RADIO MAGiC
Robert C. Mazur, VA3ROM | www.va3rom.com | va3rom@gmail.com

First published in the Jul-Aug 2020 issue of The Canadian Amateur

The OPEN-SMART Rich Shield

Introduction

OPEN-SMART is the brand name of a Chinese electronics company that produces a plethora of inexpensive µ-controller (Arduino) boards and shields (Arduino term) and various add-ons (for Arduino and other µ-controllers), all available from online retailers such as AliExpress.com, DX.com, etc. The OPEN-SMART "Rich Shield" is a shield that plugs in to and sits on top of the Arduino Uno or MEGA2560 µ-controllers and their clones, and they also make the OPEN-SMART "Nano-UNO Shield Adapter" that allows you to use the Arduino Nano µ-controller and its clones (see Figure 1).

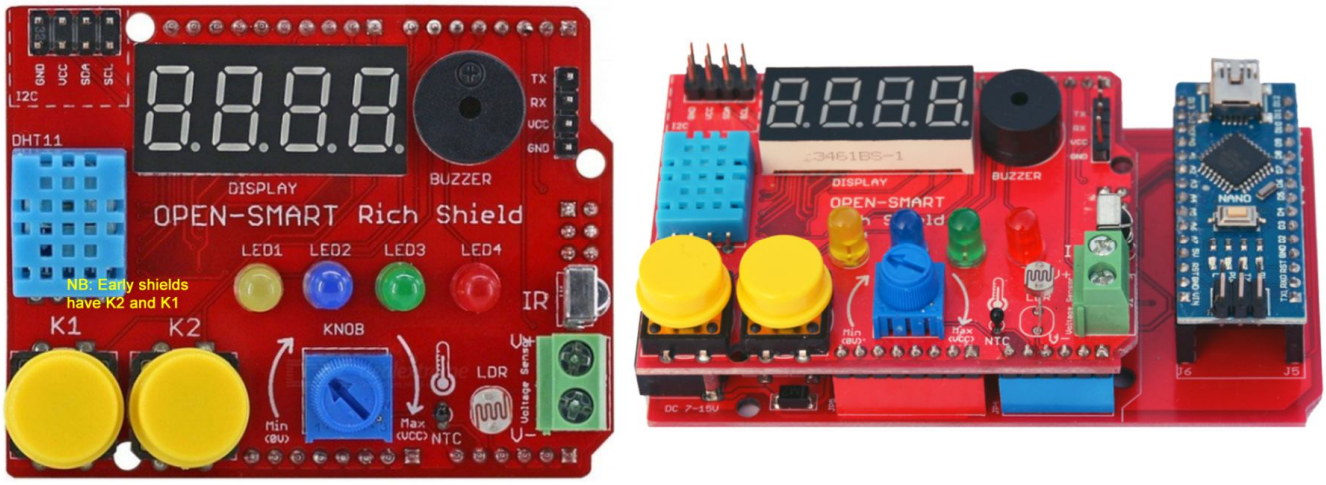

Figure 1: OPEN-SMART Rich Shield (left) and seated on the OPEN_SMART Nano-UNO Shield Adapter.

The Rich Shield is an inexpensive, neat, nifty and powerful add-on that provides the more commonly used ancillary electronic sensors and visual indicators (analog and digital), reducing the time and need to repeatedly find the parts and build/rebuild (breadboard) various Arduino "gadgets" (Arduino term). No external wiring or additional components are required because its onboard components are self-contained and independent of the others. OPEN-SMART also provides custom Arduino code libraries to handle the "heavy lifting" (behind the scenes) programming for you. This shield makes it very especially easy for electronic and programming novices alike to do a lot with it so it's great for "Pros and Joes", groups, classes and teachers alike!

Description

The Rich Shield uses almost every Arduino Uno/Nano analog and digital pin (see Table 1). It also has three sets of "breakout" (DuPont connector) male headers. One set provides separate external circuit ground, voltage (+5 volts) plus a universal asynchronous receiver-transmitter (UART) mapped to pins D0 (RX or serial receive) and D1 (TX or serial transmit); the another two sets provide additional ground, voltage (+5 volts) and the inter-integrated-circuit (I2C) serial bus or serial data line (SDA) and serial clock line (SCL) mapped to analog pins A4 and A5.

Rich Shield Device	Arduino Pin
Rotary angle sensor/potentiometer (10 kilohm – K)	A0 (analog pin 0)
Negative temperature coefficient (NTC) thermistor	A1
Light Dependent Resistor (LDR)	A2
Analog voltage divider sensor (27.75 volts max.)	A3
Infra red (IR) sensor	D2 (digital pin 2)
Buzzer	D3
Light emitting diode (LED1) (red)	D4
LED2 (green)	D5
LED3 (blue)	D6
LED4 (yellow)	D7
Pushbutton K1	D9
Pushbutton K2	D8
7-segment 4-digit LED (red) display control (TM1637)	D10
7-segment 4-digit LED (red) display control (TM1637)	D11
DHT11 air temperature and humidity sensor	D12

Table 1: Rich Shield Device and Arduino Pin Mapping/Assignments
Credit: OPEN-SMART.

*NB**:** For advanced programmers: the Rich Shield's RX/TX (pins D0/D1) may be used for external circuit digital input/output, if and only if, your Arduino program does not use RX/TX for internal serial input/output.*

I was puzzled as to why the Rich Shield's schematic diagram and library code swapped the pushbutton (K1 and K2) pin order assignment, but a closer look at my shield's silkscreen showed that the pushbuttons were mislabelled (as noted on Figure 1)! Instead of throwing out thousands of otherwise perfectly useable product, it made more sense to just change a bit of code for the later produced shields with the corrected silkscreen. You don't have to worry about the mismatch unless you write your own custom library code and map the pushbuttons to the wrong digital pins.

An onboard 7-segment, 4-digit display comes with a pre-programmed character set (0 to 9, A, b, C, d, E, F, H, U, minus, space and decimal point), which is stored in a separate 256-byte electrically erasable programmable read-only memory (EEPROM) chip. If you're really keen, you can add to it and/or design custom character sets of your own by reprogramming the EEPROM.

A Few Radio Related Uses

The Rich Shield provides a nice selection of components that can be used individually or combined to create various Arduino gadgets. Here're a few simple ones that immediately came to my mind plus a few others that are a bit more complex:

1. A direct current (DC) voltmeter using the voltage divider sensor and 4-digit display. The buzzer could be used to just audibly indicate the presence or absence of voltage if you don't need an actual value. Maximum input with the onboard voltage divider resistors is 27.75 volts DC (VDC); accuracy is +/- 0.05 volts. See http://www.ohmslawcalculator.com/voltage-divider-calculator.
   ***Note:** The Rich Shield library assumes, wrongly, that the reference voltage is always 5 volts, which may not be true if using your computer's USB port to power the Arduino µ-controller board (it's usually much less).*

2. A battery voltage monitor and alarm circuit using the voltage divider sensor, 4-digit display, and/or coloured LEDs to indicate up to four battery levels (discharge states), and the buzzer to audibly alert you when the battery is at critical discharge.
3. A Morse code visual and audible tutor using two different coloured LEDs (for dit and dah), the buzzer and potentiometer (for speed control). You can use the Arduino's random number function to generate random alphanumeric characters or commonly used words (stored in an array).
4. A code practise oscillator (CPO) for sending Morse code (similar to #2), but using a pushbutton and buzzer, and use one LED to flash code as you send. Perhaps wire an external jack for a hand key and use it.
5. An identification count down timer to remind you when to identify every 30 minutes (or whatever time you want) during radio contacts. The Arduino's internal millisecond timer (also microsecond, if needed) is fairly accurate for short term timing purposes (maybe a day). You can also use the LEDs and/or buzzer to make it as complex as you like.

Adding an external solderless breadboard makes it very easy to build and interface more complex electronic circuits to expand and enhance what you can do and control with Rich Shield.

6. A radio frequency (RF) voltage meter using the Rich Shield's voltage divider sensor. Figure 2 is the schematic diagram for a simple RF probe. In lieu of the rather "ancient" 1N34A germanium diode, you can substitute it with the more modern equivalent AA143 germanium diode (has a lower junction capacitance) or a 1N5711 Schottky diode (has a lower forward voltage drop). ***Note:*** *For measuring low RF voltages, simply connect the RF probe between the Rich Shield's voltage divider sensor input and the RF voltage source. See Figure 2, next page.*

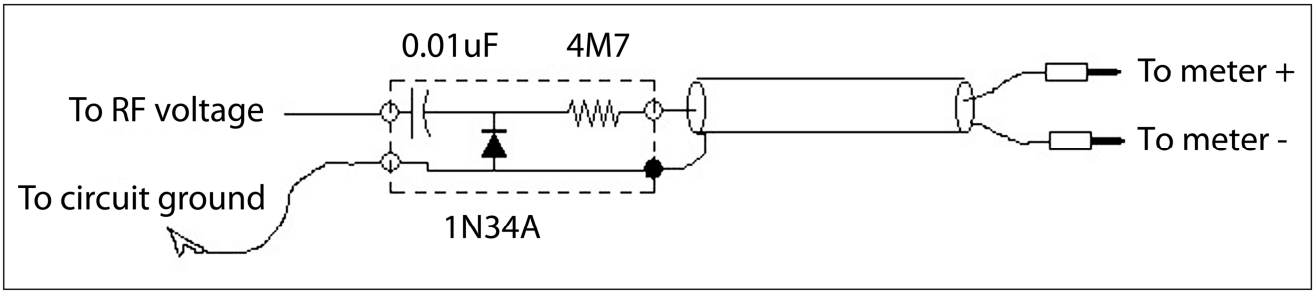

Figure 2: RF Voltage Probe

Very simple and easy to make using spare parts most of us collect like packrats. But you can also buy an inexpensive kit from QRP Kits (Pacific Antennas). Credit: Monty Northrup, N5ESE.

7. I've designed a flea power (QRPp) Morse code beacon that transmits telemetry about my campsite trailer solar/battery power supply status and the onsite temperature/humidity so I can monitor things from home (see Figure 3). It uses the Rich Shield, a solderless mini-breadboard with its RF components mounted on an acrylic plate available from AliExpress.com et al). Coloured code wiring helps minimize errors and to identify power, ground, control and data lines at a glance. And there's still some free space (and headers) to add more "stuff"! But that's for another column.

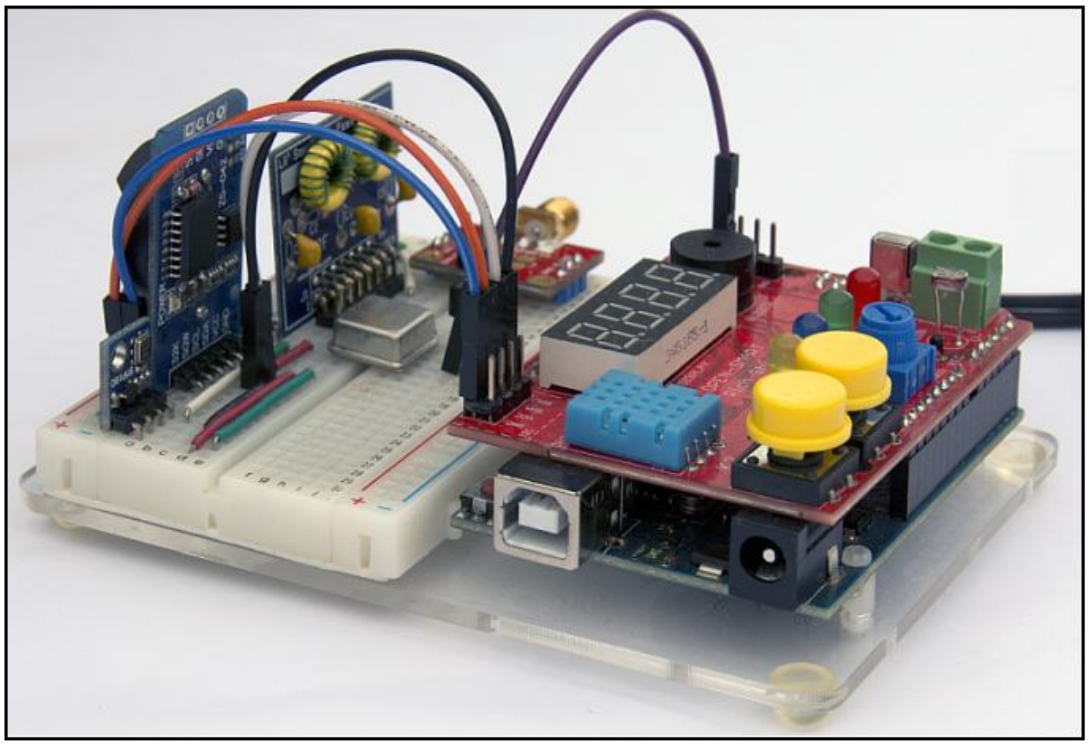

Figure 3: Rich Shield QRPp Telemetry Transmitter

8. The Rich Shield can be programmed to control a couple of off-board relays to turn various devices on/off when certain environmental conditions are met (I.E.: temperature, humidity or light). Example: The LDR is a very useful day/night on/off switch; its resistance decreases as light or lux levels increase and it increases as light levels drop allowing you to program various actions to occur at various light/lux levels. There's an example LDR program that converts its resistance to lux levels; once calibrated to my professional lux meter, the program's conversion equation results were accurate enough, especially considering the price difference!

Note: *Where you're metering (measuring) things like temperature, humidity, lux, etc., the μ-controller can be programmed as data logger and store data at intervals in its static random access memory (SRAM) or EEPROM for later retrieval, plotting and analysis. In this case, the MEGA2560 is preferable because it has four times the memory. There's also a nifty software solution using MS Excel and a free μ-controller USB interface program called "PLX-DAQ". But that's for another column.*

Example Voltmeter Program

This is the example program provided in the Rich Shield Arduino code library for a basic voltmeter gadget. I've added extra annotations and whitespace to help better explain it to novice programmers. ***Note:*** *I'm assuming that most readers know how to install Arduino libraries and use the integrated development environment (IDE). The required OPEN-SMART Rich Shield libraries and example programs are hosted on GitHub.com (also on my "Radio Magic" webpage).*

```
// Arduino required library
#include <Wire.h>

// Rich Shield voltage sensor library
#include "RichShieldVoltageSensor.h"

// Rich Shield display driver library
#include "RichShieldTM1637.h"

// TM1637 display driver clock (CLK) pin
#define CLK 10
```

```
// TM1637 display driver data in/out (DIO) pin
#define DIO 11

// create TM1637 display driver object
TM1637 disp(CLK, DIO);

// voltage (divider) sensor pin
#define VOLT_SENSOR A3

// create voltage sensor object
VoltageSensor voltage(VOLT_SENSOR);

// Arduino required setup function
void setup() {

   // initialize 4-digit display
   disp.init();

}  // end setup function

// Arduino required loop function
void loop() {

   // “volt” variable is floating (decimal) number
   float volt;

   // read voltage source and store in “volt”
   volt = voltage.read();
    // display voltage to hundredth of a volt
   disp.display(volt);

   // wait 1000 milliseconds (1 second)
   delay(1000);

}   // end loop function and repeat forever
```

Not much to it, is there? Well, actually, a lot of complex, "low-level" hardware and software control code is hidden from you inside the #include libraries. You can take this simple program and modify it to your heart's content, then combine it with as many other Rich Shield components to build for your own deluxe gadget.

Potentiometer/Rotary Angle Sensor

As an aside, the onboard linear taper potentiometer (pot) and programmable rotary angle sensor is an interesting analog control device. The Rich Shield library lets you use it as a variable linear resistor to control voltage, current or frequency, etc., but it can also be programmed to have it do something when it's turned between two angles say 180 and 270 degrees, and then do something else when it's turned between say 45 and 90 degrees. Most people can easily eyeball angles when turning control knobs but can you eyeball the actual ohmic or voltage values that the angle indicates without measuring them? If you dissect the rotary angle sensor example program and its library code, you'll learn how the "magic" works.

Note: Potentiometer "taper" is the mathematical relationship between its angular position and internal resistance ratio. For example, when a linear tape pot is turned to its mid (50%) position the output voltage is equal to half of the full voltage applied to it (assuming 0% tolerance error). But for some applications, especially volume controls, non-linear or logarithmic (audio) taper pots are used because human hearing is naturally logarithmic.

My Final

With the Rich Shield you're only limited by your imagination as to what you can make. If you can think it, you can make it. And speaking of making it, the "Makers" group is the fasting growing entity breathing new life into Amateur Radio, injecting it with different and innovative ways of doing things. They are the ones getting us back to the roots and "magic" of radio. But that's for another column.—73