect an LED in series with an 270 ohm resistor to the Digital Board via the LCD connector. Plug the cathode of the LED into the ground pin on the end of the LCD connector, next to potentiometer R1. The anode of the LED is soldered to the 220 ohm resistor. Plug the other end of the resistor into to the fourth pin from the end of the LCD connector next, to potentiometer R1. This corresponds to pin 19 on the ATMEG328 and is known as 'D13' on the Arduino Uno. The arrangement is shown in Illustration 11.
2. If not done already, attach one end of the programmer cable to the USB-to-TTL serial adapter output and the other end to the five-pin connector on the Digital Board.
3. If not done already, attach one end of the USB cable to the USB port on the computer and the other end to the USB port on the USB-to-TTL serial adapter. This will allow the computer's operating system to enumerate the adapter and supply it with a device identifier.

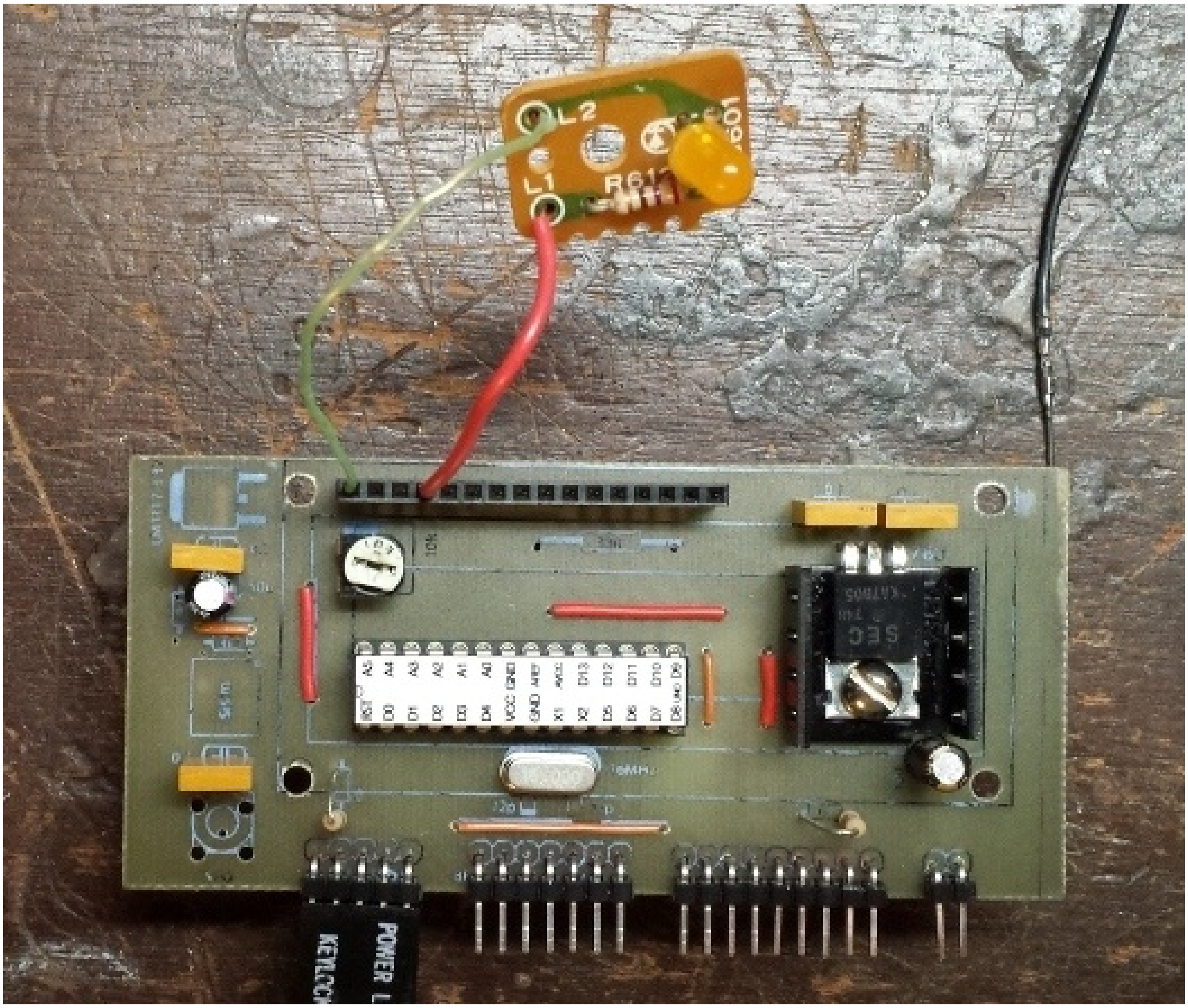

Illustration 11: Temporary connection of LED and resistor for the 'Blink' test.

4. Start the Arduino IDE and load the 'Blink' sketch. This sketch can be loaded from the Arduino IDE menu bar via Files->Examples->01.Basics->Blink. A listing of the 'Blink' sketch is provided in Text 1.
5. Ensure that the correct Arduino board type is selected. The Arduino Uno board type can be selected from the Arduino IDE menu bar via Tools->Board->Arduino Uno.
6. Ensure that the correct serial port is selected to talk to the USB-to-TTL serial adapter. The name of the serial port depends upon the operating system being used. To determine this, consult the manufacturer of the USB-to-TTL serial adapter and operating system guidelines. The serial ports available to the Arduino IDE can be selected from the menu bar via Tools->Serial Port
7. Select the 'Upload' button on the Arduino IDE button bar to compile and load the 'Blink' sketch into the Digital Board.
8. If the compile and load of the sketch is successful and there are no wiring error in the Digital Board, the LED should blink once per second. If this does not occur, there may be wiring or soldering errors on the Digital Board or the connections between the computer, USB-to-TTL serial adapter, and the Digital Board may be incorrect.

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);               // wait for a second
}
```

Text 1: Listing of the Arduino 'Blink' sketch.

Arduino Microcontroller LCD Test

The following steps describe the testing of the Arduino microcontroller with the LCD display. This will be performed using a simple Arduino sketch, lcd_testing.ino developed by Thomas Sarlandie.

1. Temporarily connect the 16x2 LCD display to the Digital Board by plugging it into the LCD connector. To avoid placing mechanical stress on the LCD connector, insert stand-offs between the LCD's mounting holes and the corresponding holes on the Digital Board and securing them with 2x56 screws and nuts.
2. If not done already, attach one end of the programmer cable to the USB-to-TTL serial adapter output and the other end to the five-pin connector on the Digital Board.
3. If not done already, attach one end of the USB cable to the USB port on the computer and the other end to the USB port on the USB-to-TTL serial adapter. This will allow the computer's operating system to enumerate the adapter and supply it with a device identifier.
4. Start the Arduino IDE and load or type in the sketch lcd_testing.ino. This sketch can be loaded from the Arduino IDE menu bar via Files->Open. A listing of the sketch is provided in Text 2 if the sketch file is not available.
5. Ensure that the correct Arduino board type is selected. The Arduino Uno board type can be selected from the Arduino IDE menu bar via Tools->Board->Arduino Uno.
6. Ensure that the correct serial port is selected to talk to the USB-to-TTL serial adapter. The name of the serial port depends upon the operating system being used. To determine this, consult the manufacturer of the USB-to-TTL serial adapter and operating system guidelines. The serial ports available to the Arduino IDE can be selected from the menu bar via Tools->Serial Port
7. Select the 'Upload' button on the Arduino IDE button bar to compile and load the sketch into the Digital Board.

8. If the compile and load of the sketch is successful and there are no wiring error in the Digital Board, the LCD should display the message 'Banana' twice, as shown in Illustration 12. If this does not occur, there may be wiring or soldering errors on the Digital Board or the connections between the computer, USB-to-TTL serial adapter, and the Digital Board may be incorrect.

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(13, 12, 11, 10, 9, 8);

void setup() {
  lcd.begin(16, 2);
  lcd.clear();
  lcd.print("Banana");
  lcd.setCursor(10, 1);
  lcd.print("Banana");
}

void loop() {
}
```

Text 2: Listing of the Arduino LCD test sketch.

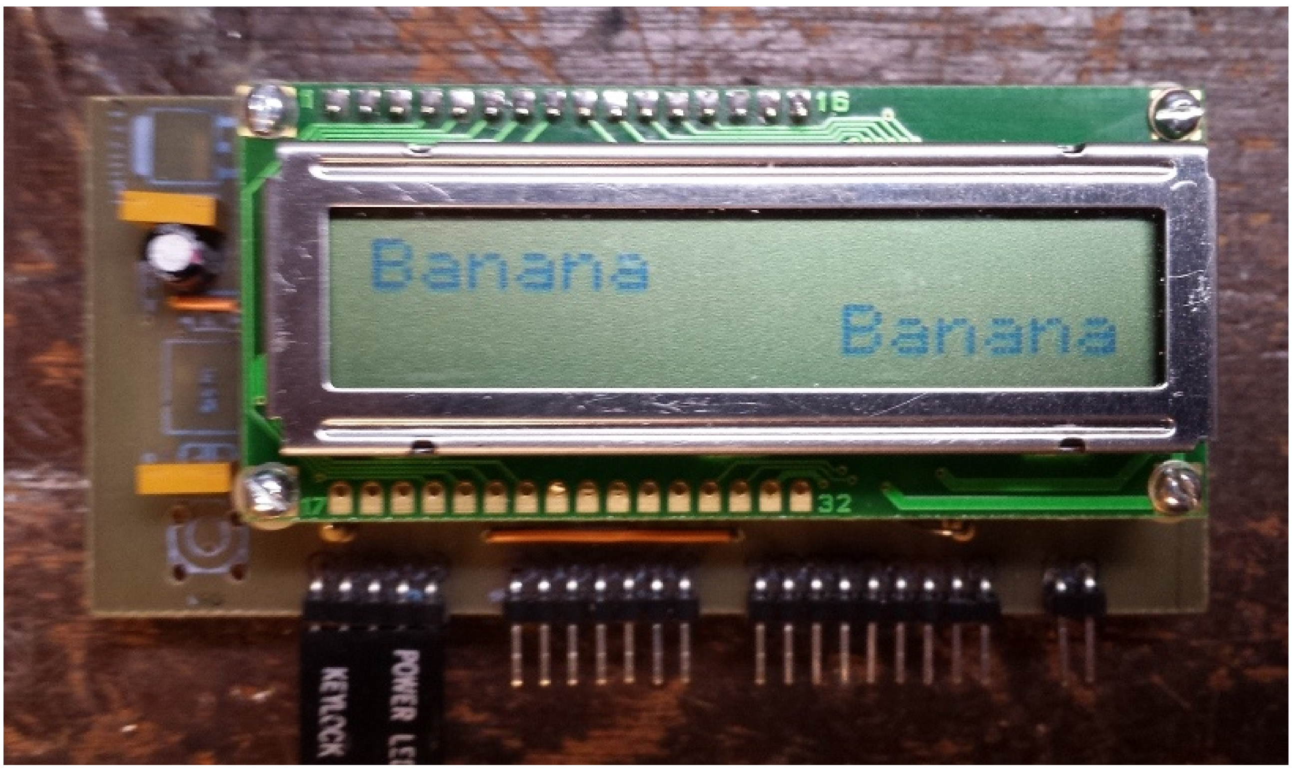

Illustration 12: LCD display results from the Arduino microcontroller LCD test.

Arduino Microcontroller I/O Test

The following steps describe the testing of digital and analog I/O of the Arduino microcontroller as it will be used by the Minima transceiver. This will be performed using a simple Arduino sketch, io_testing.ino, which is listed in Appendix C. Note that the sketch does not test the I2C interface used to communicate with the Silicon Labs Si570.

1. Testing of the Arduino I/O will involve changing inputs and monitoring outputs. This can be done through a set of control cables. A complete set of cables will consist of three with momentary pushbutton switches, one with a potentiometer, and one with four LEDs and current-limiting resistors. An example set of cables is shown in Illustration 13. The three switches cables were acquired, unmodified, from discarded PC enclosure wiring harnesses. A 100 kohm linear potentiometer was soldered to a similar cable. The potentiometer should be wired such that voltage on the wiper decreases it is rotated clockwise. The LEDs are wired so that each anode has a 270 ohm resistor and all of the cathodes are connected together to ground. The ground used is the center pin on a seven-pin connector shell. The LED/resistor pairs are connected to the shell at the locations corresponding to BFO pin and the three BAND pins.

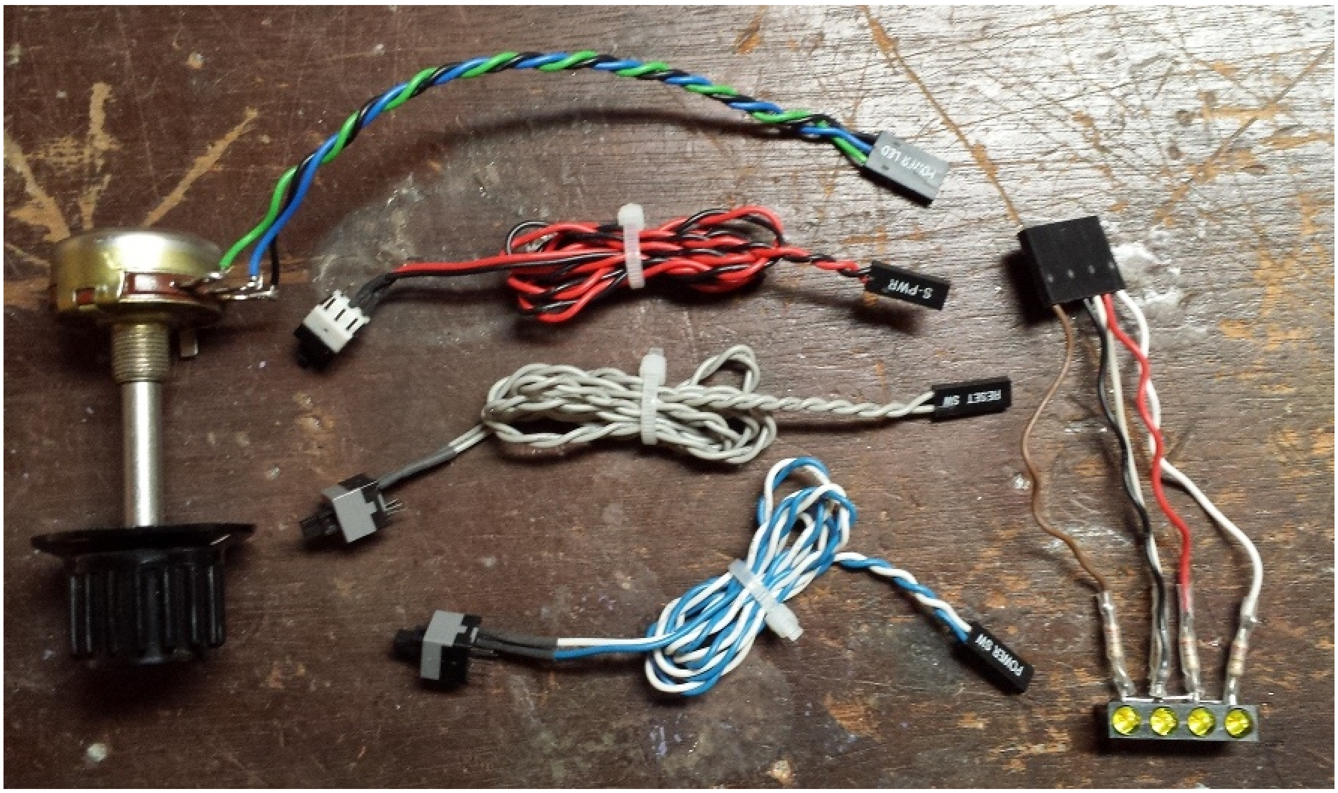

Illustration 13: Cable set used for Arduino Microcontroller I/O testing.

2. Plug one switch cable onto the PTT pin and the ground pin adjacent to it. Plug one switch cable onto the FN pin and the ground pin adjacent to it. Plug one switch onto the KEY pin and the ground pin adjacent to it. Plug the potentiometer cable onto the three TUNE pins such that voltage on the wiper will decrease as it is rotated clockwise. Finally, plug the seven-pin connector shell onto the seven-pin connector, assuring that the LED/resistor pairs are connected to the BFO and BAND pins. The proper setup is shown in Illustration 14.
3. If not done already, attach one end of the programmer cable to the USB-to-TTL serial adapter output and the other end to the five-pin connector on the Digital Board.
4. If not done already, attach one end of the USB cable to the USB port on the computer and the other end to the USB port on the USB-to-TTL serial adapter. This will allow the computer's operating system to enumerate the adapter and supply it with a device identifier.
5. Start the Arduino IDE and load or type in the sketch io_testing.ino. This sketch can be loaded from the Arduino IDE menu bar via Files->Open. A listing of the sketch is provided in Appendix C, if the sketch file is not available.

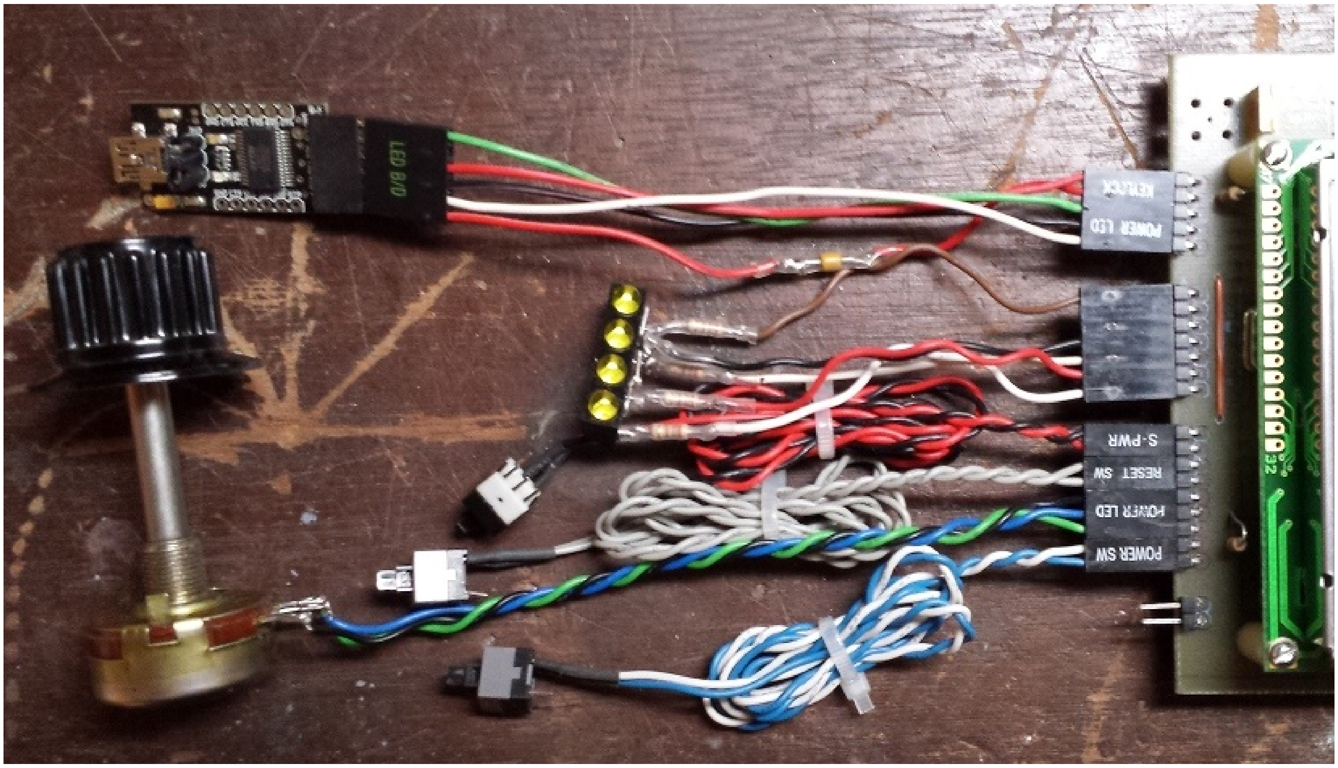

Illustration 14: Cable connections for the Arduino Microcontroller I/O testing.

6. Ensure that the correct Arduino board type is selected. The Arduino Uno board type can be selected from the Arduino IDE menu bar via Tools->Board->Arduino Uno.
7. Ensure that the correct serial port is selected to talk to the USB-to-TTL serial adapter. The name of the serial port depends upon the operating system being used. To determine this, consult the manufacturer of the USB-to-TTL serial adapter and operating system guidelines. The serial ports available to the Arduino IDE can be selected from the menu bar via Tools->Serial Port
8. Select the 'Upload' button on the Arduino IDE button bar to compile and load the sketch into the Digital Board.
9. If the compile and load of the sketch is successful and there are no wiring error in the Digital Board, the LCD should display should appear, as shown in Illustration 15. If this does not occur, there may be wiring or soldering errors on the Digital Board or the connections between the computer, USB-to-TTL serial adapter, and the Digital Board may be incorrect.
10. Adjusting the potentiometer should cause the POT reading to vary between 0 and 1023. Pressing the switch connected to the TX pin should cause the TX/RX value to display '0'. A value of '1' should be displayed the switch is released. Pressing the switch connected to the KEY pin should cause the AK value to display '0'. A value of '1023' should be displayed when the switch is released. Pressing the switch connected to the FN pin should cause the FN value to display a valuse close to '0'. A value close to '1023' should be displayed when the switch is released. The LEDs should be turning on and off at different rates, essentially counting from zero to sixteen in binary fashion repeatedly. Any errors in operation is probably caused by a bad cable connection or an error in the test sketch.

Illustration 15: Appearance of LCD display for the Arduino Microcontroller I/O Testing sketch.

Installation of the VFO Assembly

The following steps describe the installation of the components for the Silicon Labs Si570 clock chip.

1. Solder the two 4.7 kohm resistors R5 and R6. These components are the 1/8 Watt resistors shown in the center of Illustration 16.
2. Solder the SMA right angle PC board mount connector. This componemt is shown on the right-hand side of Illustration 16.
3. Invert the Digital Board so that the circuit side of the PCB is visible. Apply liquid flux to the pads where X2, the Si570 clock generator will be soldered. Carefully position the Si570 on the solder pads such that pin 1 (marked with a dot) is nearest to resistors R5 and R6. Assure that the device is centered on the pads. While holding the device in place solder pin 1 with as little solder as possible to hold the device in place. Solder the remaining pins to their corresponding pads, as shown in Illustration 17. Keep the amount of solder used to a minimum.

This completes the assembly of the clock generator componenets.

Illustration 16: Component side of Digital Board with the Si570 support components installed.

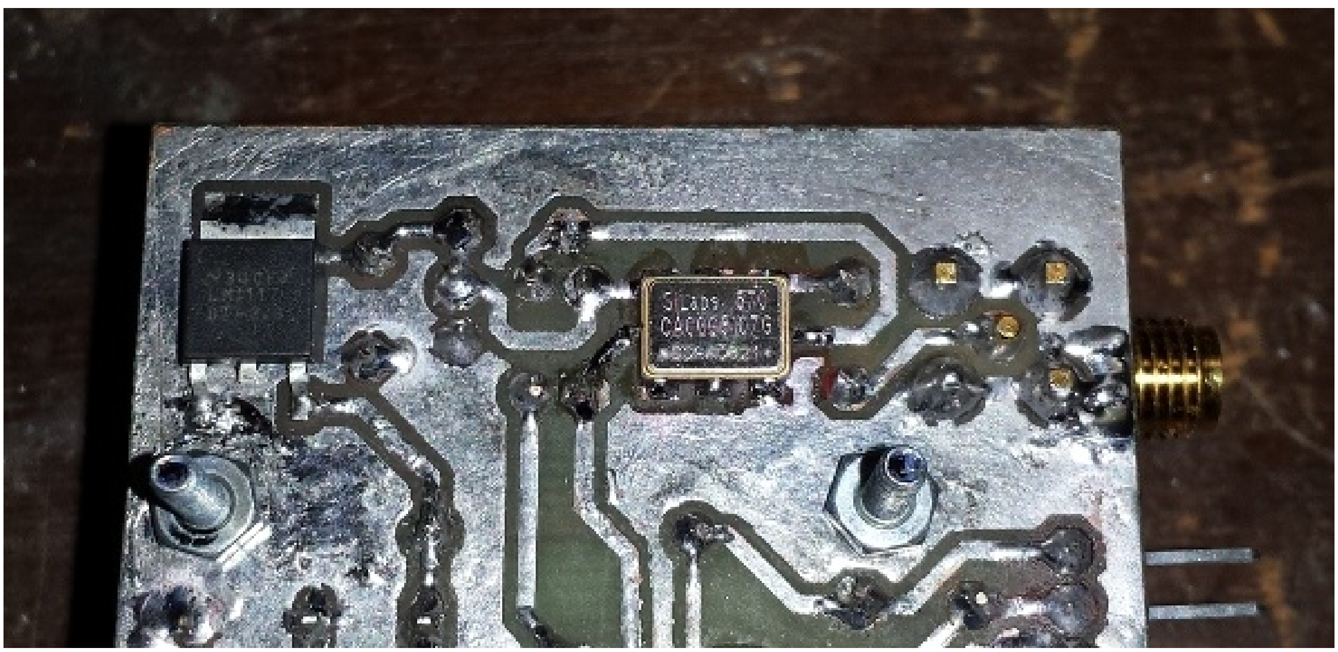

Illustration 17: Circuit side of the Digital Board with the Si570 clock generator installed.

Testing of the VFO

Verification of proper clock generator operation involves, first, an initial, power-on test. This is followed by testing using the Arduino microcontroller to verify that the clock generator can be controlled via an Arduino sketch. To do this, it will be necessary to know the start-up frequency and I2C address of the Si570 being used. If the vendor of the Si570 being used did not specify its start-up frequency and I2C address, this can be determined from the part number on the center row of text on the device package. This information can be entered into the part number field on the Silicon Labs web page at http://www.silabs.com/products/clocksoscillators/Pages/Utilityintro.aspx. For example, entering the part number 570CAC000107 yields a CMOS Si750 with a start-up frequency of 10 Mhz and and I2C address of 55 (hex format). Once the start-up frequency and I2C address have been obtained, proceed with the test procedure.

1. Connect a frequency counter to the SMA connector so that the Si570's startup clock frequency can be measured.
2. Apply power to the Digital Board either via the USB-to-TTL serial adapter. The initial starting frequency of the Si570 should be displayed on the frequency counter. Be prepared to remove power immediately, if necessary. If no frequency is genrated, remove power and check the soldering of the Si570 to the Digital Board.
3. Once the power-on test is successful, proceed to testing with the Arduino microcontroller.

The Arduino sketch si570_testing, used to test the Si570, is listed in Appendix D. At the begining of the listing are severall `#define` statements. These are `SI570_I2C_ADDRESS`, `SI570_STARTUP_FREQUENCY`, and `SI570_NEW_FREQUENCY`. Depending upon the Si570 used in the Digital Board, some of these values may need to be changed. The value of `SI570_I2C_ADDRESS` needs to be edited to list the hexidecimal representation of the I2C address of the Si570 device being used, as mentioned previously. The value of `SI570_STARTUP_FREQUENCY` needs to be edited to list the startup frequency in Hertz of the Si570 device being used as well. The value of `SI570_NEW_FREQUENCY` needs to be edited to list a frequency no less than 10 MHz and no greater than 160 MHz. This frequency also needs to be stated in Hertz and needs to be in measurable with the frequency counter being used during the test. Once the necessary changes have been made to the Arduino sketch, this test can proceed.

1. Connect a frequency counter to the SMA connector so that the Si570's startup clock frequency can be measured.
2. If not done already, attach one end of the programmer cable to the USB-to-TTL serial adapter output and the other end to the five-pin connector on the Digital Board.
3. If not done already, attach one end of the USB cable to the USB port on the computer and the other end to the USB port on the USB-to-TTL serial adapter. This will allow the computer's operating system to enumerate the adapter and supply it with a device identifier.
4. With power applied to the Digital Board, the initial starting frequency of the Si570 should be displayed on the frequency counter.
5. Start the Arduino IDE and load or type in the sketch io_testing.ino. This sketch can be loaded from the Arduino IDE menu bar via Files->Open. A listing of the sketch is provided in Appendix D, if the sketch file is not available.
6. Ensure that the correct Arduino board type is selected. The Arduino Uno board type can be selected from the Arduino IDE menu bar via Tools->Board->Arduino Uno.
7. Ensure that the correct serial port is selected to talk to the USB-to-TTL serial adapter. The name of the serial port depends upon the operating system being used. To determine this, consult the manufacturer of the USB-to-TTL serial adapter and operating system guidelines. The serial ports available to the Arduino IDE can be selected from the menu bar via Tools->Serial Port
8. Select the 'Upload' button on the Arduino IDE button bar to compile and load the sketch into the Digital Board.
9. If the compile and load of the sketch is successful and there are no wiring error in the Digital Board, the LCD should should briefly display the message 'Si570 Test'. After five seconds, the content of the LCD should change to appear as shown in Illustration 18. The information displayed is hexidecimal representations of registers 7 through 12 in the Si570. If this does not occur, there may be wiring or soldering errors of the Si570 on the Digital Board.

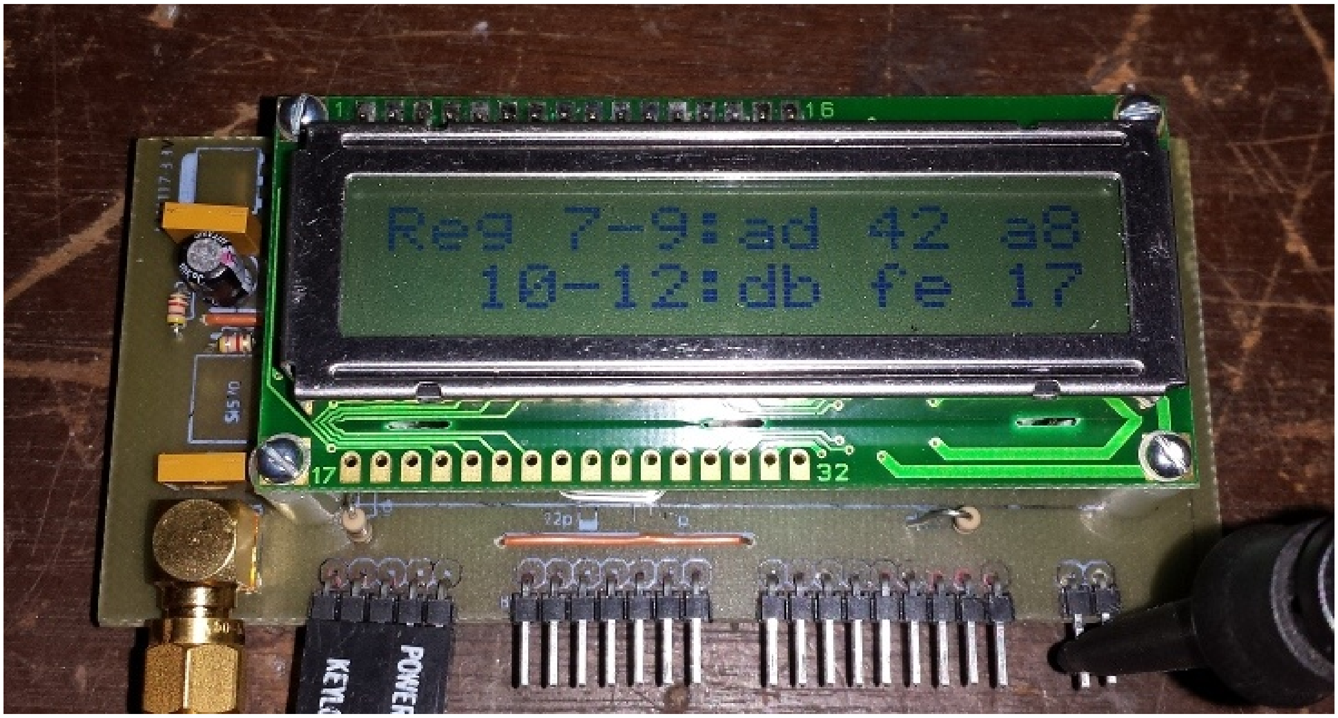

Illustration 18: Display of Si570 register values, as part of the Si570 testing sketch.

10. After fifteen seconds, the content of the LCD should change to appear as shown in Illustration 19. The information displayed are decimal representations of the startup values of the Si570's High Speed (HS) and N1 dividers, and the hexidecimal representation of the Si570's startup Reference Frequency (RFREQ) . This step should be successful if step 9. was successful.

Illustration 19: Display of HS, N1, and RFREQ, as part of the Si570 testing sketch.

11. After fifteen seconds, the content of the LCD should change to appear as shown in Illustration 20. The information displayed is the floating point representation of the Si570's startup Reference Frequency, in MHz, and the high-accuracy estimation of the Si570's crystal frequency, in Hertz. This step should be successful if step 10. was successful.

Illustration 20: Display of the Si570 Reference Frequency (in MHz) and the crystal frequency (in Hz).

12. At this point, the frequency displayed on the frequency counter should change to the value listed for `SI570_NEW_FREQUENCY`. Please note that the accuracy of the frequency displayed is dependent upon several factors, including the accuracy of the frequency counter used, the specified accuracy of the Si570, and the ambient temperature present during the test. This step should be successful if step 11. was successful.

This completes testing of the Si570 clock device and the assembly and test of the Minima Digital Board.

Minima Main Board Assembly and Testing

This section presents the instructions for the assembly and test of the Minima Main Board. Assembly and test of the Main Board board is more involved than the Digital Board, because it consists of audio and radio frequency analog circuitry rather than digital circuitry. To make the construction effort more approachable, it is broken down into several discrete sections that are tested separately. While most of the components are of the through-hole mount variety, there are a few SMD capacitors that will need to be soldered to the board.

It is assumed that the Minima Digital Board has already been assembled and successfully tested, as it will be used to assist in testing various sections of the Main Board. The programmibility of Arduion microcontroller on the Digital Board enables it to serve as a test signal generator. An Arduino sketch is provided to generate various audio and radio frequecy test signals, as well as signal measuring functions.

Minima Main Board Theory of Operation

The Main Board contains the analog audio and radio frequency circuits of the Minima transciever. A schematic of the Main Board is shown in Illustration 50 and Illustration 51 in Appendix A. Annotations have been added to the schematic to indicate the various functional elements used to implement the Minima transceiver. Similar annotations have also been made to the board layout shown in Illustration 21. These annotations will aid in the construction of the Main Board by making it easier to visualize and understand which components belong to specific functional elements and how they operate together. It is recommended to study the schematic while following this discussion.

The Main Board implements a super heterodyne transceiver with some special features to keep the design simple, yet provide good performance. The IF frequency is 20 MHz, which enables the use of commonly-available, fundamental mode crystals for the IF crystal filter and the BFO. The Si570 on the Digital Board, which provides the VFO function, generates a square wave, not a sine wave. This results in the generation of strong odd harmonics of the fundamental frequency that must be filtered to prevent out-of-band interference. The two low-pass filters are designed to attenuate harmonics, as well as image signals for either 0 to 15 MHz or 15 MHz to 30 MHz. A 20 MHz notch filter is also provided to attenuate errant signals at the IF frequency.

The Keep It Simple, Stupid (KISS) mixer, developed by Chris Trask N7ZWY, is used because it has several advantages over more traditional mixers. First, it doesn't require a post-mixer amplifer between it and the crystal filter. Second, its losses are low compared to passive mixers. Third, the JFETs are out of the signal path and merely short the transformer currents to ground as they are switched on. As a result, oscillator currents do not pass through the JFETs, so they do not mix with signal energy reflected back from the crystal filter.

The crystal filter consists of common, 20 MHz fundamental mode crystals and is designed with 50 ohm terminations. The filter was intentionally designed to have a wider passband in order to provide better quality audio sound and low pass-band ripple.

The bidirectional amplifier is borrowed from Hayward and Kopski's work on the BITX transceiver. It provides impedance matching to 50 ohms without the use of transformers. The transmit IF amplifier uses a standard feedback amplifier with higher current to handle the stronger signal coming from the modulator.

The BFO is designed to switch between two frequencies either slightly above or below 20 MHz. This is done to support both upper and lower sideband communictions. The diodes used in the BFO mixer are hand selected in order to avoid the need for carrier null adjustment circuitry.

The CW tone oscillator, mic preamp, RX audio preamp, and output audio amp are standard baseband audio designs. The RX audio preamp provides sufficient gain to the weak audio signal to drive the input of the output audio amp. The output audio amp is designed to drive headphones or an externally powered loudspeaker. Keeping most of the gain of the receiver chain in the baseband audio circuits makes construction non-critical.

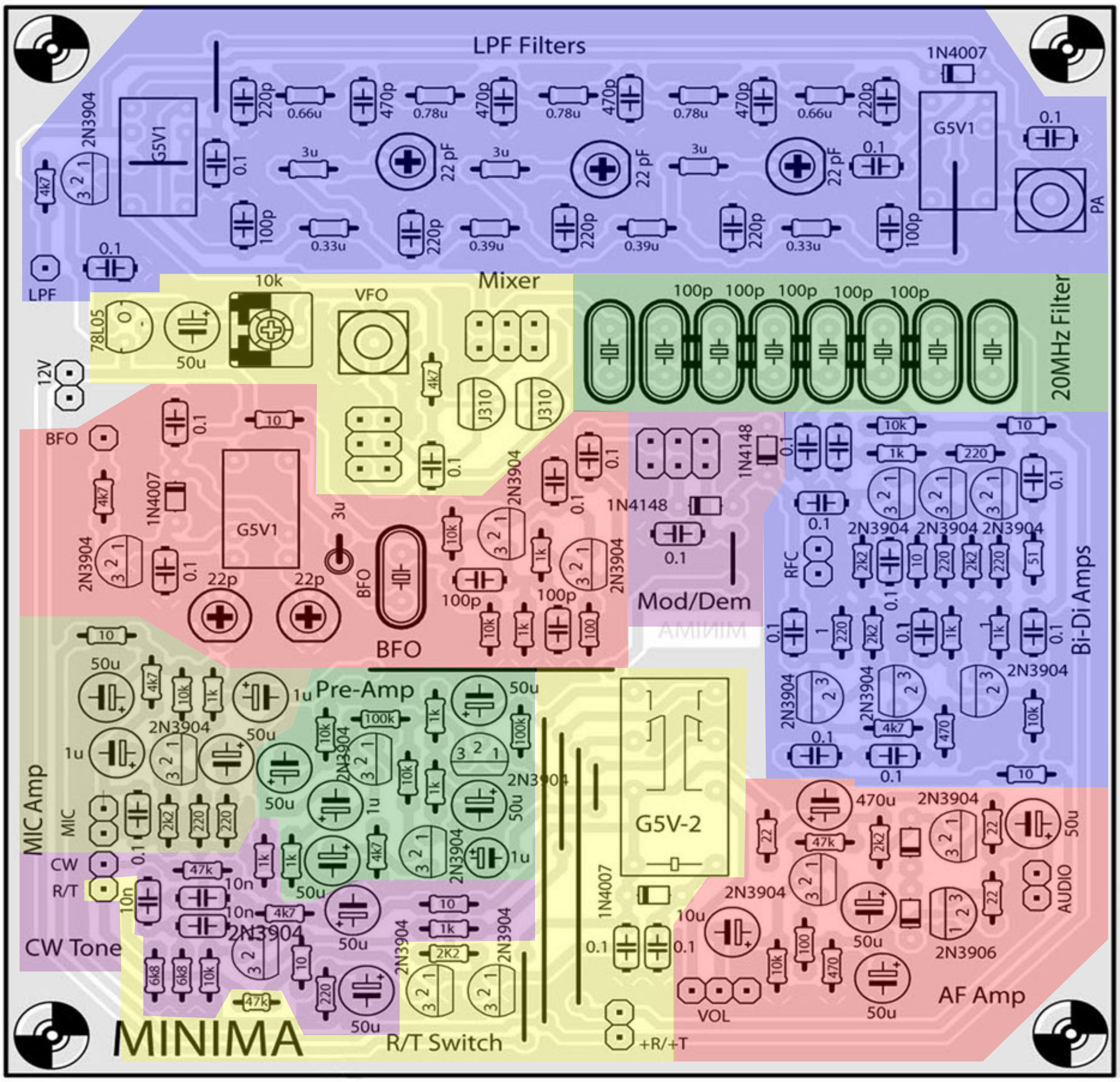

Illustration 21: Minima Main Board with annotation of functional elements.

Minima Main Board Parts List

The parts list for the Minima Main Board is presented in Appendix B. It is based on the components list generated from the Eagle CAD board projects. The component numbers listed will be used to reference the parts during construction. Note that headers JP1 through JP4 are actually mounting points for trifilar transformers used in the KISS and IF mixers and a radio frequency choke (RFC) used in the bidirectional amplifier.

Minima Main Board Assembly Instructions

Assembly of the Main Board will be performed in several sessions. In each session, a functional element of the Main Board will be assembled followed by its testing. To assist in assembly and testing, a portion of the Main Board schematic will be provided, along with the corresponding section of the Main Board layout and a parts list of those components required for that session. It is important to follow this assemble-then-test approach in order to avoid complications in getting the entire Minima transceiver to fuction properly.

Main Board Jumper Installation

The printed circuit board for the Main Board has circuit traces on only one side. In order to complete the circuit routing, it is necessary to add some wire jumpers. The placement of these wires is shown in Illustration 22. Wires of different colors were used to indicate the power buss and various signals.

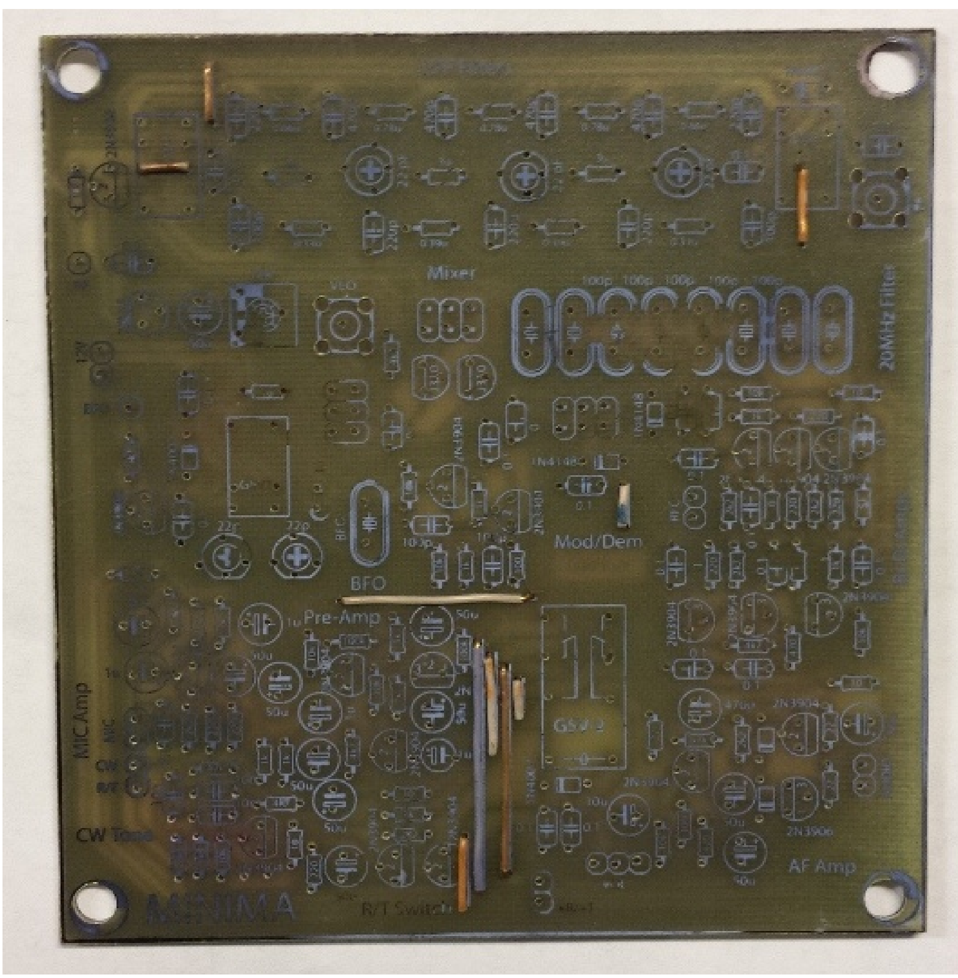

Illustration 22: Jumper placement for the Main Board.

R/T Switch Asembly

The R/T (receive/transmit) switch circuitry provides 12 Volt power to different functional elements of the Minima transceiver depending upon whether the mode of operation is reception or transmission. Proper operation of the transceiver depends upon some functional elements being unpowered while others are powered. One example of this is the bidirectional amplifier. The amplifier branch for outbound signals (i.e. transmission) must be unpowered while the amplifier branch for inboud signals (i.e. reception) is powered.

The schematic for the R/T switch circuitry is shown in Illustration 23. During receiption, the R/T input, coming from the Digital Board, is nominally 5 Volts. This will cause transistor Q28 to turn on. This pulls the base of transistor Q29 to close to ground potential, causing Q29 to turn off. This causes the solenoid on relay K2 to de-energize, switching the relay to provide the 12 Volt supply (via JP12) to those circuit functional elements requiring power during reception. Functional elements not requiring power diring reception are turned off. Conversely, when the R/T input is brought to ground, Q28 is turned off, allowing current to flow into the base of Q29, causing it to turn on. This, in turn, causes K2 to energize and switch power to transmitting circuitry, while receiving circuity is turned off.

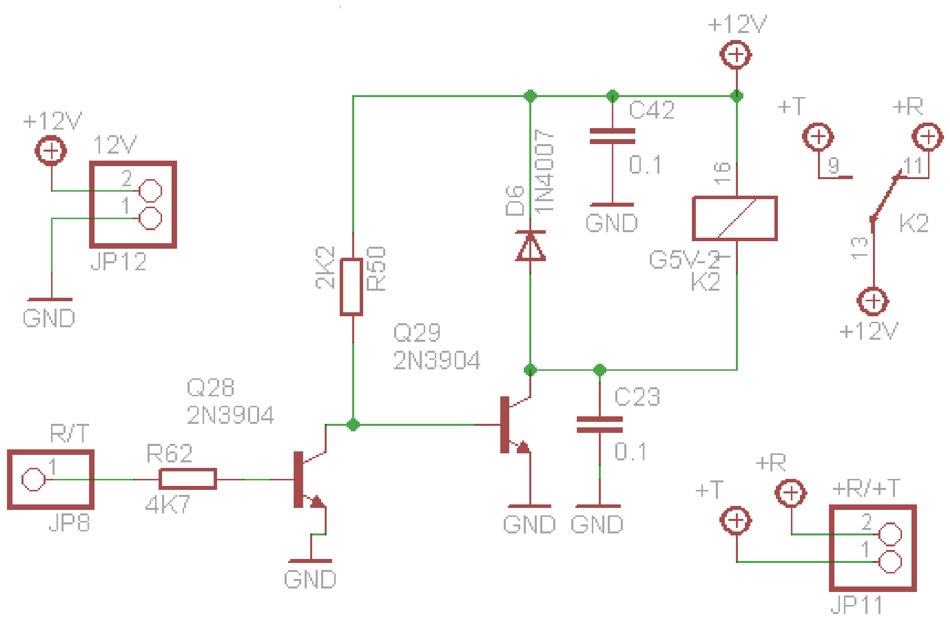

Illustration 23: Schematic of the Main Board R/T switch circuitry.

The location of the R/T switch circuity on the layout of the Main Board is shown in Illustration 24 highligheted in yellow. Note that on the layout, R62 is labled as a 47 Kohm, while the schematic in Illustration 23 indicates a value of 4.7 Kohm. There has been an issue as to the correct value of resistor to use. For this construction guide, a value of 47 Kohm has found to be satisfactory.

While the 12 Volt power header is not specifically a part of the R/T switch circuitry, it is included in this assembly session for the simple reason providing a convenient means of applying power to the Main Board. The 12 Volt power header is circled in red in the upper left hand corner of Illustration 24.

During the assembly of the R/T switch circuitry, the header JP8 (R/T) and the header JP10 (CW) will be soldered to the Main Board at the same time. While JP10 is technically a distinct header from JP8, its physical placement next to JP8 warrants treating them together as a single 1x2 physical header.

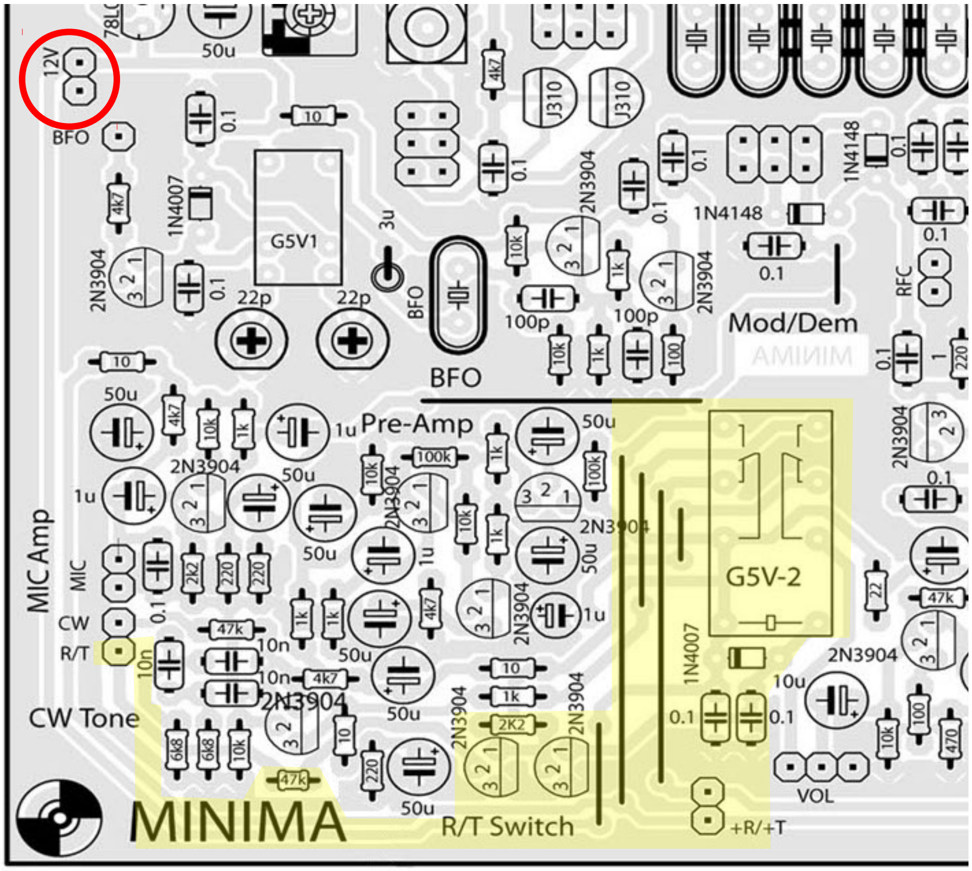

Illustration 24: Layout of the Main Board R/T Switch circuitry and 12 Volt supply header.

The parts list for the R/T Switch circuitry is presented in Table 4. As mentioned previously, the R/T header (JP8) and the CW header (JP10) are treated as one, single 1x2 pin header. In addition, R62 will be a 47 Kohm resistor.

Part	Value	Device
C23	0.1u	C-EU050-024X044
C42	0.1u	C-EU050-024X044
D6	1N4007	DIODE-D-5
JP8, JP10	R/T, CW	PINHD-1X2
JP11	+R/+T	PINHD-1X2
JP12	12V	PINHD-1X2
K2	G5V-2	G5V-2
Q28	2N3904	2N3904
Q29	2N3904	2N3904
R50	2K2	R-EU_0204/5
R62	47K	R-EU_0204/5

Table 4: Parts list for the R/T Switch circuitry of the Minima Main Board.

The following steps describe the installation of the components for the R/T Switch section of the Main Board.

1. Solder a 1N4007 diode D6 next to the G5V-2 (DPDT) relay. The diode needs to be inserted "hairpin" style because the spacing between the component pads is too close together to allow the diode to lay flat.
2. Solder two 0.1 uF capacitors C23 and C42 next to diode D6.
3. Solder a 1x2 pin header to the jumper location labeled "+R/-T" (JP11).
4. Solder a 1x2 pin header to the jumper location labeled "R/T" (JP8) and "CW" (JP10) on the lower left hand side of the Main Board.
5. Solder a 1x2 pin header to the jumper location labeled "12V" (JP12) on the upper left hand side of the Main Board.
6. Solder a 2.2 Kohm resistor R50 to the left of the two capacitors soldered in the previous step.
7. Solder the 47 Kohm resistor R62 just above the "MA" in the "MINIMA" text. The resistor needs to be inserted "hairpin" style because the spacing between the component pads is too close together to allow the resistor to lay flat.
8. Solder two 2N3904 transistors Q28 and Q29 immediately below the 2.2 Kohm resistor soldered in the previous step. Ensure that the orientation of the transistors match that of the symbol on the silkscreen on the Main Board. The flat side of each transistor should match the flat side of the flat side of the symbols.
9. Solder the G5V-2 relay K2 into the Main Board. This completes the assembly phase of the R/T Switch function of the Main Board.

The completed assembly of the R/T Switch circuitry is show in Illustration 25.

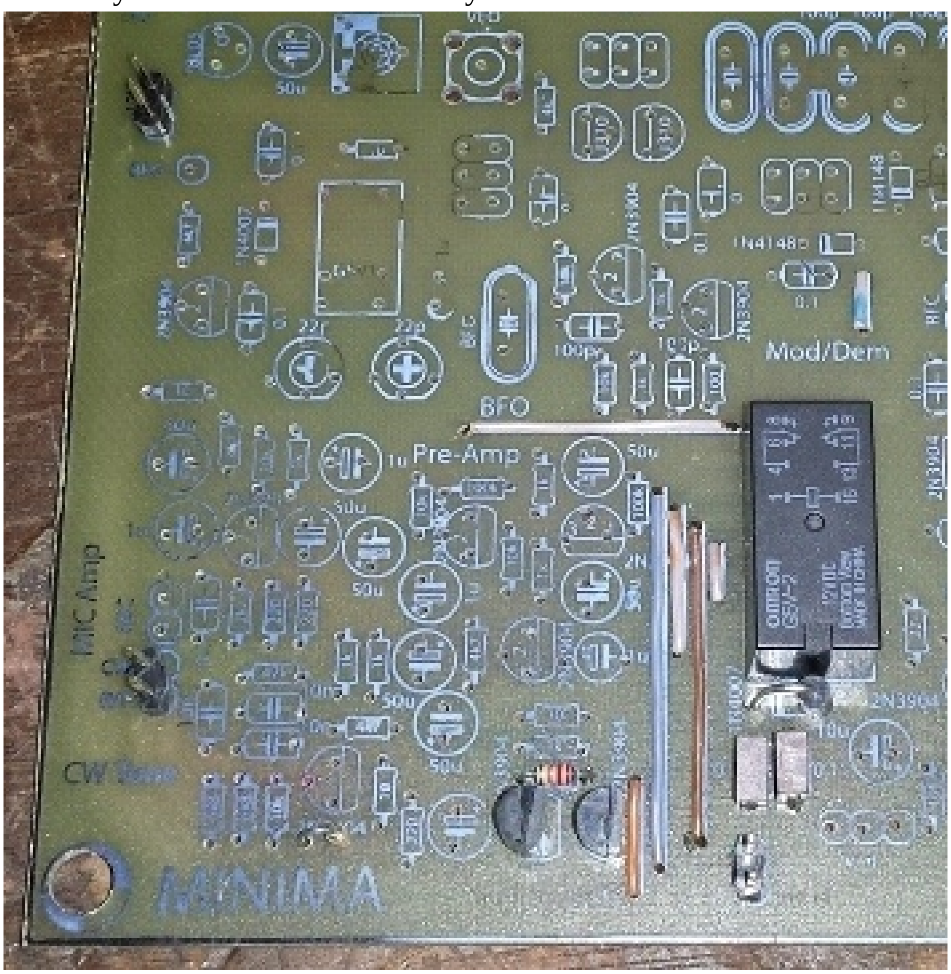

Illustration 25: Component placement for the Main Board R/T Switch circuitry.

R/T Switch Testing

The Digital Board is used to assist in the testing of the R/T Switch. The PTT (T/R) key on the Digital Board is used to control the state of the R/T switch. While there is no specific Arduino sketch required to perform this test, the sketch that is loaded must set up Arduino pin PD3 as an input with the internal pull-up enabled. The sketch listed in Appendix C is suitable for this task.

The following steps describe the testing of the R/T Switch section of the Main Board.

1. Download an appropriate Arduino sketch into the Digital Board.
2. Assemble a cable that connects the Digital Board to the Main Board through the "CW" and "R/T" lines. An example cable, constructed from a discarded PC enclosure wiring harness,is shown in Illustration 26. Also, attach a momentary pushbutton switch between the PTT pin and the ground pin adjacent to it. Finally, connect a switched, 12 volt power supply to the Digital Board and the Main Board. The completed interconnection of the Digital Board and the Main Board is shown in Illustration 27.

Illustration 26: Cable used for Main Board R/T Switch testing.

3. Apply power to Digital Board and Main Board. If the LCD on the Digital Board does not initialize and display the test messages, turn off power immediately and determine the cause.
4. Measure the voltage on the 2x1 pin header labeled "+R/-T". The pin closest to the edge of the board should measure close to zero volts. The pin furthest from the edge of the board should measure about 12 volts. If this is not the case, turn off power immediately and determine the cause.
5. Press the PTT switch connected to the Digital Board. When the switch is depressed, you should hear the G5V-2 relay switch. While keeping the switch depressed, measure the voltage on the 2x1 pin header labeled "+R/-T". The pin closest to the edge of the board should measure about 12 volts. The pin furthest from the edge of the board should measure close to zero volts. If this is not the case, turn off power immediately and determine the cause.
6. When the PTT button is released, the relay should de-energize and the voltages on the "+R/-T" header should return to their previous values. If this is not the case, turn off power immediately and determine the cause.

This completes the testing of the R/T Switch circuitry of the Main Board.

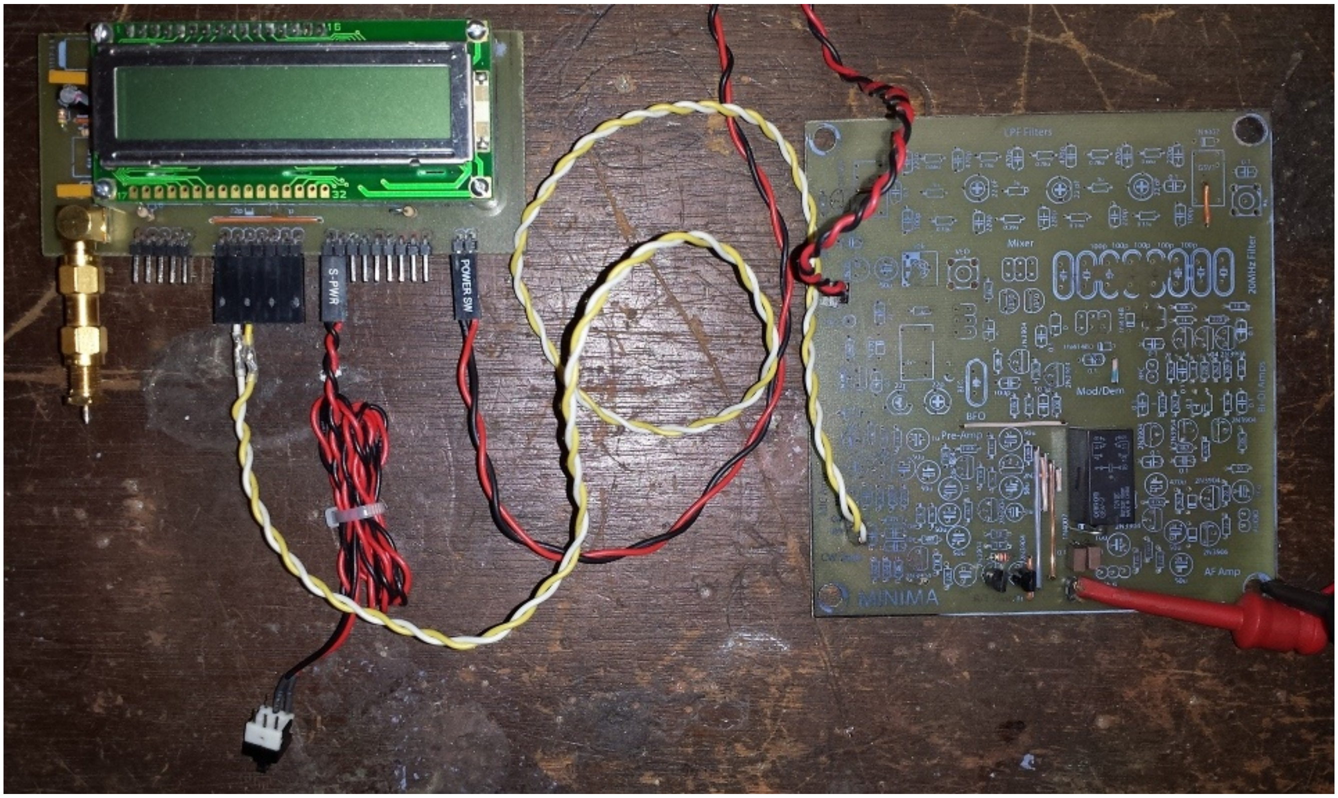

Illustration 27: Test setup for the Main Board R/T Switch assembly.

Output Audio Amp Assembly

The Output Audio Amp amplifies the output of the RX Audio Preamp to a level sufficient to drive a headphone or an externally-powered, amplified speaker. The schematic for the Output Audio Amp circuitry is shown in Illustration 28. The input level of the audio signal coming from the RX Audio Preamp is controlled by a 10 Kohm, audio taper potentiometer connected to header JP7. The input transistor Q22 provides Class A amplification, with a voltage gain of 22, based upon the load resistor R29 and the negative feedback resistor R30. The output of Q22 drives the Class AB push-pull amplifier formed by transistors Q20 and Q21. Note that diodes D3 and D4 are 1N4148s and are included in the circuit to minimize crossover distortion. The "feedback" resistors R27 and R28 are provided to limit the current flowing through R27 and R28 due to being biased to constant conduction. The current gain provide is sufficient to drive the relatively low impedance of headphone speakers. The output audio connector on the front panel of the Minima transceiver is connected to the Output Audio Amp via header JP5. **Note that the polarity of capacitor C29 is reversed from what it should be.**

The location of the Output Audio Amp circuity on the layout of the Main Board is shown in Illustration 29 highlighted in red. Note that the physical spacing between the components is fairly tight, especially any components adjacent to the 470 uF electrolytic capacitor. Care must be taken to follow the specified assembly sequence in order to simplify the insertion and soldering of components. As mentioned before, headers JP7 and JP5 are used to connect a potentiometer and a headphone connector, respectively. **Note that the polarity of the 50 uF capacitor above the AUDIO header JP5 is reversed from what it should be.**

The parts list for the Output Audio Amp circuitry is presented in Table 5. While it is not called out specifically, a 10 Kohm, audio taper potentiometer will be connected to the circuit via JP7. Likewise, a suitable panel connector for headphones or power speakers will be attached to the circuit via JP5.

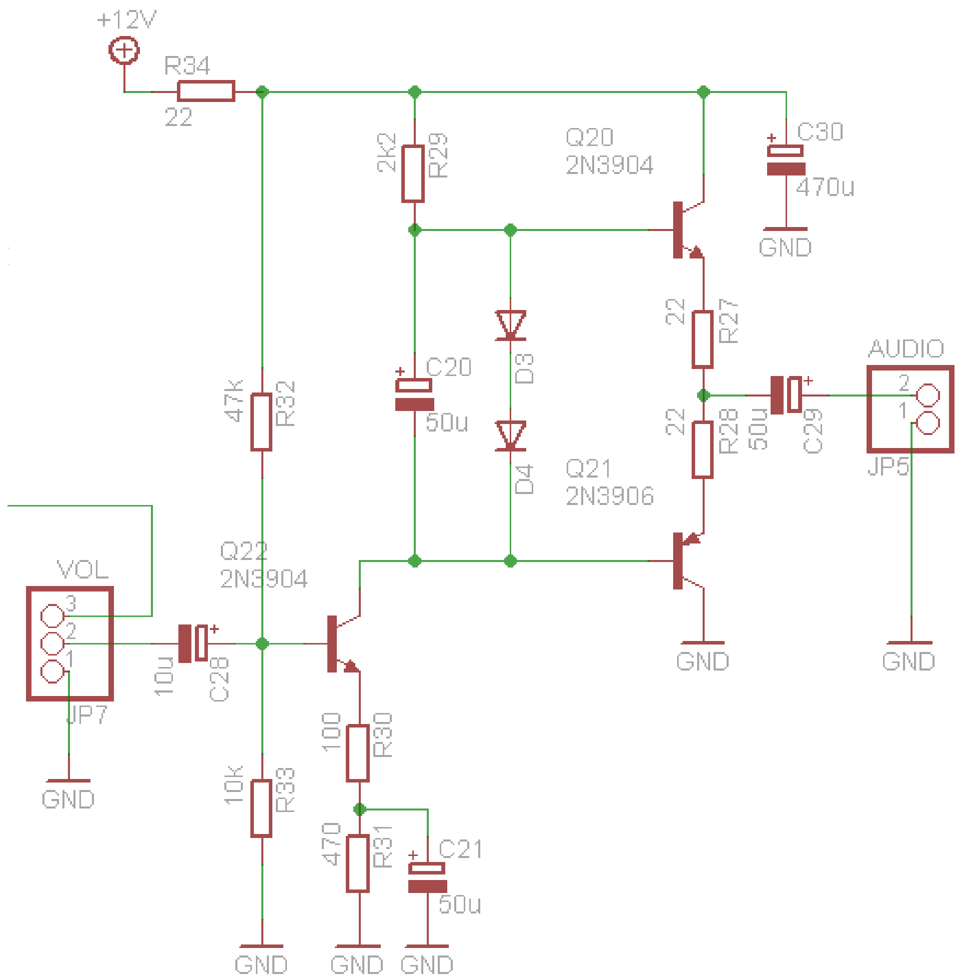

Illustration 28: Schematic of the Main Board Output Audio Amp circuitry.

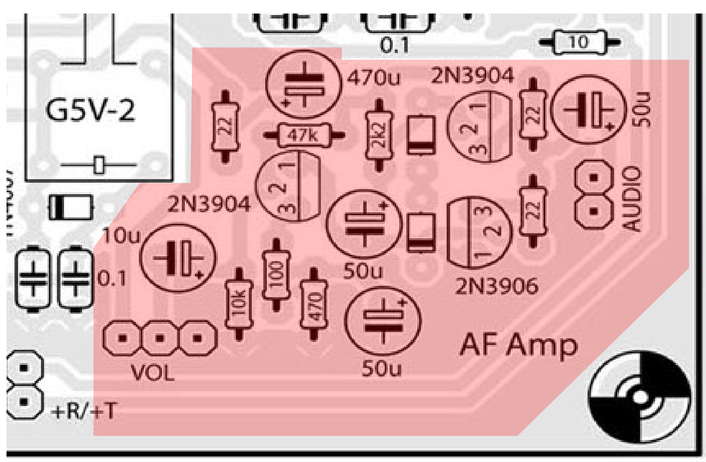

Illustration 29: Layout of the Main Board Output Audio Amp circuitry.

Part	Value	Device
C20	50uF	CPOL-EUE2.5-5
C21	50uF	CPOL-EUE2.5-5
C28	10uF	CPOL-EUE2.5-5
C29	50uF	CPOL-EUE2.5-5
C30	470uF	CPOL-EUE2.5-5
D3	1N4148	DIODE-D-5
D4	1N4148	DIODE-D-5
JP5	AUDIO	PINHD-1X2
JP7	VOL	PINHD-1X3
Q20	2N3904	2N3904

Part	Value	Device
Q21	2N3906	2N3906
Q22	2N3904	2N3904
R27	22R	R-EU_0204/5
R28	22R	R-EU_0204/5
R29	2k2	R-EU_0204/5
R30	100R	R-EU_0204/5
R31	470R	R-EU_0204/5
R32	47k	R-EU_0204/5
R33	10k	R-EU_0204/5
R34	22R	R-EU_0204/5

Table 5: Parts list for the Main Board Output Audio Amp.

The following steps describe the installation of the components for the Output Audio Amp section of the Main Board.

1. Solder R27, a 22 ohm resistor, next to Q20, a 2N3904 NPN transistor.
2. Solder R28, a 22 ohm resistor, next to Q21, a 2N3906 PNP transistor.
3. Solder R29, a 2.2 Kohm resistor, between to Q20 and Q22, 2N3904 NPN transistors.
4. Solder R30, a 100 ohm resistor, below to Q22, a 2N3904 NPN transistor.
5. Solder R31, a 470 ohm resistor, below to Q22, a 2N3904 NPN transistor.
6. Solder R32, a 47 Kohm resistor, above to Q22, a 2N3904 NPN transistor.
7. Solder R33, a 10 Kohm resistor, below to Q22, a 2N3904 NPN transistor.
8. Solder R34, a 22 ohm resistor, next to Q22, a 2N3904 NPN transistor.
9. Solder D3, a 1N4148 diode, next to Q20, a 2N3904 NPN transistor. Ensure that the cathode (black band) of the diode is oriented correctly prior to soldering.
10. Solder D4, a 1N4148 diode, next to Q21, a 2N3906 PNP transistor. Ensure that the cathode (black band) of the diode is oriented correctly prior to soldering.
11. Solder Q20, a 2N3904 NPN transistor, next to R27, a 22 ohm resistor. Ensure that the flat side of the transistor is oriented correctly prior to soldering.
12. Solder Q21, a 2N3906 PNP transistor, next to R28, a 22 ohm resistor. Ensure that the flat side of the transistor is oriented correctly prior to soldering.
13. Solder Q22, a 2N3904 NPN transistor, below to R32, a 47 Kohm resistor. Ensure that the flat side of the transistor is oriented correctly prior to soldering.
14. Solder JP7, a 1x3 pin header, to the left of R33, a 10 Kohm resistor.
15. Solder JP5, a 1x2 pin header, to the right of R28, a 22 ohm resistor.
16. Solder C20, a 50 uF electrolytic capacitor, to the right of Q22, a 2N3904 NPN transistor. Ensure that the polarity of the capacitor is oriented correctly prior to soldering.
17. Solder C21, a 50 uF electrolytic capacitor, to the right of R31, a 470 ohm resistor. Ensure that the polarity of the capacitor is oriented correctly prior to soldering.
18. Solder C28, a 10 uF electrolytic capacitor, above header JP7, the VOL connector. Ensure that

the polarity of the capacitor is oriented correctly prior to soldering.

19. Solder C29, a 50 uF electrolytic capacitor, above header JP5, the AUDIO connector. Note that the polarity of the capacitor is labeled incorrectly in the schematic in Illustration 28 and the component artwork in Illustration 29. **Solder the capacitor with the polarity reversed to what is shown on the component artwork.**
20. Solder C30, a 470 uF electrolytic capacitor, above header JP7, the VOL connector. Ensure that the polarity of the capacitor is oriented correctly prior to soldering. Note that due to the lack of clearance between C39 and R32, a 47 Kohm resistor, C30 will have to be elevated above R32.

The completed assembly of the Output Audio Amp circuitry is show in Illustration 30.

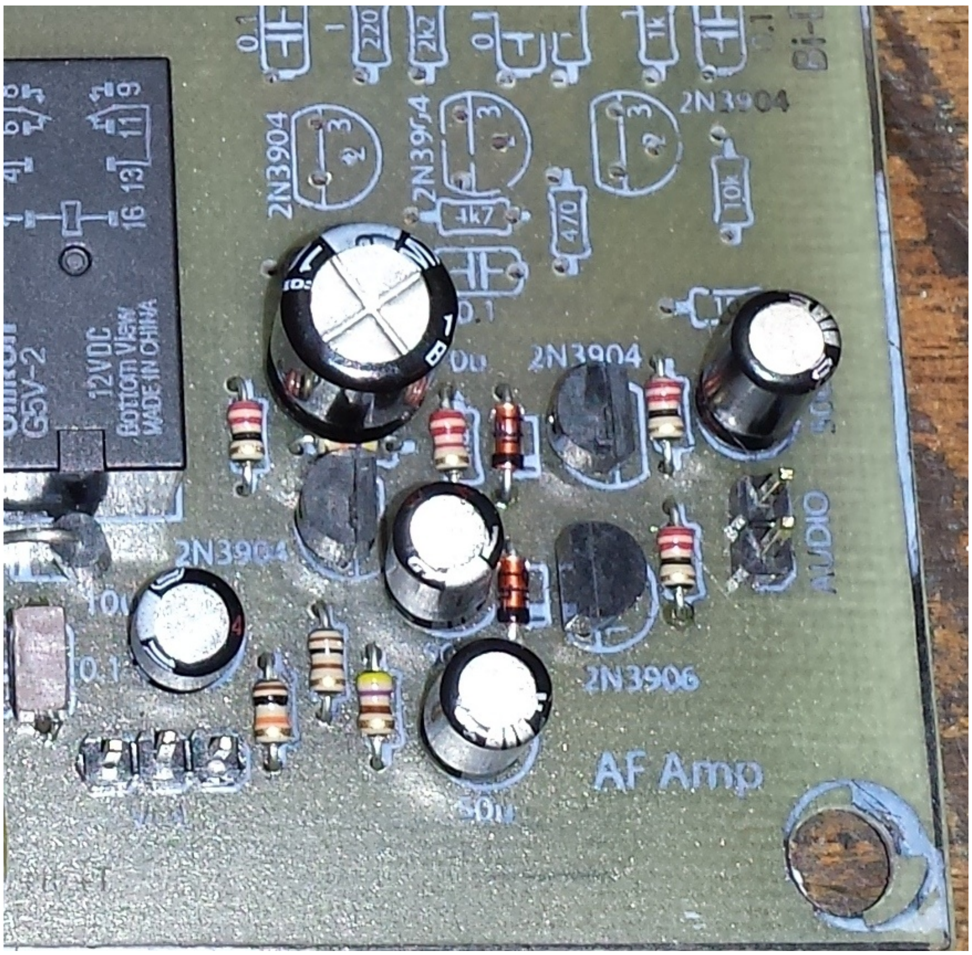

Illustration 30: Component placement for the Main Board Output Audio Amp circuitry.

Output Audio Amp Testing

The simplest means to test the Output Audio Amp is to inject an audio frequency tone into the input and listening to the amplified signal on the output. This can be done, if desired, with an audio signal generator or a tone injector typically used for audio circuit troubleshooting. The programmability of the Minima Digital Board make it possible to create an audio tone generator via an Arduino sketch. The sketch listed in Appendix E uses the `tone()` function to generate a 1 KHz squarewave signal that is output on Arduino pin PD7, which is assigned in the Radiono sketch as Band Select 0. The squarewave tone has an amplitude of about 5 volts, so it is recommended to use it cautiously when injecting it into the input of an amplifier. Overdriving the an amplifier may damage it.

The following steps describe the testing of the Output Audio Amp circuit.

1. Using the Arduino IDE, compile and download the 1 KHz Audio Tone Generator sketch into the Digital Board. The technique to do this has been describe previously and will not be repeated here.
2. Testing the Output Audio Amp will require a set of cables to be created. The cable used in the R/T Switch test will also be used. First, add a wire with a mini hook to that connector shell at the position of the output Band Select 0. This cable is shown in the bottom of Illustration 31. Next, create a cable that will deliver the amplifier output (AUDIO) of JP5 to an audio headphone receptacle. The signal pin of the receptacle will be wired to the connector shell such that it plugs into pin 2 of JP5. The ground pin of the receptacle will be wired to the connector shell such that it plugs into pin 1 of JP5. An example, using a 3.5 mm stereo jack, is shown on the left side of Illustration 31. Finally, create a cable that connects a 10 Kohm, audio taper potentiometer to the amplifier input (VOL) of JP7. When looking into the shaft of the potentiometer, the lefthand connection is wired to the connector shell so that it plugs into pin 3 of JP7. The wiper of the potentiometer is wired to the central pin of the connector shell, which will plug into pin 2 of JP7. The righthand connection of the potentiometer is wired to the connector shell so that it plugs into pin 1 of JP7. An example configuration is shown in the top portion of Illustration 31.

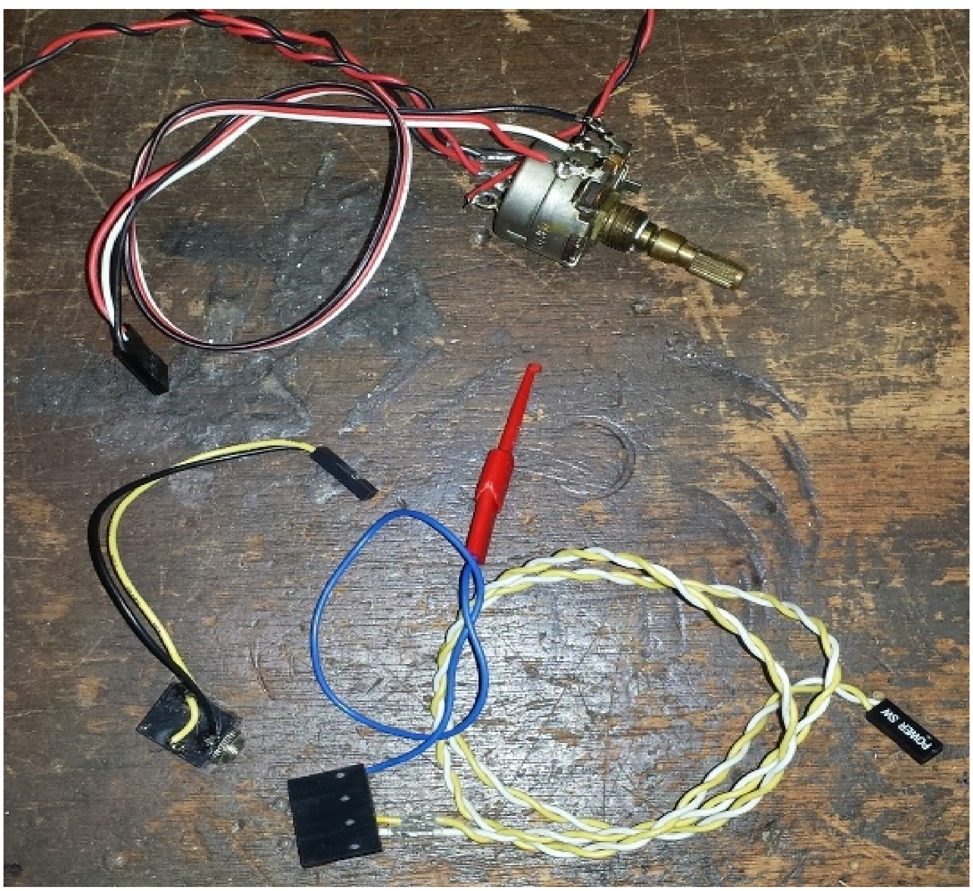

Illustration 31: Cable set used for testing the Main Board Output Audio Amp circuit.

3. Connect the Digital Board and Main Board using the cables as shown in Illustration 32. The mini hook will be attached to the lefthand terminal of the potentiometer to provide the audio tone to the amplifier circuit. **It is preferable that this be done after the potentiometer volume in set to minimum and power is applied to the test setup. It is critical that the potetiometer be set so that no signal from the tone generator is injected into the amplifier when it is turned on.** In the example setup presented here, the potentiometer shaft is rotated fully counter clockwise. Plug a headphone into the audio receptacle. **It is recommended to NOT wear the headphones during this test in order to prevent damage to hearing.**

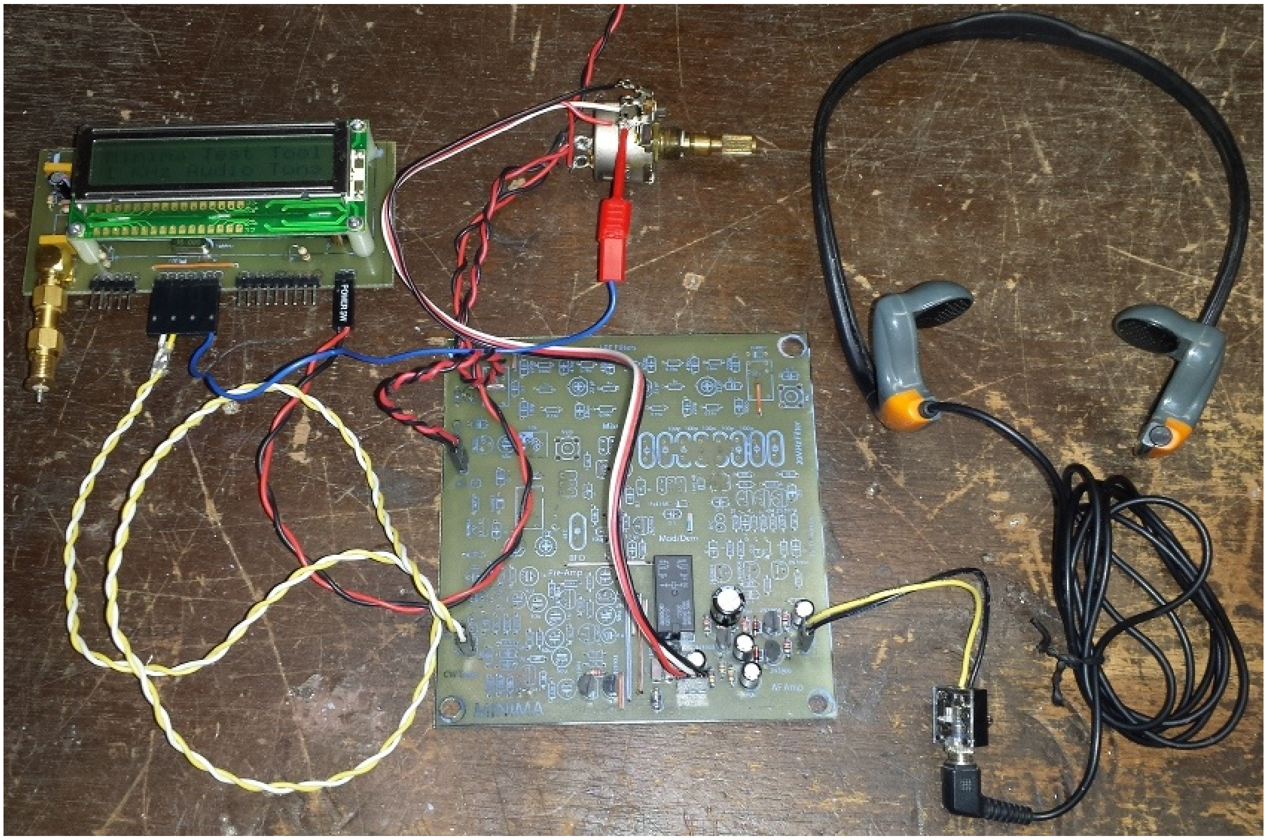

Illustration 32: Test setup for the Main Board Output Audio Amp circuit test.

4. Turn on the switch providing power to the Digital Board and Main Board. If, as shown in Illustration 32, the power switch is integral to the volume potentiometer, ensure that the volume is not increased significantly and is set to minimum. If the messages "Minima Test Tool" and "1 KHz Audio Tone" do not appear on the Digital Board's LCD, remove power from the test setup and investigate the cause.
5. Attach the mini hook to the input terminal of the potentiometer designated as signal input.
6. Gradually increase the volume of the potentiometer. A 1 KHz tone should be heard coming from the headphones. It should not be necessary to increase the potentiometer setting by very much to obtain an audiable tone.

This completes the testing of the Output Audio Amp circuitry on the Main Board.

RX Audio Preamp Assembly

The RX Audio Preamp provides amplification to the baseband audio signal received from the IF Mixer. Within the Minima, the only amplification previously given to a received signal is through the receiver branch of the Bidirectional Amp. This amplification compensates for losses in the Low Pass Filter, KISS Mixer, and IF Crystal Filter. The primary amplification of the received signal is performed at baseband audio frequencies by the RX Audio Preamp and the Output Audio Amp.

The schematic for the RX Audio Preamp circuitry is shown in Illustration 33. The signal enters the circuit via C35, a 1 uF electrolytic capacitor. The first stage of the preamp (Q25) is a voltage-biased, common-base amplifier. It presents low input impedance, to match the low impedance of the IF Mixer, moderate gain, and high output impedance to match the input impedance of the second stage of the preamp. The second stage of the preamp (Q24) is a common-emmiter amplifier, providing high gain that is dependent upon the β of the particular 2N3904 used. The combination of transistior Q23, resistor R35, and capacitor C31 form a power-up delaycircuit for the preamp. The time constant formed by R35 and C31 result in a delay of several seconds before sufficient bias is applied to Q23 to enable sufficient conduction to provide current to transistors Q24 and Q25. The output of the amplifier passes through C50, a 1 uF electrolytic capacitor, to pin 6 of the SPDT relay K2.

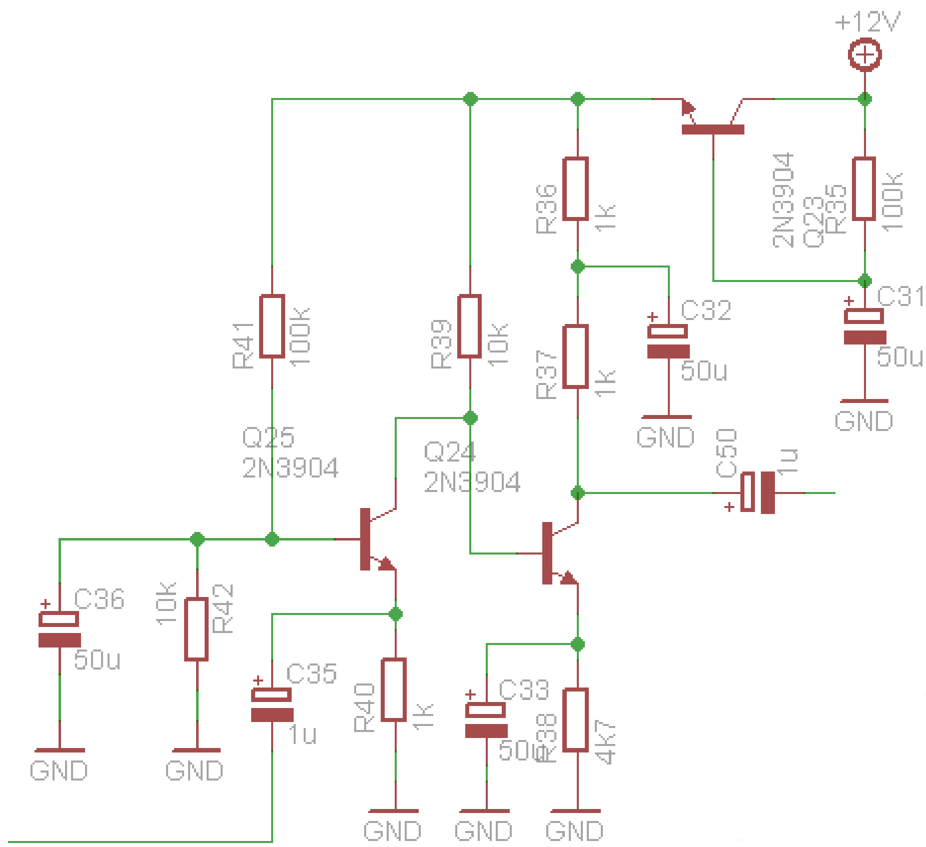

Illustration 33: Schematic of the Main Board RX Audio Preamp circuitry.

The location of the RX Audio Preamp circuitry on the layout of the Main Board is shown in Illustration 34 highlighted in green. Note that the physical spacing betweem the components is fairly tight. Care must be taken to follow the specified assembly sequence in order to simplify the insertion and soldering of components.

The parts list for the RX Audio Preamp is presented in Table 6.

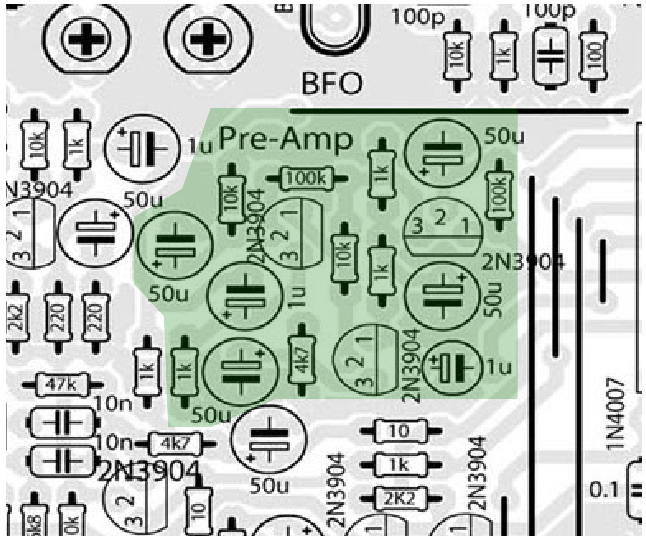

Illustration 34: Layout of the Main Board RX Audio Preamp circuitry.

Part	Value	Device
C31	50uF	CPOL-EUE2.5-5
C32	50uF	CPOL-EUE2.5-5
C33	50uF	CPOL-EUE2.5-5
C35	1uF	CPOL-EUE2.5-5
C36	50uF	CPOL-EUE2.5-5
C50	1uF	CPOL-EUE5-4
Q23	2N3904	2N3904
Q24	2N3904	2N3904
Q25	2N3904	2N3904

Part	Value	Device
R35	100k	R-EU_0204/5
R36	1k	R-EU_0204/5
R37	1k	R-EU_0204/5
R38	4k7	R-EU_0204/5
R39	10k	R-EU_0204/5
R40	1k	R-EU_0204/5
R41	100k	R-EU_0204/5
R42	10k	R-EU_0204/5

Table 6: Parts list for the Main Board RX Audio Preamp.

The following steps describe the installation of the components for the RX Audio Preamp section of the Main Board.

1. Solder R35, a 100 Kohm resistor, to the upper right of Q23, a 2N3904 NPN transistor.
2. Solder R36, a 1 Kohm resistor, to the upper left of Q23, a 2N3904 NPN transistor.
3. Solder R37, a 1 Kohm resistor, to the lower left of Q23, a 2N3904 NPN transistor.
4. Solder R38, a 4.7 Kohm resistor, to the left of Q24, a 2N3904 NPN transistor.
5. Solder R39, a 10 Kohm resistor, to the lower right of Q25, a 2N3904 NPN transistor.
6. Solder R40, a 1 Kohm resistor, to the left of C33, a 50 uF electrolytic capacitor.
7. Solder R41, a 100 Kohm resistor, above Q25, a 2N3904 NPN transistor.
8. Solder R42, a 10 Kohm resistor, to the upper left of Q25, a 2N3904 NPN transistor.
9. Solder Q23, a 2N3904 NPN transistor, to the lower left of R35, a 100 Kohm resistor. Ensure that the flat side of the transistor is oriented correctly prior to soldering.

10. Solder Q24, a 2N3904 NPN transistor, to the right of R38, a 4.7 Kohm resistor. Ensure that the flat side of the transistor is oriented correctly prior to soldering.
11. Solder Q25, a 2N3904 NPN transistor, below R41, a 100 Kohm resistor. Ensure that the flat side of the transistor is oriented correctly prior to soldering.
12. Solder C31, a 50 uF electrolytic capacitor, above Q23, a 2N3904 NPN transistor. Ensure that the polarity of the capacitor is oriented correctly prior to soldering.
13. Solder C32, a 50 uF electrolytic capacitor, below Q23, a 2N3904 NPN transistor. Ensure that the polarity of the capacitor is oriented correctly prior to soldering.
14. Solder C33, a 50 uF electrolytic capacitor, to the left of R38, 4.7 Kohm resistor. Ensure that the polarity of the capacitor is oriented correctly prior to soldering.
15. Solder C35, a 1 uF electrolytic capacitor, to the lower left of Q25, a 2N3904 NPN transistor. Ensure that the polarity of the capacitor is oriented correctly prior to soldering.
16. Solder C36, a 50 uF electrolytic capacitor, to the lower left of R42, a 10 Kohm resistor. Ensure that the polarity of the capacitor is oriented correctly prior to soldering.
17. Solder C50, a 1 uF electrolytic capacitor, to the right of Q24, a 2N3904 NPN transistor. Ensure that the polarity of the capacitor is oriented correctly prior to soldering.

The completed assembly of the RX Audio Amp circuit is shown in Illustration 35.

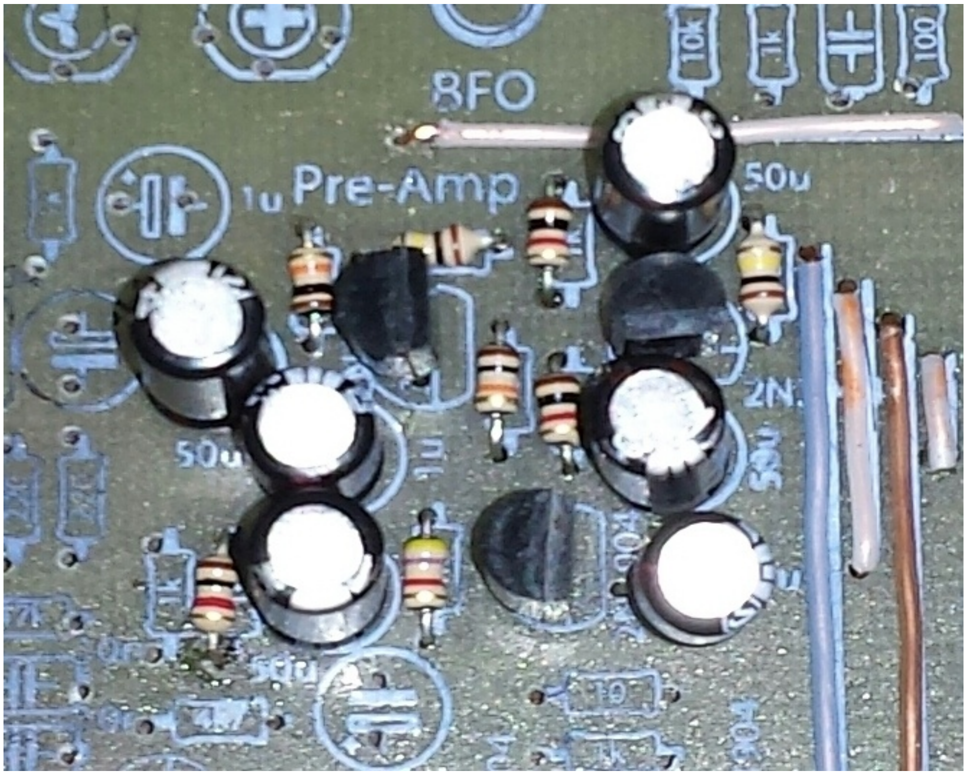

Illustration 35: Component placement for the Main Board RX Audio Preamp circuitry.

RX Audio Preamp Testing

The method for testing the RX Audio Preamp is similar to that performed for the testing of the Output Audio Amp. In this case, the 1 KHz squarewave audio tone produced by the Digital Board must be properly attenuated so it does not to overdrive the preamp. Likewise, it is undesireable for the amplified signal output by the preamp to overdrive the input of the Output Audio Amp circuit.

The following steps describe the testing of the RX Audio Preamp circuit.

1. The Arduino sketch for the 1 KHz audio test tone generator should already be downloaded into the Digital Board and will be used for testing the preamp.
2. Several cables are needed to test the preamp. These are reused from previous tests and modified where necessary. A connector wired to a momentary switch will be attached to the PTT input on the Digital Board. An example is shown on the right hand side of Illustration 36. The cable that connects a 10 Kohm, audio taper potentiometer to the amplifier input (VOL) of JP7 will be used again for this test. This cable is shown at the top of Illustration 36. The cable that delivers the amplifier output (AUDIO) of JP5 to an audio headphone receptacle will also be used again. It is shown on the left hand side of Illustration 36. Finally, the connector shell that attaches to the T/R and CW pins needs to be modified to support attenuation of the squarewave audio tone. Insert a 10 Kohm trimmer pot between the Band Select 0 output and the mini hook. The mini hook is attached to the wiper contact of the trimmer pot. The Band Select 0 output is connected to one of the fixed pins on the trimmer pot. A second mini hook is connected to the other fixed pin on the trimmer pot. This cable is shown in the bottom of Illustration 36.

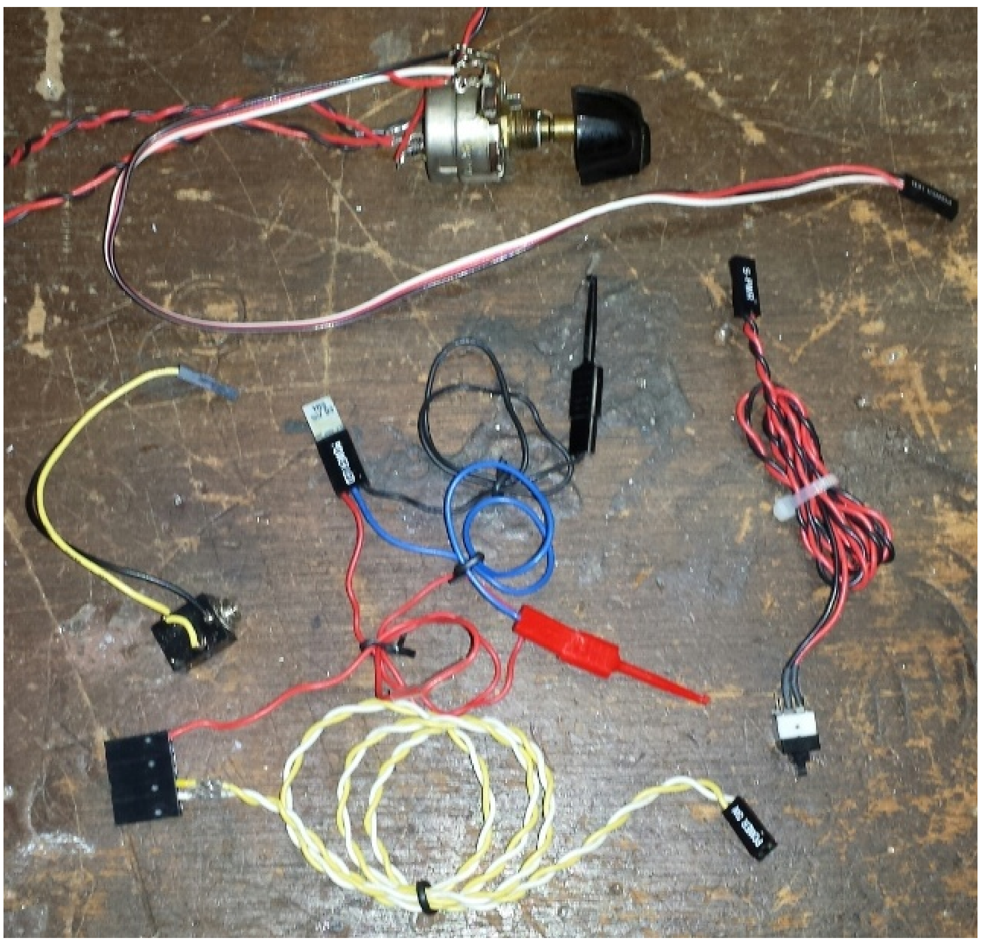

Illustration 36: Cable set used for testing the Main Board RX Audio Preamp circuit.

3. Connect the Digital Board and Main Board using the cables as shown in Illustration 37. Connect the T/R and CW cable as performed in previous tests. Connect the signal-bearing mini

hook (red in the illustration) to the wire jumper just above the RX Audio Preamp circuit to inject the audio tone into the preamp input. Connect the other mini hook (black in the illustration) to the grounded side of R40, a 1 Kohm resistor. The trimmer pot should be adjusted to inject a greatly attenuated squarewave audio tone into the input of the preamp. The wiper should be positioned such that it is closest to the trimmer lead connected to the grounded mini hook. Plug a headphone into the audio receptacle. **It is recommended to NOT wear the headphones during this test in order to prevent damage to hearing.**

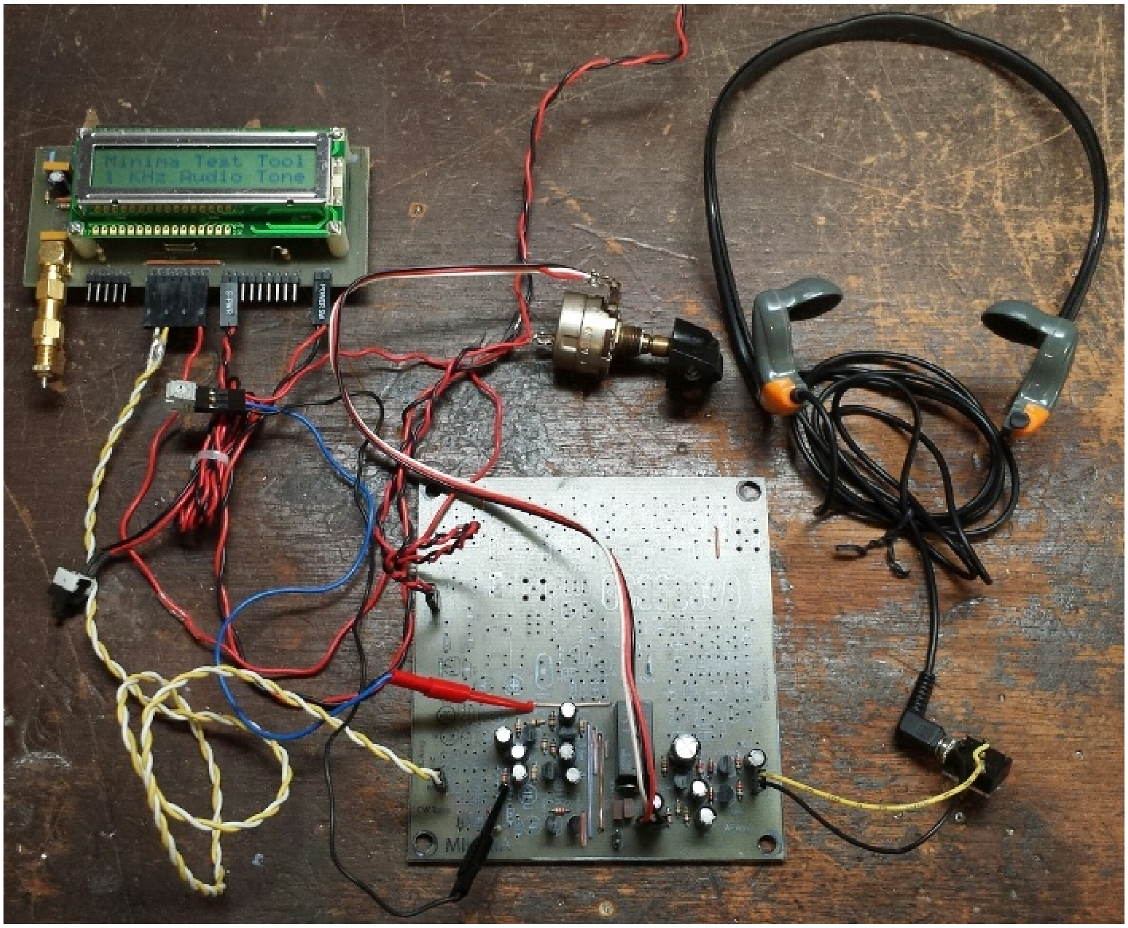

Illustration 37: Test setup form the Main Board RX Audio Preamp circuit test.

4. Turn on the switch providing power to the Digital Board and the Main Board. Note that, no sound will be heard coming from the headphones initially. The power-up delay circuit in the preamp takes a few seconds to charge up and provide current to the rest of the preamp circuit.
5. Increase volume of the Output Audio until the squarewave tone is heard coming from the headphone. It may also be necessary to very slightly adjust the trimmer pot supplying the squarewave signal to provide more signal injected into the preamp input. If no tone is heard, regardless of volume adjustment, remove power from the Digital and Main Boards and determine the cause of the problem. A problem may exist due to an error that occurred during assembly or in setting up for the test.
6. Press the PTT button and observe that the squarewave tone should stop coming from the headphone. Releasing the PTT should result in the squarewave tone being heard from the headphones again.

This completes the testing of the RX Audio Preamp circuitry on the Main Board.

Mic Preamp Assembly

The Mic Preamp provides amplification of an electret microphone to a signal level sufficient for SSB modulation and transmission. Either a discrete electret microphone or a PC-style electret microphone may be used, depending upon the wiring used to connect it to the input of the mic preamp.

The schematic for the Mic Preamp circuitry is shown in Illustration 38. Bias voltage is supplied to the microphone via R49, a 4.7 Kohm resistor. DC isolation from the bias network of the amplifier is provided by C38, a 1 uF electrolytic capacitor. Transistor Q26 is configured as a common-emitter amplifier with a voltage gain of about 4.5. The mic preamplifier output is DC isolated from the rest of the audio circuit by C41, a 1 uF electrolytic capacitor.

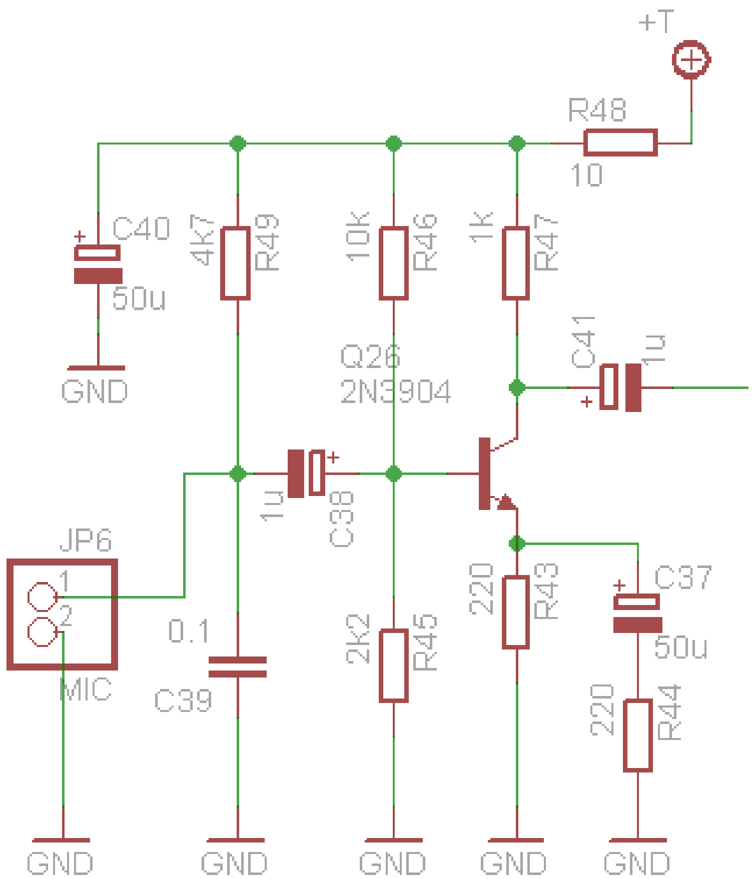

Illustration 38: Schematic of the Main Board Mic Preamp circuitry.

The location of the Mic Preamp circuity on the layout of the Main Board is shown in Illustration 39, highlighted in brown. Note that the physical spacing betweem the components is fairly tight. Care must be taken to follow the specified assembly sequence in order to simplify the insertion and soldering of components.

The parts list for the Mic Preamp is presented in Table 7.

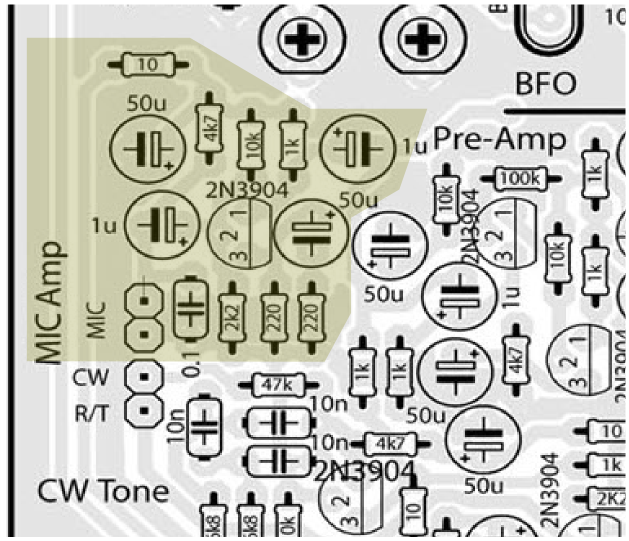

Illustration 39: Layout of the Main Board Mic Preamp circuitry.

Part	Value	Device
C37	50uF	CPOL-EUE2.5-5
C38	1uF	CPOL-EUE2.5-5
C39	0.1uF	C-EU050-024X044
C40	50uF	CPOL-EUE2.5-5
C41	1uF	CPOL-EUE2.5-5
JP6	MIC	PINHD-1X2
Q26	2N3904	2N3904

Part	Value	Device
R43	220R	R-EU_0204/5
R44	220R	R-EU_0204/5
R45	2k2	R-EU_0204/5
R46	10k	R-EU_0204/5
R47	1k	R-EU_0204/5
R48	10R	R-EU_0204/5
R49	4k7	R-EU_0204/5

Table 7: Parts list for the Main Board Mic Preamp.

The following steps describe the installation of the components for the RX Audio Preamp section of the Main Board.

1. Solder R43, a 220 ohm resistor, to the lower right of Q26, a 2N3904 NPN transistor.
2. Solder R44, a 220 ohm resistor, to the right of R43, a 220 ohm resistor.
3. Solder R45, a 2.2 Kohm resistor, to the left of R43, a 220 ohm resistor.
4. Solder R46, a 10 Kohm resistor, above Q26, a 2N3904 NPN transistor.
5. Solder R47, a 1 Kohm resistor, to the right of R46, a 10 Kohm resistor.
6. Solder R49, a 4.7 Kohm resistor, to the left of R46, a 10 Kohm resistor.
7. Solder R48, a 10 ohm resistor, to the upper left of R49, a 4.7 Kohm resistor.
8. Solder Q26, a 2N3904 NPN transistor, below R46, a 10 Kohm resistor. Ensure that the flat side of the transistor is oriented correctly prior to soldering.
9. Solder C37, a 50 uF electrolytic capacitor, to the right of Q26, a 2N3904 NPN transistor. Ensure that the polarity of the capacitor is oriented correctly prior to soldering.
10. Solder C38, a 1 uF electrolytic capacitor, to the left of Q26, a 2N3904 NPN transistor. Ensure that the polarity of the capacitor is oriented correctly prior to soldering

11. Solder C39, a 0.1 uF capacitor, to the lower left of Q26, a 2N3904 NPN transistor.
12. Solder C40, a 50 uF electrolytic capacitor, above C38, a 1 uF electrolytic capacitor. Ensure that the polarity of the capacitor is oriented correctly prior to soldering.
13. Solder C41, a 1 uF electrolytic capacitor, to the right of R47, a 1 Kohm resistor. Ensure that the polarity of the capacitor is oriented correctly prior to soldering.
14. Solder JP6, a 1x2 pin header, to the right of C39, a 0.1 uF capacitor.

The completed assembly of the Main Board Mic Preamp circuit is shown in Illustration 40.

Illustration 40: Component placement for the Main Board Mic Preamp circuitry.

Mic Preamp Testing

The Mic Preamp is easily tested by connecting a microphone to the input of the Mic Preamp and listening to what is produced at the output. To do this, it is necessary to insert a jumper to header JP11 (+R/+T) in order to apply power to both the transmit and receive power busses. The audio output from the Mic Preamp then passes through the RX Audio Preamp and the Ouput Audio Amp to a set of headphones. **The headphones need to be worn during this test so that audio feedback to the microphone does not occur. Therefore, it is important to not increase volume of the Output Audio Amp too much. Otherwise, damage to the audio amp, the headphones, or your hearing may occur. Also, do not forget to remove the jumper from JP11 after completing this test.**

While there is no specific Arduino sketch required to perform this test, the sketch that is loaded must set up Arduino pin PD3 as an input with the internal pull-up enabled. The sketches listed in Appendix C and Appendix E are suitable for this task.

The following steps describe the testing of the Mic Preamp circuit of the Main Board.

1. Download an appropriate Arduino sketch into the Digital Board.
2. Several cables are needed to test the Mic Preamp, most being reused from previous tests. The cable that connects a 10 Kohm, audio taper potentiometer to the amplifier input (VOL) JP7 will be used again for this test. This cable is shown at the upper left and bottom of Illustration 41. The cable that delivers the amplifier output (AUDIO) JP5 to an audio headphone receptacle will also be used again. It is shown on the right hand side of Illustration 41. A second cable, wired in the same fashion, is used to connect a PC-style electret microphone to the MIC input header JP6. It is shown on the left hand side of Illustration 41. The connector shell that attaches the T/R and CW pins to the Main Board will be used again for this test. This cable is shown in the upper right of Illustration 41. Lastly, a jumper will be used to connect the two pins of the +R/+T header JP11. The jumper is shown in the center of Illustration 41.

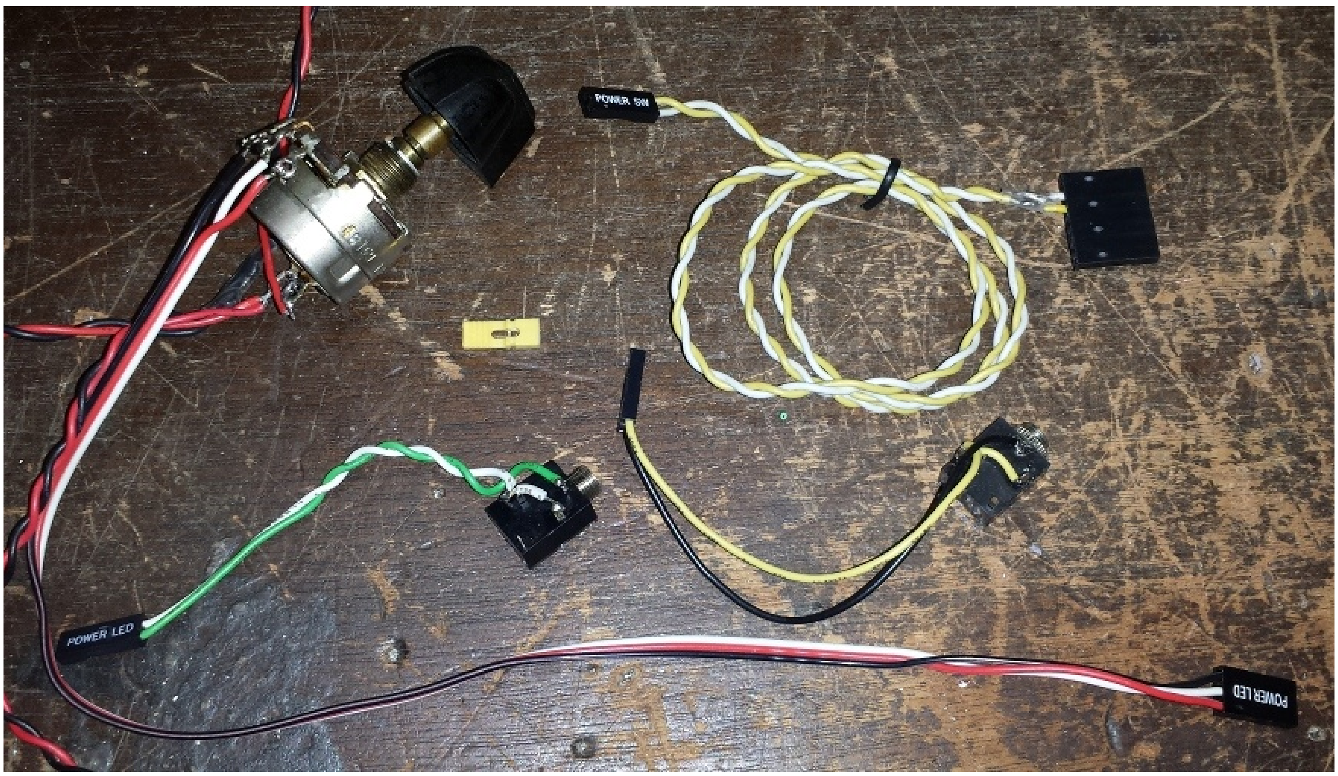

Illustration 41: Cable set used for testing the Main Board Mic Preamp circuit.

3. Connect the Digital Board and Main Board using the cables as shown in Illustration 42. Connect the T/R and CW cable, headphone cable, and output audio volume control cable as performed in previous tests. Connect the microphone cable to the MIC header JP6. Strap the jumper across the two pins of the +R/+T header JP11. Plug a PC-style electret microphone into the microphone input receptacle and a headphone into the audio output receptacle.

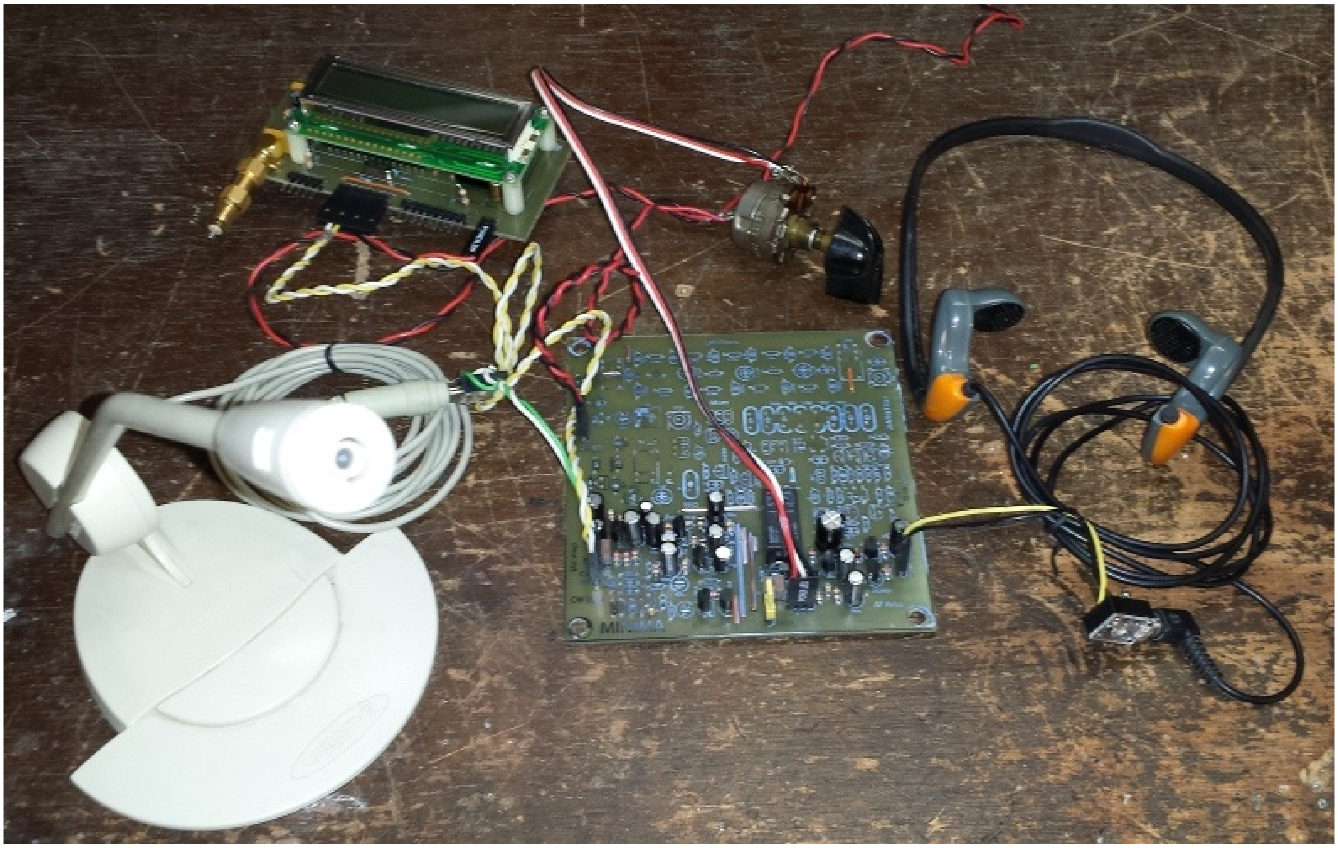

Illustration 42: Test setup form the Main Board Mic Preamp circuit test.

4. It will be necessary to wear the headphones during this test in order to minimize the possibility of audio feedback from the headphones to the microphone. **It is strongly recommended to reduce the volume of the Output Audio Amp to its minimum prior to applying power to the circuitry. This is to avoid damage to the Output Audio Amp, the headphones, or the listener's hearing.**
5. Apply power to the Minima Digital Board and Main Board. Gradually turn up the volume of the Output Audio Amp. Remember that it will take a few seconds for the capacitors in the Output Audio Amp to charge up and sound to come out of the headphones. At that point, it should be possible to hear on the headphones sounds being picked up by the microphone. If this does not occur after adjusting the output volume, remove power and investigate possible problems with the Mic Preamp circuitry.

This completes the testing of the Main Board Mic Preamp. **Do not forget to remove the jumper from JP11, the +R/+T header.**

CW Tone Oscillator Assembly

The CW Tone Oscillator generates an audio frequency tone for Morse code (CW) transmissions. The SSB modulation shifts the tone from baseband modulation to the intermediate frequency and then to the transmit radio frequency. The signal transmitted is the tuned frequency plus or minus the CW tone oscillator frequency depending upon whether the transmission is upper or lower side band, respectively.

The schematic for the CW Tone Oscillator circuitry is shown in Illustration 43. A phase shift oscillator is used to generate the audio tone. The three RC stages (C47 and R56, C48 and R57, C49 and R58) provide a 60° phase shift to the signal and Q30, a 2N3904 transistor, is configured as a common-emitter amplifier with a voltage gain of about 470. Oscillation is controlled via CW header JP10, which is driven by the digital output 'CW' on the Digital Board. An Arduino sketch can set the CW output to logic 'high' to turn on the CW Tone Oscillator. Setting the CW output to logic 'low' turns off the oscillator. The output of the CW Tone Oscillator is connected to the IF Mixer and RX Audio Preamp at all times and is also switched, via relay K2, to the Output Audio Amp during transmission.

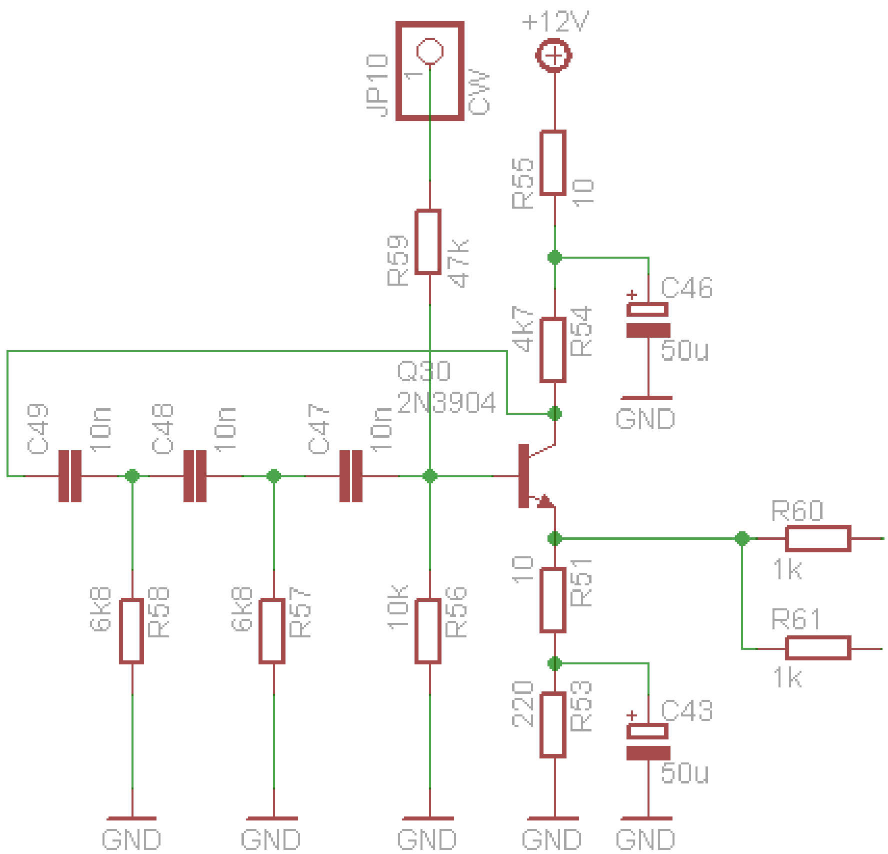

Illustration 43: Schematic of the Main Board Tone Oscillator circuitry.

The location of the CW Tone Oscillator circuity on the layout of the Main Board is shown in Illustration 44, highlighted in purple. Note that the physical spacing betweem the components is fairly tight. Care must be taken to follow the specified assembly sequence in order to simplify the insertion and soldering of components.

The parts list for the CW Tone Oscillator is presented in Table 8.

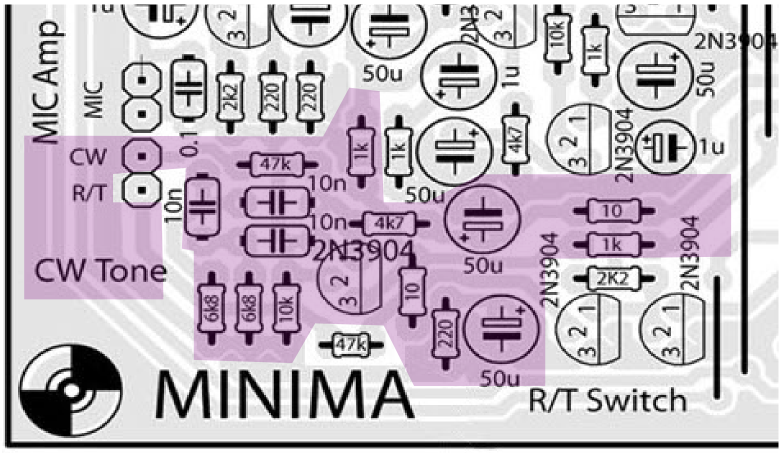

Illustration 44: Layout of the Main Board CW Tone Oscillator circuitry.

Part	Value	Device	Part	Value	Device
C43	50uF	CPOL-EUE2.5-5	R54	4k7	R-EU_0204/5
C46	50uF	CPOL-EUE2.5-5	R55	10R	R-EU_0204/5
C47	10nF	C-EU050-024X044	R56	10k	R-EU_0204/5
C48	10nF	C-EU050-024X044	R57	6k8	R-EU_0204/5
C49	10nF	C-EU050-024X044	R58	6k8	R-EU_0204/5
Q30	2N3904	2N3904	R59	47k	R-EU_0204/5
R51	10R	R-EU_0204/5	R60	1k	R-EU_0204/5
R53	220R	R-EU_0204/5	R61	1k	R-EU_0204/5

Table 8: Parts list for the Main Board CW Tone Oscillator.

The following steps describe the installation of the components for the CW Tone Oscillator section of the Main Board. Note that the CW header JP10 was installed during the assembly of the R/T Switch circuitry.

1. Solder R51, a 10 ohm resistor, to the right of Q30, a 2N3904 NPN transistor.
2. Solder R53, a 220 ohm resistor, to the lower right of R51, a 10 ohm resistor.
3. Solder R54, a 4.7 Kohm resistor, to the upper right of Q30, a 2N3904 NPN transistor.
4. Solder R55, a 10 ohm resistor, to the right of C46, a 50 uF electrolytic capacitor.
5. Solder R56, a 10 Kohm resistor, to the lower left of Q30, a 2N3904 NPN transistor.
6. Solder R57, a 6.8 Kohm resistor, to the left of R56, a 10 Kohm resistor.
7. Solder R58, a 6.8 Kohm resistor, to the left of R57, a 6.8 Kohm resistor.
8. Solder R59, a 47 Kohm resistor, to the upper left of R54, a 4.7 Kohm resistor.
9. Solder R60, a 1 Kohm resistor, to the right of R59, a 47 Kohm resistor.
10. Solder R61, a 1 Kohm resistor, to the right of C46, a 50 uF electrolytic capacitor.
11. Solder Q30, a 2N3904 NPN transistor, to the left of R56, a 10 Kohm resistor. Ensure that the flat side of the transistor is oriented correctly prior to soldering.
12. Solder C43, a 50 uF electrolytic capacitor, to the right of R53, a 220 ohm resistor. Ensure that

the polarity of the capacitor is oriented correctly prior to soldering.

13. Solder C46, a 50 uF electrolytic capacitor, to the right of R54, a 4.7 Kohm resistor. Ensure that the polarity of the capacitor is oriented correctly prior to soldering.
14. Solder C47, a 10 nF capacitor, to the upper left of Q30, a 2N3904 NPN transistor.
15. Solder C48, a 10 nF capacitor, to the upper left of C47, a 10 nF capacitor.
16. Solder C49, a 10 nF capacitor, above C47, a 10 nF capacitor.

The completed assembly of the Main Board Mic Preamp circuit is shown in Illustration 45.

Illustration 45: Component placement for the Main Board CW Tone Oscillator assembly.

CW Tone Oscillator Testing

To test the CW Tone Oscillator circuit, it is necessary provide a way to pull up the voltage on the CW header JP10 to around five volts. The easiest way to do this is to use an Arduino sketch to control the CW pin on the Digital Board. The Arduino sketch used previously to generate a 1 KHz audio test tone also monitors the analog to digital converter on the Analog Key input. When the value measured is less than 512, the CW pin is turned on. This turns on the CW Tone Oscillator. The CW pin is turned off, along with the tone oscillator, when the value measured is greater than 512. The listing for the Arduino sketch is provided in Appendix E.

The following steps describe the testing of the CW Tone Oscillator circuit of the Main Board.

1. Compile and download the Arduino sketch listed in Appendix E into the Digital Board.
2. Several cables are needed to test the CW Tone Oscillator, most being reused from previous tests. The cable that connects a 10 Kohm, audio taper potentiometer to the amplifier input (VOL) JP7 will be used again for this test. This cable is shown at the upper left and bottom of

Illustration 46. The connector shell that attaches the T/R and CW pins to the Main Board will be used again for this test. This cable is shown in the top of Illustration 46. The cable that delivers the amplifier output (AUDIO) JP5 to an audio headphone receptacle will also be used again. It is shown on the right hand side of Illustration 46. A cable should be constructed with a momentary switch to connect to the Analog Key input on the Digital Boad. An example of such a cable is shown in the center of Illustration 46. Finally a cable should be constructed with a toggle switch to connect to the PTT input on the Digital Board. An example of such a cable is shown in the lower center of Illustration 46.

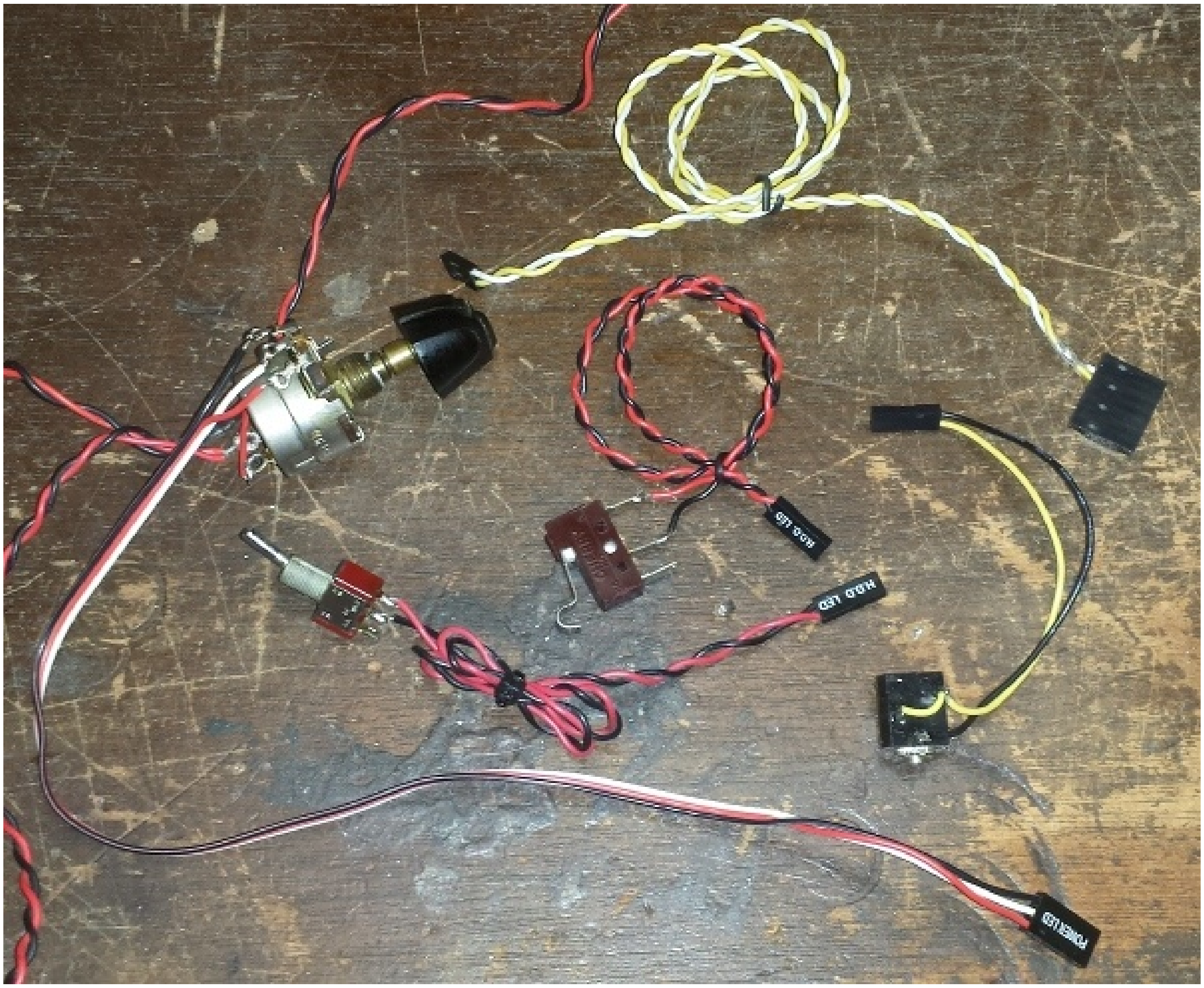

Illustration 46: Cable set used for testing the Main Board CW Tone Oscillator circuit.

3. Connect the Digital Board and Main Board using the cables as shown in Illustration 47. Connect the T/R and CW cable, headphone cable, and output audio volume control cable as performed in previous tests. Connect the cable with the momentary switch to the Analog Key input on the Digital Board. Connect the cable with the toggle switch to the PTT input on the Digital Board. Plug a headphone into the audio output receptacle. **It is recommended to NOT wear the headphones during the first portion of the test in order to prevent damage to hearing.**

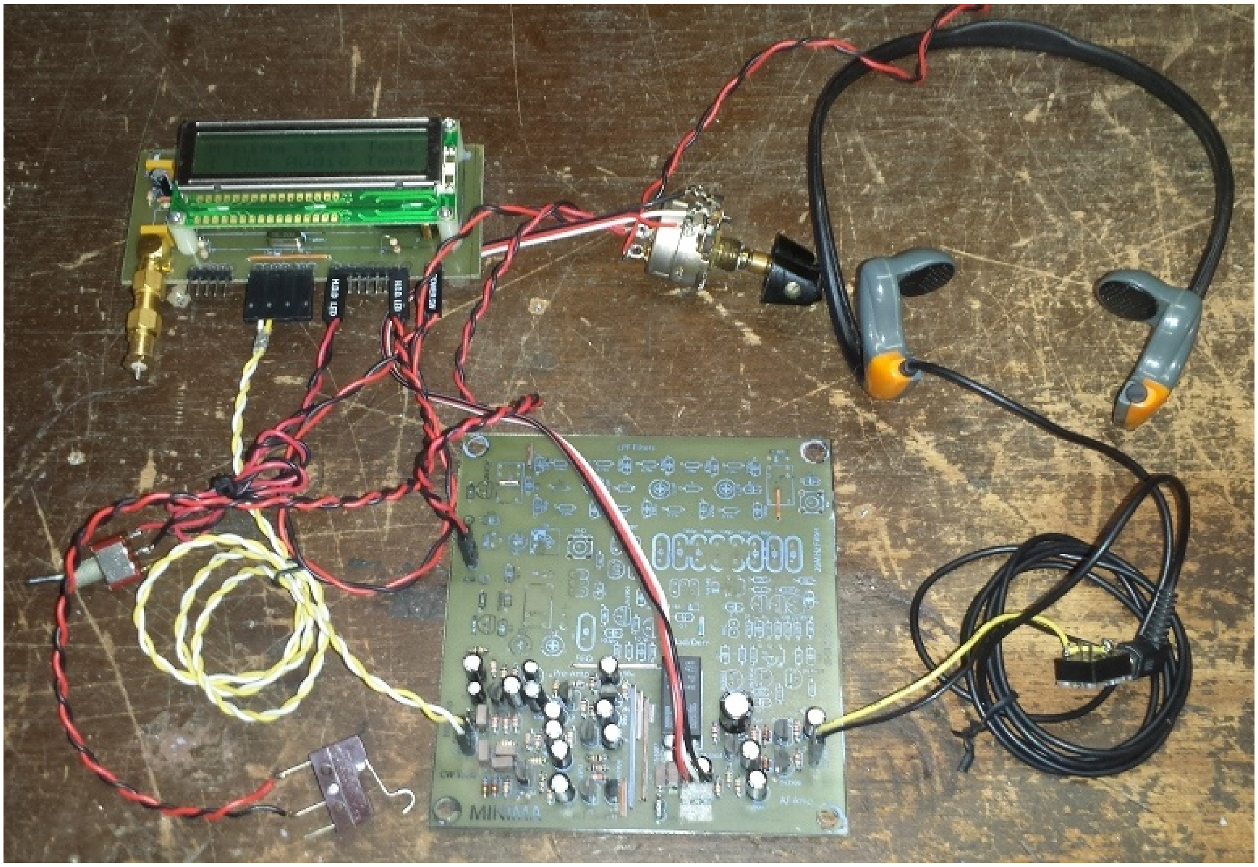

Illustration 47: Test setup form the Main Board CW Tone Oscillator circuit test.

4. Turn on the switch providing power to the Digital Board and the Main Board. Note that, no sound will be heard coming from the headphones initially. The power-up delay circuit in the preamp takes a few seconds to charge up and provide current to the rest of the preamp circuit.
5. Make sure that the toggle switch is in the off position, that is, the contacts are not closed and the Main Board is in the receive state. Depress the momentary switch connected to the Analog Key input and increase the volume of the Output Audio until the audio tone is heard coming from the headphone. If no tone is heard, regardless of volume adjustment, remove power from the Digital and Main Boards and determine the cause of the problem. A problem may exist due to an error that occurred during assembly or in setting up for the test.
6. Switch the PTT button to the on position, that is, the contacts are closed and the Main Board is in the transmit state. Observe that the audio tone coming from the headphone should become inaudiable. Increasing the volume of the Output Audio Amp significantly should enable the audio to to be heard again. **Use caution while listening to the headphone durng this portion of the test. Damage to hearing could result if the PTT switch is accidentally switched back to the receive state.**

This completes the testing of the CW Tone Oscillator circuitry on the Main Board.

Appendix A - Minima Transceiver Schematics

The schematics for the Minima Serial, Digital, and Main boards were originally created by Adrian Preda YO4HHP using Eagle CAD. The schematics displayed below have been reorganized by the author to improve readability. No circuit changes have been made otherwise.

Serial Board Schematic

The schematic for the Serial Board is presented in Illustration 48.

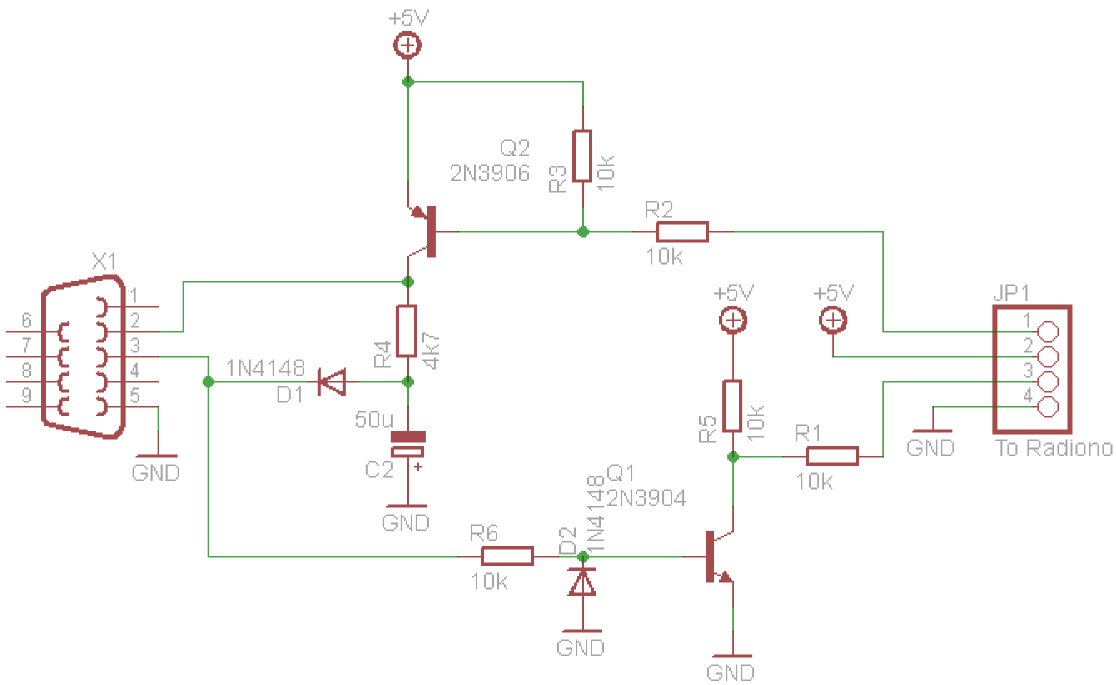

Illustration 48: Schematic of the Minima Serial Board.

Digital Board Schematic

The schematic for the Digital Board is presented in Illustration 49.

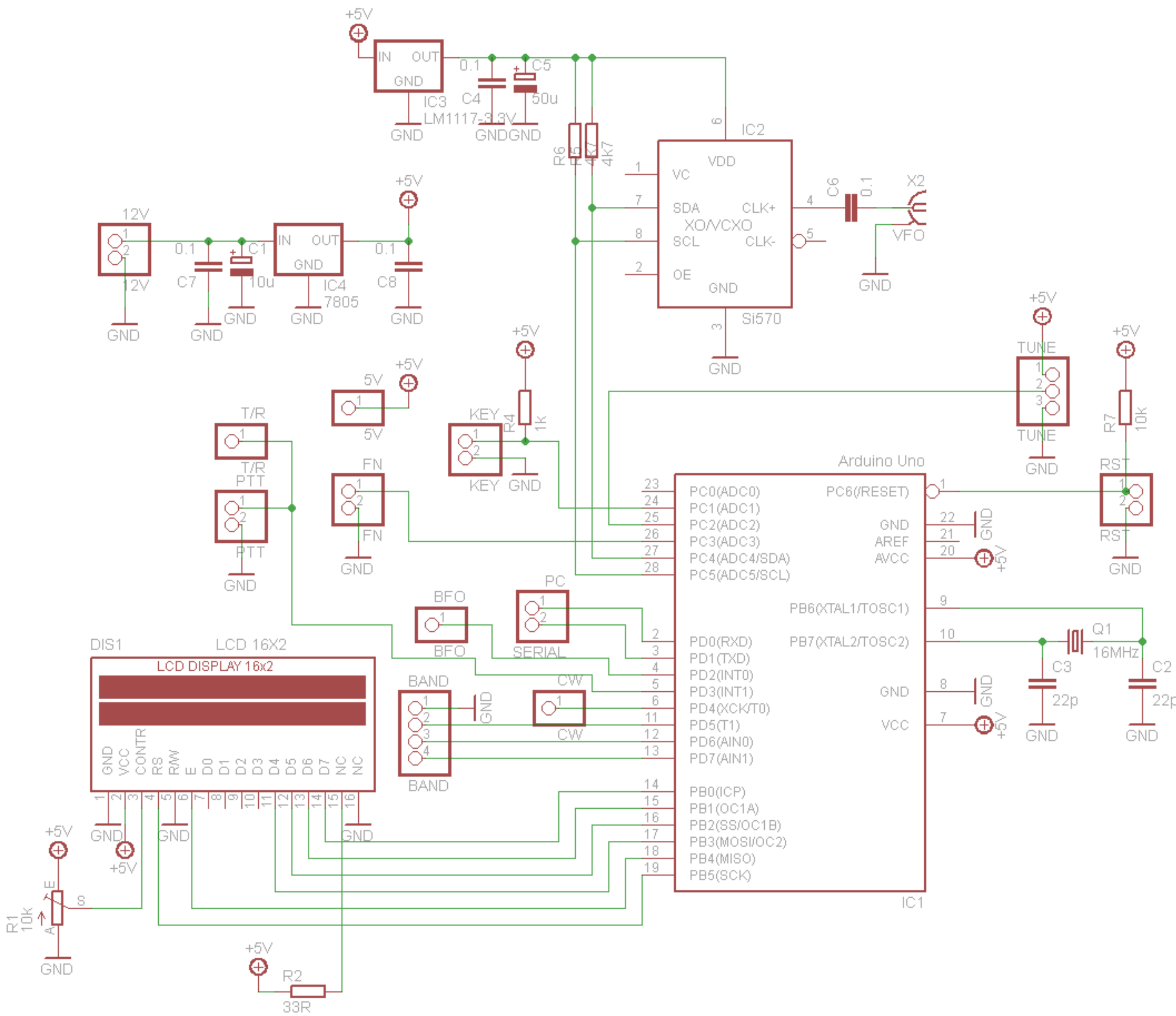

Illustration 49: Schematic of the Minima Digital Board.

Minima Main Board Schematic

The schematic for the Main Board is presented in two parts in Illustration 50 and Illustration 51 on the pages 51 and 52, respectively. Annotations have been added by the author to indicate different functional elements. Note that JP1, JP2, JP3, and JP4 are not pin headers, but are actually toroidal inductors. Also note that the polarity of capacitor C29, a 50 uF electrolytic capacitor is reversed.

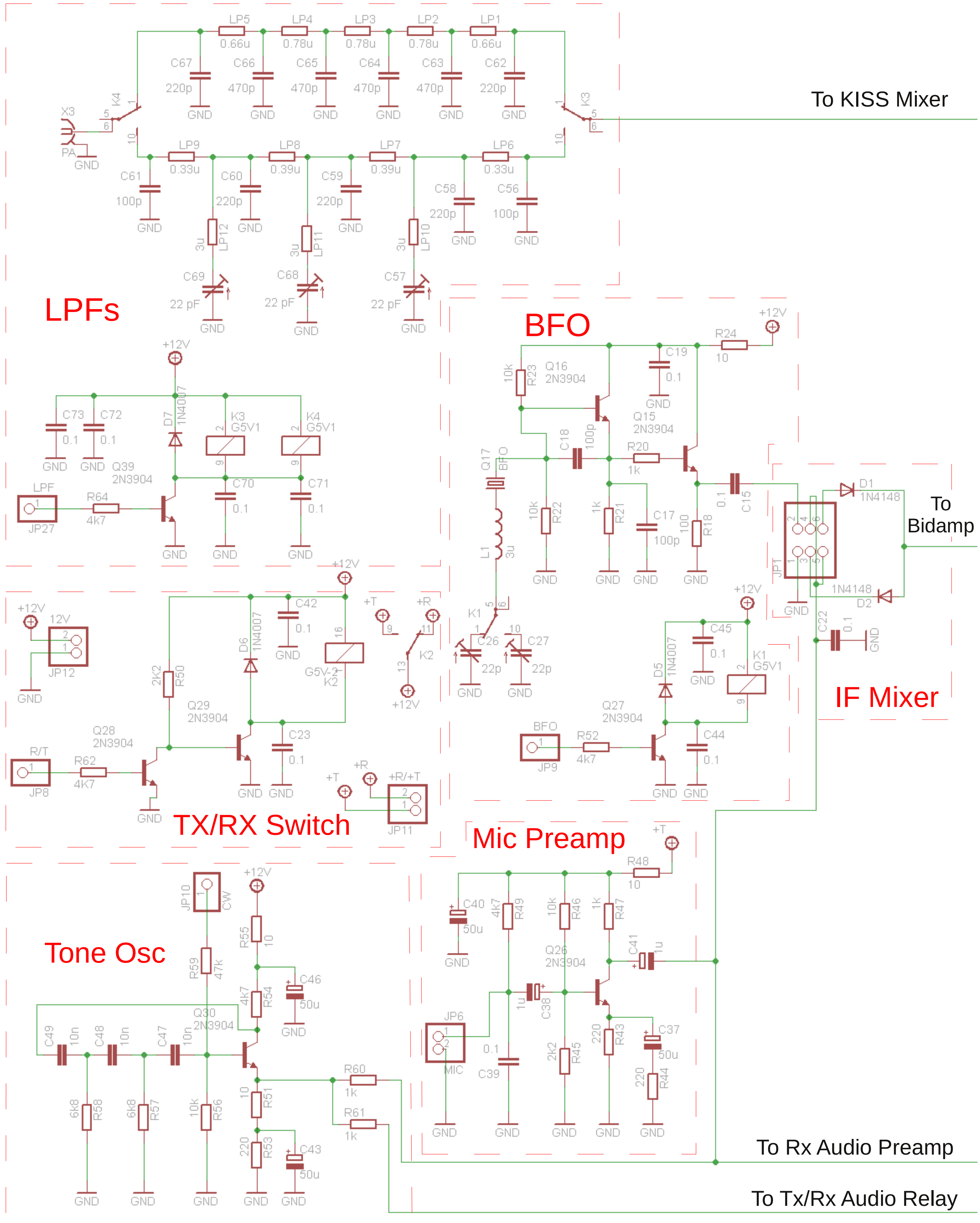

Illustration 50: Schematic for the Minima Main Board (Part 1 of 2).

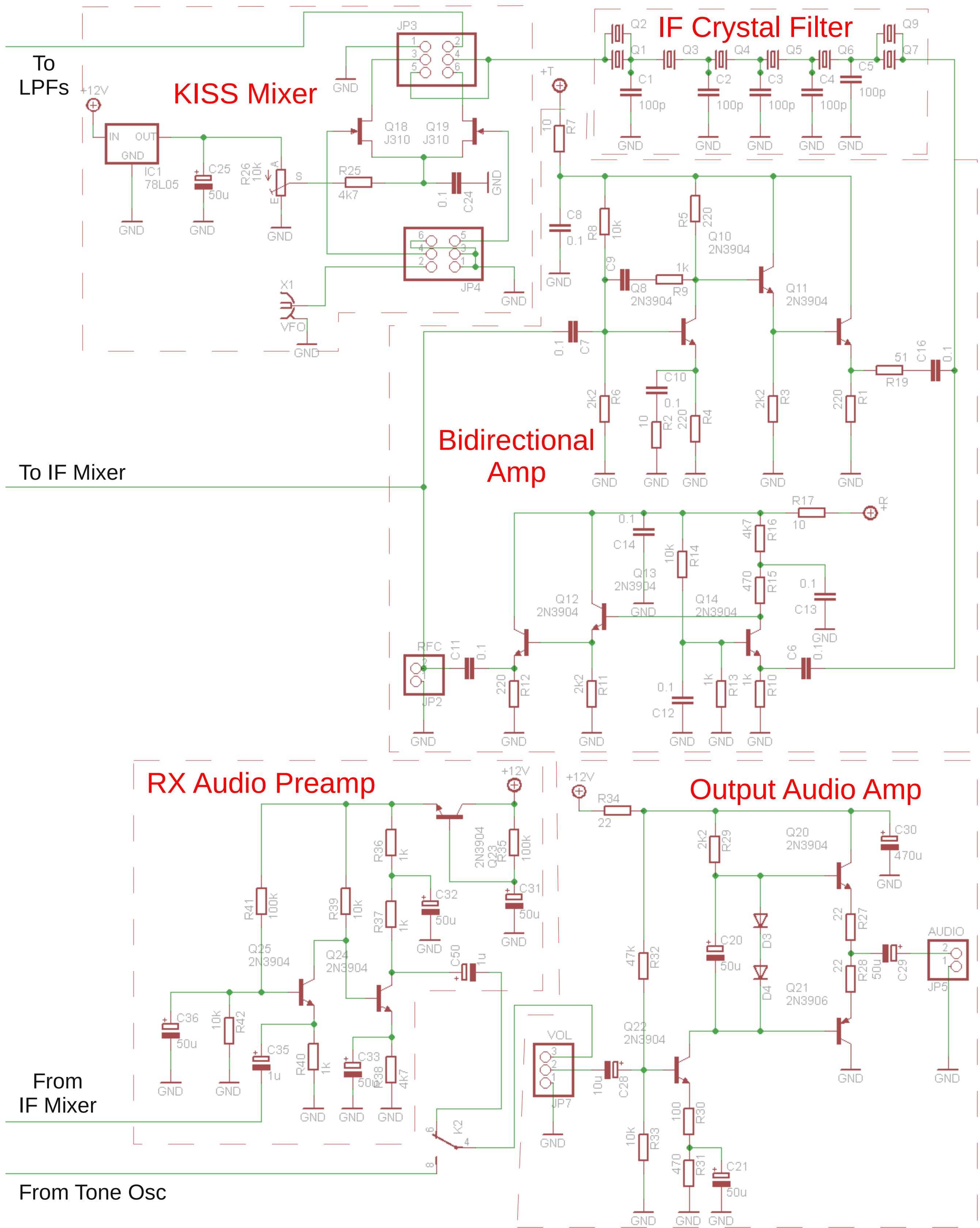

Illustration 51: Schematic for the Minima Main Board (Part 2 of 2).

Appendix B - Minima Transceiver Parts Lists

The Minima transceiver consists of three printed circuit boards, the Serial Board, the Digital Board, and the Main Board. The Serial Board provides an interface between RS232 serial data voltage levels and the 5 volt data voltage levels of the Arduino microcontroller on the Digital Board. The Digital Board holds the Arduino microcontroller and the Si570 VFO. The Main Board holds all of the analog audio and radio frequency circuitry. Note that constructing the Serial Board is optional, as the user may want to program the Arduino using a USB to Serial TTL interface.

Below are three parts lists corresponding to these three boards described above. The part numbering is based upon the EagleCAD schematics created by Adrian Preda YO4HHP.

Serial Board Parts List

Part	Value	Device
C2	50u	CPOL-EUE2.5-5
D1	1N4148	DIODE-D-5
D2	1N4148	DIODE-D-5
JP1	To Radiono	PINHD-1X4
Q1	2N3904	TO92
Q2	2N3906	TO92
R1	10k	R-EU_0204/5
R2	10k	R-EU_0204/5
R3	10k	R-EU_0204/5
R4	4k7	R-EU_0204/5
R5	10k	R-EU_0204/5
R6	10k	R-EU_0204/5
X1	F09D	con-subd

Digital Board Parts List

Part	Value	Device
C1	10 uF, 50 V	CPOL-EUE2.5-5
C2	22 pF SMD	C-EUC1206
C3	22 pF SMD	C-EUC1206
C4	0.1 uF	C-EU050-024X044
C5	50 uF 6.3 V	CPOL-EUE2.5-5
C6	0.1 uF	C-EU050-024X044
C7	0.1 uF	C-EU050-024X044
C8	0.1 uF	C-EU050-024X044
DIS1	LCD	16X2 TUXGR_16X2_R2
IC1	Arduino Uno	ATMEGA328-P
IC2	Si570	SI570
IC3	LM1117-3.3V	LM1117TO252
IC4	LM7805	78XXL

Part	Value	Device
P1	12V	PINHD-1X2
P2	PTT, FN, TUNE, KEY	PINHD-1X9
P3	BFO, T/R, CW, BAND	PINHD-1X7
P4	5V, RST, SERIAL	PINHD-1X5
Q1	16 Mhz	CRYSTALHC49S
R1	10k	TRIM_EU-CA6V
R2	33R	R-EU_0207/15
R4	1k	R-EU_0204/5
R5	4k7	R-EU_0204/5
R6	4k7	R-EU_0204/5
R7	10k	R-EU_0204/5
X2	VFO	BU-SMA-V

Main Board Parts List

The Main Board parts list is presented on the following page.

Part	Value	Device
C1	100pF	C-EUC1206
C2	100pF	C-EUC1206
C3	100pF	C-EUC1206
C4	100pF	C-EUC1206
C5	100pF	C-EUC1206
C6	0.1uF	C-EU050-024X044
C7	0.1uF	C-EU050-024X044
C8	0.1uF	C-EU050-024X044
C9	0.1uF	C-EU050-024X044
C10	0.1uF	C-EU050-024X044
C11	0.1uF	C-EU050-024X044
C12	0.1uF	C-EU050-024X044
C13	0.1uF	C-EU050-024X044
C14	0.1uF	C-EU050-024X044
C15	0.1uF	C-EU050-024X044
C16	0.1uF	C-EU050-024X044
C17	100pF	C-EU050-024X044
C18	100pF	C-EU050-024X044
C19	0.1uF	C-EU050-024X044
C20	50uF	CPOL-EUE2.5-5
C21	50uF	CPOL-EUE2.5-5
C22	0.1uF	C-EU050-024X044
C23	0.1uF	C-EU050-024X044
C24	0.1uF	C-EU050-024X044
C25	50uF	CPOL-EUE2.5-5
C26	22pF	C-TRIMMTZ03
C27	22pF	C-TRIMMTZ03
C28	10uF	CPOL-EUE2.5-5
C29	50uF	CPOL-EUE2.5-5
C30	470uF	CPOL-EUE2.5-5
C31	50uF	CPOL-EUE2.5-5
C32	50uF	CPOL-EUE2.5-5
C33	50uF	CPOL-EUE2.5-5
C35	1uF	CPOL-EUE2.5-5
C36	50uF	CPOL-EUE2.5-5
C37	50uF	CPOL-EUE2.5-5
C38	1uF	CPOL-EUE2.5-5
C39	0.1uF	C-EU050-024X044
C40	50uF	CPOL-EUE2.5-5
C41	1uF	CPOL-EUE2.5-5
C42	0.1uF	C-EU050-024X044
C43	50uF	CPOL-EUE2.5-5
C44	0.1uF	C-EU050-024X044
C45	0.1uF	C-EU050-024X044
C46	50uF	CPOL-EUE2.5-5
C47	10nF	C-EU050-024X044
C48	10nF	C-EU050-024X044
C49	10nF	C-EU050-024X044
C50	1uF	CPOL-EUE5-4
C56	100pF	C-EU050-024X044
C57	22pF	C-TRIMMTZ03
C58	220pF	C-EU050-024X044
C59	220pF	C-EU050-024X044
C60	220pF	C-EU050-024X044
C61	100pF	C-EU050-024X044
C62	220pF	C-EU050-024X044
C63	470pF	C-EU050-024X044
C64	470pF	C-EU050-024X044
C65	470pF	C-EU050-024X044
C66	470pF	C-EU050-024X044
C67	220pF	C-EU050-024X044
C68	22pF	C-TRIMMTZ03
C69	22pF	C-TRIMMTZ03
C70	0.1uF	C-EU050-024X044
C71	0.1uF	C-EU050-024X044
C72	0.1uF	C-EU050-024X044
C73	0.1uF	C-EU050-024X044

Part	Value	Device
D1	1N4148	DIODE-D-5
D2	1N4148	DIODE-D-5
D3	1N4148	DIODE-D-5
D4	1N4148	DIODE-D-5
D5	1N4007	DIODE-D-5
D6	1N4007	DIODE-D-5
D7	1N4007	DIODE-D-5
IC1	78L05	78LXX
JP1	T37-43	8T-trifilar, 32 AWG
JP2	T37-43	10T 32 AWG
JP3	T37-43	8T-trifilar, 32 AWG
JP4	T37-43	8T-trifilar, 32 AWG
JP5	AUDIO	PINHD-1X2
JP6	MIC	PINHD-1X2
JP7	VOL	PINHD-1X3
JP8, JP10	R/T, CW	PINHD-1X2
JP9	BFO	PINHD-1X1
JP11	+R/+T	PINHD-1X2
JP12	12V	PINHD-1X2
JP27	LPF	PINHD-1X1
K1	G5V1	G5V1
K2	G5V-2	G5V-2
K3	G5V1	G5V1
K4	G5V1	G5V1
L1	3uh	L-US0207/2V
LP1	0.66uh	R-EU_0204/5
LP2	0.78uh	R-EU_0204/5
LP3	0.78uh	R-EU_0204/5
LP4	0.78uh	R-EU_0204/5
LP5	0.66uh	R-EU_0204/5
LP6	0.33uh	R-EU_0204/5
LP7	0.39uh	R-EU_0204/5
LP8	0.39uh	R-EU_0204/5
LP9	0.33u	R-EU_0204/5
LP10	3uh	R-EU_0204/5
LP11	3uh	R-EU_0204/5
LP12	3uh	R-EU_0204/5
Q1	20 MHz XTAL	CRYSTALHC49S
Q2	20 MHz XTAL	CRYSTALHC49S
Q3	20 MHz XTAL	CRYSTALHC49S
Q4	20 MHz XTAL	CRYSTALHC49S
Q5	20 MHz XTAL	CRYSTALHC49S
Q6	20 MHz XTAL	CRYSTALHC49S
Q7	20 MHz XTAL	CRYSTALHC49S
Q8	2N3904	2N3904
Q9	20 MHz XTAL	CRYSTALHC49S
Q10	2N3904	2N3904
Q11	2N3904	2N3904
Q12	2N3904	2N3904
Q13	2N3904	2N3904
Q14	2N3904	2N3904
Q15	2N3904	2N3904
Q16	2N3904	2N3904
Q17	20 MHz XTAL	CRYSTALHC49S
Q18	J310	J310
Q19	J310	J310
Q20	2N3904	2N3904
Q21	2N3906	2N3906
Q22	2N3904	2N3904
Q23	2N3904	2N3904
Q24	2N3904	2N3904
Q25	2N3904	2N3904
Q26	2N3904	2N3904
Q27	2N3904	2N3904
Q28	2N3904	2N3904
Q29	2N3904	2N3904
Q30	2N3904	2N3904

Part	Value	Device
Q39	2N3904	2N3904
R1	220R	R-EU_0204/5
R2	10R	R-EU_0204/5
R3	2k2	R-EU_0204/5
R4	220R	R-EU_0204/5
R5	220R	R-EU_0204/5
R6	2k2	R-EU_0204/5
R7	10R	R-EU_0204/5
R8	10k	R-EU_0204/5
R9	1k	R-EU_0204/5
R10	1k	R-EU_0204/5
R11	2k2	R-EU_0204/5
R12	220R	R-EU_0204/5
R13	1k	R-EU_0204/5
R14	10k	R-EU_0204/5
R15	470R	R-EU_0204/5
R16	4k7	R-EU_0204/5
R17	10R	R-EU_0204/5
R18	100R	R-EU_0204/5
R19	51R	R-EU_0204/5
R20	1k	R-EU_0204/5
R21	1k	R-EU_0204/5
R22	10k	R-EU_0204/5
R23	10k	R-EU_0204/5
R24	10R	R-EU_0204/5
R25	4k7	R-EU_0204/5
R26	10k	TRIM_EU-CA6V
R27	22R	R-EU_0204/5
R28	22R	R-EU_0204/5
R29	2k2	R-EU_0204/5
R30	100R	R-EU_0204/5
R31	470R	R-EU_0204/5
R32	47k	R-EU_0204/5
R33	10k	R-EU_0204/5
R34	22R	R-EU_0204/5
R35	100k	R-EU_0204/5
R36	1k	R-EU_0204/5
R37	1k	R-EU_0204/5
R38	4k7	R-EU_0204/5
R39	10k	R-EU_0204/5
R40	1k	R-EU_0204/5
R41	100k	R-EU_0204/5
R42	10k	R-EU_0204/5
R43	220R	R-EU_0204/5
R44	220R	R-EU_0204/5
R45	2k2	R-EU_0204/5
R46	10k	R-EU_0204/5
R47	1k	R-EU_0204/5
R48	10R	R-EU_0204/5
R49	4k7	R-EU_0204/5
R50	2K2	R-EU_0204/5
R51	10R	R-EU_0204/5
R52	4k7	R-EU_0204/5
R53	220R	R-EU_0204/5
R54	4k7	R-EU_0204/5
R55	10R	R-EU_0204/5
R56	10k	R-EU_0204/5
R57	6k8	R-EU_0204/5
R58	6k8	R-EU_0204/5
R59	47k	R-EU_0204/5
R60	1k	R-EU_0204/5
R61	1k	R-EU_0204/5
R62	4K7	R-EU_0204/5
R64	4k7	R-EU_0204/5
X1	VFO	BU-SMA-V
X3	PA	BU-SMA-V

Appendix C – Arduino Sketch for Digital Board I/O Testing

```
/*
 I/O Testing
 This sketch tests most of the digital and analog inputs and outputs on the
 Arduino microcontroller as they will be used on the Minima transceiver.
 Note that the I2C port, which communicates with the Si570, is not tested.
 */

#include <LiquidCrystal.h>
#include <stdlib.h>

/*
 The 16x2 LCD is connected as follows:
   LCD's PIN    Raduino's PIN  PURPOSE      ATMEGA328's PIN
   4            13             Reset LCD    19
   6            12             Enable       18
   11           10             D4           17
   12           11             D5           16
   13            9             D6           15
   14            8             D7           14
*/
LiquidCrystal lcd(13, 12, 11, 10, 9, 8); // LCD configuration.

/*
 Digital Controls
   Raduino's PIN  PURPOSE              Orientation  ATMEGA328's PIN
   D2             LSB/USB BFO select   Output        4
   D3             TX/RX (PTT)          Input         5
   D4             CW Key               Output        6
   D5             Band High            Output       11
   D6             Band Select 1        Output       12
   D7             Band Select 0        Output       13
*/
#define BFO_LSB (2) // Logic 0 - USB, logic 1 - LSB.
#define TX_RX (3) // Logic 0 - RX, logic 1 - TX.
#define CW_KEY (4) // Logic 0 - unkeyed, logic 1 - keyed.
#define BAND_HI (5) // Logic 0 - low band, logic 1 - high band.
#define BAND_SEL_1 (6) // Currently unused.
#define BAND_SEL_0 (7) // Currently unused.

/*
 Analog Controls
   Raduino's PIN  PURPOSE            ATMEGA328's PIN
   A1             Analog Keyer       24
   A2             Analog Tuning      25
   A3             Function Button    26
*/
#define ANALOG_KEYER (A1)
#define ANALOG_TUNING (A2)
#define FBUTTON (A3) // Why does 'A3', versus '3' make a difference?

int analogKeyer = 0; // Analog keyer variable.
int tuningPosition = 0; // Tuning pot variable.
```

```
int functionButton = 0; // Function button varible.
int lineTX_RX = 0; // TX/RX line variable.
int blinkenLights = 0; // Blinking lights counter.
int bitValue; // Used to isolate bits from the counter.
char str[20]; // Storage for strings generated by sprintf.

void setup() {
  lcd.begin(16, 2);
  lcd.clear();

  // Set up the digital pins.
  pinMode(BFO_LSB, OUTPUT);
  pinMode(TX_RX, INPUT);
  pinMode(CW_KEY, OUTPUT);
  pinMode(BAND_SEL_0, OUTPUT);
  pinMode(BAND_SEL_1, OUTPUT);
  pinMode(BAND_HI, OUTPUT);

  // Set the side-tone off, put the transceiver to receive mode.
  digitalWrite(CW_KEY, 0);
  digitalWrite(TX_RX, 1); // Old way to enable the built-in pull-ups.
  // For some as yet unknown reason, the following makes the F-button
  // signal levels more stable.  If left out, the voltage level tends
  // to drift.  A pullup resistor may solve this issue.
  digitalWrite(FBUTTON, 1); // Old way to enable the built-in pull-ups.
}

void loop() {
  lcd.setCursor(0, 0); // Position cursor for first line of text.
  lcd.print("POT:     AK:    ");
  lcd.setCursor(0, 1); // Position cursor for second line of text.
  lcd.print("FN:     TX/RX:  ");

  // Read the tuning pot A/D.
  tuningPosition = analogRead(ANALOG_TUNING);
  // Display the digitized value.
  lcd.setCursor(4, 0);
  sprintf(str, "%4d", tuningPosition);
  lcd.print(str);

  // Read the Analog Keyer.
  analogKeyer = analogRead(ANALOG_KEYER);
  // Display the digitized value.
  lcd.setCursor(12, 0);
  sprintf(str, "%4d", analogKeyer);
  lcd.print(str);

  // Read the Function Button.
  functionButton = analogRead(FBUTTON);
  // Display the digitized value.
  lcd.setCursor(3, 1);
  sprintf(str, "%4d", functionButton);
  lcd.print(str);

  // Read the TX/RX Line.
  lineTX_RX = digitalRead(TX_RX);
  // Display the value.
```

```
  lcd.setCursor(14, 1);
  sprintf(str, "%d", lineTX_RX);
  lcd.print(str);

  bitValue = blinkenLights; // Copy the current count for bit extraction.
  digitalWrite(BAND_SEL_0, bitValue & 1); // Isolate lowest bit and write out.
  bitValue = bitValue >> 1; // Shift remaining bits to right by one bit.
  digitalWrite(BAND_SEL_1, bitValue & 1); // Isolate lowest bit and write out.
  bitValue = bitValue >> 1; // Shift remaining bits to right by one bit.
  digitalWrite(BAND_HI, bitValue & 1); // Isolate lowest bit and write out.
  bitValue = bitValue >> 1; // Shift remaining bits to right by one bit.
  digitalWrite(BFO_LSB, bitValue & 1); // Isolate lowest bit and write out.
  blinkenLights = (blinkenLights + 1) % 16; // Count from 0 to 15 and repeat.
  delay(100);  // Provide some delay to slow down the counting and display.
}
```

Appendix D – Arduino Sketch for Digital Board Si570 Testing

```
/*
 * Wire is only used from the Si570 module but we need to list it here so that
 * the Arduino environment knows we need it.
 */
#include <Wire.h>
#include <LiquidCrystal.h>

#include <avr/io.h>
#include <stdlib.h>
#include <math.h>

/*
 The 16x2 LCD is connected as follows:
    LCD's PIN   Raduino's PIN  PURPOSE      ATMEGA328's PIN
    4           13             Reset LCD    19
    6           12             Enable       18
    11          10             D4           17
    12          11             D5           16
    13           9             D6           15
    14           8             D7           14
*/

#define FREQ_XTAL (114292532l)
#define SI570_I2C_ADDRESS 0x55
// #define SI570_STARTUP_FREQUENCY 56320000L  // Most Ham Si570 have this startup
frequency.
#define SI570_STARTUP_FREQUENCY 10000000L // My Si570 has this startup frequency.
Select as appropriate.
#define SI570_NEW_FREQUENCY 12000000L

unsigned char si570_i2c_address = SI570_I2C_ADDRESS;
unsigned char dco_reg[13], dco_status='s';
uint8_t hs_reg_value;
uint8_t n_reg_value;
uint64_t refFrequencyUint64;
double refFrequency;
double dcoCurrentFrequency, freqXtalDbl;
unsigned long bitval[38];
unsigned long f_center=0, frequency=14200000, dco_freq=0;
unsigned int hs, n1;
unsigned long freq_xtal;
unsigned char wasSmall = 1;

LiquidCrystal lcd(13, 12, 11, 10, 9, 8);

int count = 0;
char b[20], c[20], printBuff[32];

void i2c_write (char slave_address,char reg_address, char data )  {
  int rdata = data;
  Wire.beginTransmission(slave_address);
  Wire.write(reg_address);
  Wire.write(rdata);
```

```
  Wire.endTransmission();
}

char i2c_read ( char slave_address, int reg_address ) {
  unsigned char rdata = 0xFF;
  Wire.beginTransmission(slave_address);
  Wire.write(reg_address);
  Wire.endTransmission();
  Wire.requestFrom(slave_address,1);
  if (Wire.available()) rdata = Wire.read();
  return rdata;
}

void read_si570(){
  //we have to read eight consecutive registers starting at register 5
  for (int i = 7; i <= 12; i++)
    dco_reg[i] = i2c_read( si570_i2c_address, i);
}

void write_si570()
{
  int idco, i;

  // Freeze DCO
  idco = i2c_read( si570_i2c_address,137);
  i2c_write(si570_i2c_address, 137, idco | 0x10 );

  i2c_write(si570_i2c_address, 7, dco_reg[7]);

  //Set Registers
  for( i=7; i <= 12; i++){
    i2c_write(si570_i2c_address, i, dco_reg[i]);
    idco = i2c_read( si570_i2c_address, i);
  }

  // Unfreeze DCO
  idco = i2c_read( si570_i2c_address, 137 );
  i2c_write (si570_i2c_address, 137, idco & 0xEF );

  // Set new freq
  i2c_write(si570_i2c_address,135,0x40);
}

void qwrite_si570()
{
  int i, idco;
  //Set Registers
  for( i=8; i <= 12; i++){
    i2c_write(si570_i2c_address, i, dco_reg[i]);
    idco = i2c_read( si570_i2c_address, i);
  }
}

void setBitvals(void){

      //set the rfreq values for each bit of the rfreq (integral)
      bitval[28] = (freq_xtal) / (hs * n1);
```

```
        bitval[29] = bitval[28] << 1;
        bitval[30] = bitval[29] << 1;
        bitval[31] = bitval[30] << 1;
        bitval[32] = bitval[31] << 1;
        bitval[33] = bitval[32] << 1;
        bitval[34] = bitval[33] << 1;
        bitval[35] = bitval[34] << 1;
        bitval[36] = bitval[35] << 1;
        bitval[37] = bitval[36] << 1;

        //set the rfreq values for each bit of the rfreq (integral)
        bitval[27] = bitval[28] >> 1;
        bitval[26] = bitval[27] >> 1;
        bitval[25] = bitval[26] >> 1;
        bitval[24] = bitval[25] >> 1;
        bitval[23] = bitval[24] >> 1;
        bitval[22] = bitval[23] >> 1;
        bitval[21] = bitval[22] >> 1;
        bitval[20] = bitval[21] >> 1;
        bitval[19] = bitval[20] >> 1;
        bitval[18] = bitval[19] >> 1;
        bitval[17] = bitval[18] >> 1;
        bitval[16] = bitval[17] >> 1;
        bitval[15] = bitval[16] >> 1;
        bitval[14] = bitval[15] >> 1;
        bitval[13] = bitval[14] >> 1;
        bitval[12] = bitval[13] >> 1;
        bitval[11] = bitval[12] >> 1;
        bitval[10] = bitval[11] >> 1;
        bitval[9] = bitval[10] >> 1;
        bitval[8] = bitval[9] >> 1;
        bitval[7] = bitval[8] >> 1;
        bitval[6] = bitval[7] >> 1;
        bitval[5] = bitval[6] >> 1;
        bitval[4] = bitval[5] >> 1;
        bitval[3] = bitval[4] >> 1;
        bitval[2] = bitval[3] >> 1;
        bitval[1] = bitval[2] >> 1;
        bitval[0] = bitval[1] >> 1;
}

//select reasonable dividers for a frequency
//in order to avoid overflow, the frequency is scaled by 10
void setDividers (unsigned long f){
  int i, j;
  unsigned long f_dco;

  for (i = 2; i <= 127; i+= 2)
    for (j = 4; j <= 11; j++){
      //skip 8 and 10 as unused
      if (j == 8 || j == 10)
        continue;
      f_dco = (f/10) * i * j;
      if (480000000L < f_dco && f_dco < 560000000L){
        if (hs != j || n1 != i){
          hs = j; n1 = i;
        setBitvals();
```

```
        }
        //f_dco = fnew/10 * n1 * hs;
        return;
      }
    }
  }

void setRfreq (unsigned long fnew){
  int i, bit, ireg, byte;
  unsigned long rfreq;

  //reset all the registers
  for (i = 7; i <= 12; i++)
    dco_reg[i] = 0;

  //set up HS
  dco_reg[7] = (hs - 4) << 5;
  dco_reg[7] = dco_reg[7] | ((n1 - 1) >> 2);
  dco_reg[8] = ((n1-1) & 0x3) << 6;

  ireg = 8; //registers go from 8 to 12 (five of them)
  bit = 5; //the bits keep walking down
  byte = 0;
  rfreq = 0;
  for (i = 37; i >= 0; i--){
    //skip if the bitvalue is set to zero, it means, we have hit the bottom of the
bitval table
    if (bitval[i] == 0)
      break;

    if (fnew >= bitval[i]){
      fnew = fnew - bitval[i];
      byte = byte | (1 << bit);
    }
    //else{
    // putchar('0');
    //}

    bit--;
    if (bit < 0){
      bit = 7;
      //use OR instead of = as register[7] has N1 bits already set into it
      dco_reg[ireg] |= byte;
      byte = 0;
      ireg++;
    }
  }
}

void setDCO(unsigned long newfreq){

  //check that we are not wasting our time here
  if (dco_freq == newfreq)
    return;

  //if the jump is small enough, we don't have to fiddle with the dividers
  // This should not be an absolute value comparison, but a relative
```

```
  // ppm deviation from the current center frequency.
  if ((newfreq > f_center && newfreq - f_center < 50000L) ||
    (f_center > newfreq && f_center - newfreq < 50000L)){
    setRfreq(newfreq);
    dco_freq = newfreq;
    qwrite_si570();
    wasSmall = 1;
    return;
  }
  //else it is a big jump
  setDividers(newfreq);
  setRfreq(newfreq);
  f_center = dco_freq = newfreq;
  write_si570();
  wasSmall = 0;
}

void setup() {
  // Initialize the LCD.
  lcd.begin(16, 2);
  printBuff[0] = 0;
  lcd.setCursor(3, 0);
  lcd.print("Si570 Test");

  Wire.begin();

  // Disable internal pullups - You will need external 3.3v pullups.
  digitalWrite(SDA, 0);
  digitalWrite(SCL, 0);

  // Force Si570 to reset to initial freq.
  i2c_write(si570_i2c_address,135,0x01);
  delay(20);
  read_si570();

  delay(5000);

  // Display the raw register values.
  lcd.clear();
  sprintf(c, "Reg 7-9:%02x %02x %02x", dco_reg[7], dco_reg[8], dco_reg[9]);
  lcd.setCursor(0, 0);
  lcd.print(c);
  sprintf(c, "  10-12:%02x %02x %02x", dco_reg[10], dco_reg[11], dco_reg[12]);
  lcd.setCursor(0, 1);
  lcd.print(c);

  delay(15000); // Wait a good long time to admire the results!

  // Obtain the High Speed register value.
  hs_reg_value = dco_reg[7] >> 5;

  hs_reg_value += 4;
  // Obtain the N1 register value.
  n_reg_value = ((dco_reg[7] & 0x1F) << 2) + (dco_reg[8] >> 6);
  if (n_reg_value == 0) {
    n_reg_value = 1;
  } else if ((n_reg_value & 0x01) != 0) {
```

```
    n_reg_value += 1;
  }

  // Obtain the reference frequency value in hex.
  refFrequencyUint64 = (uint64_t)(dco_reg[8] & 0x3F);
  refFrequencyUint64 = (refFrequencyUint64 << 8) | (uint64_t) dco_reg[9];
  refFrequencyUint64 = (refFrequencyUint64 << 8) | (uint64_t) dco_reg[10];
  refFrequencyUint64 = (refFrequencyUint64 << 8) | (uint64_t) dco_reg[11];
  refFrequencyUint64 = (refFrequencyUint64 << 8) | (uint64_t) dco_reg[12];

  // Display the HS, N1, and RFREQ values.
  lcd.clear();
  sprintf(c, "HS: %d  N1: %d", hs_reg_value, n_reg_value);
  lcd.setCursor(0, 0);
  lcd.print(c);
  sprintf(c, "RFREQ:%lX%lX", (unsigned long) (refFrequencyUint64 >> 32), \
    (unsigned long) (refFrequencyUint64 & 0x00000000ffffffff));
  lcd.setCursor(0, 1);
  lcd.print(c);

  delay(15000); // Wait a good long time to admire the results!

  // Determine the reference frequency in floating point.
  // The implied decimal point for RFREQ is between bits 28 and 29.
  // Convert to double and divide by 2^28 to get the reference frequency in MHz.
  refFrequency = (double) refFrequencyUint64 / (double) 268435456.0;
  lcd.clear();
  // Third parameter in the sprintf is the integer part of the reference
frequency.
  // Forth parameter is the fractional part of the reference frequency multiplied
by 10^6.
  sprintf(c, "RFREQ: %lu.%lu", (unsigned long) refFrequency, \
    (unsigned long) ((refFrequency - floor(refFrequency)) * 1000000.0));
  lcd.setCursor(0, 0);
  lcd.print(c);

  // Determine the DCO current frequency, based on the startup frequency, HS, and
N1 values.
  dcoCurrentFrequency = (double) SI570_STARTUP_FREQUENCY * (double) hs_reg_value *
(double) n_reg_value;
  // Determine the more accurate crystal frequency.
  freqXtalDbl = dcoCurrentFrequency / refFrequency;
  freq_xtal = (unsigned long) freqXtalDbl;
  sprintf(c, "fXTAL: %lu", freq_xtal);
  lcd.setCursor(0, 1);
  lcd.print(c);

  // Set the new frequency.
  setDCO(SI570_NEW_FREQUENCY);
}

void loop(){
}
```

Appendix E – Arduino Sketch for 1 KHz Audio Tone Generator

```
/*
This sketch implements a 1 KHz tone generator, using the Minima Digi Board,
to assist in testing the Minima Main Board.  In order to use this sketch,
the Digi Board must be assembled and tested first.  The squarewave tone
is output on Arduino pin PD7, which is used by the Radiono as the BAND_SEL_0.
The loop process continuously reads the Analog Keyer analog-to-digital
converter.  If the value read is below 512, the CW Key line is set to 1
(turned on).  Otherwise the CW Key line is set to zero (turned off).
*/
#include <LiquidCrystal.h>
#include <stdlib.h>

/*
 The 16x2 LCD is connected as follows:
   LCD's PIN   Raduino's PIN  PURPOSE      ATMEGA328's PIN
   4           13             Reset LCD    19
   6           12             Enable       18
   11          10             D4           17
   12          11             D5           16
   13           9             D6           15
   14           8             D7           14
*/
LiquidCrystal lcd(13, 12, 11, 10, 9, 8); // LCD configuration.

/*
 Digital Controls
   Raduino's PIN  PURPOSE             Orientation  ATMEGA328's PIN
   D2             LSB/USB BFO select  Output        4
   D3             TX/RX (PTT)         Input         5
   D4             CW Key              Output        6
   D5             Band High           Output       11
   D6             Band Select 1       Output       12
   D7             Band Select 0       Output       13
   D17            Function Button     Input        26
*/
#define BFO_LSB (2) // Logic 0 - USB, logic 1 - LSB.
#define TX_RX (3) // Logic 0 - RX, logic 1 - TX.
#define CW_KEY (4) // Logic 0 - unkeyed, logic 1 - keyed.
#define BAND_HI (5) // Logic 0 - low band, logic 1 - high band.
#define BAND_SEL_1 (6) // Currently unused.
#define BAND_SEL_0 (7) // Currently unused.
#define FBUTTON (17) // Function Button in digital mode.

/*
 Analog Controls
   Raduino's PIN  PURPOSE            ATMEGA328's PIN
   A1             Analog Keyer       24
   A2             Analog Tuning      25
*/
#define ANALOG_KEYER (A1)
#define ANALOG_TUNING (A2)

#define ANALOG_TONE (7) // Square wave tone generator.
```

```
int analogKeyer = 0; // Analog keyer variable.

void setup() {
  lcd.begin(16, 2);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Minima Test Tool");
  lcd.setCursor(0,1);
  lcd.print("1 KHz Audio Tone");
  // Set up the digital pins.
  pinMode(BFO_LSB, OUTPUT);
  pinMode(TX_RX, INPUT);
  pinMode(CW_KEY, OUTPUT);
  pinMode(BAND_SEL_0, OUTPUT);
  pinMode(BAND_SEL_1, OUTPUT);
  pinMode(BAND_HI, OUTPUT);
  pinMode(FBUTTON, INPUT);

  // Set the side-tone off, put the transceiver to receive mode.
  digitalWrite(CW_KEY, 0);
  digitalWrite(TX_RX, 1); // Old way to enable the built-in pull-ups.
  digitalWrite(FBUTTON, 1); // Old way to enable the built-in pull-ups.

  tone(ANALOG_TONE, 1000);
}

void loop() {
  // Read the Analog Keyer.
  analogKeyer = analogRead(ANALOG_KEYER);
  if (analogKeyer < 512) {
    digitalWrite(CW_KEY, 1);
  } else {
    digitalWrite(CW_KEY, 0);
  }
}
```