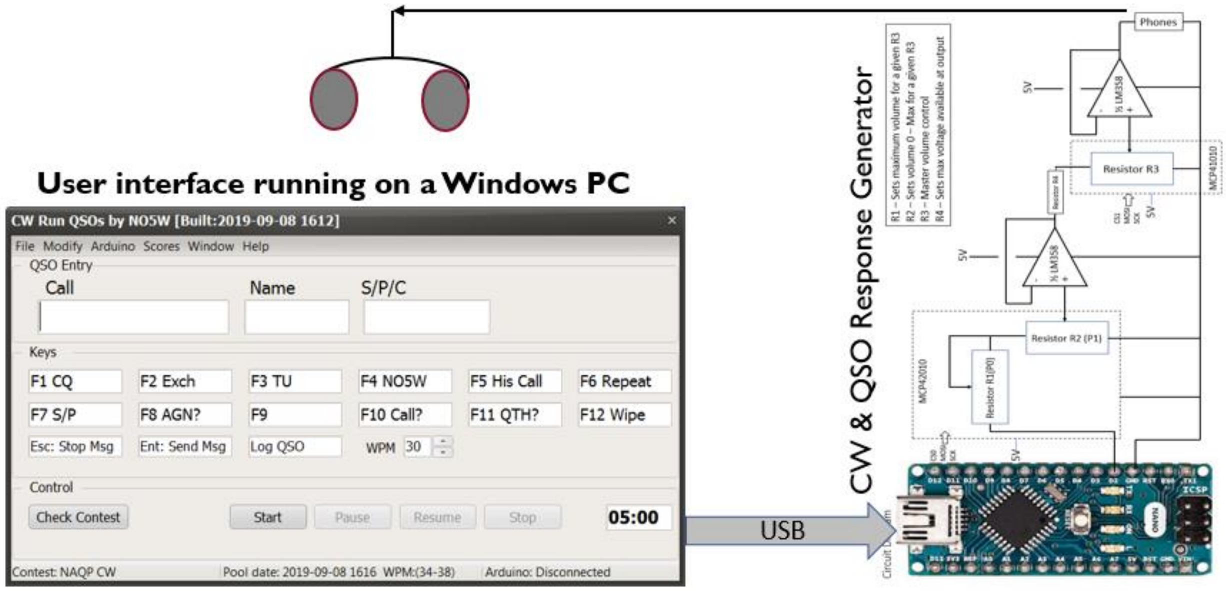
Notes for Building the Simulator

Overview of the Simulator – As shown in the following diagram the Simulator consists of three main parts:

- an application running on the Arduino Nano that generates Morse in response to commands from the PC application received over the USB connection;
- a signal conditioning subsystem consisting of three digital potentiometers and a buffer amplifier that provides the final audio to the user headphones.
- an application running on the PC which provides the logging interface for the user;

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. In this application the SPI interface is used by the Nano to control the digital potentiometers to provide calling stations with variable signal strength under commands generated in the PC.



Constructing the Simulator – Building the simulator requires only a small amount of construction using software for the PC and parts and software for the Arduino that are available from a number of sources. The Arduino Nano and the signal conditioning subsystem are built on a PCB that is available from NO5W. Here is an overview of what you will need and the steps involved. Following this overview there is a section that provides more detailed steps. The parts list contained in the Appendix provides a detailed list of parts along with part numbers and sources. Here is a description of the software and hardware you will need.

- **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 (7, 8, 10). The executable and other files needed by the application will be installed when you run the setup exe contained in the download package.
- **Arduino Code** – The Arduino Nano available from NO5W comes with the necessary code pre-installed. The source code for the Arduino is also supplied in the download package in case it becomes necessary to reinstall the software or you wish to supply your own Nano and use the Arduino IDE to install the

software on the Nano. Following installation of the PC application the Arduino code, which consists of a single file (RunQso-Simulator.ino), may be found in the Arduino subdirectory of the PC application. If it becomes necessary to use this code you will need to setup an Arduino development capability. This is easy to do since the Arduino IDE 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 Nano and Associated Hardware** – The subsystem for generating Morse consists of the following:
  - Printed Circuit Board
  - Arduino Nano
  - Dual Digital 10K ohm Potentiometer (MCP-42010)
  - Single Digital 10K ohm Potentiometer (MCP-41010)
  - Dual Op-amp (LM-358)
  - 470k ohm ¼ watt resistor
  - Headers for the Nano
  - Sockets for the Potentiometer and Op-amp Chips
  - Stereo Panel Connector for Headphone Output
  - Stereo Panel Connector for Paddle Input

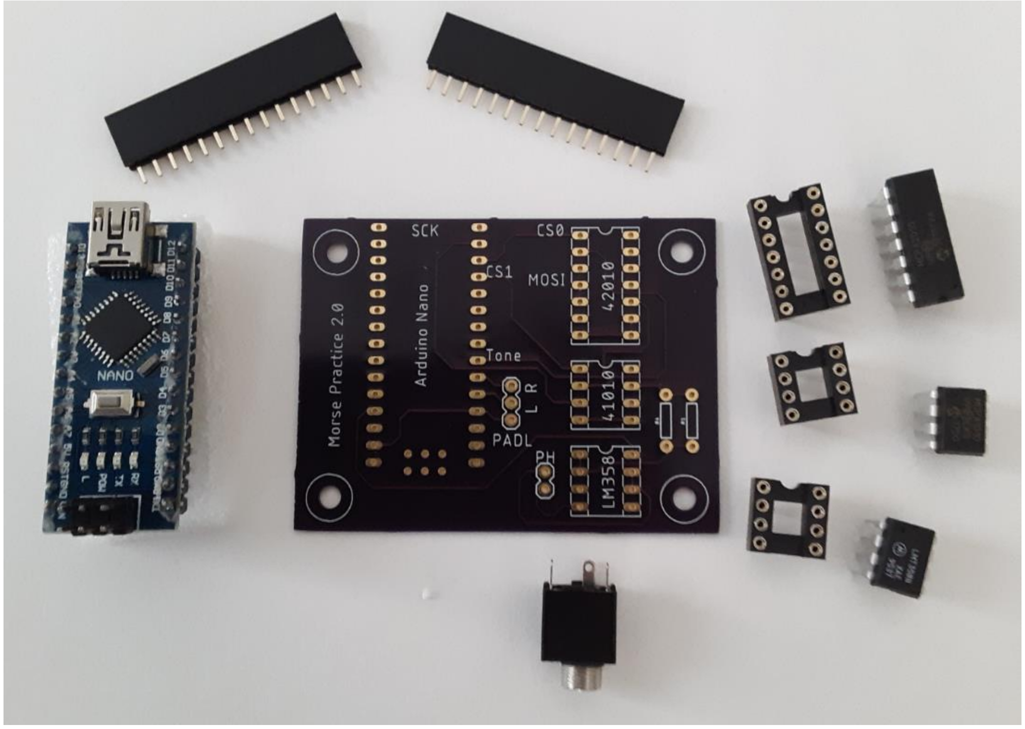
  Populating the PCB with these parts requires soldering about 60 points on the PCB and, with proper soldering equipment, can be easily done in a couple of hours or less. Since the Nano fits very well on a breadboard you could also elect to build the Arduino and associated hardware on a breadboard at first and, after using that version for a while, make a decision whether or not to move to a more permanent device using the PCB. Notes on this approach are presented in Appendix B.

- **Enclosure** – A custom 3D-printed enclosure has been designed and the resulting STL file for printing the enclosure is included in the install package in case it is desired to print your own or to submit the file to your local library or an on-line 3D printing house.

Detailed Construction Steps – The only construction involved is related to the PCB containing the Nano and associated parts. There are two basic options for acquiring the hardware including:

1. Provide your own Nano and associated parts for assembly on a breadboard
2. Provide your own Nano and associated parts and assemble them on a PCB from NO5W

Sources for these options are included in Appendix A. In the following it will be assumed that you have elected to assemble the hardware from a collection of parts (Option 2). The photo below shows the parts you will be using.



Populate the Printed Circuit Board

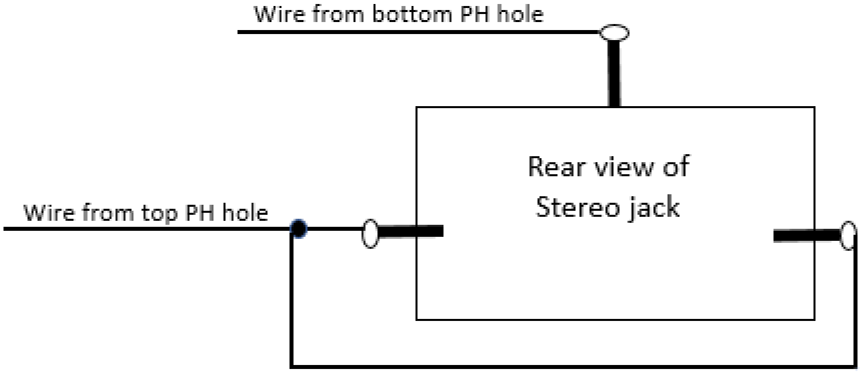
The sequence used to install the above components is not that critical but I have found that it is easiest to save the Nano headers and the headphone and key connectors for last. Some common tips for being careful in your construction are:

- When installing the sockets on the PCB be careful to observe the orientation shown on the board.
- In all of the soldering steps below be careful not to create solder bridges between pins.
- When inserting the three ICs in their sockets be careful to observe the correct orientation and make sure that all the pins are properly aligned before pressing the IC into the socket.
- You will probably need to bend the IC pins inward a bit by placing one side of the IC on a flat hard surface and slightly "rolling" the free side. Then check the alignment of the pins in the socket and, if necessary, repeat until all pins fit. Then press the IC into the socket and check that all pins are properly inserted.

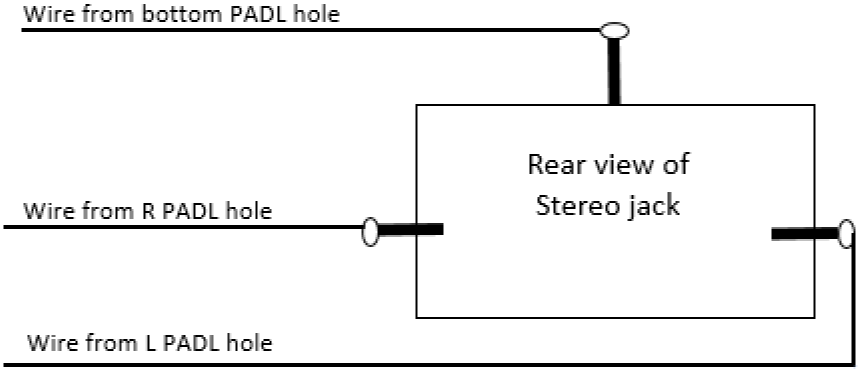
Here is the sequence of steps that has worked for me.

- **Step 1:** Solder the 8-pin socket for the LM-358 on the PCB
- **Step 2:** Solder the 8-pin socket for the MCP-41010 on the PCB
- **Step 3:** Solder the 14-pin socket for the MCP-42010 on the PCB
- **Step 4:** Solder the two 15-pin headers in the holes provided for the Nano. Be sure the headers are fully inserted and are lying flat on the PCB. To facilitate their proper alignment, it helps to attach the headers to the Nano (leaving a little play) and then insert the headers into the PCB holes and solder.

- **Step 5:** Solder the 470K ohm ¼ watt resistor between two of the pins to the right of the MCP-41010. The slot for the second resistor is not used.
- **Step 6:** Carefully insert the LM358 in its socket. You may need to slightly bend the pins inward as described above.
- **Step 7:** Repeat Step 6 for the MCP-41010.
- **Step 8:** Repeat Step 6 for the MCP-42010.
- **Step 9:** Insert the Nano firmly in the headers making sure it is fully seated.
- **Step 10:** Prepare a twisted pair of hookup wire about 3 inches long using two colors if available. Solder the end of one of the wires of the pair to the top hole labeled PH on the PCB and the other wire to the bottom hole. Solder the other ends of the twisted pair as shown in the diagram



- **Step 11:** Prepare a twisted trio of hookup wire about 3 inches long using three colors if available. Solder the end of one of the wires of the trio to the top (R) hole labeled PADL on the PCB and the other wire to the middle (L) PADL hole and the third wire to the remaining PADL hole. Solder the other ends of the twisted trio as shown in the diagram



Install the PC User Interface Software – Regardless of which of the above options you chose, the following two steps will be necessary. These steps will install the user interface on your PC.

1. Browse to the CWT-Simulator page of the NO5W website and download the installation zip file to a directory on your PC.
2. Extract the files from the package and run the setup file (Setup-RunQso-Simulator-200.exe) to install the PC application

Note: If you obtained your Nano from NO5W the software for the Nano should already be installed and you can skip steps 3-10. Otherwise you will need to do the following.

Setup and Test the Arduino Integrated Development Environment (IDE) –

3. Browse to the Arduino site and follow the instructions to download and install the Arduino IDE on your system. A good on-line resource of detailed instructions for this is at this site.
4. Use a USB cable to connect your Arduino Nano to your PC. You should see a red LED light up on the Nano and remain steady indicating that the Nano is receiving power via the USB connection.
5. Compile and upload the simple blink sketch that comes with the Arduino IDE to check that your Nano is working. When you do the compile be sure to select the Arduino Nano. If all goes well you should see a slowly blinking LED and you will know that the Nano is working and can be programmed via the IDE.

You will now need to install (reinstall) the Simulator software on the Nano by performing steps 6-10 below.

Compile and Upload the RunQso Simulator Code to the Arduino

6. The Arduino IDE will keep track of where your source code (sketches in Arduino terminology) is located if you follow its scheme of placing your sketch in a separate directory having the same name as the sketch and where the directory is a sub-directory of the Arduino documents directory which is normally located in the MyDocuments folder. Browse to your MyDocuments folder and you should find a sub-directory called Arduino. Make a subdirectory under the Arduino directory called RunQso-Simulator.
7. Move or copy the Arduino source code (RunQso-Simulator.ino) found in the Arduino sub-directory of the RunQso-Simulator that you installed in step 2 to the …MyDocuments\Arduino\ RunQso-Simulator\ directory.
8. Start the Arduino IDE and notice that if you use menu item File | Sketchbook one of the sub-menus of Sketchbook will be called RunQso-Simulator. Click on that menu item and the IDE will load the RunQso-Simulator source code. At the top of the source code file you will notice two #include statements which tell the compiler to go out, find the two named files and include them in the code:
    - NewTone which is an improved version of the tone library that is responsible for creating the tones used in the Morse generator
    - The SPI library is the one that allows the Arduino to talk to external devices (in our case the digital pots) using the SPI communication protocol.

    The two files are called libraries and contain code that was written by others saving the developer a lot of work. Of course it is very important that the compiler be able to locate and use the two libraries. Some libraries are included as part of the Arduino IDE installation performed in Step 3 above. You can find information on libraries including how to use them here . If you go there you will note that the SPI library is part of the standard but the NewTone library is not. So we need to make sure that the NewTone library is accessible to the IDE. There is capability within the IDE to manage libraries and this will enable the IDE to install contributed libraries that you find on the internet. The steps required to install a contributed library may be found here.

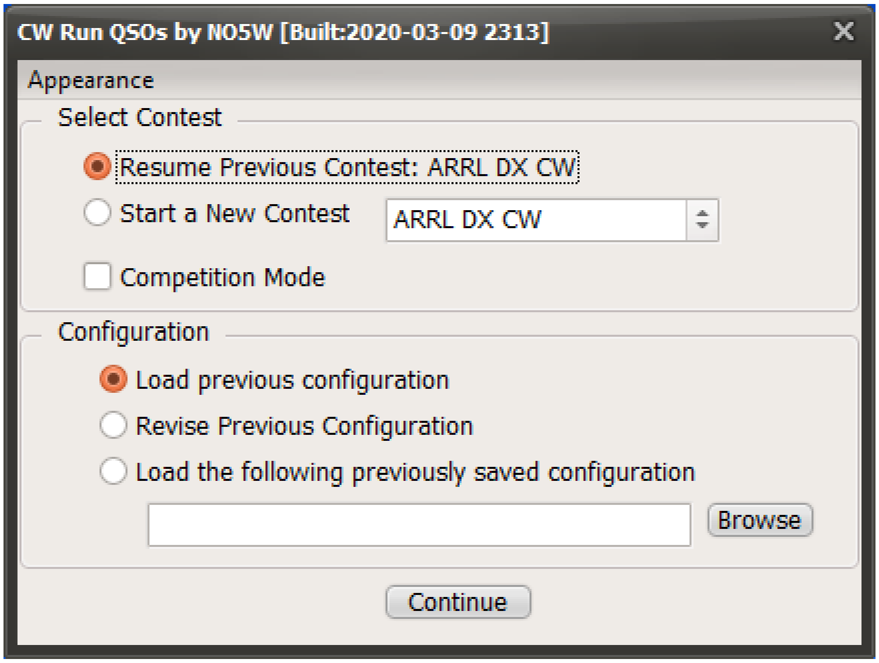
Use the menu item Tools | Board and verify that the selected board agrees with the board you are using (Arduino Nano). If it does not agree use that same menu to select the correct board.

9. Use menu item Sketch | Verify/Compile to verify that the sketch will compile.
10. Use menu item Sketch | Upload to upload the compiled code to the Arduino. If you encounter problems getting the sketch to load you might have a Nano with an older version of the boot loader. In that case use the Tools menu item of the IDE to select the item whose label contains "Old Bootloader" and select the submenu that also contains "Old Bootloader".

Check the Operation of the PC and Arduino

Now that you have the PC application installed and the Arduino Morse generator installed it is time to check out how they operate together. To do this perform the following steps using the PC application CWRunQso.

12. Make sure the Arduino IDE is off
13. Use device manager to determine the comm port the Nano is connected to.
14. Double-click on the CW Run Qso icon on the desktop which will open the Start Up form



CW Run QSOs by NO5W [Built:2020-03-09 2313]

Appearance

Select Contest

● Resume Previous Contest: ARRL DX CW

○ Start a New Contest   ARRL DX CW

□ Competition Mode

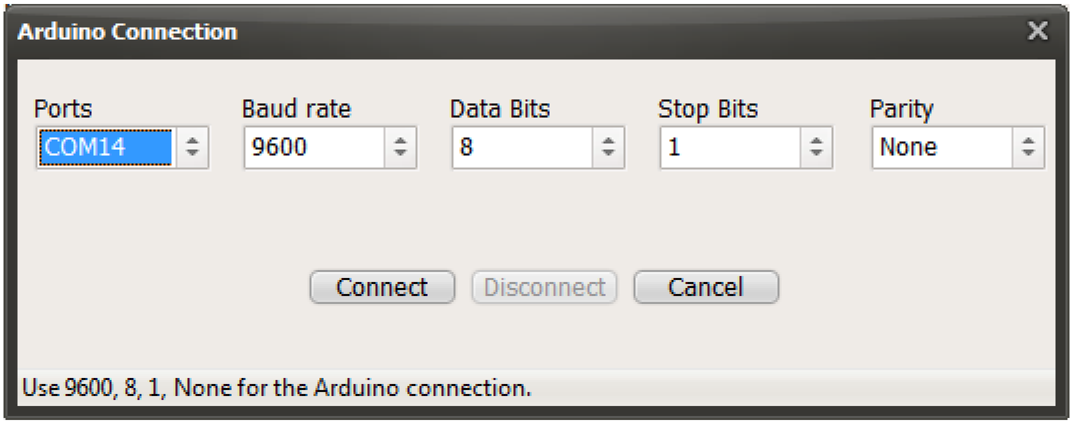
Configuration

● Load previous configuration

○ Revise Previous Configuration

○ Load the following previously saved configuration

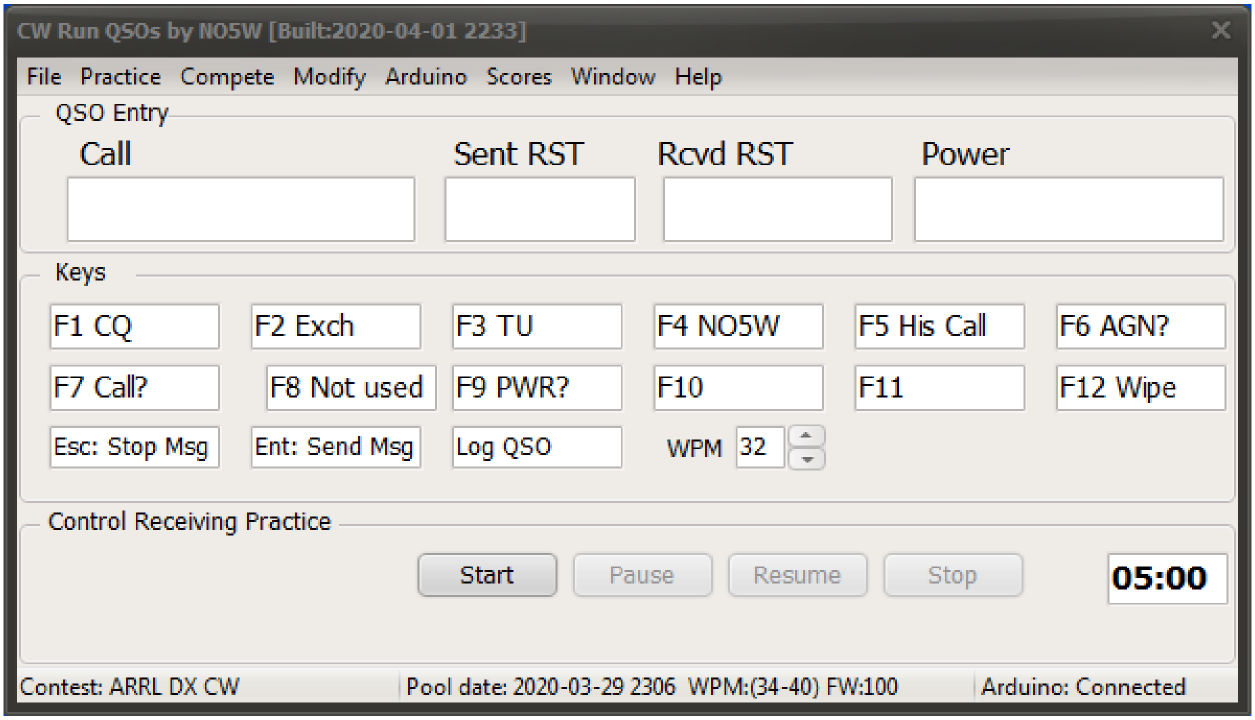
15. On the first use of the program you will need to personalize some settings by revising the configuration included in the install package. However, for the purpose of checking the operation of the PC and Arduino just choose to Resume Previous Contest and to Load Previous Configuration. Another document will describe in detail configuration and use of the application.
16. On clicking on the Continue button the main form for running the simulator and entering QSOs will be presented along with the following dialog for setting up the Arduino Connection. Use the Ports dropdown to select the port, leaving all the other items unchanged.



Ports	Baud rate	Data Bits	Stop Bits	Parity
COM14	9600	8	1	None

Use 9600, 8, 1, None for the Arduino connection.

Clicking on the Connect button will allow access to the main form.

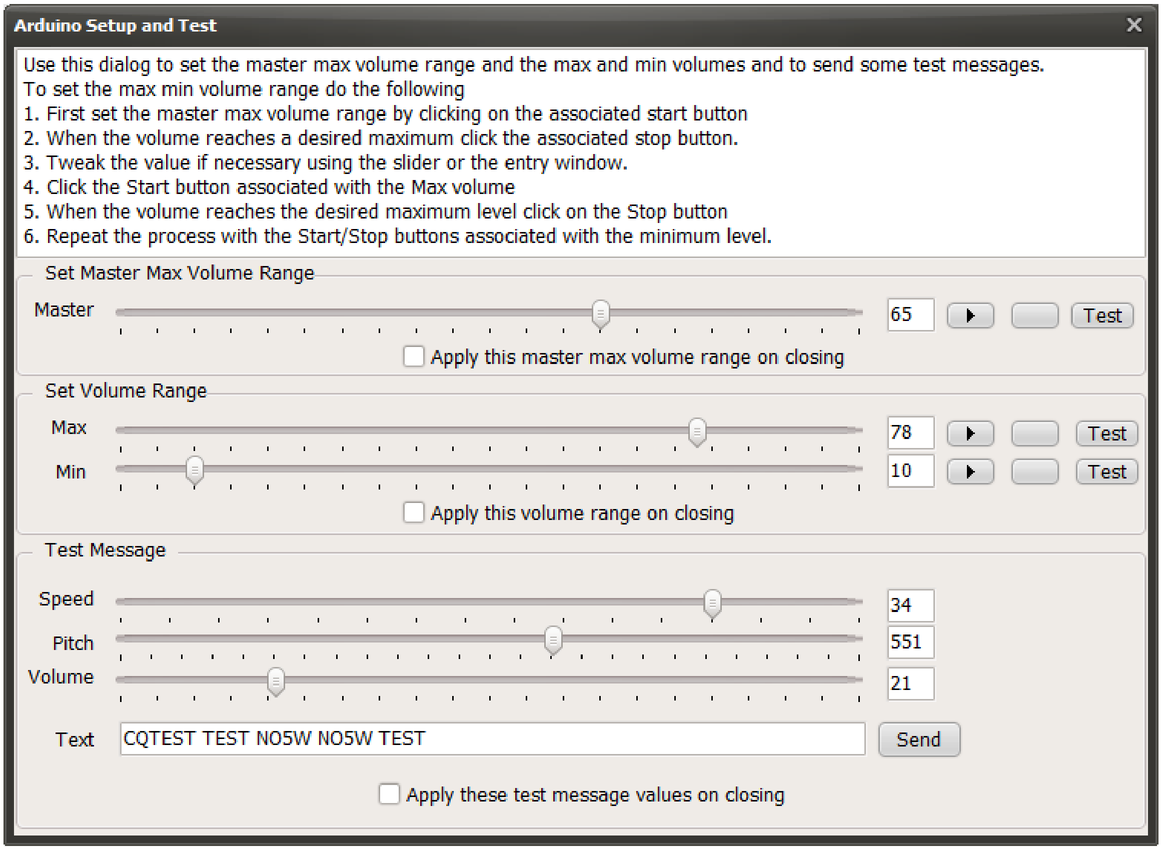


WPM 32

05:00

Contest: ARRL DX CW | Pool date: 2020-03-29 2306 WPM:(34-40) FW:100 | Arduino: Connected

Make sure your headphones are plugged in and use the menu item **Arduino | Setup and Test** to access the following dialog.



Use this dialog to set the master max volume range and the max and min volumes and to send some test messages.
To set the max min volume range do the following
1. First set the master max volume range by clicking on the associated start button
2. When the volume reaches a desired maximum click the associated stop button.
3. Tweak the value if necessary using the slider or the entry window.
4. Click the Start button associated with the Max volume
5. When the volume reaches the desired maximum level click on the Stop button
6. Repeat the process with the Start/Stop buttons associated with the minimum level.

Set Master Max Volume Range

Master 65

☐ Apply this master max volume range on closing

Set Volume Range

Max 78

Min 10

☐ Apply this volume range on closing

Test Message

Speed 34

Pitch 551

Volume 21

Text CQTEST TEST NO5W NO5W TEST

☐ Apply these test message values on closing

Use this dialog and the process at the top of the form to set the Master Max Volume Level, the Volume Min-Max Range of responding stations as a percentage of the Master Max Level and to generate a Test Message. Start with the values already loaded in the program and make adjustments and experiment as desired, using the Test and Send buttons to check how the settings sound.

That completes the initial checkout. The next step is to personalize the application and begin actually using it. For this you should access the User Guide found under the Help menu.

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.

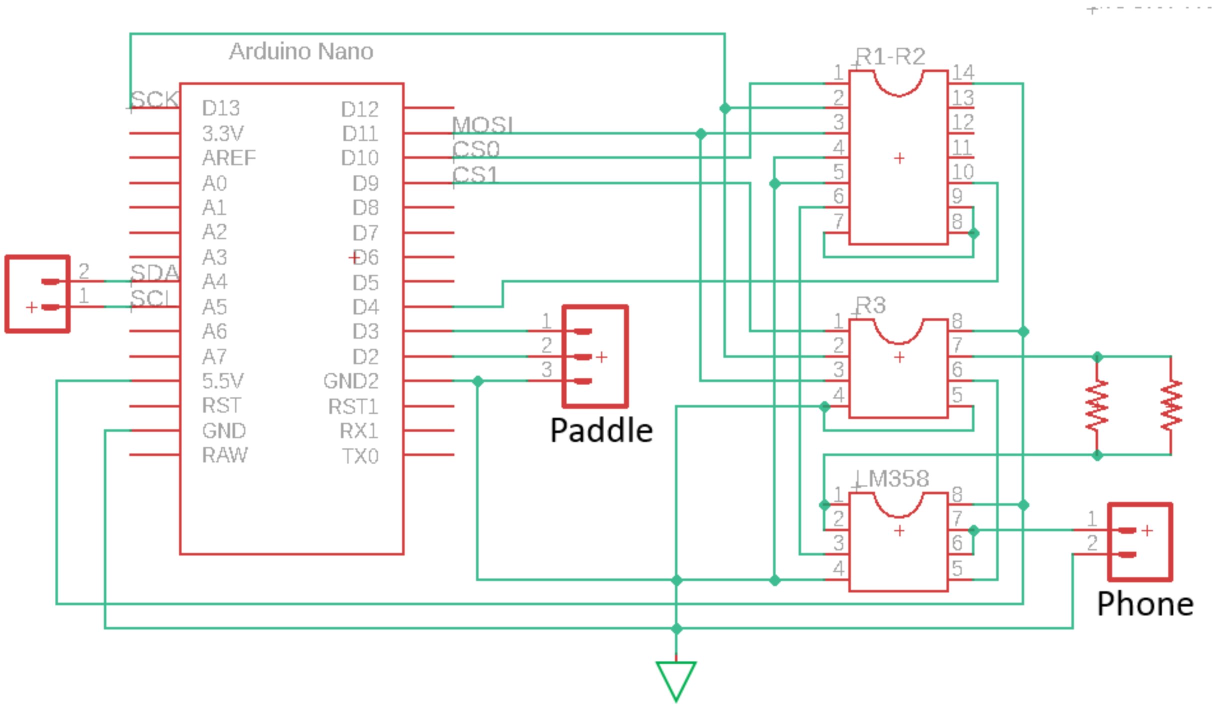
1. MicroChip, Datasheet DS11195C, MCP41XXX/42XXX Single/Dual Digital Potentiometer with SPI Interface, 2003.
2. National Semi-conductor, Datasheet DS007787, Low Power Dual Operational Amplifiers, LM358, October 2005.

Appendix A - Parts Sources: Here is a detailed list of software and hardware you will need along with comments on the source for each. The software can be obtained via download from the CWT-Simulator page of my website. You can obtain all parts except the PCB from the sources listed below.

Part	Source
PC User Interface Exécutable Code	Contained in Download from NO5W*
Arduino Morse Generator Source Code	Contained in Download from NO5W*
Arduino IDE Code	Obtain from https://www.arduino.cc/en/Main/Software
Arduino Drivers	Contained in the Arduino IDE and also in NO5W download
Arduino Nano	There are many sources including the Arduino store. Many clones are available via Amazon, New Egg, Adafruit and others. Adafruit PRODUCT ID: 50
MCP 41010 Digital Resistors (1)	Mouser Electronics part 579-MCP41010-I/P
MCP 42010 Dual Digital Resistors (1)	Mouser Electronics part 579-MCP41010-I/P
470K ohm ¼ watt resistor	Jameco is a good source
LM358 Dual Operational Amplifier	Mouser Electronics part 512-LM358AN
Audio Connector 1/8 inch (2)	Jameco part 2095437
8-pin dip socket (2)	Phoenix Enterprises HWS9027
14 pin dip socket	Phoenix Enterprises HWS9031
15 pin headers (2)	Phoenix Enterprises HWS1879
Printed Circuit Board	Available from NO5W
Project Enclosure	Can be 3D printed using the STL file in download. Some libraries offer this service to members
Miscellaneous hookup wire (22 gauge)	Provided by the user
USB A to mini b Cable	Provided by the user

NO5W CWT Simulator page* https://www.no5w.com/CWT-Simulator.php

Appendix B – Constructing the Simulator on a Breadboard: Here is a overview description of information needed to build the simulator on a breadboard. Here is a wiring diagram to follow.



Note that the connections to the Nano SDA and SCL pins (A4 and A5) are for future use and are not used in this version of the simulator. Also note that typically only one of the resistors shown in the diagram is used and that value is 470K ohms. If it is found that more volume is needed you can try placing a second 470K ohm resistor in parallel as shown.