All Things Digital

Amateur Radio for the 21st Century

026

Robert C. Mazur, VA3ROM
E: va3rom@gmail.com
W: http://www.va3rom.com

First published in the September-October 2016 issue of The Canadian Amateur

BUILD AN ARDUINO ANALOG SDR CONTROL GADGET

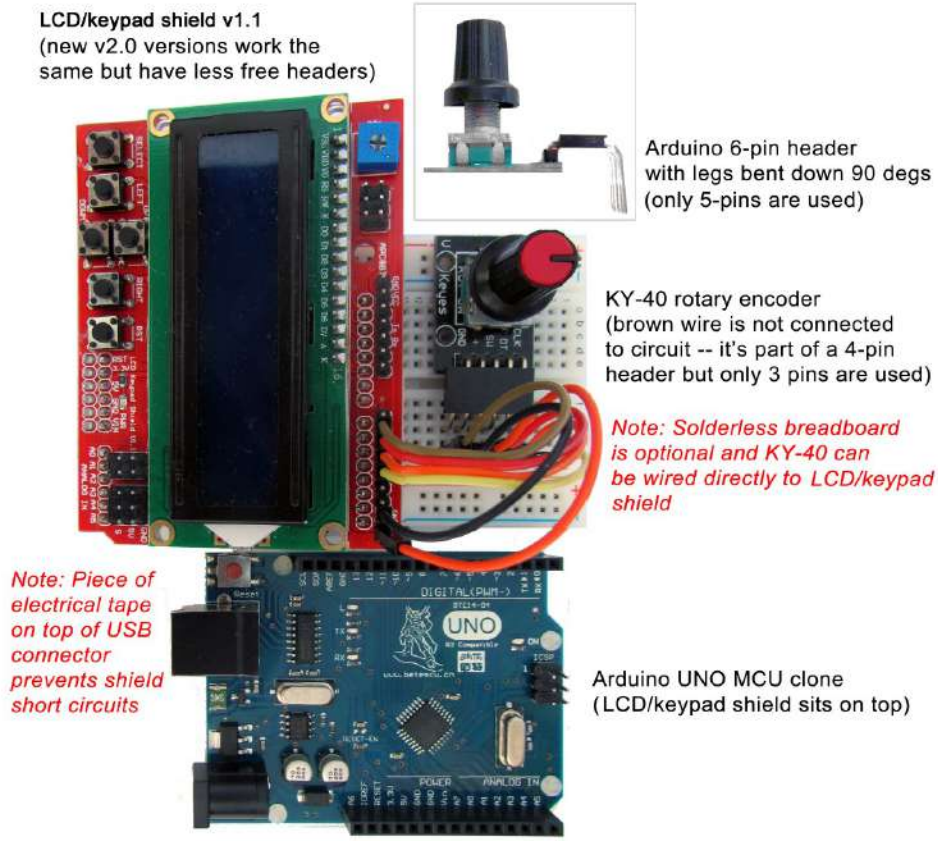
INTRODUCTION

Our radios are becoming more miniaturized along with virtual digital controls on a computerized display controlled by keyboard, mouse, and/or joystick. As a result, large analog switches and knobs, etc. are disappearing from 21st century "black box" software defined radios (SDRs). It may be "heresy" to think about adding external analog controls to these digital black boxes, but it's human nature to have and hold them—even if they aren't really required—at least from the radio's point of view.

SDR# NET REMOTE GADGET

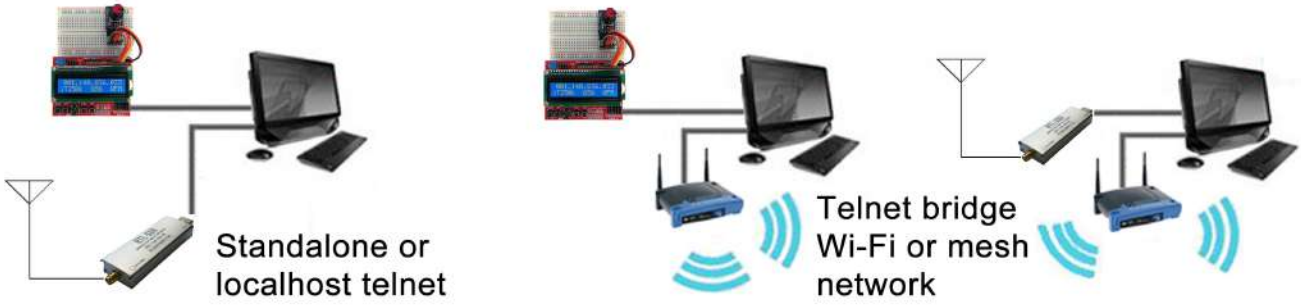
I dislike using computer controls to "drive" my Flex-1500 and various SDR# (SDRSharp) controlled USB RTL-SDR dongles. The idea for this Arduino gadget came from my Flex-1500 SDR's FlexControl analog control USB interface (see above) I had purchased to make a more ergonomic human/radio control interface.

It's an awesome analog add-on device so I wondered if there was some way to implement a similar type of control for SDR#. It didn't take too long to search for and find a free programming add-on module (plug-in) for SDR# called "Net Remote", which is written by Al Brown. Net Remote provides an external serial port "hook" to any Arduino MCU using a free add-on library for the Arduino Integrated Development Environment (IDE). It also provides local or wide area network (LAN or WAN) wired and/or Wi-Fi remote control using "telnet", which is a 1960's command-line interface whereby you keyboard instructions to control a remote device. Al's website shows how to wire a basic Arduino control circuit (he uses an Arduino Nano) and also provides a simple sample program. He distributes Net Remote with an installation package so you don't have to do manually configure SDR# yourself, as is the normal case when installing new plug-ins. A liquid crystal display (LCD) isn't used in his example, but I feel it's handy to have to see what's going on with SDR# when it's sent to the background when you bring up another Windows foreground program. And a display is especially essential for remote SDR# control where you can't physically see the remote computer using for telnet control. *Note: Microsoft dropped built-in telnet support after Windows 7 and it must be reinstalled manually.*

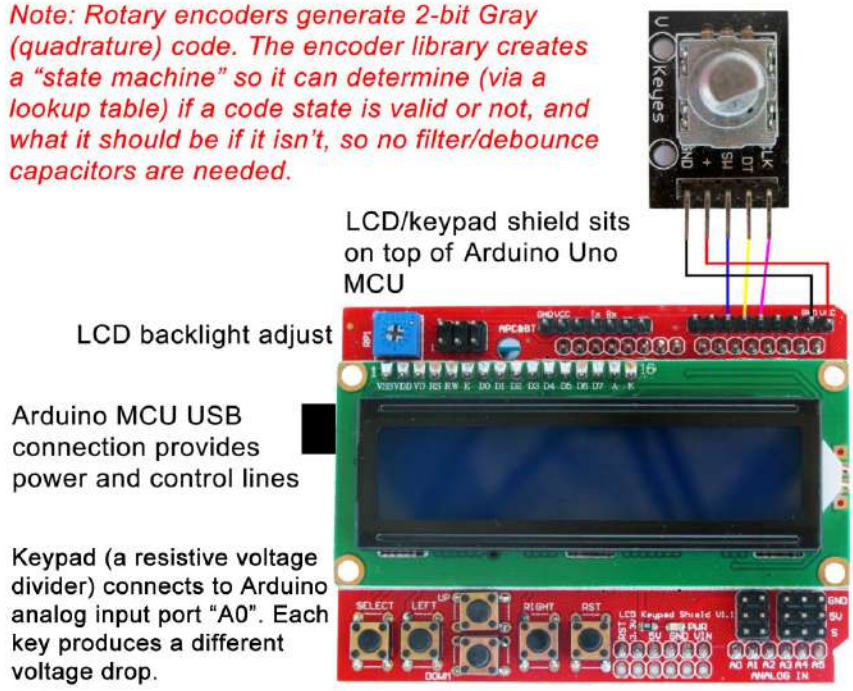


The initial idea was to create a "proof-of-concept" Arduino-based analog control prototype gadget (see left), connected to a host computer using a short USB serial cable.

Long cable runs means using repeater cables (or a CAT-5 Ethernet cable) so a radio-based network (Wi-Fi or Bluetooth/BT) is preferable the distance increases between the control gadget and the remote SDR# system (see below).



Net Remote uses JSON (JavaScript Object Notation) while the typical Arduino "speaks" a variation of C, so a lot of "translation" between the two had to be figured out because Al's website only had basic JSON programming information. But, after a few hours of trial and error (I had no prior experience with JSON) it all worked out very well, and I thought others would be interested in building and using this useful gadget, especially since SDR# is very popular with RTL-SDR users.



BUILDING THE GADGET

There's not much to building and wiring the gadget (see left), and it can be "naked" or put in a suitable housing.

The KY-40 rotary ("half-step") encoder I used is mounted on a small printed circuit board (PCB) with pull-up resistors, and have a very handy built-in shaft push button (normally open or "NO").

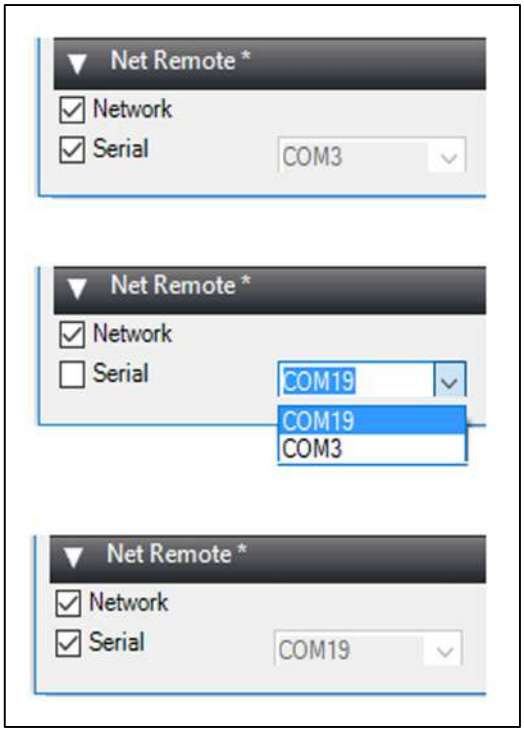
The free Arduino IDE encoder library can be programmed to use full or half-step encoders, and implements a virtual "state machine" ("debounce" capacitors aren't required). All features and functions can be performed entirely by the KY-40, so you can build the gadget with just an LCD (and appropriate driver module) if you prefer not to use the LCD plus keypad combo shield. I just happened to have several on hand and wanted to learn how to use them. The breadboard is optional, and you can use DuPont or similar jumpers to wire the encoder directly either to an LCD plus keypad shield or to the Arduino (as in Al's example). *Note: My prototype was developed using Net Remote v1.2.5947, SDR# v1.1430, and Arduino IDE v1.6.5. The required source code, Arduino IDE support libraries and other information are available on my website*

USING THE GADGET

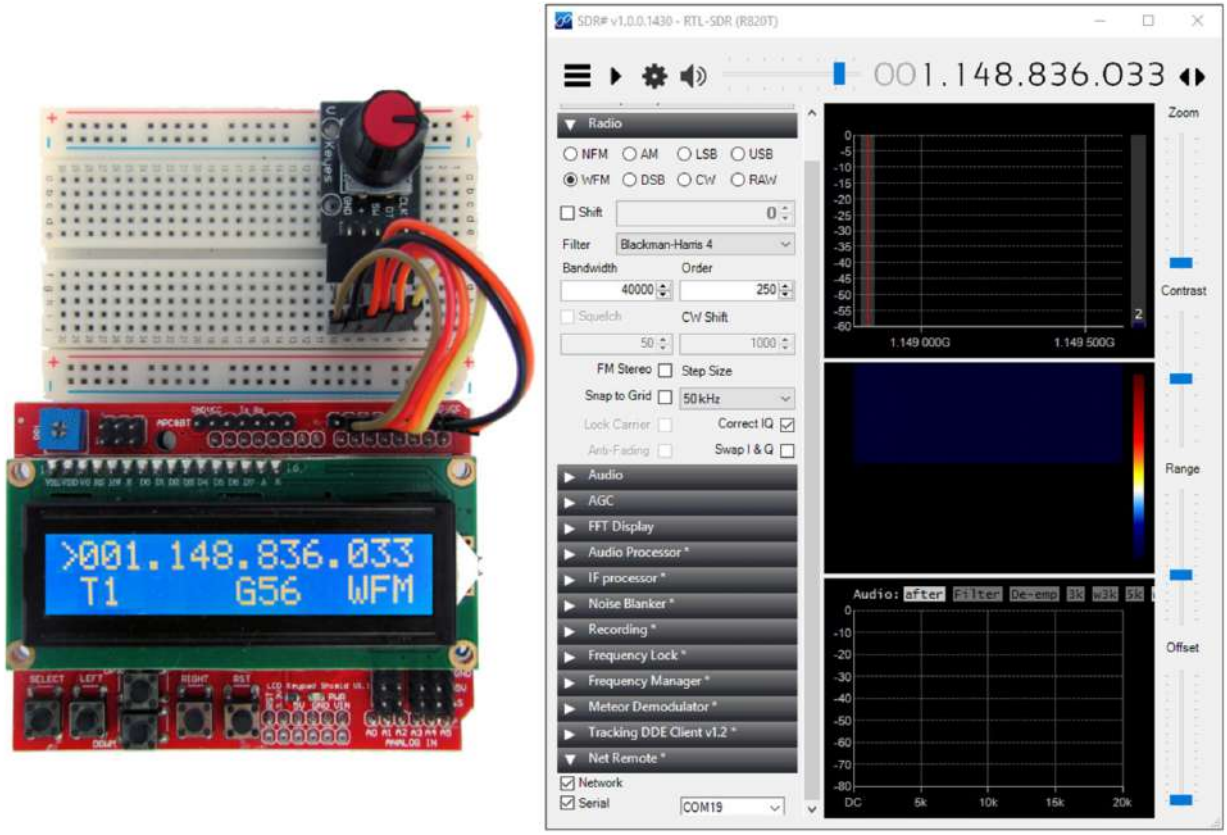
I won't cover the various aspects of installing SDR# plug-ins, Arduino IDE libraries, or programming MCUs, but the source code is fully commented, and if you have any programming experience, you shouldn't have problems using it, or expanding on and adding more functionality to the code. As programmed, a 2-line LCD displays SDR#'s tuned frequency (first line), while the second line shows the current tuning step in hertz (Hz), volume control gain in decibels (dB), and receiver mode. The rotary encoder (or keypad buttons) let you select what SDR# feature you want to change (indicated by a right angle ">" pointer) and also lets you turn the SDR# program on/off (play mode).

Here are the quick operating procedures for the local or host computer:

1. Connect USB RTL-SDR dongle and remote control gadget to host computer.
2. Load SDR#. *Note: First time use of Net Remote may trigger the Windows network permissions allow/deny request.*
3. For first time use or changing serial port connection, open the Net Remote plug-in settings and uncheck Serial box; set it to the gadget's serial (USB) port; check the Serial box (see right). Leave the Network checkbox checked.



4. Press gadget's reset (RST) button; this establishes/reestablishes connection between it and SDR#. The gadget's LCD will show the current SDR# settings (see below). *Note: Anytime you lose sync with SDR# or have some other problem, press the RST button to reestablish the network link.*



5. The default startup receive frequency is 24 megahertz (MHz) for R820T/T2 RTL-SDRs (maximum 1700 MHz) and this can be used to indicate a connection problem if the receive mode (lower right corner LCD display) is also blank.
6. Activate SDR# (play mode) by pressing and long holding (≈ 1/2 second) the rotary encoder's shaft button then release it. If you keep holding the shaft button down, SDR# cycles between play/off modes (a circular buffer with a delay is used for all functions/features).
7. A right angle ">" pointer indicates the SDR# field value you can change. Adjust this by turning the rotary encoder shaft left or right or pressing the keyboard up/down buttons. Pressing the rotary encoder shaft button down and releasing or pressing the keypad's Select button cycles the ">" pointer from frequency, tuning step, volume, and mode then back to frequency.

8. Programmed tuning (T) steps in Hz cycles as follows: 1, 10, 25, 50, 100, 250, 500, 1000 k (kilohertz), 2k5, 5k, 10k, 25k, 50k, 100k, 250k, 500k, 1M, 2M5, 5M, 10M then back to 1.
9. SDR# volume control gain (G) cycles as follows: 25 (mute), 26 to 60 in 1 dB increments then back to 25 dB (mute).
10. SDR # receiver mode cycles as follows: NFM, AM, LSB, USB, WFM, DSB, CW, and RAW then back to NFM.

Note: In mid-2015, SDR# and its support plug-ins were recompiled using Microsoft's dotNet 4.5+ programming framework. This decision means Windows XP won't work with later SDR# versions or plug-ins. SDR# v1.1361 the last XP to W10 compatible version, but v1.3333 is the most stable one to use, IMHO.

MY FINAL

Feel free to make any modifications or additions to the gadget source code. It's only a basic template showing you how to use most of the JSON plug-in programming features. One nifty addition would be adding an external keypad to allow for direct frequency entry, or a BT module for remote control via smartphone (this could also be used to replace the LCD). You will have to add the required Arduino/JSON commands but it's not that hard to do.–73

REFERENCES AND RESOURCES

Arduino libraries

http://tinyurl.com/mau8hqp (JSON)

http://tinyurl.com/qh38vr2 (rotary encoder)

FlexRadio FlexControl

http://tinyurl.com/jabqk7m

KY-40 Rotary Encoder

http://tinyurl.com/jhutsy7

Net Remote

http://tinyurl.com/h6vo7n7

Telnet (for Windows 8 and later)

http://tinyurl.com/zf36xpg

VA3ROM (All Things Digital)

http://tinyurl.com/og2acxq