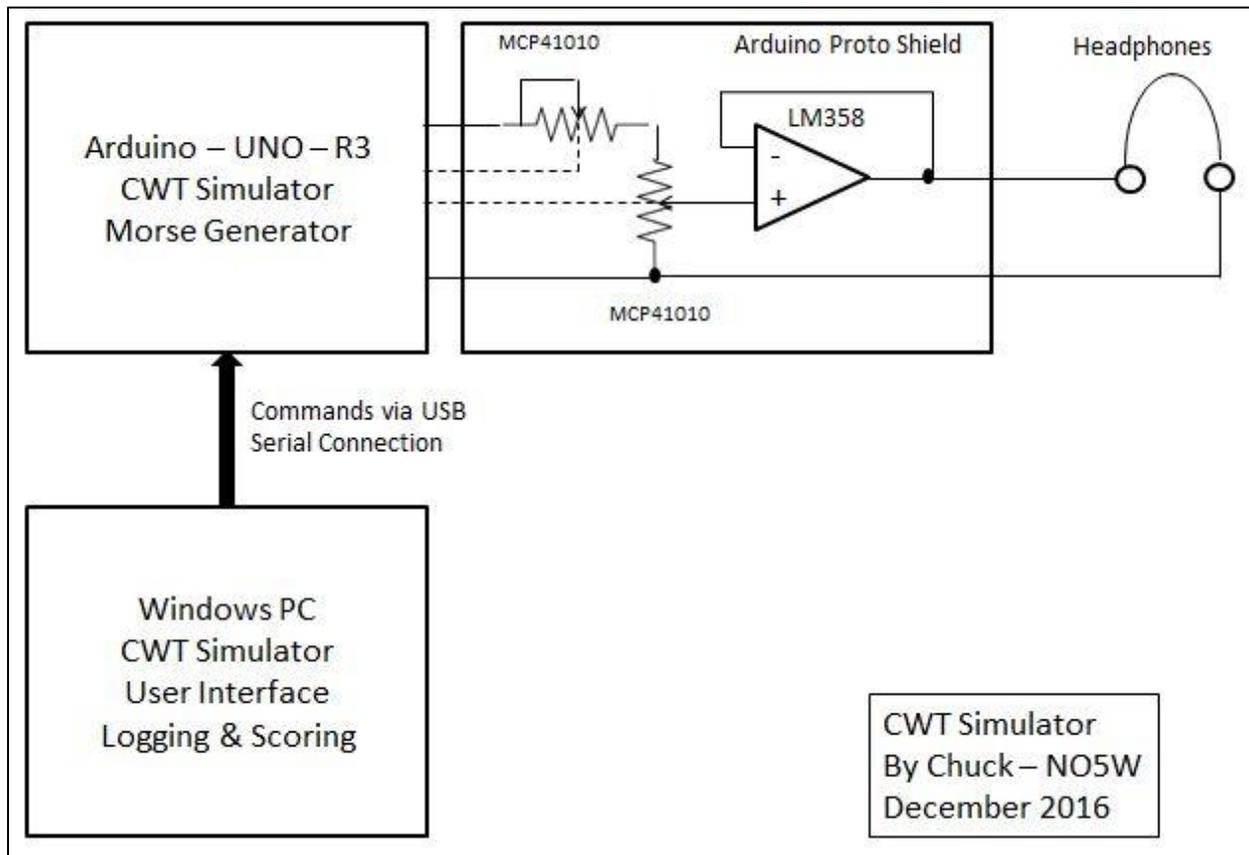


User Notes for the NO5W CWT Simulator

Overview of the Simulator – As shown in the following diagram the Simulator consists of three main parts: an application running on the PC (or Mac) which provides the logging and bookkeeping interface to the user; an application running on the Arduino that responds to commands from the PC application to generate various Morse signals; and a signal conditioning subsystem consisting of two digital potentiometers and a buffer amplifier that provides the final audio to the user headphones. The need for the digital pots came about only after it was realized that controlling the volume of tones generated by the Arduino is not a simple matter – tones are either full signal strength or no signal strength. There are software methods for controlling the volume using PWM but after some experimentation it was determined that their use would complicate the design of the Morse part of the Arduino code. It became apparent that use of digital pots which can be controlled by the Arduino’s Serial Peripheral Interface (SPI) was the way to keep the Arduino code simple.

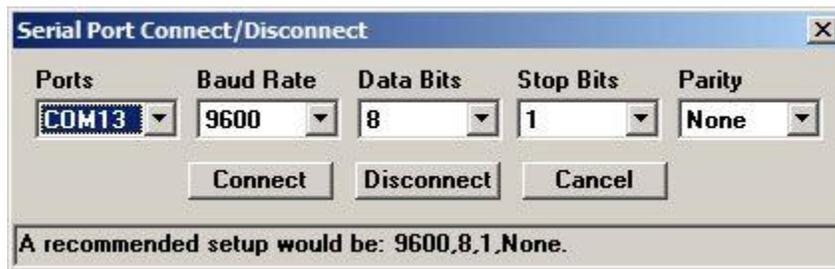
The Arduino is essentially a microcontroller (ATmega 328) with a few peripherals including a USB interface, a Serial Peripheral Interface (SPI) and a relatively large amount of digital I/O and analog inputs. An integrated development environment is freely available for download and is used to develop code using C/C++ for compiling and uploading the necessary hex code to the device over a USB interface. It has a limited amount of on board program memory (32K) and dynamic data memory (2K) and runs at a 16MHz clock rate which is completely adequate for the generation of Morse.



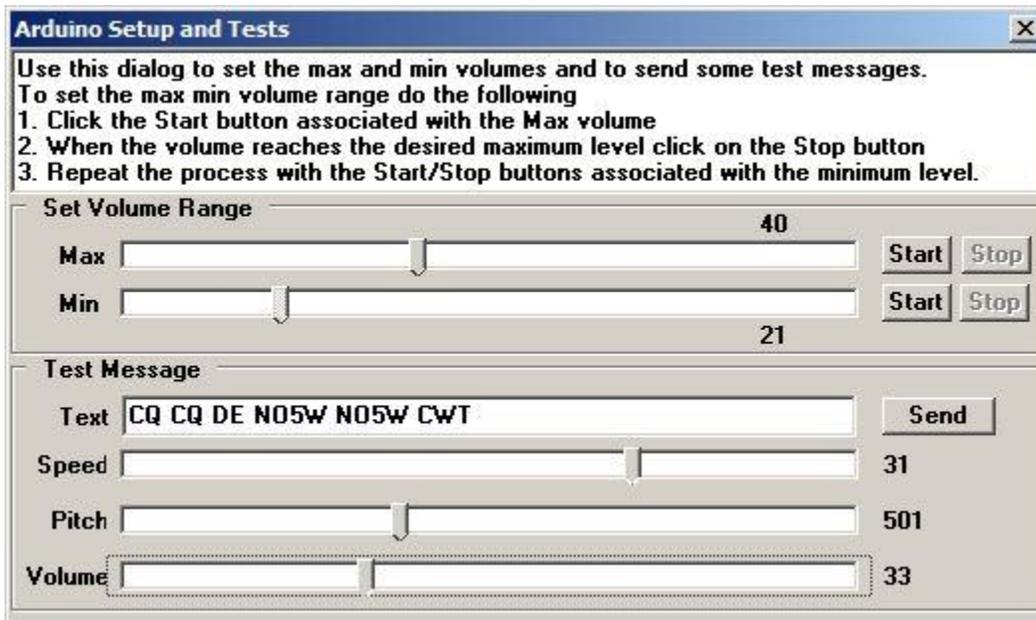
User Notes for the NO5W CWT Simulator

Using the Simulator – To operate the Simulator the user performs a configuration step which involves connection to the Arduino, creation of a pool of potential callers, definition of information about the user's operation (call, name, number, etc.), and definition of the type of session the user wishes to run. All of these steps are performed from the user interface on the PC as described below.

Connection to the Arduino – To begin a session the user must connect to the Arduino using the following dialog. It is of course important to make sure that the com port and related parameters are consistent between the PC and the Arduino. Your com port may vary and you may need to use Device Manager to find which of your ports is for the Arduino connection but the other parameters shown below should not be changed since those are what the Arduino code is expecting.



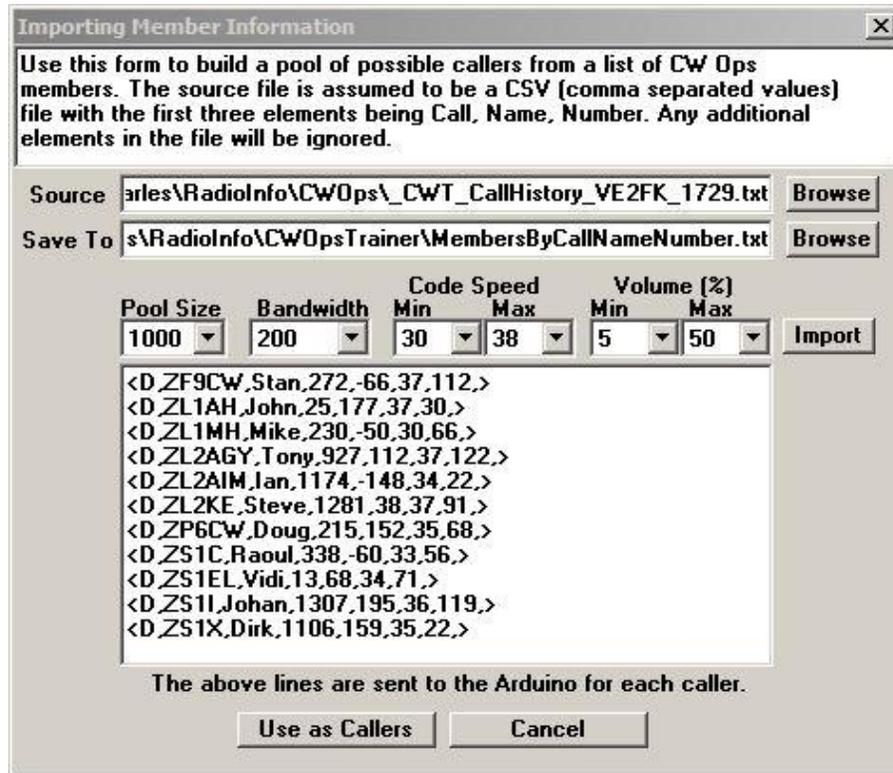
To check the connection to the Arduino and to obtain an idea of the sound of the Simulator the following dialog, accessible from the Setup menu, is used.



Once satisfied that the Arduino will produce Morse with a desired range of volumes it is time to build a pool of callers.

Building a Pool of Potential Callers - The pool is built from a call history file similar to the ones that Claude-VE2FK maintains and updates frequently. Here is the user interface for building the pool.

User Notes for the NO5W CWT Simulator



The user chooses a source file for the members which in the example above is a call history produced by VE2FK that contains members with numbers through 1729. The only requirement is that the file must be a CSV text file with the first three entries being the call, the name, and the number of the member. So if you have a favorite call history file that meets those requirements it will work also. The file shown will be adequate for a lot of play with the Simulator but as new members are added the user may want to download and use new files. Next the user specifies the name of the file where the pool is to be saved. The user then chooses a pool size (1000 in the example), a bandwidth of caller frequencies (50 – 200 Hz), a range of code speeds (10-40 wpm) and a range of signal strengths or volumes in the range 0-100. Once the parameters are set and the import button is clicked the program will build the pool by randomly selecting members from the source file and assigning them random bandwidths (frequency deviations about the side tone frequency of the user), code speeds, and signal strengths within the specified ranges. The resulting CSV lines shown in the above dialog are the commands that will be sent to the Arduino each time a new caller is needed. The right-most element in the above lines is the signal strength converted to a range of 0-256 for use by the digital potentiometers.

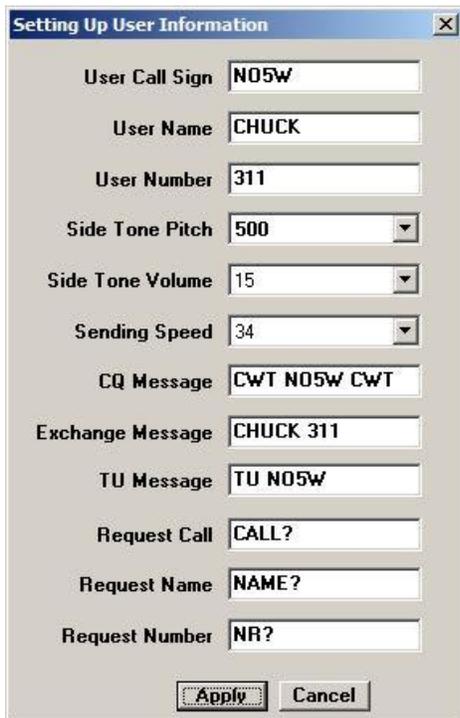
Setup User and Session Information - It is also necessary for the user to define some information regarding his operation as well as certain parameters defining how the session will be run. This is done using the following dialogs which are accessible from the Setup menu. This information is mostly self-explanatory and will generally remain fixed.

User Notes for the NO5W CWT Simulator

A session is one running of the Simulator with parameters that are fixed during the session. It is possible to run the Simulator for a fixed duration session of from 1 minute to 60 minutes or to run it open-ended with no definite ending time.

- If the **Allow Completion of QSO** box is checked then any QSO that is started prior to the end of the session will be allowed to complete (as long as it is completed within 30 seconds of the ending of the session).
- If the **Give Bonus for Pre-Filled Names** box is checked a bonus will be given for any name that is provided by the user prior to the responding station beginning his exchange. This is a “nod” to the original objective of this project. If you can remember the other operators name within the 5-6 seconds that elapses prior to the other station sending his name then you get a bonus.
- If the **Randomize the Callers** box is checked then the callers will be chosen at random from the pool. For personal use where it is desired to avoid memorization of the callers this box should be checked. On the other hand if you are using the Simulator to run a competition you might want to uncheck this box so that each participant will be receiving calls from the same stations.

Once the parameters have been set, clicking on the Apply button will complete the configuration of the Session and this configuration will apply and will be loaded each time the user starts the program until the user changes it.



The 'Setting Up User Information' dialog box contains the following fields and controls:

- User Call Sign:
- User Name:
- User Number:
- Side Tone Pitch:
- Side Tone Volume:
- Sending Speed:
- CQ Message:
- Exchange Message:
- TU Message:
- Request Call:
- Request Name:
- Request Number:

Buttons:



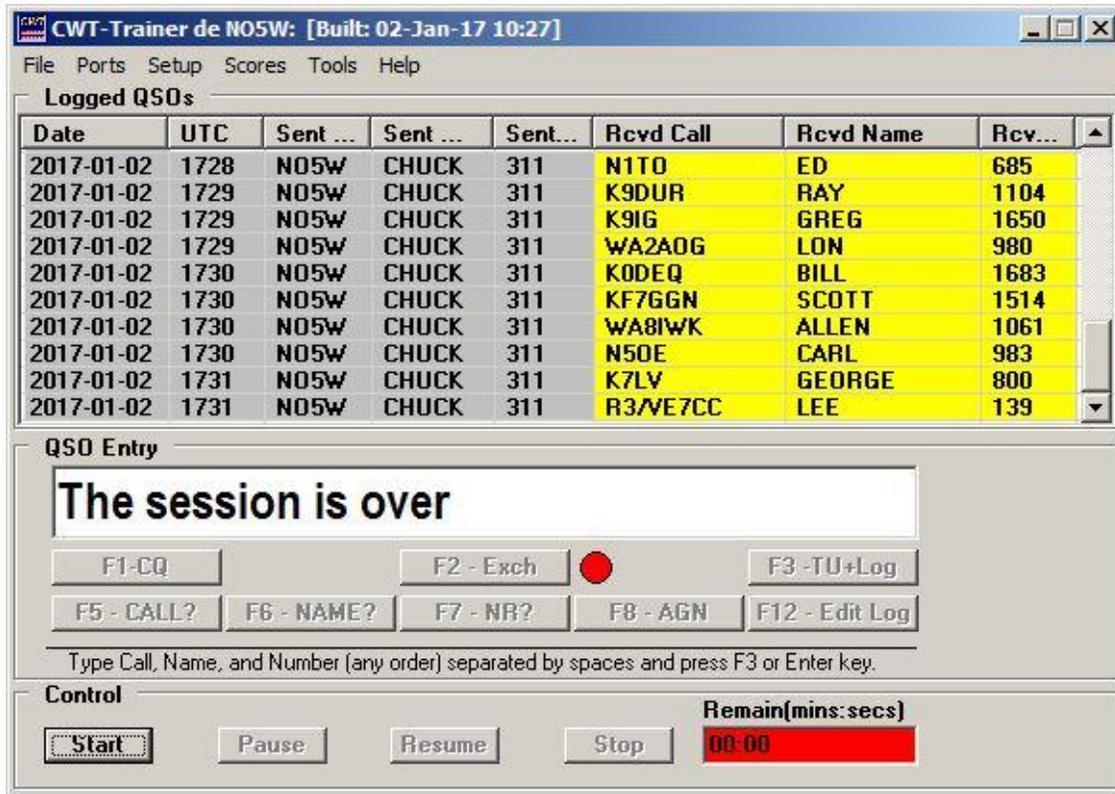
The 'Configuring the Session' dialog box contains the following controls:

- Session Duration (mins):
- Allow Completion of QSO:
- Give Bonus for Pre-Filled Names:
- Randomize the Callers:

Buttons:

Operating a Session – A session is run from the QSO entry window shown below which shows a session that has just ended.

User Notes for the NO5W CWT Simulator



To start the next session, press the F1 key or the Start button which will begin the session with a QRL. If the user has not yet saved the log from the just completed session a prompt asking whether to save the log is issued. Once the log is saved or the save is declined a press of F1 or the Enter key will result in a CQ message being issued. When a station responds type the call into the QSO Entry window and either press the F2 key or the Enter key to send the exchange, or press F8 to request another call by the responding station. If the call has been copied correctly the responding station will just send his exchange but if there is an error in his call he will repeat his call twice before sending his exchange. Upon copying his exchange type it into the QSO Entry window in any order and press F3 or the Enter key. If it is necessary to ask for a repeat of his call, name, or number or to simply send AGN use the indicated keys F5-F8. As each QSO is completed it will appear in the Logged QSOs window. If it is necessary to edit a previously logged QSO use F12 to access the logging area with the last QSO logged. If the QSO to be revised is further up the log use the cursor keys to move up to the QSO that needs to be revised and press the Enter key. This will take you to a dialog for making the desired change.

Reviewing Your Log – Once a session is over you can use the Scores menu item to pull up the following dialog containing an analysis of how you did. The top portion of this dialog shows your log along with an indication of any errors you made and any bonus you received.

User Notes for the NO5W CWT Simulator

Viewing Your Current Log

Show Complete Log Show Only Log Errors

Session of 10 minute duration started at: 2017-01-02 1721

DATE	UTC	CALL	NAME	NUMBER	CALL	NAME	NUMBER	PTS	PITCH	SPEED
2017-01-02	1721	UU5JZ	STAN	206	: OK	OK	OK	3	659	35
2017-01-02	1722	K7MDA/4	KEY	1594	: OK	OK	OK	3	664	37
2017-01-02	1722	N9TK	JIM	1309	: OK	OK	OK	3	314	32
2017-01-02	1722	W3WW	DON	896	: OK	OK	OK	3	471	31
2017-01-02	1723	K8ZZ	ED	1706	: OK	OK	OK	3	306	37
2017-01-02	1723	KZ5D	ART	12	: OK	OK	OK	3	308	36
2017-01-02	1723	AD4EB	JIM	867	: OK	OK	OK	3	402	30
2017-01-02	1724	VE2FK/M	CLAUDE	1181	: OK	OK	OK	3	550	35
2017-01-02	1724	W6RKC	RICK	576	: OK	OK	OK	3	475	30
2017-01-02	1724	K5GQ	MARK	672	: OK	OK	OK	3	430	30
2017-01-02	1724	K4RO	KIRK	387	: OK	OK	OK	3	528	35
2017-01-02	1725	WA2BCK	TOM	762	: OK	OK	OK	3	516	35
2017-01-02	1725	PA/W1NN	HAL	785	: OK	OK	OK	3	372	36
2017-01-02	1725	K5AB	ALAN	199	: OK	OK	299*	2	590	37
2017-01-02	1726	RA3CQ	ICOR	20	: OK	OK	OK	3	568	31
2017-01-02	1726	AE7I	CRAIG	875	: OK	OK	OK	3	567	37
2017-01-02	1726	K1CX	BILL	1038	: OK	OK	OK	3	659	31
2017-01-02	1726	W5XU	DAVID	539	: OK	OK	OK	3	584	36
2017-01-02	1727	K0HX	STAN	1721	: OK	OK	OK	3	658	36
2017-01-02	1727	K1RX	MARK	1122	: OK	OK	OK	3	522	30
2017-01-02	1727	WA2ADG	LON	980	: OK	OK	OK	3	497	30

Note 1: Your logging entries are on the left followed on the right by review comments.
 Note 2: In the case of logging errors the correct items are shown instead of OK
 Note 3: The last three entries on the right are points, and pitch and speed of the responder
 Note 4: Bonus! indicates that the name was entered before other station started his exchange

A summary of the scoring results may be obtained by scrolling down past the end of the log area where information like the following is presented. Although not shown in the screen shot below each request for a fill is recorded and is summarized for the user to review.

Viewing Your Current Log

Show Complete Log Show Only Log Errors

Scoring summary:

Available QSOs: 34
 Correct QSOs: 33
 Percent Correct QSOs: 97.00

Total QSO Points: 101
 Bonus Names: 9
 Total Points (incl Bonus): 128
 Note: Operator gets 3 bonus points per bonus name.
 A bonus name is one that is entered before the other station begins his exchange.

Correct Calls: 34
 Correct Names: 34
 Correct Numbers: 33
 Total Correct Elements: 101
 Percent Correct Elements: 99.00

Top Speed on Correct QSOs: 37
 Avg Speed on Correct QSOs: 33

Highest Pitch on Correct QSOs: 697
 Lowest Pitch on Correct QSOs: 306

Note 1: Your logging entries are on the left followed on the right by review comments.
 Note 2: In the case of logging errors the correct items are shown instead of OK
 Note 3: The last three entries on the right are points, and pitch and speed of the responder
 Note 4: Bonus! indicates that the name was entered before other station started his exchange

Saved logs from previous sessions can also be reviewed using an item from the Scores menu.

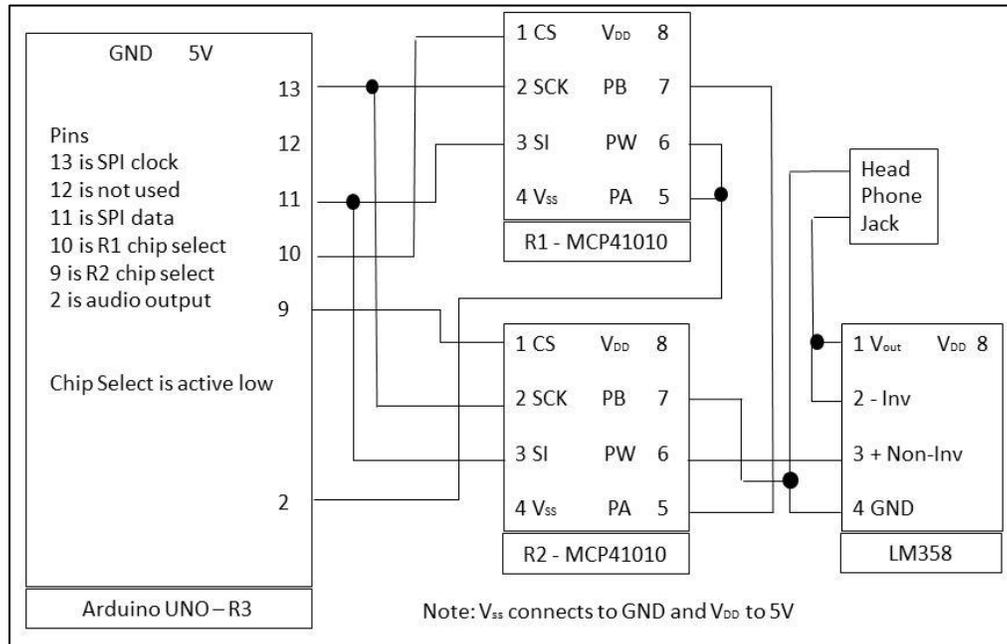
User Notes for the NO5W CWT Simulator

Constructing the Simulator – Building your own version of the simulator requires only a small amount of construction, primarily in the wiring of the digital pots. You will need the following software and hardware.

- **PC Application Code** – The executable for the PC that provides the user interface as well as the command interface to the Arduino is available for download from the CWT Simulator page of the [NO5W website](#). This code was developed for the Windows operating systems (XP, 7, 10) but tests have confirmed that it will run fine on a Mac under the Parallels desktop software.
- **Arduino Code** – The download package obtained from the author's website also contains source code for the application that runs the Arduino Morse generator. To use this code you will need to download the Arduino IDE which is available for free from [this site](#). Using this IDE and the source code, compiling and uploading the resulting hex code to the Arduino is straightforward and is handled completely by the Arduino IDE over the USB connection.
- **Arduino Hardware** – There are several versions of the pre-assembled Arduino. The one used in constructing the Simulator is Rev 3 of the Arduino UNO (UNO-R3). Since this is open-source hardware it is available from a number of manufacturers with prices ranging between \$5.00 and \$25.00. The version used in the development of the prototype was purchased at a ham fest from [MCM Electronics](#) and cost about \$22.00. I have since also found it available from [Adafruit](#), [Sparkfun Electronics](#), and several other vendors. During development a version of the Simulator was also built using the Arduino Mega 2560 which is a more capable device with more memory and more I/O but of course is also more expensive (~\$50.00). For the CWT Simulator in its current form it would be overkill. It may also be possible to use other Arduino devices lower in the food chain than the UNO including for example the Pro-Mini but in order to do that you would need to modify the pin definitions in the Arduino source code as well as changing the pins used in connecting to the digital pots on the proto-shield.
- **Miscellaneous Parts** – Enclosures are available with cutouts pre-made for the USB and power connectors on the Arduino. One such enclosure is available from Sparkfun as part PRT-10088. You will also need a USB –AB cable for connecting between the PC and the Arduino. A headphone jack will also be needed. I added a 10K manual pot between the output of the op-amp and the headphone jack for convenience as a final master volume control. The USB-AB cable and the headphone jack are available from many vendors and may already be in your junk box. Depending on how you decide to install the three 8-pin ICs on the proto-shield you might need sockets.
- **Proto-shield Construction** – Daughter boards for the Arduino are called shields and provide a way of adding functionality to that of the Arduino and a reliable way to make connections to the Arduino board and easy removal if you find you have other applications for the Arduino. The additional functionality needed in the Simulator consists of two MCP 41010 SPI- controlled 256-level 10K digital potentiometers and a single LM358 operational amplifier. These parts as well as the proto-shield (Sparkfun DEV-07914) were obtained from [Mouser Electronics](#). The digital pots and the op-amp were in the \$1.50 price range in single quantities and the proto-shield was about \$10.00. The digital pots are also available in a dual 14-pin package as MCP 42010. Some

User Notes for the NO5W CWT Simulator

soldering is required in assembling the proto-shield and in connecting the parts. Since I did not have a ready source of sockets I chose to use a breadboard during development and then to solder the ICs directly to a proto-board (Radio Shack 276-150) in the final assembly. An option to the Radio Shack board would be a Quarter-size breadboard PCB (Product 589) available from [Adafruit](http://adafruit.com). The source code for the Arduino assumes certain pins are used for the audio output and SPI interface lines to the digital pots on the proto-shield. Here is the wiring diagram for the output section.



The audio tone output from the Arduino is obtained from pin 2. During key-down it has a 5-volt amplitude and the eventual volume/signal strength produced for the user is controlled by digital pots R1 and R2 using the SPI interface provided by the Arduino. The SPI interface uses four pins on the Arduino: the clock from pin 13 connects to both of the digital pot clock inputs (SCK) to synchronously load data from pin 11 which connects to both of the digital pot data inputs (SI). The digital pot receiving a given data setting is controlled by the active-low chip select inputs from Arduino pins 9 and 10. R1 is the digital pot wired in rheostat mode and is used to set the maximum volume. R1 is fixed during a session whereas R2 is wired as a voltage divider and is used to dynamically vary the “signal strength” of the callers. R2 provides output to the user via the op-amp which is simply a buffer amplifier to isolate R1-R2 from loading by any headphone(s).

If you decide to build the CWT Simulator and have questions, encounter problems, or have suggestions for additional features and/or areas of improvement feel free to contact the author using the CQ/X support email found on the [NO5W website](http://no5w.com).

73/Chuck/NO5W

User Notes for the NO5W CWT Simulator

References – The following were found useful in the development of the CWT Simulator and may be of interest to those wishing to dig deeper into certain areas.

1. MicroChip, Datasheet DS11195C, MCP41XXX/42XXX Single/Dual Digital Potentiometer with SPI Interface, 2003.
2. National Semi-conductor, Datasheet DS007787, Low Power Dual Operational Amplifiers, LM358, October 2005.