**Date:** 2/4/2022

**To:** Katherine Haines

**From:** Jonathan Ceis (KJ7ELA)

**Subject:** Final Project Proposal - Ham Radio Morse Code Transmitter

**Project Overview:**

Morse code was created in 1830 by Samuel F.B. Morse. It has long been used by amateur radio operators, the military, and other entities to transmit messages since its creation. It can be more easily understood when a radio operator is experiencing signal interference and degradation than when sending a voice transmission. While advancements in radio technology have immerged since it’s creation, morse code has not lost its place as an acceptable and advantageous method of communication.

As my final project for ENGR121 I will build a ham radio morse code transmitter. This circuit will be capable of capturing user input via a computer keyboard, interpreting the user input text, encoding that input as morse code, and sending that data to a Ham Radio. As a stretch goal, the circuit will automatically acuate the radio’s push-to-talk (transmit button) functionality. The circuit will not be capable of receiving signals from the ham radio, nor will it be capable of controlling any other functionality that the radio possesses such as frequency selection.

The completion of this project will allow me to make contact with other ham radio operators at longer distances that I can using voice transmissions.

**Project Architecture:**

The radio being used is a Baofeng UV-5R, and the microprocessor to be used is a Sparkfun RedBoard. The user input will be collected from a keyboard attached to a computer. The computer will be running a custom coded script via the Arduino IDE compiler. The compiler in turn will establish a serial connection to the Sparkfun Redboard for the purpose of transmitting the transcoded morse code message input by the user.

**Project Limitations:**

This project will not address controlling the radios functions, receiving signals, nor decoding signals received by the radio. The Baofeng UV-5R is capable of being programmed via a data connection to an Arduino. This connection however requires two-way communication with the radio using established communication protocols which are outside the current scope of this project. The circuit and Arduino program will not receive or process any signals received by the Baofeng UV-5R. The processing of morse code transmissions would require the implementation of several sophisticated algorithms to account for radio signal interference, morse code tone variation, and variation in the timing and speed of morse code transmission. Algorithms such as these are outside the scope of this class and resulting project.

**Materials List:**

* Baofeng UV-5R Ham Radio
* Sparkfun Redboard
* Circuit Breadboard
* Jumper Wires
* 3.5mm TRS Plug
* 2.5mm TRS Plug
* ALTERNATIVE: As an alternative to two TRS plugs one 3.5mm/2.5mm TRS plug may be used
* Resistors
* Relay(s)

**Circuit Abstract:**

The Baofeng UV-5R ham radio will be connected to the Sparkfun Redboard via a breadboard and two TRS (Tip-Ring-Sleeve) connectors or a single connector possessing two TRS tips. The 3.5mm TRS connector is traditionally used for a microphone and the 2.5mm TRS connector is traditionally used for a speaker. The morse code signal will be injected into the radio via the 3.5mm TRS connector.

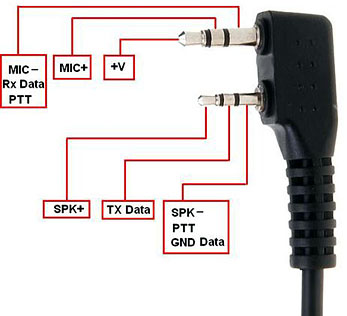


Figure - https://ham.stackexchange.com/questions/7398/how-to-make-a-ptt-button-for-baofeng-uv-5r

The circuit may require resistors, capacitors, diodes, potentiometers, and other components to establish the correct voltage, amperage, isolate separate ground sources, and filter interference as required.

To achieve the stretch goal of automatically actuating the Push-to-Talk functionality of the radio required to transmit a relay and two separate ground sources may be required. The push-to-talk functionality is actuated by grounding the sleeve of the 3.5mm TRS connector to the sleeve of the 2.5mm TRS connector. This can be achieved using a relay and the introduction of resistors as needed.

**Program Abstract:**

The program to accept user keyboard input, transcode that input to morse code, and instruct the Redboard to generate the proper signals will be written using the Arduino IDE. The program will utilize the “Serial” function to establish a connection with the RedBoard. The Arduino IDE Serial Monitor will be used to accept keyboard input from the user. Once the user has entered input and pressed the enter key the program will automatically transcode the keyboard input into morse code. This will be accomplished by parsing the user input into an array and iterating over that array. This iteration will match the keyboard input to the appropriate morse code signal. Once the appropriate morse code pattern has been established the program will send the signal to a pinout on the Redboard.

The program will require 5 variables to store data for timing. These variables will determine the length of the following: Morse Code “dots”, Morse Code “Dashes”, the delay between “dots” and “dashes”, the delay between letters in a word, and the delay between words in a sentence. An additional two variables will be required to store the frequency tone which must be generated and the pin to output that tone. The morse code alphabet will be stored using at least one array, and the user input will be stored in an array as well.

**Testing:**

Only I, Jonathan Ceis, will be operating the experimental circuit. I currently hold a General Class Ham Radio license under the call sign KJ7ELA. All experiments and testing will be conducted according to standard courtesy, convention, regulation, and law. Transmissions will be conducted according to the [Code of Federal Regulations Title 47, Chapter I, Subchapter D, Part 97](https://www.ecfr.gov/current/title-47/chapter-I/subchapter-D/part-97). One-way transmissions are specifically allowed via 97.111(b) parts 1, 2, 5.

Transmissions will be conducted within the appropriate spectrums within the 2 meter or 70cm ranges. Channels will be monitored for use prior to attempting any transmission to avoid disruption to other radio frequency users.

Testing will be conducted by transmitting on the Baofeng UV-5V and monitoring the resulting signal on a second ham radio tuned to the same frequency.

**Possible Roadblocks:**

There are several possible roadblocks I may encounter while building this circuit:

Every precaution will be made to avoid damaging the RedBoard or Baofeng radio during testing and operation. However, due to the complexity of ensuring that grounds are appropriately isolated and “buffered” it is possible that at some point the circuit may exceed it’s allowed voltage or amperage. This may result in damage to the microprocessor or radio. In the event that this occurs I plan on purchasing a new radio and/or board. I plan on building the circuit as far in advance as possible so if this were to happen I would be able to have replacement components delivered with enough time to finish the project.

One of the functions of the Baofeng radio is that when a microphone and speaker are connected the built-in microphone and speaker are disabled. It may be required during testing that I include the circuitry required to output sound through a speaker in order to monitor incoming responses. If this proves to not be possible then I will utilize a secondary ham radio to monitor for responses.

The final and most significant roadblock relates to grounding and interference. In this circuit there will be three grounds which must be accounted for. These consist of the RedBoard ground, the microphone ground coming from the radio, and the speaker ground coming from the radio. It is possible that in the process of sharing a ground between these three circuits that I could damage a component or introduce interference to an extent that makes the signal generated by the circuit unrecognizable. In the event I am not able to resolve an issue such as this. I will pivot to designing a circuit which will output a tone through the piezo buzzer only. This will allow me to manually key the radio and play the sound into the radio’s built-in microphone.

**Outcome:**

If I can successfully complete this circuit, I will be able to communicate with other ham radio operators using morse code transcribed by my computer and the RedBoard microprocessor. This will allow me to contact operators located further away than possible with voice and/or at times of unusually high radio interference.

**References:**

* <https://ham.stackexchange.com/questions/7398/how-to-make-a-ptt-button-for-baofeng-uv-5r>
* <http://www.miklor.com/COM/UV_Technical.php>
* <https://www.ecfr.gov/current/title-47/chapter-I/subchapter-D/part-97>