Trailer Voltage Monitor

One of the guys at the office here asked if I could build an Arduino device for monitoring the voltage of the 12V battery in his trailer- as now required for registration of some trailers which have electric brakes. Here's what we built- and it's a bit cheaper than some of the offerings out there.

Shopping list:

XC4430 Leonardo (An Uno will work too, but read on to see why we chose a Leonardo)
XC4428 RGB LED Module
RR0596 10kOhm Resistor Pack
RR0588 4.7kOhm Resistor Pack
HM3211 Header Strip

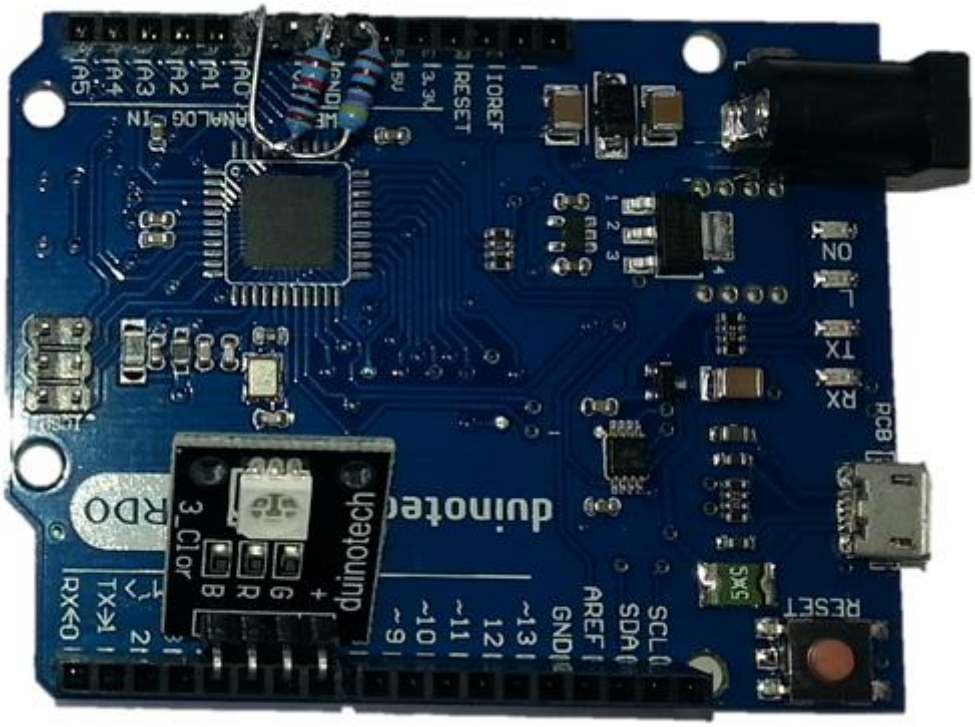
Construction:

As you can see, the construction is quite straightforward. Snap off a length of four header pins, and insert them into the headers on the Arduino to hold them steady (it doesn't matter where for now). Solder the resistors like this:

You might find it easier to trim one leg of the resistor while you hold the other leg, then trim and bend the other leg before soldering. Note that the 4.7kOhm resistor goes between the two outside legs, and that the 10kOhm resistor goes between the first and third legs.

Insert the above ‘module’ into the Leonardo so that the 4.7kOhm resistor is above GND, the 10kOhm resistor is above VIN and the shared resistor leg goes into A0:

Then plug the RGB LED Module into D4-D7 as below:



See how we’ve gently bent the resistors and LED Module so they sit a bit flatter. This is the end of our construction.

Sketch:

This sketch is quite straightforward and doesn't need any external libraries. It's a bit long because we've set a few defines to make it customizable if we need to. Select your board and serial port from the Arduino IDE and press upload. The light should come on red (because it's being fed from less than 11.5 Volts). If nothing happens, check that the LED wiring is right. If you get a green light, your resistor wiring might be wrong.

Connection and Use:

The simplest way to connect the Trailer Voltage Monitor is to run some wires from the battery you are monitoring to a PA3711 Screw Terminal DC Jack, and plug that into the DC jack on the Leonardo. This will cause the LED to light up green if the voltage is above 11.5V, and red if the voltage is below 11.5V. Below about 7V, the LED will not light up at all, and this is what will happen if you have a breakaway.

If you want a better indication that there is a breakaway, you can also feed 5V to the Leonardo through its Micro USB socket. That way, if no power feeds in from the monitored battery, power is supplied from the Micro USB Socket. That's also why we chose the Leonardo- it's easier to find a car charger with Micro USB than the USB-B on the Uno.

Modifications:

The onboard voltage regulator of the Leonardo has a maximum input voltage of 20V, so unfortunately, cannot be used directly with a 24V system. If you wanted to rig up an external regulator to bring the 24V down below 20V, and also change out the 10kOhm resistor for something 30kOhm or higher, then this is possible- note you'll also need to change the RVIN definition below.

If you do want to change any of the other configuration parameters, such as resistor value or voltage threshold, this can be done in the #defines in the code.

Another enhancement you could make is to add a buzzer like XC4424 to give audible warning of the voltage. If you had a buzzer wired up with GND going to - on the buzzer and D12 going to S on the buzzer (which can be done by simply pushing it into the Leonardo headers), you could put:

```
tone(12, 400, 100); //sound tone
```

in the line after

```
ledset(1,0,0);//red
```

So that any time the red LED is on, the tone is triggered.

Code:

```
//Trailer Voltage Monitor
//For power brakes etc.
//supply Arduino through VIN/GND or DC jack from Trailer battery
//LED is green if OK, red if low and off!! if disconnected

//define led module pins here, polarity is automatically handled by presence of LEDPLUS or LEDMINUS
//if LED's have common negative, use LEDMINUS
//if LED's have common positive, use LEDPLUS
//#define LEDMINUS 3
#define LEDPLUS 7
#define LEDBLUE 4
#define LEDRED 5
#define LEDGREEN 6

//Set resistors here:
#define RVIN (10000.0)
#define RGND (4700.0)
#define VTRIGGER (11.5)

void setup() {
  ledsetup();                       //set up pins
}

void loop() {
  int a=analogRead(A0);             //read input
  float v;
  v=(a*5*(RVIN+RGND))/RGND/1023;    //work out v at VIN based on resistors
  if(v<VTRIGGER){                   //trigger LED
      ledset(1,0,0);//red
    }else{
      ledset(0,1,0);//green
    }
  delay(200);
}

void ledsetup(){      //set up led pins depending on whether they are common + or common -, turn all
LED's off
#ifdef LEDPLUS
  pinMode(LEDPLUS, OUTPUT);
  digitalWrite(LEDPLUS, HIGH);
  pinMode(LEDRED, OUTPUT);
  digitalWrite(LEDRED, HIGH);
  pinMode(LEDGREEN, OUTPUT);
  digitalWrite(LEDGREEN, HIGH);
  pinMode(LEDBLUE, OUTPUT);
  digitalWrite(LEDBLUE, HIGH);
#endif
#ifdef LEDMINUS
  pinMode(LEDMINUS, OUTPUT);
  digitalWrite(LEDMINUS, LOW);
  pinMode(LEDRED, OUTPUT);
  digitalWrite(LEDRED, LOW);
  pinMode(LEDGREEN, OUTPUT);
  digitalWrite(LEDGREEN, LOW);
  pinMode(LEDBLUE, OUTPUT);
  digitalWrite(LEDBLUE, LOW);
#endif
}

void ledset(byte r, byte g, byte b){
#ifdef LEDPLUS
  r=!r;         //invert if we're using common +
  g=!g;
  b=!b;
#endif
  digitalWrite(LEDRED, r);    //set outputs
  digitalWrite(LEDGREEN, g);
  digitalWrite(LEDBLUE, b);
}
```