Project 2 - Speedometer

April, 2020
David Ackmann – Gateway Division NMRA

You know you always wanted one. From the time you placed your first locomotive on the loop of your Christmas “Plywood Pacific” and cranked the knob of the throttle all the way to the right, you wanted one. You wanted a train speedometer. I know, because I wanted one too.

Recently, sixty years later, I still wanted one, but what I want now is a bit different than what I had seen described in the past. I could indeed build a point-to-point speedometer, a “speedometer-on-a-stick”, to be taken out and used only when I could remember where I stored it, but that wouldn’t fit my needs. I wanted a speedometer permanently embedded in a loop of my layout, always at hand, always operational. Something simple, that would work well with momentum enabled throttles or locomotives. I made it happen, and so can you.

I built my speedometer using an Arduino Nano microprocessor (AMAZON ASIN: ***Photo*** B07G99NNXL, 3 for $13.99), a Nano Expansion Board (AMAZON ASIN: B07MGVC18K, 5 for $11.99), a 4 line display (AMAZON ASIN: B01GPUMP9C, $12.99), a single light dependent resistor (LDR; AMAZON ASIN: B01N7V536K; 30 for $4.15), a power supply (AMAZON ASIN: B07VBR327W, 2 for $7.99) , a 10K resistor, some jumpers, and a 4 position barrier strip, permanently embedded in my layout, all for about $30.

> If you are not yet familiar with an Arduino, it is an inexpensive microcomputer on a 1.5 to 5 square inch printed circuit board. Arduinos were originally developed in Italy about 15 years ago, and were intended to teach young people how to program a computer. Arduinos are available in several form factors, including the Nano (and it’s more familiar cousin, the “Uno”), and is available on AMAZON.COM.
>
> The Arduino plugs into a USB port of an Apple or Windows-based personal computer. It has many labeled connection points to which you can connect a switch, MP3 player, motor or display. Arduino “sketches” (aka: programs) are written in a variant of the “C” computer programming language and make the attached components work

together as desired. Once a sketch is developed, the Arduino may be disconnected from the PC.

"Sketchs" are entered into a free application called the "Interactive Development Environment", the "IDE" (which is similar to a word processor), and then downloaded into the Arduino (you can download the IDE itself to your PC from https://www.arduino.cc/en/Main/Software). In this article, our sketch is made to sense when a locomotive passes over a light dependent resistor, and when we also provide the length of our loop and the scale of our layout, it can calculate and display the scale speed of any locomotive in our fleet.

For a good introductory book, consider "Programming Arduino: Getting Started with Sketches" by Simon Monk; many good YouTube videos also exist,

To get started, I selected a spot where I wanted to mount the speedometer display and the Arduino microcomputer. Anywhere on the layout fascia, or on the top of the layout, something about 4 by 8 inches, would work fine. Because the display can not mount directly on the board, I used plastic "standoffs" and 1" #4-40 sheet metal screws to fasten the display to the layout; at this point you may want to only attach the display with two screws, because you will probably want to remove it once all components are in place to adjust its brightness. ***Photo***

I attached the expansion board for the Nano close by, again with #4-40 sheet metal ***Photo*** screws, but standoffs were not needed for this board.

I screwed the 4 position barrier strip somewhere close to the Arduino and labeled the 4 positions for "DC+", "GND", "5V" and "A0", which correspond to the places where we need more than two wires to come together.   ***Photo***

Next I found a place close by where I could mount the light dependent resistor between the rails of the track. I removed enough of a tie to fit the LDR between the ***Photo*** rails. I drilled a hole about 1/8th of an inch in diameter all the way through the layout to create a channel where the wires connecting the LDR to the barrier strip will eventually go. I soldered hookup wires to the LDR, sufficiently long to easily reach the barrier strip, and used 3/32" heat shrink tubing to insulate where the wires met the LDR. I ran the wires through the hole and attached the LDR with a bit of glue, making sure the LDR was facing straight up. I soldered spade lugs to the other ends, and attached the lugs to the barrier strip at positions A0 and 5V; it doesn't matter which end goes where.
Photo

I soldered spade lugs to each end of the 10KΩ resistor and attached the resistor to the barrier strip at positions A0 and GND; again, it doesn't matter which end goes where. **Photo**

From the barrier strip's GND position, I ran a wire to one of the Nano's GND positions and tightened it into place with its screw. From the same position on the barrier strip, I ran a jumper with a spade lug on one end and a female jumper socket to the GND pin of the display. **Photo**

From the barrier strip's 5V position, I ran a wire to the Nano's expansion board's 5V position, and tightened the screw. From the same position on the barrier strip, I ran a jumper with a spade lug on one end and a female jumper socket to the VCC pin of the display. **Photo**

Now I ran a male-female jumper from the Nano's A4 socket to the display's SDA pin. Then another male-female jumper from the Nano's A5 socket to the display's SCL pin. **Photo**

The wiring diagram is found below. I generally use 24 gauge wires, and it's just good practice to color code your wires. I use black for ground wires, red for 12VDC, white for 5VDC, and any other colors for the other connections. I often make a spreadsheet of all connections, along with the colored wired I used.

```
                       Barrier                Arduino
                        Strip                  Nano                   Display

                      +--------+            +--------+               +--------+
                      |        |            |        |               |        |
Power                 |        |            |        |               |        |
Supply ---------------0  VIN   O------------O  VIN   |         +-----O  VCC   |
  +                   |        |            |        |         |     |        |
                      |        |  /---------|        |----+    |     |        |
                      |        | /          |        |    |    |     |        |
Power                 |        |/           |        |    |   (      |        |
Supply----------------O  GND   O------------O  GND   O    +--(-------O  GND   |
  -                   |        |\---+       |        |        (      |        |
                      |        |   10K      |        |    |+---|     |        |
                      |        |/---+       |        |    |          |        |
     /----------------O  A0    O------------O  A0    O    |  +-------O  SDA   |
    /                 |        |            |        |    |  |       |        |
 LDR                  |        |            |        |    |  |       |        |
    \                 |        |            |        |    |  |       |        |
     \----------------O  5V    O------------O  5V    O----+  |   +---O  SCL   |
                      |        |            |        |       |   |   |        |
                      |        |            |        |       |   |   |        |
                      +--------+            |  A4    O-------+   |   +--------+
                                            |        |           |
                                            |        |           |
                                            |  A5    O-----------+
```

I cut off the connector end of the power supply and soldered spade lugs to both wires, and drilled a hole through the fascia to allow the wall wart to be hidden. I wired the power supply into the barrier strip, plugged it in and using a volt meter, I verified the proper polarity at the barrier strip. ***Photo***

Now, before we can mount the Arduino into its expansion board, we must download a program, or "sketch" as they are called in Arduino-speak, into the Arduino to make the speedometer work. Using the free Arduino Interactive Development Environment that I downloaded to my Personal Computer (see "Part 3" of this tutorial series), I wrote a program (aka: "sketch") that did three main things: 1) when the locomotive passed over the light dependent resistor, it started a timer; 2) when the locomotive again passed over the LDR, it would check the timer, display the time it took to complete the loop, and then calculate the average speed and display it as well; 3) finally, and immediately, the sketch would restart the timer and do it all again. But, to make this work for your layout, you will need to modify the sketch by entering the length of your track loop and the scale of your layout, which will let the Arduino do the proper calculations for your loop; these values can be found around line 178 of the sketch. One Other thing: before downloading the sketch, make sure you have installed the "LiquidCrystal" library so that your display will work properly, as described in Part 11 of this tutorial. You ***MAY*** need to alter the address of the display around line 180, as ***Screen Shot*** described in Part 11 of this tutorial; most displays have an address of "0x27", but some are "0x3F", so if yours does not display properly, either change the address and try again, or use the SCANNER program as described in Part 11 to verify the proper address setting . Yes, the sketch is quite lengthy, but you don't need to worry about the details. All you have to do is:

1) Visit http://www.github.com/daackm,
2) Open the "Project-2-Speedometer" repository,
3) Click on the "Speedometer.ino" file, and the code will open
4) Highlight and copy the code to the clipboard (Control/C),
5) Open the Interactive Development Environment (IDE)
6) Click on the FILE tab
7) Highlight all the default contents
8) Hit the "delete: key
9) Paste the code into a sketch of your own (Control/V)
10) Replace my values for scale and length of loop with your own.
11) Change the display address, if necessary
12) Save the sketch to your hard drive
13) Download the sketch to the Arduino

After downloading the sketch and disconnecting the Arduino from my PC, I inserted the Nano into its expansion board, making sure the proper end of the Nano went in to the proper socket holes (hint: the USB connector goes near the D12 socket). ***Photo***

When power is turned on, the display should show the software version number, and the scale and the length of the loop that is provided in the IDE; if it does not, you may need to turn the small blue potentiometer on the back of the display clockwise a bit. If the display shows a bunch of bright squares, you need to turn it counter-clockwise (if nothing lights, I hope you did check the display's address with the SCANNER program, didn't you?). The display should be showing a status of "Awaiting Transit..". When the locomotive passes over the light dependent resistor, the display status changes to "Now Timing Transit" and also shows the elapsed time since the transit started. When the engine again passes over the LDR, the display shows the total transit time and the average scale speed, and then restarts the timer. To see it in action, visit YouTube and search for "daackm BVD Arduino speedometer".

In order not to trigger the timer when light hits the sensor between the locomotive and any cars that it may be pulling, a "minimum interval" was written into the sketch, so that there must be at least 2 seconds of continuous light before the transit would be declared to be complete, for smaller scales and shorter loops this value may need to be decreased. There is also a variable which can be set to accommodate layouts with more or less ambient light than I have on my layout.

Best of all, it works and I enjoy it! I have a DCC layout and can use the data collected from the speedometer to set the speed profile for each locomotive, as I see fit. Or, I can just run the trains and see how fast they go. I have been surprised at the difference in maximum speeds as the motors in my locomotives warm up. A difference of 10-15% between the first transit and the tenth is not unusual.

The creative power of Arduinos is enormous, and for me it was well worth the effort. I now have a speedometer, something I have wanted for over 60 years.

-30-

Dave Ackmann, Ackmanns@charter.net, 314-355-8184