YouTube Model Builders eMag

A Free YouTube Model Builders e-Magazine Produced by YouTube Model Builders.

Be Part of the Community
YTMB Live! Shows
YTMB Hangout Shows

VOLUME 3 www.YouTubeModelBuilders.com JULY 2017

ARTICLES YOUTUBE CHANNELS COMMUNITY TIPS & TRICKS

Micro Control Systems For Model Railroading

INSIDE THIS ISSUE:

- A Tribute to Jim Ross
- A Choice is a Good Thing
- An Introduction to Arduino
- Raspberry Pi, Anyone?
- The Table Turns...
- A SPROG Solution

Be Sure To Check Out Columns From Jack Hykaway, Geno Sharp, The Track Planner, Harry M. Haythorn, and Andy Crawford

Cover Photo: Courtesy of JD (Loggin' Locos)

Be Sure To Check Out YouTube Model Builders Live! Join Us LIVE Every Month

Welcome YouTube Model Builders!

We are excited to present the July, 2017 issue of the **YouTube Model Builders eMag** to the community. The YouTube Model Builders "Team" is committed to putting the "eMag" together with the assistance from the model railroading community at large.

We deliver a useful and informative publication for model railroaders who travel this vast net of information. In this publication, we include many informative, tutorial-based articles, information on happenings in the community, listings of up-and-coming YouTube channels, information about the **YouTube Model Builders LIVE!** show, **Hangout Presentations**, along with general information that is inspirational in building of our model railroads.

Our Vision:

To establish free, online resources as a primary source for model railroad techniques and inspiration in an ad-free, selfless service environment.

Our Mission:

The mission of YouTube Model Builders is to inspire individuals for sharing model railroad building techniques through the use of YouTube and other free online resources. Our goal is not only to share knowledge in a community but also assist individuals who are learning or looking for inspiration through the online model railroading community.

— The YouTube Model Builders Team

Editor's *Note...*

We begin this issue with a tribute to Stewart Jim Ross "jiminmichigan" who passed away in May of this year from pancreatic cancer. As a tribute to Jim's friendship and his commitment to the hobby and this community, we dedicate this issue to him. Many thanks to Phillip Wyman for putting together a personal tribute to "jiminmichigan" (see page 6).

There are many challenges we modelers face when building our railroad empires. Some of those challenges are well understood by the model railroading community at large and there are many well-known and commonly available solutions. However, there is a new buzz within this community about using micro control systems such as Arduino and Raspberry Pi and custom-built solutions using these controllers to take our layouts to the next level with the use of animation and automation of lighting, signals, track control, and sound. The challenge of leveraging new technologies and techniques can seem quite daunting and frustrating for many. The key to overcoming these challenges, and for some the fear of the unknown, is to gain and build incremental knowledge on the subject.

To help modelers with this challenge, we have focused this issue of the eMag on the topic of micro control systems for model railroading. We have enlisted the help of the community's technocrats and knowledge mongers to help explain, for example, what is a microcontroller or a Single Board Computer (SBC), what is an Arduino versus a Raspberry Pi, and provide examples of some custom-built solutions being utilized within the model railroading community.

The issue is laid out into two main parts. Part one begins with an overview article by Andy Crawford entitled "A Choice is a Good Thing," covering several micro controllers and SBC boards that are currently most popular. Chris Rood, an engineer by trade, follows with his two-article set introducing the Arduino and the Raspberry Pi boards respectively. Part two showcases two examples of custom-built solutions using the Arduino and the Raspberry Pi boards. Greg and Donna Heinz describe how they are building an automated turntable using an Arduino microcontroller in their article "As the Table Turns…" and Martyn Jenkins discusses how he built a self-packaged plug-and-play throttle solution using the Raspberry Pi 3, JMRI, and SPROG DCC in his article "A SPROG Solution."

The Track Planner in his column discusses several types of prototypical rail yards and how you can implement these on your layout. Harry Haythorn describes the use of truck and freight service by the Union Pacific and Geno Sharp discusses in this installment of "Geno's Corner" his plans to change his layout 'One Mo' Time' in order to be more capable of operations. Jack Hykaway outlines how an internal combustion diesel engine (the backbone of North American railroad freights) works and Andy Crawford discusses various factors that impact our decision to either build it or buy it in his "Food for Thought…" column entitled "DIY vs. DIBuy."

Everyone within our community knows and loves Barry Rosier and we celebrate his marriage to Dorine Menchhofer in the Community Collage with photographs from their wedding. Please be sure to check it out. Also be sure to check out the YouTube channels listed in the Pick 3 section.

Happy Model Railroading!

– **Loggin' Locos**
Editor-In-Chief

Table of Contents...

About the Cover

LEDs flash and blare red on a breadboard as the circuit comes to life during the Arduino Workshop. The circuit uses a shift-register that is controlled by an Arduino UNO and code to make the LEDs flash in one of several chase sequences. Photograph Courtesy of JD (Loggin' Locos).

YouTube Model Builders eMag Team:

Editor-in-Chief	JD - Loggin' Locos
Content Editor	Blayne Mayfield
Content Editor	Jack Hykaway
Content Editor	Jen Hanson
eMag Articles Ambassador	Harry M. Haythorn - UPHS #4043
eMag Photographer	Jack Hykaway

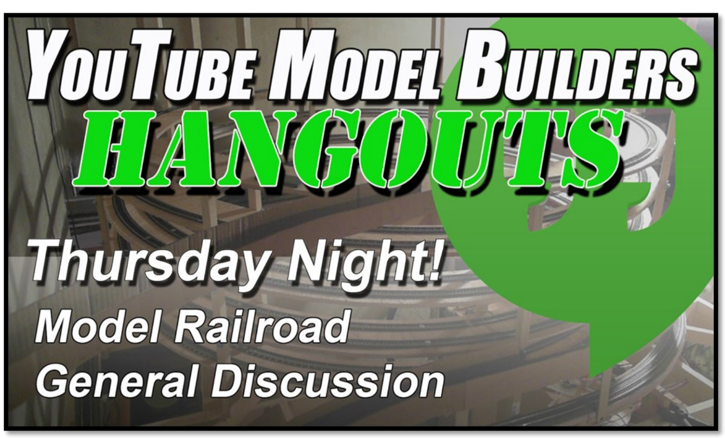
Contributing Authors:

Jack Hykaway
William (Bill) J. Beranek
Harry M. Haythorn - UPHS #4043
Geno Sharp
Andy Crawford

We have several different types of shows each month!

For the latest schedule updates please go to www.YouTubeModelBuilders.com.

Tuesday night shows are topic-driven and are hosted by Johnny of Southeast Rails. The shows include various topics and interesting guests such as Miles Hale and Bill Beranek (The Track Planner).



During this Thursday night show, open presentations are topic driven and fellow YouTube modelers join in to discuss various model railroading topics.

A Tribute to Stewart Jim Ross "jiminmichigan"

By Phillip Wyman

Stewart Jim Ross "jiminmichigan" was a member of the YouTube Model Builders community for over two years. In May of this year, he passed away from pancreatic cancer.

He was a good guy – honest, funny, and stubborn. Jim modeled the "Stu Lines Railroad," which was fashioned after his native state of Arizona. He was active on the YouTube Model Builders hangouts. He always wanted to learn; avidly asking questions and trying to figure out the best way to make things work on his layout. Jim was open to suggestions and willing to try new things. Once he learned something, he would pay it forward by openly sharing his knowledge and inspiring the rest of us to try new modeling techniques.

Although Jim and I came from different backgrounds, I came to learn that we shared the same core beliefs. After meeting him in person, there was a strong bond of trust and friendship that formed between us; we became like brothers. As I got to know him better, I realized that Jim was a very giving person – he was the type of guy that would give you the shirt off his back if he knew it would help you out.

Trust was the basis of our friendship. I knew I could trust him and his opinion. He told it how it was, and didn't sugarcoat it, no matter the situation.

Jim was the type of guy that was greater than words can describe. I am a better person for having known him, and I will miss Jim dearly.

YouTube Model Builders Presents

Next Show—August 2nd, 2017

8 PM Central / 9 PM Eastern

All Newbies Welcome!

Please join YouTube Model Builders along with Chris Heili and guest panelists, who will build live, various projects using the Arduino platform. *Projects include lighting, sensors, servo control, and animation. Each project is quite easily accomplishable by any beginner hobbyist.

Let's explore and learn together some of the coolest projects in model railroading. We encourage you to ask questions directly to the presenters and chat live with them during the workshops. Share videos, pictures, and comments of your accomplishments on our YouTube Model Railroaders Google+ community page.

* Participation in workshops requires an Arduino based project kit and breadboards. For more information, please see the YouTube Model Railroaders Google+ community page and posts announcing the workshops.

Exploring Micro Control Systems for Model Railroading

Part One

Photograph by JD (Loggin' Locos)

A Choice is a Good Thing

By Andy Crawford

Thankfully we hobbyists in the model railroading electronics space have ample choice in platforms on which to build control systems for our model railroads. In fact, there are hundreds of platforms that can be leveraged to build control systems for model railroading projects. However, in this article, I will summarize my top five platforms which, to me, show the most promise, traction, community support, and the most available resources on the web. From these five, I have experimented and have developed model railroading solutions with the Arduino board, the ESP chips, and the Raspberry Pi board. Both Arduino and Raspberry Pi seem to have the largest following and interest in the model railroading community as well as other electronics hobbies (aka the hobbyist electronics space), and the maker space.

Each of these platforms listed below offers their own suitability and fit to any given project at hand.

Below is a short overview of each of the five of the most popular and well-supported microcontroller and Single Board Computer (SBC) platforms.

Arduino

http://www.arduino.cc
http://mrrwa.org

Arduino has gained massive popularity lately with model railroaders, and for good reason. The Arduino is a microcontroller. A microcontroller differs from a microprocessor in that a microcontroller is a set of switches, inputs, and outputs with simple logic to operate it, whereas a microprocessor is at the core of a general purpose computational device (i.e., a computer). Arduino has a wide array of code libraries and software called sketches (firmware programs that reside on the hardware itself) available on the Internet; many of these were developed (and are still being developed) by those in the model railroading community for other modelers to use. You can even write your own sketches or modify the ones you find on the Internet to meet your needs. There is also support for Arduino hardware components, breakout boards, and a special kind of breakout board called a shield. A shield stacks on top of the Arduino Uno or the Arduino Mega (two of the many Arduino variants available) and extends the functionality of the Arduino. An example shield is the motor control shield, which provides greater electrical currents than the Arduino alone can supply; there are many code libraries available to help simplify model railroaders working with motor shields and many types of motors.

Microcontrollers are very well suited to repetitive tasks that don't require complex math or variable interpretation. And Arduino is very well suited for model railroading purposes. There's even a complete DCC command station sketch available that uses an Arduino Uno or Mega plus a motor shield. The learning curve to do basic Arduino development for a custom application on your layout is quite low and you easily can get started with sketches that are available for free on the Internet.

YouTube Model Builders also proudly hosts a monthly Arduino Workshop in addition to the many Arduino-related topics explored together on general video chats nearly every night.

ESP8266, ESP32, and NodeMCU

https://howtoesp8266.com

There is a wide selection of "Arduino-compatible" microcontrollers available. While the ESP series of chips aren't, strictly speaking, Arduinos, they can execute Arduino code. The ESP chips are very small form-factor microcontrollers with integrated wireless networking paired with the standard assortment of inputs and outputs. This makes them very well suited to controlling your layout components and accessories over a wireless network, which will massively reduce the amount of cabling you need to install on your layout.

There are many variations and options for these boards, and a quick review of YouTube videos will give you some opinions on the best particular board for your application. I'm personally quite fond of the NodeMCU firmware running on an ESP8266 board, because of its ease of use and a simple web-based interface.

While you may run either an Arduino sketch or Python code, these boards are still limited by the same premise as any microcontroller – a limited command set.

PicAxe

http://www.picaxe.com

PicAxe chips are a quite simplistic microcontroller in the Pic family of chips. They offer a slightly more limited set of logic available for the modeler and, most critically, a reduced family of communication technologies to integrate several of them into a centrally-controllable system. However, they do meet the most common application needs for a model railroader, such as structure lighting, crossing gates, and other similar projects. They also are cheaper than nearly any of the other common options. But possibly the most interesting thing about PicAxe is its size. You easily could integrate a PicAxe into some of the smallest of N-scale structures or rolling stock.

PicAxe uses a different programming methodology than other microcontrollers. Support and widely available resources are somewhat limited, but the amount of information available within the modeling community continues to grow.

PicAxe also has a very intuitive programming environment that supports a graphical programming interface that would allow even those with little or no programming skills whatsoever to build a project with the little Pics.

Raspberry Pi SBCs

https://www.raspberrypi.org
http://dietpi.com

A few years ago a British company began producing small, microprocessor-based computers called Raspberry Pi. For $35 US, you can get a fully-functional, general-purpose computer with HDMI out, sound, USB, both wired and wireless network, and Bluetooth (at least for the current Raspberry Pi 3 version), all on a board the size of a credit card. There's also a smaller version (roughly half the size) called the Raspberry Pi Zero which has many of the features and most of the functionality of the "larger" Raspberry Pi 3 at cost of $10. For wireless networking applications, the Raspberry Pi Zero W model (which also costs $10) works best. (Wired networking with the Zero or Zero W requires a USB Ethernet adapter.) Because of their smaller size and lower price, the Zero is the perfect device for creating high-quality sound effects on your layout and for other projects that require less processing power than the Raspberry Pi v3 but would benefit from a dedicated, general-purpose computer.

The Raspberry Pi foundation also sells small 5 and 8-megapixel camera boards that are just perfect for integrating into your layout for overhead viewing or for keeping an eye on a hidden staging yard – again, at quite a cost savings.

In general, Raspberry Pi is best suited for projects that require more complex computations, higher memory usage, better-quality sound output, or the ability to utilize a high-resolution LCD display. Unlike Arduino or the other platforms, Raspberry Pi leverages a full-featured operating system such as Linux or MS Windows IoT (Internet of Things). However, Raspberry Pi fails to provide real-time operations, which the Arduino can execute perfectly – one of its core competencies.

DietPi

http://dietpi.com

DietPi is a distribution of the Linux operating system tailored for small, Single-Board Computers (SBCs) that I favor and from which I have built my JMRI core system. At the bottom of the DietPi home page, you'll also find a wide-ranging list of similar SBC systems that run this distribution, and this will give you some ideas about the popularity of this hobby and the SBC market. Most of these SBCs would make a great JMRI system, a dispatcher station, etc. Look at the choices to see if any of the other variants of these SBCs may best fit your particular project. I'm particularly interested in the Nano-Pi Air (http://nanopi.org/NanoPi-NEO-Air_Feature.html) as the basis of my layout camera system. At just over an inch and a half square and for $28 US, they will make a very compact camera base to hide behind the valance or for monitoring staging yards.

SPROG

http://www.sprog.us.com

SPROG DCC makes small DCC command stations and decoder programming devices that connect over USB. Sprog has recently begun offering a device affectionately named Pi-SPROG (http://www.sprog.us.com/pisprog.html). The Pi-SPROG connects to the GPIO pins of a Raspberry Pi 3 and provides a complete DCC command station and a computer running JMRI in a package about the size of a pack of playing cards. And that full DCC system, along with a Raspberry Pi host computer running JMRI, can be had for less than $200 US, and that's pretty impressive.

While I personally use an Arduino-based DCC++ system, I do use a Raspberry Pi 3 for my JMRI system. Martyn Jenkins, a member of the YouTube Model Builders community from down under, uses a Raspberry Pi 3 and Pi-SPROG for his club's modular layout. Look for his article elsewhere in this issue of the eMag for the very impressive solution he's concocted.

LaunchPad

https://www.ti.com/ww/en/launchpad/launchpads.html
http://launchpad4mrr.blogspot.com

LaunchPad is yet another microcontroller board; it is produced by Texas Instruments and is definitely worth considering for your model railroad. Terry Terrance has created a great website for model railroaders to discuss projects based around this board and its add-on boards (similar to Arduino shields), known as booster packs. Users are encouraged to post code and tutorials so that other modelers can learn from them, and there are some awesome modeling projects that can be found on the site.

LaunchPad has some particular niches that I find particularly interesting. While LaunchPad would make a great platform for many microcontroller applications such as crossing flashers or structure lighting, I find it's a great fit for some specific tasks, such as Radio-Frequency Identification (RFID). While RFID readers for Arduino are now available commercially for model railroaders, I am very impressed with what I've seen in the RFID space around LaunchPad.

As model railroaders, we are often recreating systems from the prototype that were originally built with relay logic. And this makes microcontrollers well-suited for recreating those systems in miniature. The boards listed above, as well as many others, may fit your need for controlling, automating, animating or signalizing your layout.

There are a tremendous number of choices of hardware and software solutions available to help you build your railroad's control system that the short list above could do it no justice. But never fear, because YouTube Model Builders community members have a wealth of information to share about their experiences with many of these platforms to help ease your adoption. Explore or talk to members of our community for more information.

Your choice may come down to what meets your needs, but much of your choice may be defined by your personal preferences as much as it does with DCC vendors. But a choice is a good thing.

About the Author

Andy Crawford, 38 years old, is a technology provider to mid-sized businesses and financial institutions, and a lifetime model railroader. Starting young in the hobby with a train set, like many others, and after spending a decade as an armchair modeler, he returned to the hobby a few years ago, in full force. He models a very exacting replica of a 15 mile section of the Clinch Valley District of the famous Pocahontas Division of the Norfolk & Western Railway in 1952.

His interest in exacting replication of the prototype, fine scale craftsmanship, weathering, and prototypical operation can all be seen in his work. For him, recreating the experience of being a railroad professional, 1/87th the size, in the 50's is all the focus that is needed. You can check out Andy's YouTube channel here: https://www.youtube.com/channel/UC8l2bTYfzVY37328sGPD9Bw.

An Introduction to Arduino for Model Railroading

By Chris Rood

All photographs by Chris Rood (unless indicated).

We live in a world where technology is constantly evolving, and we should look for ways to apply this new technology to model railroading. Did you ever want to create your own animations or signaling system, or have your computer communicate with your layout? If your answer is "yes" to such questions, then you are going to like what I have in store for you. In this article, we are going to discuss a microcontroller board called the *Arduino*, which is very popular with hobbyists and makers.

What is a Microcontroller?

A *microcontroller chip* is a simple computer within a single integrated circuit, sometimes referred to as a system-on-a-chip. The chip contains one or more CPUs (central processing units) along with memory and programmable input and output connections. A *microcontroller board* is comprised of a microcontroller chip and its peripheral chips and input/output pins mounted on a small printed circuit board.

What is an Arduino?

The Arduino is one type of microcontroller board. (More correctly, it is a family of microcontroller boards, each with different features.) While the group that originally developed the boards also is named Arduino, the boards have an open source design; this means that other manufacturers also can produce and sell the boards. Regardless of the manufacturer, boards based on these designs generally are referred to as Arduinos.

Arduinos provide an easy-to-use hardware and software platform. The boards can be programmed to read input signals – such as that from a button attached to pins on the board – and then turn on a LED or activate a motor connected to other pins that are set as output. You can look at an Arduino board as a series of inputs and outputs that can interact with and control certain parts of its physical environment, which in our case is your model railroad.

Arduino was developed to teach students electronics and programming. These boards have quickly become very popular with many hobbyists, artists, programmers, and professionals. In turn, this has created a very large community around the Arduino platform, which means there are a ton of FREE resources available to help you learn and use it.

What is great about the Arduino is that you quickly can take your idea for a project and prove that the concept works by rapid prototyping. Once you are satisfied that your idea works, you can take it to the next level by building a permanent circuit that can be installed on your layout. The prototyping is done with an Arduino board, a breadboard, and common electrical components and parts. A breadboard is used to prototype electronic circuits without the need for soldering by using jumper wires to make connections. The breadboard has power and ground rails, terminal strips, and support for integrated circuits. (See Figure 1.)

An Arduino Project

There are two major aspects of any Arduino project. These aspects work together, in tandem to accomplish the tasks you want the Arduino to perform. The first aspect is the hardware, which is made up of the physical parts used in the project, such as resistors, capacitors, sensors, and the Arduino board itself. The second aspect is the software. This is the "brains" of the project, and your idea would not work without it. The software lets you tell the Arduino what to do, which input

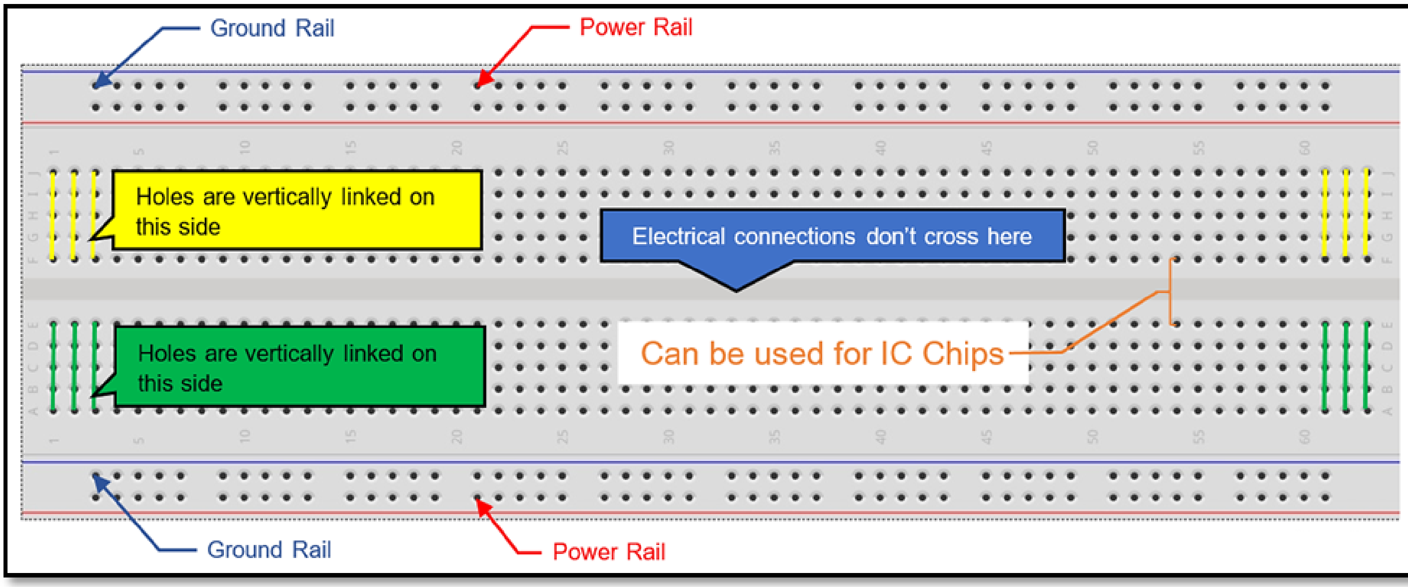


Figure 1. Breadboard Layout. Diagram by Chris Rood.

signals to read and outputs to turn on/off, and how to respond to different events.

The Hardware

There literally are hundreds or even thousands of hardware components that can be used in your Arduino projects. What I want to do here is introduce you to some of the beginner components, such as Arduino microcontrollers, shields, and breakout boards. This will give you a better understanding of the Arduino ecosystem.

First, there are several different models of Arduino microcontrollers available for use in your projects. The two boards most commonly used by beginners are the Arduino UNO and the Arduino MEGA. Most folks begin with the UNO, and they may progress to the MEGA for larger projects or when more program space is required. Arduino boards can be powered either through a USB connection (usually connected to a computer) or from a "wall wart" (i.e., a power supply that plugs into an electrical outlet). Most Arduino boards, such as the UNO and the MEGA, use a 5V (5 Volts) USB input; but a few Arduinos (such as the Pro Mini board) require a 3.3V FTDI (Future Technology Devices International's standard based) connection. You connect your UNO or MEGA to your computer while programming it, and once you've finalized the project and install it on your layout, you can disconnect it from the computer and probably will connect it to a wall wart.

An Arduino receives input signals from and sends output signals to external components through a collection of I/O (input/output) pins. There are two types of pins, the first being the digital I/O pins. The signal on each digital pin is binary, either being "on" (5V) or "off" (ground). The digital pin functions are set in an Arduino program, known as a *sketch*. (More on sketches later.) For example, a pin might be programmed to receive the binary status of a button ("pressed" or "not pressed"). When a digital pin is programmed to generate output, it sends a signal from the board; for example, it could be used to turn a LED on or off.

Some of the digital input and output pins also can do pulse width modulation (PWM). PWM takes an analog (i.e., non-binary) message and turns it into a pulsing signal that can represent many different values, not just "on" and "off". This is used to drive RC servo motors, can motors, and RGB LEDs, for example. When using PWM on standard LEDs it allows you control the brightness. An Arduino board also has analog input pins that allow analog sensors to be read. The analog pins can be used as general input and output pins, as well.

The Arduino UNO board (pictured in Figure 2) offers 14 digital I/O pins, of which 6 can be used for PWM. It also offers 6 analog inputs. The Arduino MEGA (pictured in Figure 3) has 54 digital I/O pins, of which 15 can be used for PWM. It also offers 16 analog inputs. Another feature I like about the MEGA is that it also offers four serial ports for communication. To learn more about these and other official Arduino boards, visit the products page on the official Arduino website.

As mentioned earlier, Arduino boards are based on open source designs. There are several companies other than Arduino that manufacture and sell their own versions of the boards; these boards are known as "Arduino clones" or "Arduino-compatible boards". Arduino clones can be used anywhere that a "genuine" Arduino board is needed, but the clones tend to cost less. These clones can be purchased from eBay and other websites.

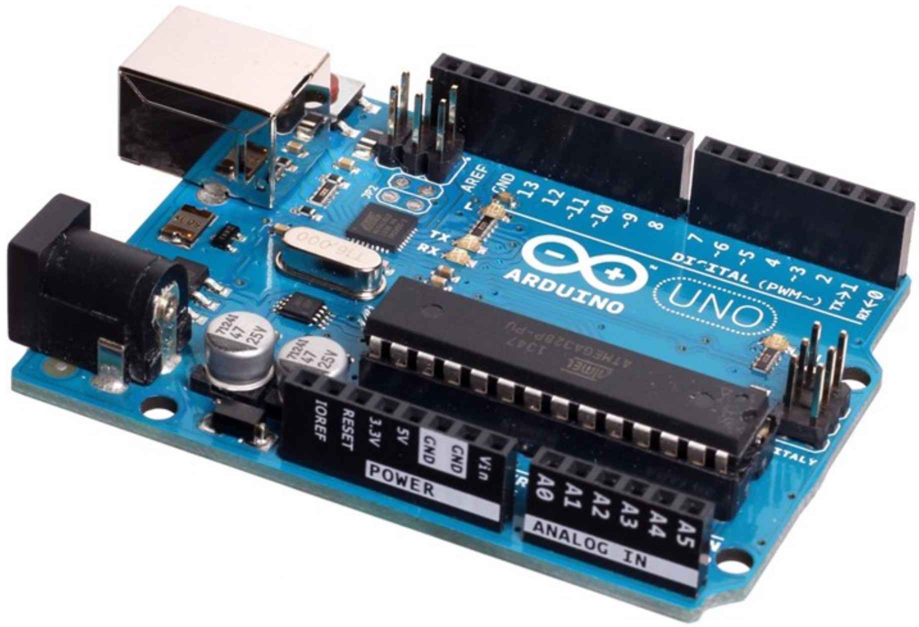
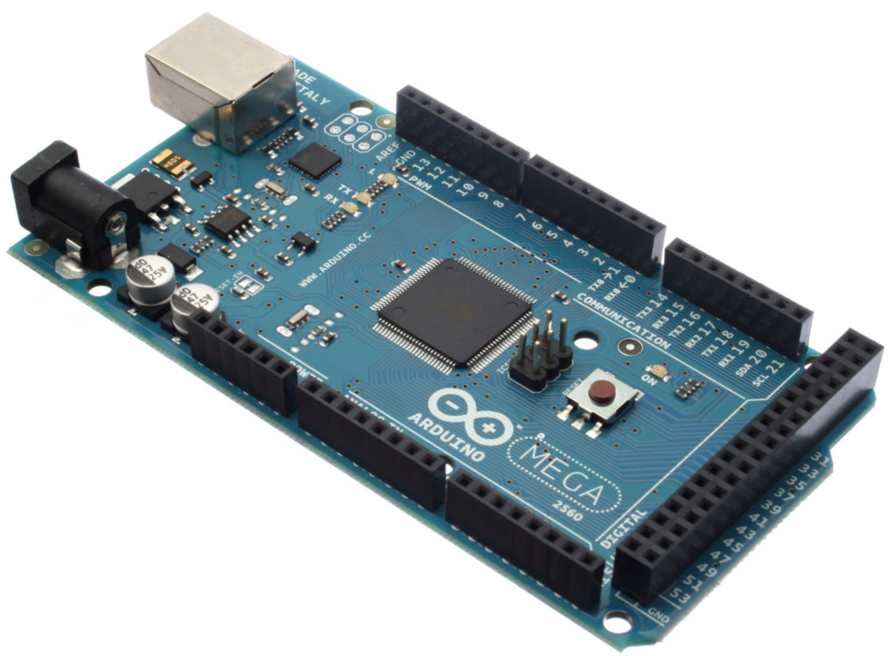


Figure 2 (above). Arduino UNO
Figure 3 (below). Arduino MEGA
Images used under the Creative Commons Attribution-Share Alike 2.0 Generic license.



ESP8266 Boards

Another board that I want to mention, because it is becoming very popular, is the ESP8266. (See Figure 4.) These boards are smaller than the Arduino UNO, but they come with a Wi-Fi chip onboard, allowing them to connect to the Internet and/or your local network. They aren't expensive and can be programmed just like an Arduino. One thing to note is that ESP8266 boards usually have an input voltage of 3.3 volts.

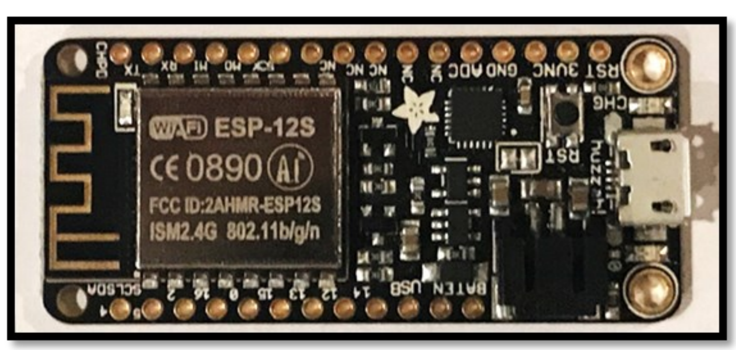


Figure 4. Adafruit Feather Huzzah, which is an ESP8266 board

Shields and Breakout Boards

Since you now have a basic understanding of Arduino microcontrollers, we can discuss *shields*. An Arduino shield is a secondary board that can be plugged on the top of the Arduino microcontroller, extending its capabilities. An example of a shield is a motor shield. (See Figure 5.) The motor shield allows the microcontroller to drive inductive loads such as relays, solenoids, DC

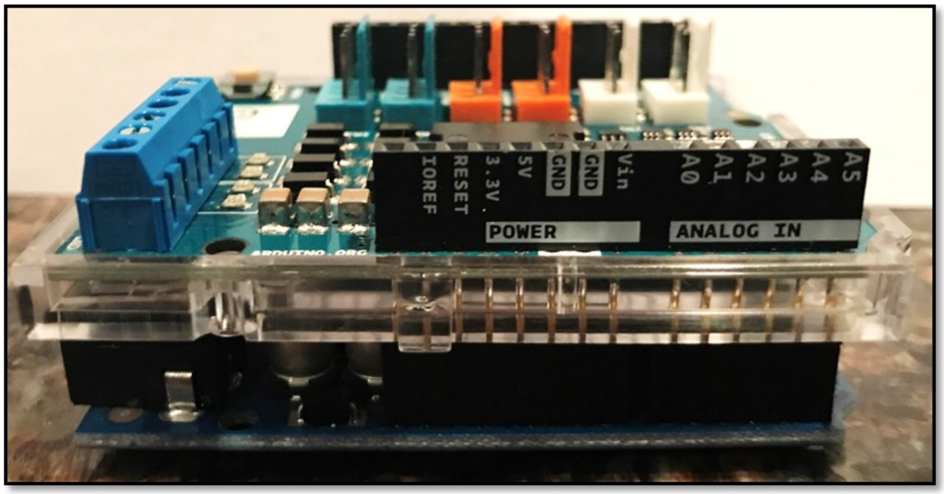


Figure 5. An Arduino motor shield mounted on top of an Arduino UNO.

motors, and stepper motors that usually draw more current than the Arduino output pins can provide.

A *breakout board* is very similar in function to a shield, but it doesn't mount on top of the Arduino. However, unlike a shield, a breakout board can be used with other types of microcontrollers that are not part of the Arduino ecosystem.

The Software

Now that we have discussed the hardware aspect of an Arduino-based project, we will turn our attention to the software aspect. As I mentioned previously, software is how you tell the Arduino what to do. I am not going to dive into this too deeply here, but I will give you a basic understanding of the software.

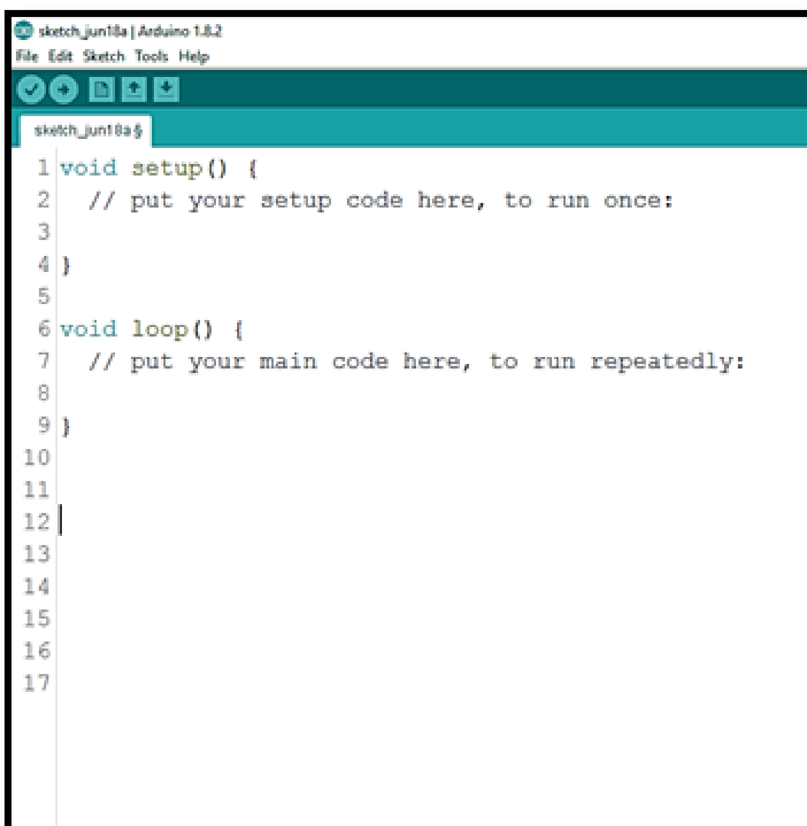
An Arduino microcontroller uses the C/C++ programing language for its programs, and the files containing the program code are called *sketches*. It isn't the easiest thing to learn to write sketches, but it also isn't the hardest.

The easiest way to create Arduino sketches is by using an IDE (Integrated Development Environment), which is a computer program that allows you to write, test, debug, and make changes to your code. The Arduino has its own IDE (see Figure 6), which also allows you to compile and upload your sketch to an Arduino microcontroller and run the sketch. In addition, it can interact with your Arduino microcontroller in order to perform tests on your projects or for real time debugging. The IDE is available from the software page of the official Arduino website. The IDE comes in three varieties: one that you install on your computer, another that you can install on a flash drive so that it is portable (just plug your flash drive into another computer running your type of operating system, and you're ready to program), and a third that runs in your Web browser.

The Makeup of a Sketch

Let's take a look at the two core functions of every Arduino sketch: the *setup* and *loop* functions. (See Figure 7.) The setup function con-

Figure 6. The basic layout of the Arduino IDE. No matter which version you use, the interface will be the same.



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Figure 7. Initially, the setup and loop functions in an Arduino sketch are empty. Note the keyword "void" in front of "setup( )" and "loop ( )"; this means that these functions do not return a value for use by other parts of the sketch.

tains code that you want running only once; this function is executed at the beginning of the sketch run. For example, when we want to set up particular pins as inputs or outputs, this usually is done in the setup function.

The loop function runs repeatedly until the microcontroller is turned off. (Such code is known as an *infinite loop* in computer lingo.) This is where the magic happens within the program. For example, let's say that you want to turn on a LED when a button is pressed and turn it off when the button is released. Then you would write code in the loop function that reads the status of the button, and if the button is pressed, it turns on the LED, or if the button is released, it turns off the LED. Since the loop function keeps repeating until the board is turned off, your operation will continue to work until the board is powered down.

The code that would be placed into the setup function of the button and LED example above consists of the declarations to set up specific pins for input or output. As you can see in Figure 8, line 5 of the setup function executes another function named *pinMode* (which is part of the built-in Arduino code libraries) to tell the Arduino that pin 2 is an input pin; we would attach one terminal of the button to pin 2, and the other to the Arduino's 5V power source pin. And line 3 of the setup function designates pin 3 as an output; we attach the positive terminal of the LED to this pin and the negative terminal to the Arduino's ground pin. The declaration of the pin only needs to happen once, which is why it's in the setup function.

Now let's examine the code in the loop function. In line 10, we declare a variable named *buttonState* into which we read our button's state by calling the built-in function *digitalRead*, telling it to read from pin 2,

Figure 8. A simple sketch to control a LED based on whether or not a button is pressed.

```
1.  void setup() {
2.    // initialize the LED pin as an output:
3.    pinMode(3, OUTPUT);
4.    // initialize the pushbutton pin as an input:
5.    pinMode(2, INPUT);
6.  }
7.
8.  void loop() {
9.    // read the state of the pushbutton value:
10.   int buttonState = digitalRead(2);
11.
12.   // check if the pushbutton is pressed.
13.   // if it is, the buttonState is HIGH:
14.   if (buttonState == HIGH) {
15.     // turn LED on:
16.     digitalWrite(3, HIGH);
17.   }
18.   else {
19.     // turn LED off:
20.     digitalWrite(3, LOW);
21.   }
22. }
```

which is attached to the button. Notice that we declared that the variable *buttonState* is an integer by placing "*int*" in front of it; this means that it can contain only whole number (integer) values. Since we are using a digital pin, the value returned by *digitalRead* will be either HIGH (1) or LOW (0). (HIGH and LOW also are built-in values.) After the button state has been read, an "if-else" statement (lines 14-21) checks the value of *buttonState* to see what to do with the state of the LED pin. If *buttonState* is equal to HIGH, we turn on the LED by calling *digitalWrite* to set pin 3 (which is attached to the LED) to HIGH; otherwise (i.e., if *buttonState* is equal to the value LOW), a different call to *digitalWrite* turns the LED off. Since the loop function is executed over and over, pushing the button will turn on the LED, and releasing the button will turn it off.

Arduino Libraries

In the description of the sketch above, I mentioned that there are built-in Arduino *libraries*. Libraries provide extra functionality for the sketches in the form of pre-written, pre-tested, commonly-used functions; these functions help us perform complex operations within the Arduino sketch without having to write those functions ourselves. There are lots of libraries that come with the Arduino IDE, such as the RC servo library, which provides the heavy lifting for operating a servo using an Arduino microcontroller. There also are libraries that have been developed by members of the Arduino community and can be downloaded from various Arduino forums on the Internet.

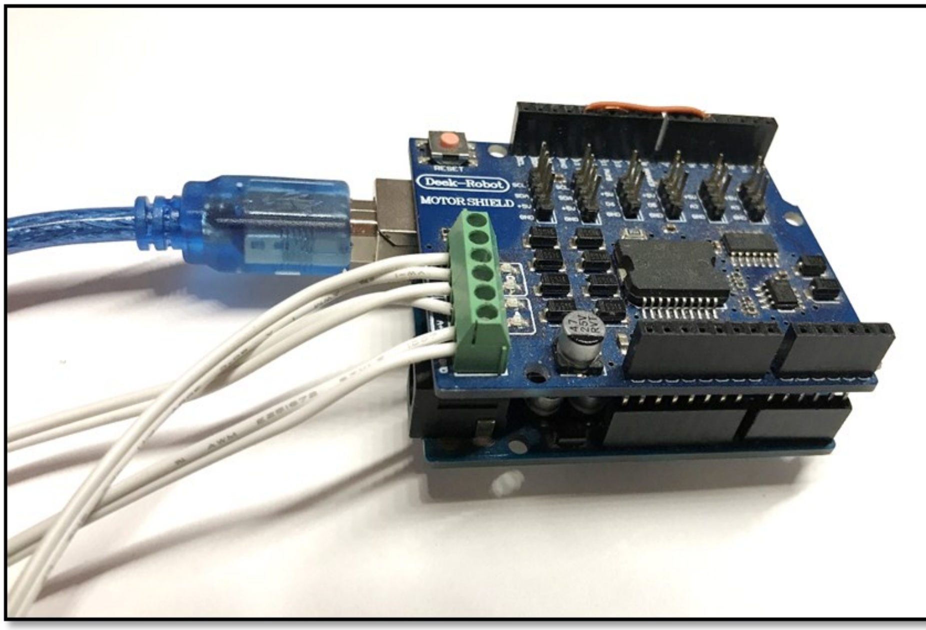


Figure 9. An Arduino DCC++ base station. Photograph by Andy Crawford.

Applying Arduinos to Model Railroading

Since you now know what an Arduino is and have a basic understanding of how it works, let's talk about a few ways we can use it in model railroading.

The Arduino is a great tool that can be used on every model railroad and can be used for a multitude of things. Some things that it can be used for include lighting, block detection, signalization, a DCC command station, and nodes for Java Model Railroad Interface (JMRI).

When I got back into the hobby after several years of not being active, I started to think about some things I wanted on my next layout. One thing I always wanted to do was create my own custom animations; that is how I discovered the Arduino. I have several animation projects that I plan on prototyping and building on Arduino, one being railroad crossing gates. The Arduino would determine when the train is approaching the crossing utilizing block detection or one of the many types of sensors available. Once the Arduino has determined that the train is approaching, it then would start to bring the gates down over the roadway using an RC servo or another type of motor. Finally, it would play the crossing gate bell sound.

On the March 1st, 2017 YouTube Model Builders Arduino Workshop, both JD (Loggin' Locos) and Dave Heili showed their versions of the automated railroad crossing gates using an Arduino. You can view the show on YouTube by clicking here.

Sketches for Model Railroading

Sketch programming may seem overwhelming at first, especially if you've not programmed before. However, on the Web you will find examples of sketches that perform many common operations for model railroads, so you can get by with writing very little or even no code at all. For example, maybe you want a sketch to control a crossing signal, as mentioned above. There are fully-written sketches for that on the Web!

DCC++

Another good project (though more complex) for the Arduino is building your very own DCC base station. (See Figure 9.) This is done by using an Arduino library called *DCC++*, usually with an Arduino UNO or MEGA that has a motor shield attached. The motor shield allows the Arduino to be connected directly to the tracks of your model railroad, thereby allowing the DCC signals to be sent out to the decoders. This system can be connected natively to JMRI software running on another computer, though it doesn't have to be used with JMRI. For more information on the DCC++ open source system, visit its GitHub page.

Our final example is to use Arduinos as nodes for JMRI to control turnouts and signals. Arduinos can act like CMR/I nodes by using this library found on its GitHub page. You also can have JMRI communicate with Wi-Fi-enabled Arduino boards, such as ESP8266, to control turnouts and signals.

These examples are just the tip of the iceberg; there are a hundred, if not a thousand, ways to use an Arduino on your model railroad empire.

Why Use Arduino?

You now have a basic understanding of what an Arduino is and how you can apply it your model railroading.

So why should you use it? The Arduino is great for people who aren't very fluent in electronics, and it gives them an easy way to learn it. For example, I wasn't very fluent in electronics myself prior to learning Arduino. I knew more about writing sketches, but since I tackled the hardware aspect, I have learned more about it and now consider myself to have an intermediate knowledge of electronics. I now am able to design my own circuits around the Arduino for my projects.

Another reason you should use it is that it's easy to learn. Remember when I earlier stated that there is a HUGE community around Arduino? That means there are a ton of hardware and software resources available from many websites and many YouTube videos that will help you learn. The great part is that all this information is FREE. And in many cases, you may not need to modify the hardware diagrams and sketches you download in order to apply them to your layout; all you will need to do is upload them to your Arduino board.

Using the Arduino also will save you money. Say that you are able to build an Arduino-based project for under $50; well, its commercially-available counterpart could cost you $100 or more. Not only can you save some money, but you can also get the self-satisfaction of having built it yourself.

Learn More about Arduino Boards

If you are interested in learning more about the Arduino I have provided below some links to some good resources. Also, make sure you check out the Arduino Workshop series presented by YouTube Model Builders and hosted by Chris Heili of MRL Trains.

- YouTube Model Builders Arduino Workshop
- Official Arduino Website
- Arduino Tutorials by Adafruit
- Arduino Video Tutorials by Jeremy Blum
- Arduino Video Tutorials by Paul McWhorter
- Arduino Tutorials by Dave Bodnar (Some good stuff on DCC++).

About the Author

Chris Rood is a Structural Engineer and is a Volunteer Firefighter. Ever since he can remember he has had an interest in model railroading. After a break for several years, he is now back in the model railroad hobby.

Chris is planning a very large modern-day layout where he plans to freelance model the area in which he lives.

He is interested in all aspects of the hobby, but his specific interests are electronics, command control, layout automation, and structure building (real and model).

He has a very special interest in finding ways to bring non-model railroading technology into the hobby. Chris models in 1/48 scale (specifically 3-rail O Scale - Hi Rail), and On30.

You can follow Chris on his Google+ page here.

Raspberry Pi, Anyone?

An Introduction to Raspberry Pi for Model Railroading

By Chris Rood

All photographs by Chris Rood (unless indicated).

Computers aren't just those things that sit on our desks (or laps) at work or at home. Today, most of us are using train control command systems, such as those manufactured by Digitrax, NCE, etc., and many of us also use microcontrollers and computers to run our model railroading empires; these allow us to interface protocols such as DCC with software such as the Java Model Railroad Interface (JMRI) package.

Computers continue becoming smaller and smaller over time, and there are now computers on the market that are the size of a credit card.

These computers are called *Single Board Computers (SBCs)*. They are somewhat similar to an Arduino microcontroller, but are full-fledged computers and can do a lot more than microcontrollers can. An SBC contains pretty much everything the computer needs to run, including a *CPU (Central Processing Unit)*, memory, video card, sound card, and other components. One of the more popular SBCs available today is the Raspberry Pi.

What is a Raspberry Pi?

The Raspberry Pi was developed to teach students electronics and programming using a full-fledged computer. These SBCs are used to create everything from basic file servers to robots, and they are very popular with computer and electronics hobbyists. This has created a large community around the Raspberry Pi, which means there are a ton of free resources available for it … just like the Arduino!

There are several variations of the Raspberry Pi, with some being older than others. In this article, I will discuss specifically the most current models: Raspberry Pi 3 (see Figure 5), Raspberry Pi Zero (see Figure 6), and Raspberry Pi Zero W (see Figure 7). Before we dive into the different models, we need to explore how the Raspberry Pi stores the *OS (Operating System)*, what OS is used most commonly on the Raspberry Pi, and its *GPIO (General Purpose Input/Output)* pins.

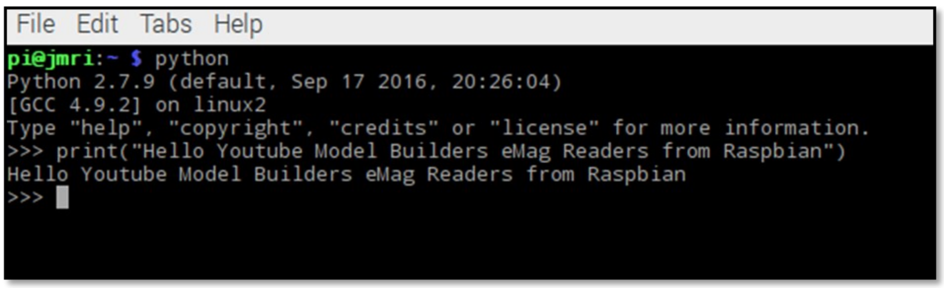
No matter which model of the Raspberry Pi you decide to use, they all store their operating systems and files onto a micro SD card, just like the ones your digital camera uses. You can use 8GB (8 Gigabytes), 16GB, and 32GB cards with a Raspberry Pi. (See Figure 1.)

Typically, the Raspberry Pi is loaded with the Raspbian OS, the official operating system for the Raspberry Pi. There are two variants of the OS: one (see Figure 2) has a full desktop *UI (User Interface)*, while the other doesn't (see Figure 3). The latter is called *Raspbian Lite*. Raspbian Lite is a lightweight OS that is navigated by using the *Linux CLI (Command Line Interface)*. There are other operating systems that can be used with the Raspberry Pi such as *Windows 10 IoT (Windows 10 for Internet of Things)* or *UBUNTU*. Downloadable images for

Figure 1. A micro SD card is smaller than a US quarter!

Figure 2 (above). A screenshot of the Raspbian Desktop.

Figure 3 (below). A screenshot of the Raspbian CLI (or terminal window).



these third-party operating systems are available on the Raspberry Pi official downloads page.

All Raspberry Pi models have GPIO pins, which allow you to connect the SBC to other components, such as a microcontroller. You can control LEDs, servos, motors, read sensors, and do other things. All Raspberry Pis have 40 GPIO pins, as shown in Figure 4; the details of these pins are described on the Pinout page for Raspberry Pi.

The Raspberry Pi 3 (see Figure 5) is the third generation of this SBC and has the following specifications:

- Quad Core 1.2 GHz 64-bit CPU
- 1 GB of RAM
- Wireless and Bluetooth on board
- 40-pin GPIO header
- 4 USB ports
- Stereo output
- Composite video port
- Full-size HDMI port
- CSI camera port
- DSI display port for connecting a touchscreen display
- Micro SD port

This version is the most powerful computer in the Raspberry Pi family.

A Raspberry Pi Zero (see Figure 6) is about half the size of the Raspberry Pi 3. This SBC has the following specifications:

- 1 GHz, Single-core CPU
- 512MB RAM
- Mini-HDMI port
- Micro-USB OTG port
- Micro-USB power
- 40-pin GPIO header (**Note:** header pins will need to be soldered to the Pi Zero)
- Composite video and reset headers
- CSI camera connector

The Raspberry Pi Zero W (see Figure 7) is the same as the Pi Zero, except that it includes both wireless LAN and Bluetooth capabilities.

One of the great things about the Raspberry Pi boards is that they are very affordable. The Raspberry Pi 3 costs around $35 (US), the zero is usually around $5 (US), and the Zero W is $10 (US).

The Raspberry Pi Foundation (the group that designs and manufactures the boards) also makes some neat accessories that can be used for model railroading. These include an 8-megapixel camera module (see Figure 8), a 5-megapixel infrared camera module, and a touchscreen display. For more information on the Raspberry Pi computer and accessories, visit the official Raspberry Pi product web page.

Very much like the shield boards that attach to an Arduino, the Raspberry Pi has HATs and pHATs, which are special-purpose boards that attach to the Raspberry Pi through its GPIO pins to extend the functionality of the Pi. A pHAT is a HAT sized for the Raspberry Pi Zero. (**Note:** most pHats also can be made to work with the Raspberry Pi

3.) There are special HATs (Figure 9) and pHATs (Figure 10) made by SPROG DCC that turn a Raspberry Pi into a low-cost DCC interface that can be used with JMRI.

Using a Raspberry Pi for Model Railroading

As stated in other articles in this issue, microcontrollers are very practical for model railroading, but there are certain scenarios where you are going to exceed their practicability and will require something more powerful. This is where a Raspberry Pi comes into play. Also, just like a microcontroller, there are multiple ways you can use the Raspberry Pi in model railroading. There are three that I would like to mention in this article: sound, video, and JMRI.

We model railroaders are always looking for ways to make our layouts more like the real world. One way of doing this is by adding ambient sound to specific scenes on the layout. The Raspberry Pi is a great way to do this since it gives you the hardware resources and connections to handle the task. Let's say you have a scene on your layout that has a sawmill, and you want to add some sound to it. But you only want the sound to come on when someone is close it. You could use a Raspberry Pi Zero or Zero W with a Passive Infrared (PIR) Motion Sensor and an audio speaker to trigger the audio playback.

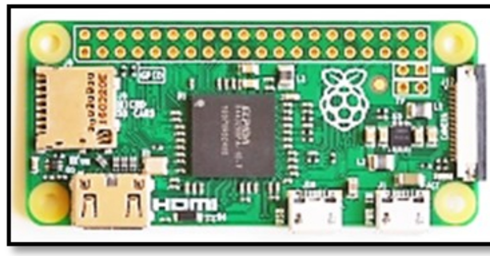
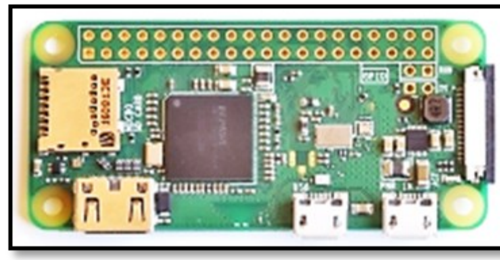


Figure 6 (above). A Raspberry Pi Zero.

Figure 7 (below). A Raspberry Pi Zero W

Photographs by Raspi.tv and are used under the Creative Commons License.



When the PIR sensor detects that someone is close to the scene, it will send a signal back to the Raspberry Pi, which will then turn on the sound. Another thing you could do is have the same Raspberry Pi control the lighting of the scene; that is, you can have the same Raspberry Pi control or do multiple things. (One thing that I should mention is that the Raspberry Pi doesn't have super-high-quality audio output, so you may want to get a third-party sound card to get better sound quality. These are relatively cheap and easy to install.

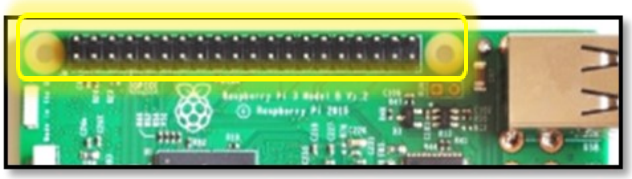


Figure 4 (above). GPIO pins on a typical Raspberry Pi. The pins are highlighted in yellow.

Figure 5 (below). A Raspberry Pi 3.

Photographs by Raspi.tv and are used under the Creative Commons License.

Another cool thing that you can do with Raspberry Pi and its camera module is to add cameras all around the layout in order to give people the feeling that they are right next to real tracks or standing on the sidewalk in your downtown area. Another idea for which you can use the cameras is to view hidden areas of the layout – say on a helix or inside a tunnel – to verify that the train is still on the track and running.

Probably the best thing that the Raspberry Pi can be used for on a model railroad is as a JMRI serv-

Figure 8. An 8-megapixel camera module.

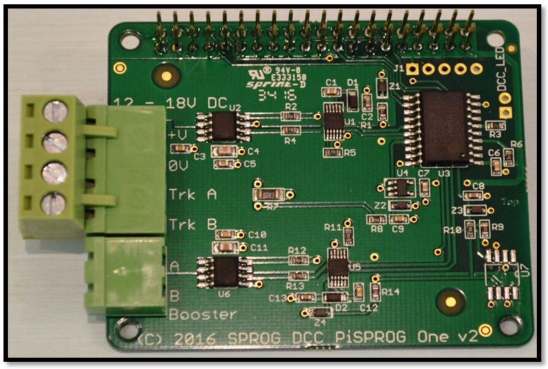
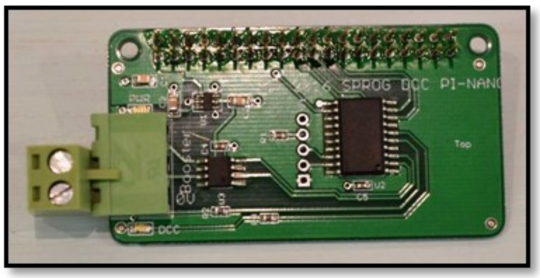


Figure 9 (above). A Pi SPROG One HAT for the Raspberry Pi 3.

Figure 10 (below). A Pi SPROG Nano pHAT for the Raspberry Pi Zero or Zero W.

Photographs by Martyn Jenkins.



er. (See Figure 11.) The Raspberry Pi is affordable and has the resources needed for this task. You can use the Raspberry Pi to communicate interactively with the layout through the USB, GPIO pins, serial communication, and Wi-Fi communication, for example. If you are interested in learning how to install JMRI on the Raspberry Pi, then please visit the official JMRI on Raspbian installation page. One of the great things about using the Raspberry Pi as the JMRI server is that you easily can configure remote access to the Raspberry Pi desktop from your Windows computer or Mac over a Wi-Fi network, meaning that you don't need to attach a monitor and keyboard to the Raspberry Pi.

If you want to go even further, the Pi can be used with DCC++ on an Arduino or a SPROG to create your own low-cost DCC command system using JMRI as the brains of the operation.

Learn More about Raspberry Pi

This article has provided a basic description of the Raspberry Pi SBC, but this is just the beginning. There is a ton more to learn and explore when it comes to these low-cost computers. They are great for model railroading and don't break your wallet.

Just like the Arduino, there is a huge community around the Raspberry Pi, meaning it will be easy to learn how to use it and integrate it into your model railroad. The best part is that these resources are FREE.

If you are interested in learning more about the Raspberry Pi, I have provided links below to some great resources.

- Official Raspberry Pi Website
- Raspberry Pi Tutorials by Adafruit
- Raspberry Pi Tutorials by Paul McWhorter on YouTube.

Figure 11. JMRI Panel Pro running on a Raspberry Pi 3.

About the Author

Chris Rood is a Structural Engineer and is a Volunteer Firefighter. Ever since he can remember he has had an interest in model railroading. After a break for several years, he is now back in the model railroad hobby.

Chris is planning a very large modern-day layout where he plans to freelance model the area in which he lives.

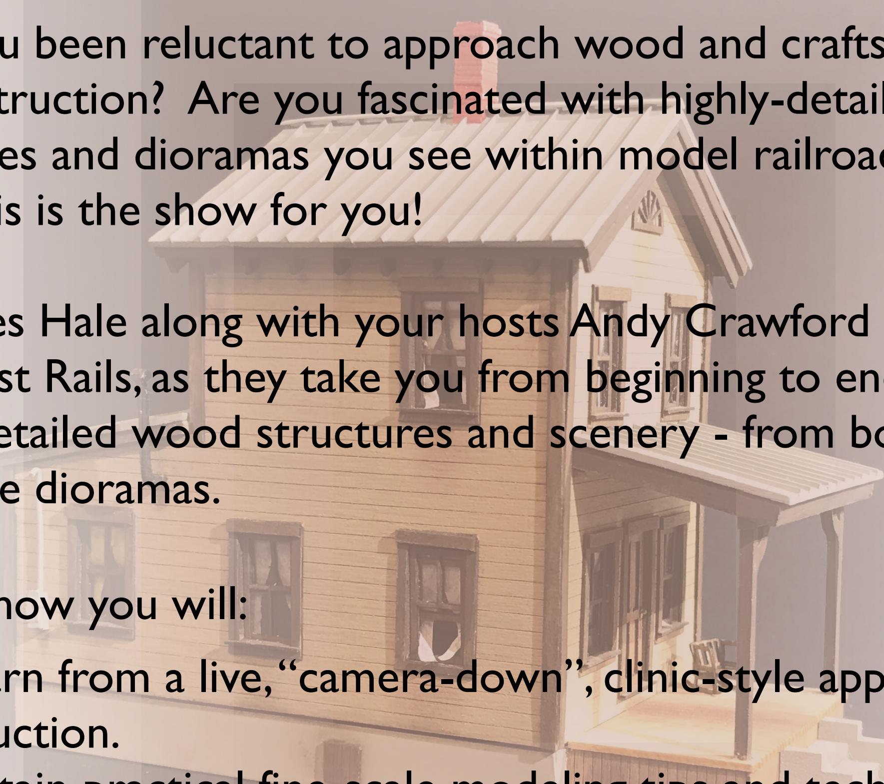
He is interested in all aspects of the hobby, but his specific interests are electronics, command control, layout automation, and structure building (real and model).

He has a very special interest in finding ways to bring non-model railroading technology into the hobby. Chris models in 1/48 scale (specifically 3-rail O Scale - Hi Rail), and On30.

You can follow Chris on his Google+ page here.

YouTube Model Builders
Proudly Presents
The FineScale Live Build Show!

Next Show Is August 22nd, 2017



Have you been reluctant to approach wood and craftsman structure kit construction? Are you fascinated with highly-detailed weathered structures and dioramas you see within model railroading publications? Then this is the show for you!

Join Miles Hale along with your hosts Andy Crawford and Johnny of Southeast Rails, as they take you from beginning to end of constructing highly-detailed wood structures and scenery - from box opening to complete dioramas.

In this show you will:

- Learn from a live, "camera-down", clinic-style approach to construction.
- Obtain practical fine-scale modeling tips and techniques.
- Understand multiple methods for each step of the construction process.
- Develop your modeling skills and overcome the fear of fine-scale modeling.
- Build confidence in your own abilities as a modeler.

Please join us, build along, and learn as you go. We encourage you to ask questions directly to the presenters and chat live with them during the show. Then share videos, pictures, and comments of your progress on our YouTube Model Railroaders Google+ community page.

A Perspective On Track Planning

By William (Bill) J. Beranek —The Track Planner

Rail Yards and Track Planning

When designing prototypical track plans, numerous elements make up the finished product. On my website, www.thetrackplanner.com, I talk about seven design elements that I use on most track plans: walk-along mainlines, narrow shelves, trains traveling through scenes only once, view blocks, etc. What I don't discuss are "elements within elements"; one of the most important of these is the rail yard. If you want your model railroad to operate in a prototypical manner, you will need a rail yard.

I'll limit the scope of this article to the traditional freight rail yard. I won't discuss hump yards, gravity yards, passenger yards, intermodal yards, etc. (Those discussions are for another day.) I will discuss the various elements that make up a freight rail yard and the things you'll need to consider when incorporating a rail yard on your layout.

What is a Rail Yard?

In its simplest form, a rail yard is a series of parallel tracks designed to sort, store, make up, and break down consists. I will limit my discussion to the four major elements of a rail yard: parallel tracks, yard ladders, the yard lead track, and arrival and departure tracks.

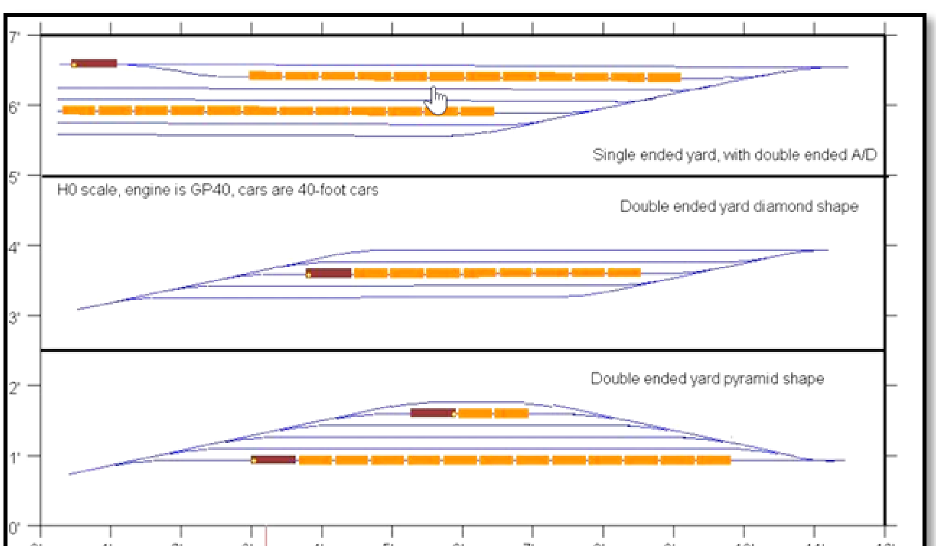
The two most popular yard types are the stub-end (that is, single-ended) yard and the double-ended yard. Stub-end yards normally take up the least amount of space, since they require only one yard ladder (see the description of yard ladders below). Double-ended yards are more prototypical, but because of the space required for two yard ladders, they aren't always used on model railroads. Double-ended yards can either be diamond-shaped or pyramid-shaped, depending on the location at which they're being used. (See Figure 1.)

Depending on the era, rail yards may also include caboose tracks, RIP (repair-in-place) tracks, MOW (maintenance-of-way) tracks, etc. In some cases, turntables, roundhouses, engine servicing facilities, car repair shops, etc. are considered part of the rail yard. Again, for this article, I'll limit the discussion just to the tracks used to sort, store, make up, and break down freight trains.

Stub-End vs. Double-Ended Yards (Advantages & Disadvantages)

Stub-end yard: This is the simplest yard to design and build on a model railroad. The primary disadvantage of this yard is that you only can work the yard (i.e. switch cars) from one end. Stub-end yards usually are

Figure 1. Examples of various yard types.



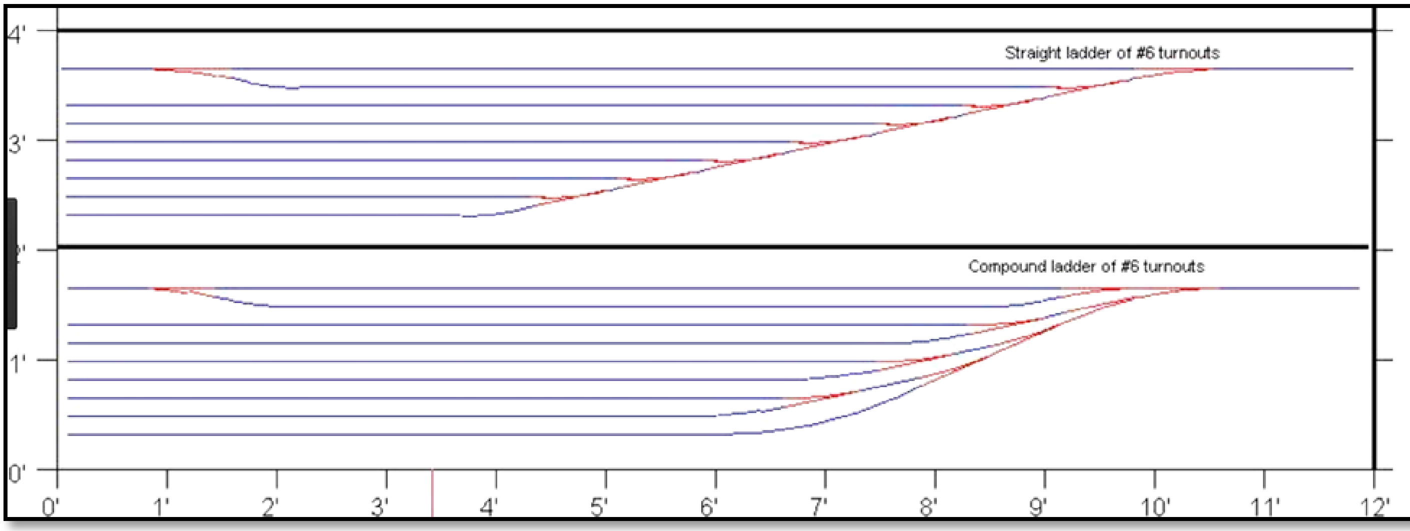


Figure 2. Straight vs. compound yard ladders.

found on smaller layouts where space is at a premium.

Double-ended yard: This type of yard can be worked from either end. The disadvantage is that a yard ladder is required at each end. If space is at a premium, a double-ended yard may be out of the question. Most medium-to-large layouts are designed with double-ended yards. On very large layouts, double-ended yards are very common. If the layout operation generates a lot of traffic, the yard may require two yardmasters, switching either end of the yard.

Parallel Tracks

Every rail yard has a series of parallel tracks. On a model railroad, the number and length of parallel tracks are determined by the type of railroad being modeled and (most importantly) the available space.

If you model a Class I railroad – heavy on freight hauling – the yard should be the focal point of the layout. For a Class I model railroad to operate in a prototypical manner, you will have to allocate enough yard space to handle the daily traffic that the layout generates.

If you're modeling a short line or branch line railroad, you probably can get by with as few as three or four parallel yard tracks. Again, the yard size would be dependent on the size of the short line or branch line railroad you're modeling. If you're modeling logging operations, you may need only two parallel tracks.

Regardless of the size or type of railroad being modeled, parallel yard tracks are used to perform the same functions: sorting, storing, making up, and breaking down consists.

Yard Ladders

Yard ladders are the series of turnouts that connect the parallel tracks to the rest of the railroad. Yard ladders can come in many configurations and can contain any combination of turnout types. Most contain traditional straight turnouts with the single diverging track; however, you also can design yard ladders using curved turnouts, wyes, and specialty turnouts such as three-ways and double slip turnouts. I could write a whole column just on the various types of yard ladders and switches used.

There are two keys to designing a successful yard ladder: 1) design the ladder to take up the least amount of space, and 2) have it connect to as many parallel tracks as your railroad needs.

The two most common types of yard ladders are the straight ladder and the compound ladder. The straight ladder is the easiest for the novice to build but takes up the greatest amount of real estate. The compound ladder, while not popular

Video Chats! If you like real time video chats with other model railroaders, watch for these LIVE chats to join. Ask questions, help others with their modeling videos, or just join in live chat and simply "Chat!"

on real railroads, is a great way to save space on a model railroad and – as seen in Figure 2 – uses fewer turnouts and creates longer parallel tracks.

The biggest mistake novices make when designing a rail yard is using pencil and paper. They never seem to calculate the correct length of the ladder. They start out wanting some particular number of parallel tracks in the yard – which is fine – but then they do not accurately measure the turnout lengths.

Why is using pencil and paper a problem? Because it is too easy to draw six parallel tracks and connect them with five (hand drawn) turnouts, none drawn to scale!

For years, I've designed track plans using a CAD (computer-aided design) program. Even today, many years later, I occasionally get surprised at how much real estate yard ladders take up. Unfortunately, most novices discover this <u>after</u> they've started construction.

Yard Lead Track

The yard lead track is another element I see novices frequently get wrong. The yard lead track is <u>the</u> most important track in the yard. If you design a yard lead correctly, your yard and layout will operate smoothly. Design the yard lead wrong, and your railroad will grind to a halt.

What is a yard lead? A yard lead is nothing more than a single section of track at one or both ends of a yard. (Yard leads at both ends are usually the norm for double-ended yards.) The yard lead is needed so that a yard switcher can pull a string of cars onto the yard lead while making various switching moves.

I have seen many rail yards designed by novices where there is no yard lead at all! Unfortunately, it's not until the layout is up and running and the first operating session is underway that the novice modeler realizes the yard doesn't function as he or she envisioned. The yard switcher is continually fouling the mainline while sorting cars, blocking traffic on the mainline.

If the novice does include a yard lead, they invariably do not design it to be long enough. For my designs, I use the 125% to 150% rule, meaning that yard leads need to be <u>at least</u> 125% longer than the longest track in the yard. An alternative rule is to the make the yard lead 125% to 150% longer than the longest freight train that uses the rail yard.

I see many layout designs in which the yard lead is simply too short. This causes unnecessary switching moves; instead of being able to pull a complete train onto the yard lead, operators have to break the train into sections and pull only parts of the train onto the lead.

If you're an operator who likes to do a lot of switching, you're probably thinking, "This is a good thing; I get to make lots of switching moves!" But wait until your first operating session; because of all the extra switching moves you're making, other operators (and probably the rest of the railroad) have ground to a halt waiting on you.

Arrival & Departure Tracks

Arrival and departure tracks are other key elements of good yard design. Like the absence of an adequate yard lead, not having arrival and departure tracks can quickly bring a railroad to its knees.

Arrival and departure tracks are designated tracks within the yard, designed to hold complete consists that are either arriving or departing from the yard.

Here is a typical scenario for an arriving train: before entering yard limits, the engineer will be instructed by the yardmaster to pull his/her consist onto the arrival track, thus allowing traffic to continue moving (on the main). After pulling his/her train onto the arrival track, the engineer will uncouple the engine(s). At that point, the engineer will move the motive power to the engine servicing facilities, or a person called the "yard holster" will transfer the engine(s) to servicing. The engineer's job is now done; he/she has turned over the consist to the yardmaster and the motive power to the yard holster.

Ideally, you want at least one arrival and one departure track. In this way, the yardmaster can be making

Figure 3. A double slip turnout.

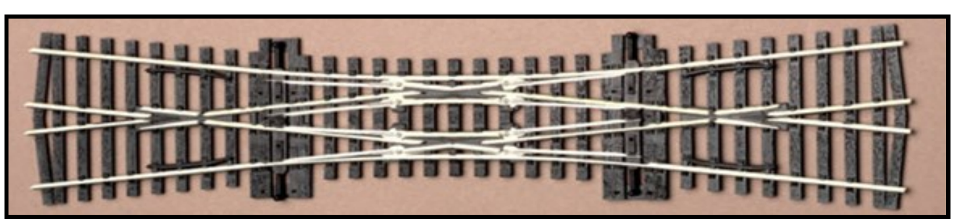


Figure 4. A curved turnout.

Figure 5. A three-way turnout.

up a train on the departure track while at the same time, a train is arriving at the yard.

If space is limited, you can get by with just one track that serves for both arrival and departure. While not ideal, a single arrival and departure track will work on smaller model railroads. If the layout is medium to large, with multiple trains running at the same time, a single arrival/departure track does not work. You'll want at least one arrival and one departure track.

Arrival/departure tracks usually sit between the mainline(s) and the parallel sorting tracks. This makes it easy for engineers to bring their trains into the yard without fouling the mainline or getting in the way the yardmaster sorting cars and/or making up a train for departure.

Specialty Turnouts for Yards

Because a rail yard takes up a large amount of space, using specialty turnouts can save a lot of precious real estate.

As I mentioned earlier, yard ladders eat up the most space, and anything you can do to reduce their length is a big plus. Below are a list and descriptions of specialty turnouts.

Double slip: This offers the most space savings of any turnout. The double slip turnout takes the place of three turnouts, allowing the operator to traverse one of four tracks within a rail yard. (See Figure 3.)

Curved: Curved turnouts also can save space over the traditional turnout. (See Figure 4.)

Three-way: Three-way turnouts work well in a small logging camp yard, where only three parallel tracks are needed for prototypical operations. (See Figure 5.)

Wye: Like the three-way, not many modelers consider using a wye in a yard. But the wye track also can be used to save space. (See Figure 6.)

In Conclusion

On model railroads, rail yards do not need to be overly complicated. A fairly simple design based on the key elements described above is all the modeler needs. The challenge is designing the rail yard so it operates like a real railroad.

When planning your next layout or reconfiguring your current layout concentrate on the rail yard first. Get it right and the rest of the railroad will fall into place.

Figure 6. A wye turnout.

About the Author

Bill Beranek - The Track Planner has over forty years in the model railroading hobby. Bill enjoys golfing, traveling, and of course designing "prototypical operations" focused track plans. He has been a member of a local 135+ member model railroad club since 2003 and has served twice as the club's president, twice as a board member, and is currently serving as the club's treasurer.

Bill is currently working on his latest triple-deck HO scale layout depicting the SP&S (Spokane, Portland & Seattle Railway) in southern Washington and the OTL (Oregon Trunk Line) on the upper level in northern Oregon in the mid-50s.

You can find more about Bill—The Track Planner at:

www.thetrackplanner.com.

HARRY'S UP-HUB

Harry M. Haythorn, UPHS #4043

Union Pacific Rail Truck Service

Welcome, all to the UP Hub! In this issue, we are going to look at truck freight service. We also will cover some equipment and how you can incorporate truck freight service on your layout. I hope you all will come along with me on this short but informative piece.

Truck Freight

Many of us order items for our railroads on the Internet, especially if we don't have a local hobby shop. Then, we wait (anxiously) for the mail carrier or for the step vans of UPS or FedEx to deliver our orders; these are the modern day equivalents of Less Than Car Load (LCL) service.

In the financial boom of the post-WWII days, many people were ready to transition from a period of a war ration books and hard luck to a more normal economy. As the public acquired more money for the finer things in life, the railroads answered the call of the people, shipping more and more items that the American public needed and wanted. In 1947, Union Pacific started its own rail-to-truck service that coincided with the LCL service that was moving everything from baby chicks to washing machines, to pallets of flour – and everything in between (see Figures 1 and 2). In those days, the railroads owned the trailers and private trucking companies provided the trucks to pull them.

Figure 1. Dock workers at the Brady Island Depot.

In Nebraska, the Nielsen and Petersen Trucking Company was contracted to be the carrier for UP trailers. They ran routes in and around the towns and cities to deliver items that people needed and wanted. Most of these routes were 50 miles or less, and the trucks ran on schedules that delivered goods between 8 AM and 8 PM. The reason that this type of service was hired out was that most railroads didn't have the proper ICC (Interstate Commerce Commission) permits to operate any type of trucking, so most of these operations were handled through multi-year lease agreements between a trucking company and the railroad. Lease terms varied from company to company, but most included maintenance of the trailers and trucks paid for by the railroad, and performed by or at the discretion of the trucking company.

Trucks and Trailers

The most common equipment used across the UP system was Fruehauf Trailers, and GMC/Chevrolet or International Tractors; after 1949, White, Mack, and just about every other known semi-tractor manufac-

Figure 2 (above). An example of the three UP Express Service boxcar schemes.

Figure 3 (below). Here we see (from left to right) a 1946 Chevrolet, a 1949 Mac, and a 1948 International I-290.

turer weren't uncommon, either (see Figure 3). The trailers were painted in the Streamliner Scheme. Some trailers were lettered for U.S. Mail and Railway Express to keep them separate from the standard trailers, which were lettered for Rail-Truck Service only. The early tractors were yellow with the UP shield on the doors; many had black front fenders (see Figure 4), and some also had a black hood. These trucks usually were used on the Mail Service trailers but were not limited to them. By 1956, most of the trucks with black fenders/hoods either were repainted all yellow or had run their last miles.

Figure 4. This 46' Chevy sports the black fenders.

Modeling these trucks in both HO and N scale has been made a breeze thanks to multiple manufacturers, with models available from Athearn, GHQ, Ulrich, Classic Metal Works, and many others. These models are some of the best on the market and can easily be detailed to accurately depict a truck or two that was used by a local trucking company; even if you don't model the Nebraska Division or even the UP (all railroads

Figure 5 (left above). A few early Trailers on Flat Cars (TOFC), including two 25' Fruehauf rail-truck service trailers.

Figure 6 (left below). The local switcher has spotted a few baggage cars and a truck is unloading a large piece of furniture that will be delivered later the same day.

had their own rail-truck service in some form or another).

Incorporating rail-to-truck service on your layout can be as simple as adding a few trucks and trailers at the depot, trans-load facility, or sitting at a grade crossing with an express boxcar with a door open spotted there. Now if you "play the game" as deeply as I do in my operating sessions, you have a huge fleet of express boxcars and reefers, trailers on flat cars (see Figures 5 and 6), and multiple trucks and trailers; these serve towns from the depot, the freight house, the Railway Express buildings, and all over the layout. I move my trucks around the layout all the time based on the loads that they have picked up from the express service and LCL cars. This gives life to the layout beyond the trains.

I hope that you all have enjoyed this trip off of the rails and onto the roads and highways. In the coming months, I plan to cover a few more topics along these lines, including the Express Service boxcars that were used on the UP from the late 1930s through the 1950s, and beyond. I also will cover the UP Bus fleet and other less covered topics (see Figure 7).

Figure 7. This shot depicts a Railway Express Agency (REA) delivery truck and a UP bus.

About the Author

Harry is a rancher in Nebraska who works with his father and grandfather to help run their 22,000-acre, 1,500-head of mother cow, ranch. Harry has been model railroading for over 20 years and models the Union Pacific Steam era from the 1930's to the 1960's, in central and western Nebraska. Harry is a Sustaining Member of the Union Pacific Historical Society and a member of the UPHS Streamliner 100 club. He is a National Model Railroad Association member currently working on his Master Model Railroader Certificate. Harry regularly posts videos on his YouTube page. You can follow Harry as he works on his 7th layout at https://www.youtube.com/channel/UC6-MPHmYU3Cc2uEVfjZDIcQ.

YouTube Model Builders LIVE!

Join Us LIVE Every Month

Air Dates

YouTube Model Builders LIVE! show is aired monthly with a great line up of events and panel members.

The main focus of YouTube Model Builders LIVE! is to provide a Q&A style forum for YouTube modelers to interact with their favorite YouTube model builders.

Come watch and remember to register for great door prizes during the show! For the latest schedule updates go to www.YouTubeModelBuilders.com.

YouTube Model Builders HANGOUTS

Live Chat / Presentations

YouTube Model Builders Tuesday night Topic-Driven presentation now start at 8 PM CST / 9 PM EST and are scheduled for up to 2 hours so you have plenty of time to ask questions and learn. These presentations are also recorded for later viewing. For the latest schedule updates go to www.YouTubeModelBuilders.com.

Space is limited so make sure you don't miss these popular Hangouts!

Community Collage

In this issue, we present photographs from the marriage of Dorine Menchhofer to our very own Barry Rosier. The enchanting affair was held on June 24th, on the beach at Cape May, NJ. Barry is a founding and an active member of the YouTube Model Builders community. We thank Barry and Dorine for sharing their wedding with us and wish them both a long and happy life together!

If you would like to share pictures of your layout in the Community Collage, please contact us at YTMBeMag@gmail.com.

YouTube SHARE

PICK 3

In each issue we share with you three YouTube Model Builders' channels that provide the community interesting ideas, tips, tricks, and resources. Here are three channels that will help you be more creative in your modeling efforts.

Scale Model Trains & Colorado's Joint Line

https://www.youtube.com/user/RailfanLayout

This site is full of how-to, product reviews, op sessions, and railfanning videos. The layout highlighted on the channel is in HO scale and is based on Colorado's Joint Line along the front range of the Rockies; specifically, it depicts the Denver to Palmer Lake section, primarily in the 1992-1995 era.

Valerie Valley RR (Vlad)

https://www.youtube.com/user/CTPAHHuKChannel

Vlad's modeling efforts are centered around his N-scale layout. On his channel you will find videos that highlight Arduino projects, layout construction, decoder installation, structure scratch-building, and more. The audio in these videos is Russian, but the English captions on the video make this channel worth checking out for those of us who don't speak Russian.

Da'Bobs N-Scale Mancave

https://www.youtube.com/user/DaBobsNScaleMancave

Bob is in the process of expanding the Mancave N-scale layout into a whole-house layout. The mainline is on a shelf 7 1/2 inches from the ceiling, and it connects downward via helix to the staging yard and an 18-stall roundhouse. Bob shares how he creates benchwork, makes electrical connections, scratch-builds structures, etc. In addition, he reviews N-scale products and visits model train shows.

Into Facebook?
Check out the YouTube Model Railroaders Facebook page!

Geno's Corner

Out With The Old & In With The New One Mo' Time!

By Geno Sharp

All Photos by Geno Sharp

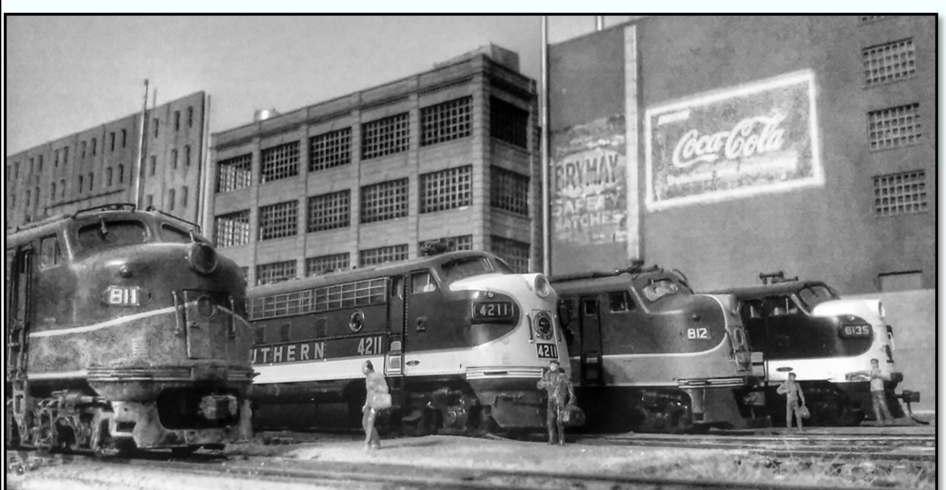
Hello all, welcome back to the corner! I hope everyone is well and enjoying the summer weather.

Those of you who have followed my modeling for any length of time know that I am not afraid of change. I have changed everything from track arrangements to scales on my layout more than just a few times – I get much flack for that. I started out with HO-scale on the Central City Beltline design, then I had a short stint with two-rail O-scale. I returned to modeling HO on my Birmingham Beltline layout. Why did I change, you ask? The answer is quite simple: I do not plan for the long term.

Sure, I have a vision of what I want the railroad to look. However, once I achieve my vision of a completed layout, I find that it no longer serves a purpose. I was the first to carry the torch promoting the idea that "operations are not every modeler's thing," but in reality, I have found that my layout satisfaction will never be until I achieve some level of planned operation. A planned operation would give the design an actual purpose of being, justifying its existence.

My vision is a perfect segue to this month's edition of the corner. I want to go a different route, by covering various topics, focusing on the creation, and the construction of a new layout project while documenting step-by-step the whole process in detail. I hope that you will follow along with this new adventure and if you are floating around in the model railroading world, lost – a bit like I am – maybe we can find our direction together and create a complete and fruitful layout that will operate for years to come.

Figure 1. Working on the South Dixie layout has allowed me to learn how to create scenes which lend themselves to model railroad photography.



Let us start with the current South Dixie design which needs to come down, to-the-bare-walls-down! For the few that already know what is happening, the burning question is "why?" My reasoning for starting fresh is that the layout is not functional in its current state - there is no predetermined operation or purpose for the current design.

It is not a lost cause because the South Dixie layout has allowed me

Figure 2. The new layout will keep the old Birmingham Beltline theme, modeling street running and other scenes of the Deep South, while incorporating a more operational design.

to practice with modeling. I improved my structure-building and scenery-laying skills. I learned how to design certain scenes which favor model railroad photography. I hope to continue this on the new layout (see Figure 1).

Now that the decision has been made to scrap the current plan, what's next? The first step is to deconstruct the first design and brainstorm what parts could be salvageable for the new scheme. Although the thought of moving into the "designed for operations" area of model railroading has been on my mind for several months now, it is not an area of the hobby that I understand. I want to incorporate an operationally-diverse track plan into the thought process and design of the new layout. My belief is that I will keep the old Birmingham Beltline theme – modeling street running and other scenes of the Deep South – while incorporating a more functionally operational design. Keeping with the old theme allows me to recycle region-specific structures that I have already modeled, and to continue recreating in miniature a region that I enjoy modeling (see Figure 2).

Now I must say that I had help with the pending operational design. I hope most of you are familiar with Kevin Smith and his "Scale Model Trains & Colorado's Joint Line" YouTube Channel. Kevin has offered ideas for my new layout. I owe Kevin a debt of gratitude because, without him, it would have been unsuccessful.

In the next edition, we will kick off part one of my layout project. I will cover design elements and operational ideas for the new plan, including the pre-construction. This new project will include a video, made specifically for each edition of the corner. The video will give readers an overview of that month's Geno's Corner article, and it will give viewers the chance to leave their input in the comments section. I hope that you enjoy following along with the building of the new Birmingham Beltline layout. I look forward to seeing you next time in the Corner!

About the Author

Geno Sharp is a retired law enforcement officer with 21 years of service.

Geno has been involved in model railroading for over 30 years and is now a YouTube channel owner.

He produces a monthly layout blog video for his YouTube channel, Gknos Custom Models, as well as various "how-to" and structure build videos.

Geno is currently working on his HO-scale South Dixie Railroad which is home to the Central City Belt Line. The layout is set in the late 1940s to early 1950s. It features steam and first generation diesels from many of the deep south railroads His layout features many highly detailed and weathered scenes, and hand-laid track.

You can learn more about Geno's weathering techniques and about his Central City Belt Line on his YouTube channel Gknos Custom Models.

Welcome to All New Subscribers!

Here are some themes we are developing for our upcoming issues:

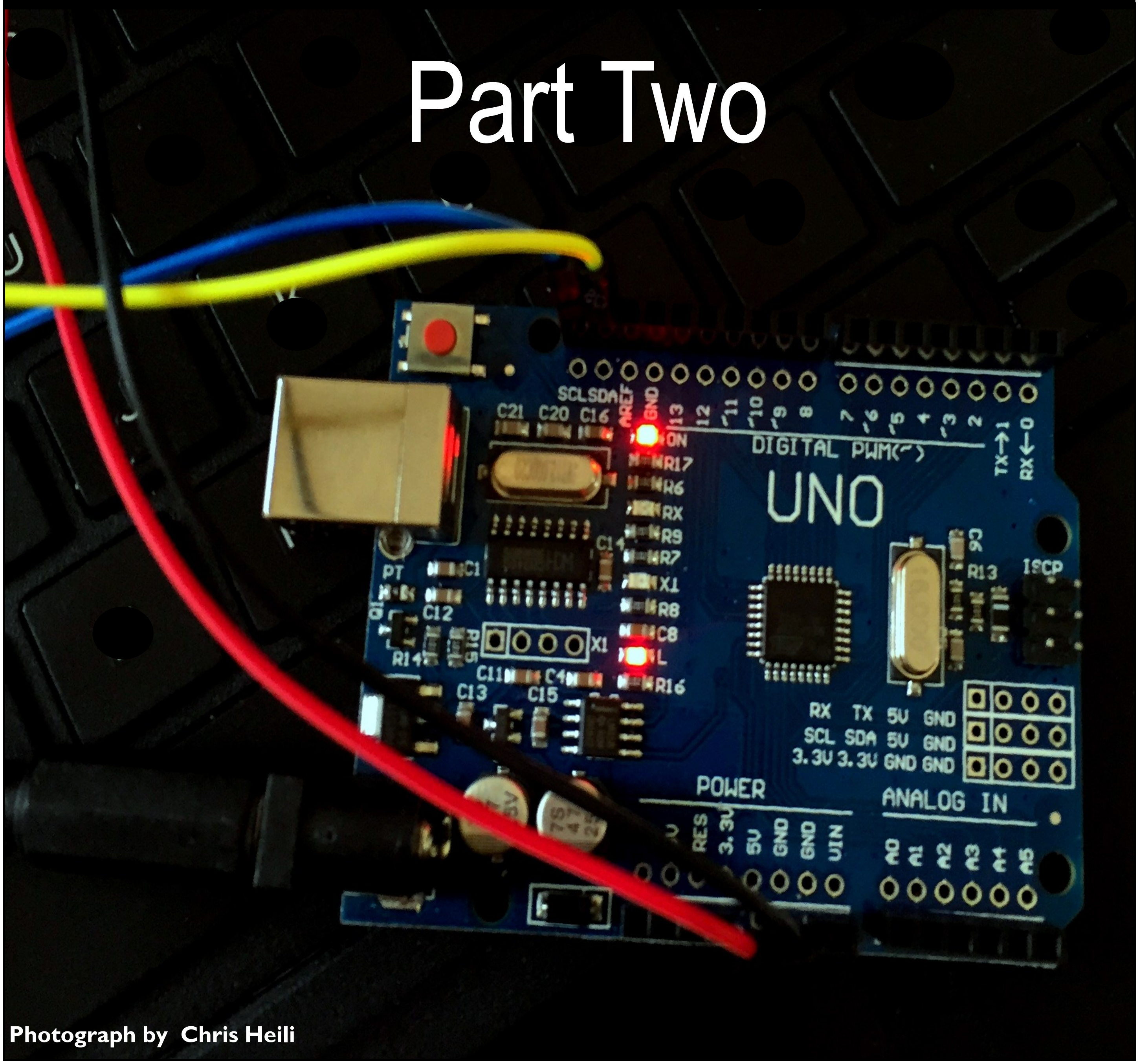
- **Modeling Passenger Trains**
- **Prototypical Operations**
- **Using 3D Printing and 3D Modeling in Model Railroading**
- **Model Railroad Photography**
- **Modeling Narrow Gauge**

If you are interested in contributing to the eMag an article or photographs related to any of the above topics, or even a topic you would like for us to consider, please drop us a note at YTMBeMag@gmail.com. To learn more about the eMag, or to subscribe now, please visit www.YouTubeModelBuilders.com/emag.

We look forward to hearing from you soon!

Exploring Micro Control Systems for Model Railroading

Part Two

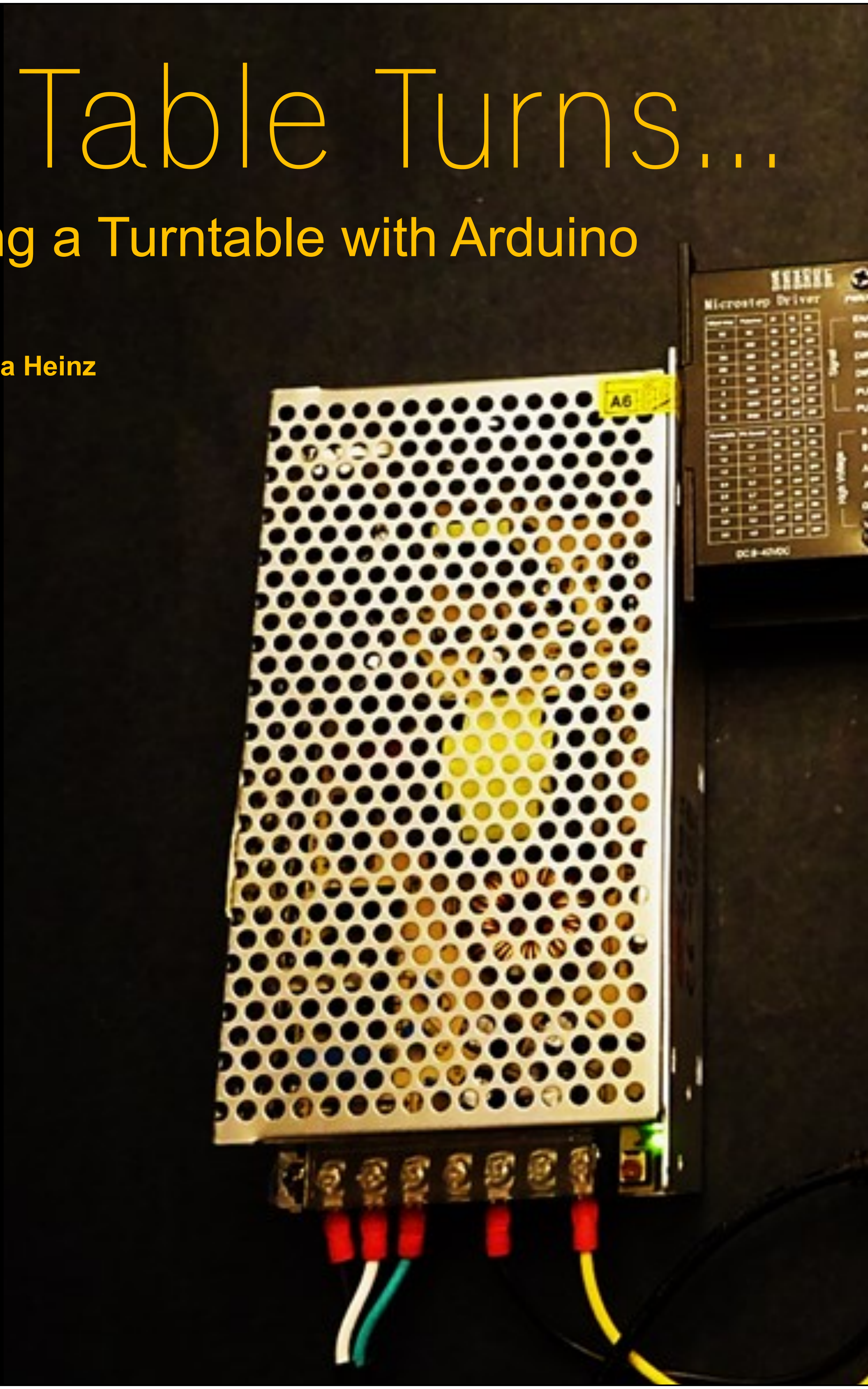


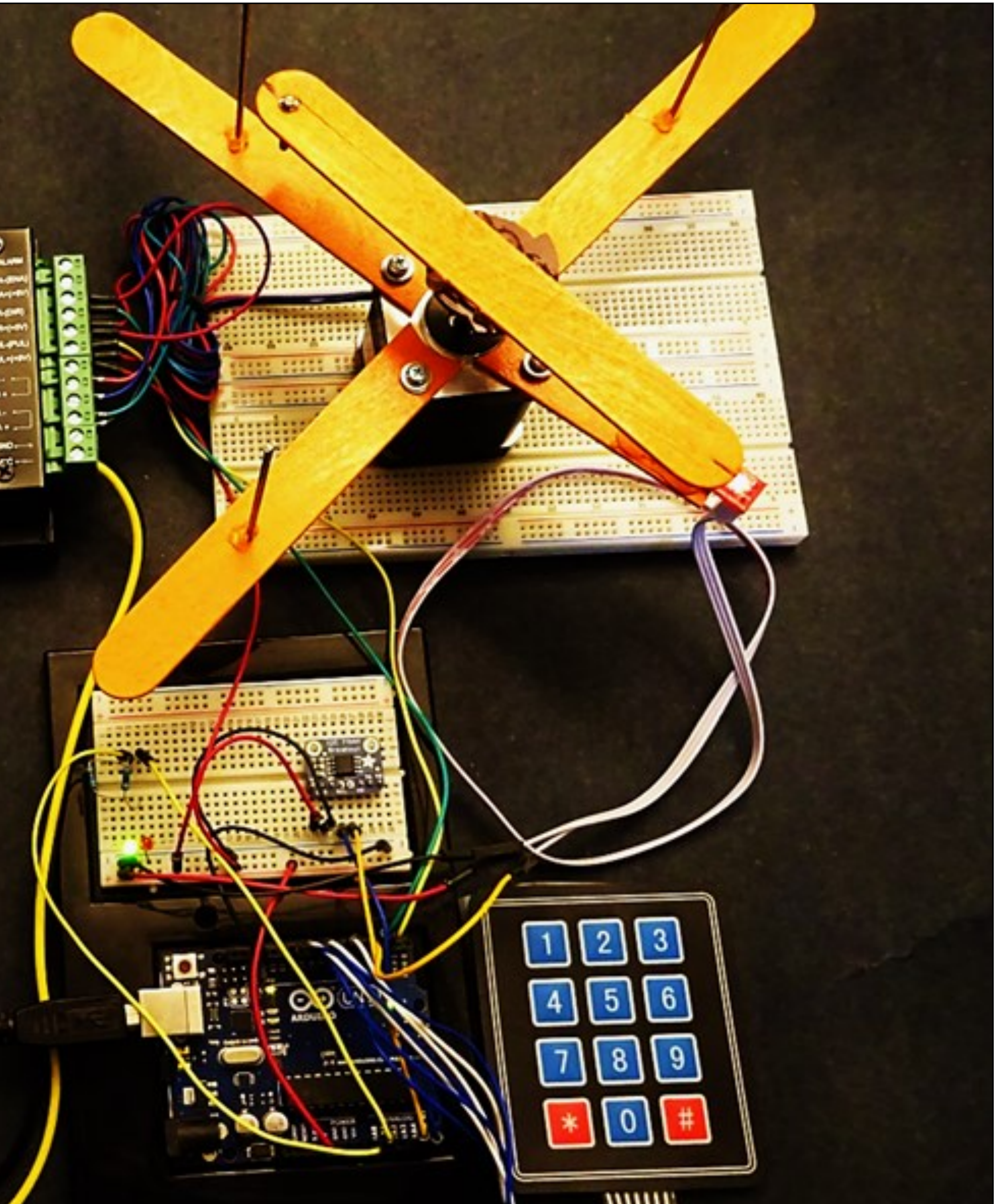
Photograph by Chris Heili

The Table Turns...

Automating a Turntable with Arduino

By Greg and Donna Heinz




1 2 3
4 5 6
7 8 9
* 0 #

For years, modelers have looked for ways to integrate electronics into their layouts in order to bring realism to their railroad operations. Despite the electronic age in which we live, this is often difficult to do, as there are few practical off-the-shelf components that are available – especially for the narrow-gauge modeler. Searching to automate the turntable on our D&RG-inspired layout was no different. The off-the-shelf systems that are available are either too complicated to use or cost prohibitive for a modeler who is building a small empire on a shoestring budget.

After carefully researching the few turntables that are available for the narrow-minded modeler, I chose to purchase two narrow-gauge HO turntables from Kitwood Hill Models in the UK. I found that they most closely matched the era and line after which our freelance railroad is modeled. However, there is no operating system or components to match.

The motor provided with the turntable is a straight DC motor which would be too fussy to use because there is no way to set exact stops. Thus, I will keep it in my parts file in case I find some other project for which it is better suited. Choosing a more desirable motor for my turntable was easy. A stepper motor is the motor of choice for this application because it has a 360° range of operation. It would also fit easily underdeck, and unlike a servo motor where there is chatter from the meshing of gears, it would operate virtually silent. I chose a *Nema17* stepper motor because I wanted something with some torque.

For my configuration (see the FRITZ Turntable Setup Diagram and the parts list), I wanted a motor controller that could do micro-stepping, as well as would be adjustable for current. This would allow for the widest selection of motors.

For this application, I chose a *TB6600 Single Axis CNC Micro-Step* stepper motor driver/controller. I set the controller to 6,400 steps and a current rating of 2.0 Amps, as the stepper motor uses 1.7 Amps. The stepper motor controller setup is achieved by looking at the two tables printed on the top of the controller (see Figure 1), and setting the two sets of three switches (see Figure 2) to the appropriate positions as indicated in the table.

The next step was to find an indexing system for my turntable. The indexing system is the operating program that allows the computer to find a certain point when asked. In this case, the indexing system will be used to locate the right track position turned to the desired end of the turntable. Most turntable indexing systems are designed to operate as part of a specific brand or model of a standard gauge turntable. Others are sold together with a turntable as a complete package. There was only one indexing system that I could find available for a narrow gauge turntable, but it is based on outdated technology (1996), and it seems to be

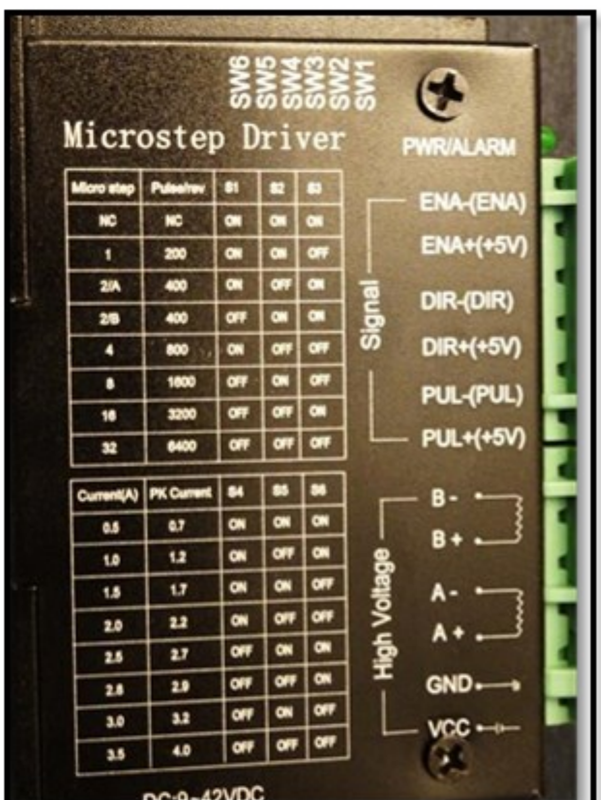
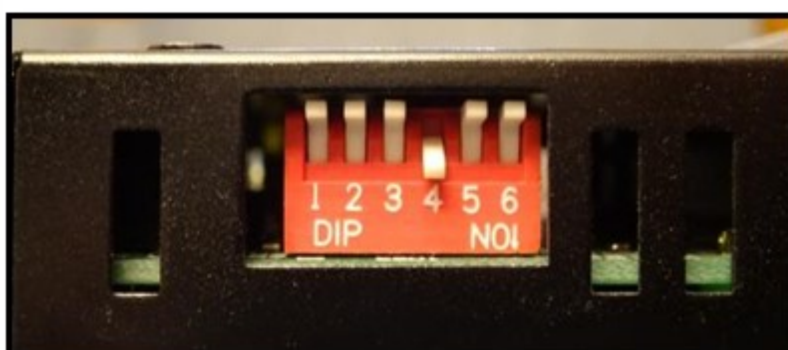
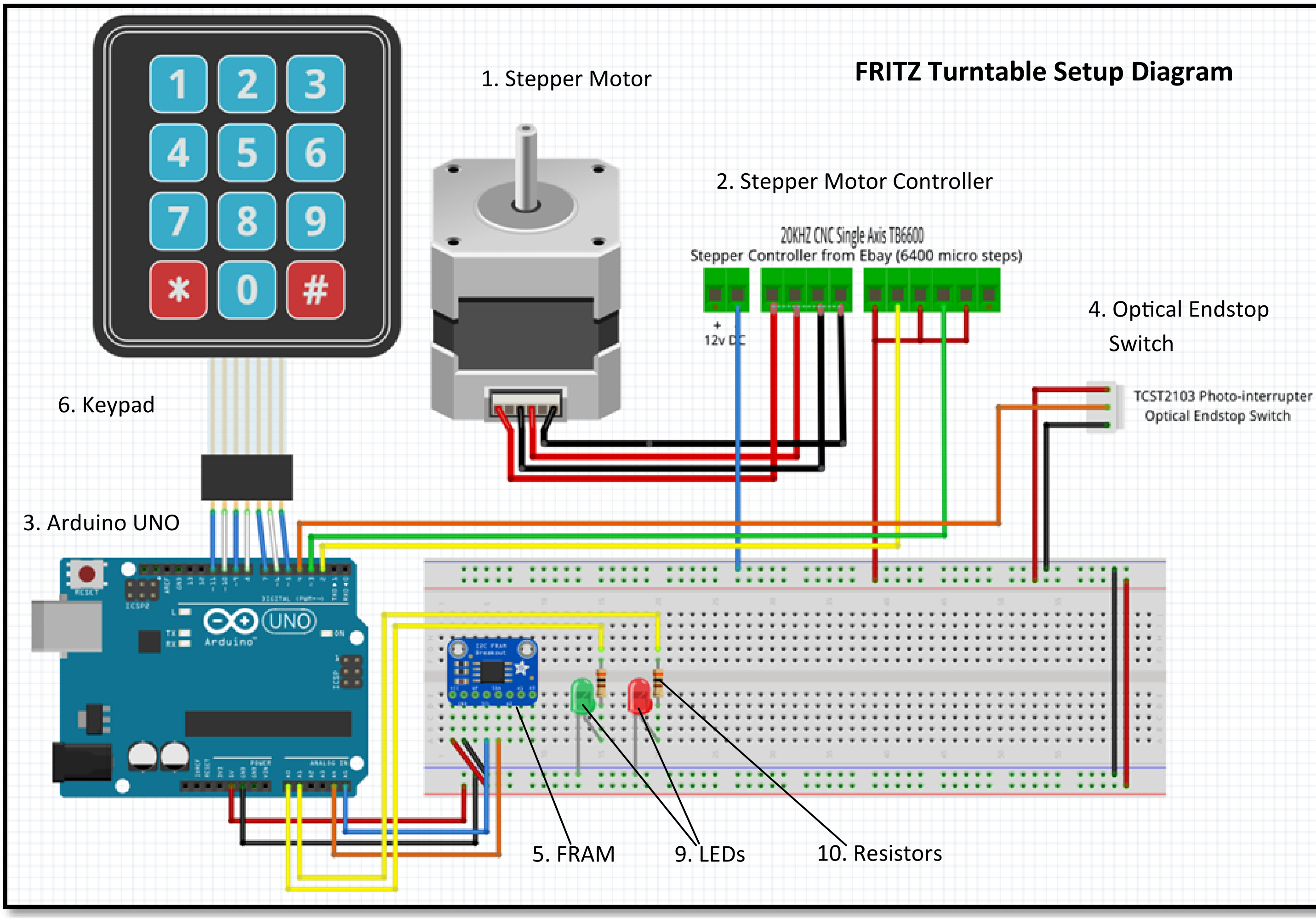


Figure 1 (left). Stepper motor driver configuration matrix.

Figure 2 (below). Stepper motor configuration switches.





Parts List:

1. Nema17 Stepper Motor 1.7 A 0.59 Nm 84 Ozin: This company is based here in the US. I needed 2 motors and I purchased 3 for $25.

2. 20KHZ CNC Single Axis TB6600 2/4 Phase Hybrid Stepper Motor Driver Controller DI: I used this controller as it could handle any stepper motor and supported micro stepping of 6,400 steps.

3. Arduino UNO: It was what I had on hand at the time, but while I don't need the form factor, I'm thinking of moving to an Arduino MEGA, or another processor. I'm very close to maxing out the memory of the Arduino UNO.

4. 5 pieces Optical Endstop Switches - RepRap Mendel Prusa RAMPS V1.4 Arduino 3D Printer.

5. Adafruit I2C Non-Volatile FRAM Breakout - 256Kbit / 32Kbyte: I picked this memory chip so that I can write to it over and over again.

6. Membrane 3x4 Matrix Keypad + extras - 3x4: I will be switching to the 3x4 Phone-style Matrix Keypad.

7. 12V, 10A, 150W LED Switching Power Supply Transformer.

8. External 9-12v ("wall wort") power supply to drive the Arduino UNO.

9. Red and Green LEDs.

10. Two 300 Ohm resistors.

11. Miscellaneous hook up wires.

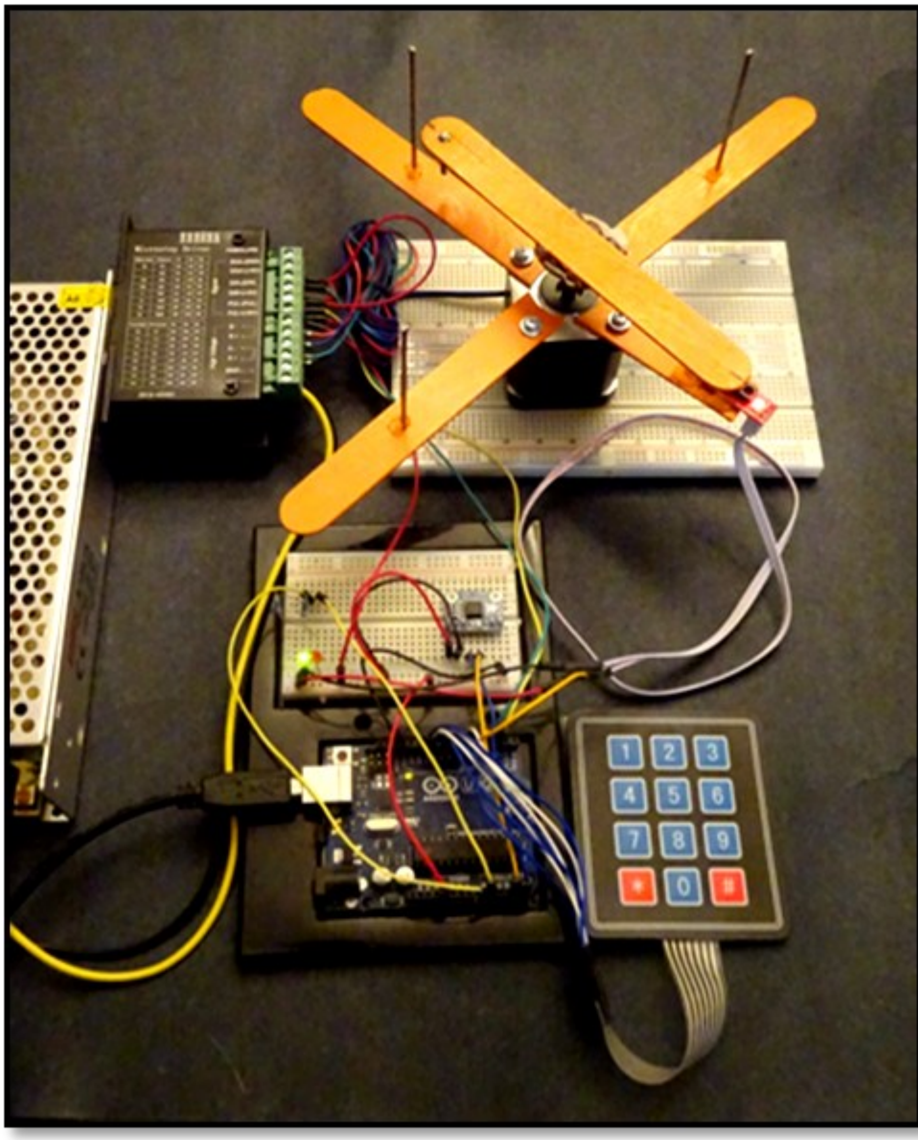


Figure 3. Mock turntable setup.

much more complicated to operate than needed for the task. On top of these drawbacks, this system has a 6-8 week turn-around time from the vendor and comes with a hefty price tag. None of these systems met the criteria I had set for myself.

The criteria I set for my system is:

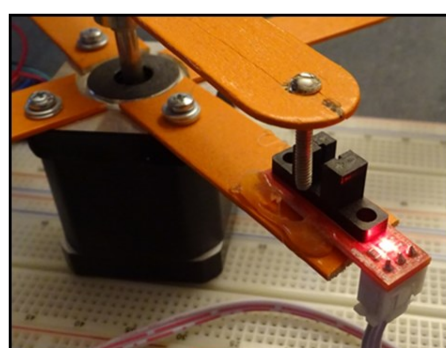
- ✓ Uses only one controller, not multiple buttons
- ✓ Capability to manage a large number of stalls or tracks (up to 25)
- ✓ Must work in both clockwise and counter-clockwise directions
- ✓ Able to select the shortest distance to turn the table
- ✓ Able to find the exact stop alignment every time (repeatability of alignment is critical)
- ✓ Must remember the last stop whenever turned on

The only remaining option was to build my own circuit by hand and program it to do only the tasks needed, as precisely as possible. I found this challenge the most appealing, as I could get just what I needed and program it to function exactly the way I wanted.

In my quest to find a controller for the turntable in my main yard, I researched a lot of mini-computers, along with different systems and components, many of which were woefully inadequate for my application (such as drivers that couldn't drive the *Nema17* stepper motor I had selected). I also watched dozens of YouTube videos on what others had done to control their turntables, and what issues they had run into. I didn't find any that actually ended up with a completed system.

I had been playing with Arduino boards for years, and therefore, it was a no-brainer. It was clearly the platform of choice for my application. For those who are not familiar, Arduino produces affordable, open-source microcontrollers and the programming environment needed to operate them. Unlike a minicomputer, which is not only costly but overkill for the task, an Arduino is designed solely to control hardware; to do one task and do it exceptionally well.

Figure 4. Optical sensor for finding the "0" stop position.



My next task was to find a stepper library to drive my Arduino. The library contains the blocks of code that tell the hardware what specific functions to do and how to execute them. I got online and researched different libraries to determine what each would allow me to do. I needed a library that would allow for acceleration and deceleration. In addition to the stepper library, I needed a FRAM library (Ferroelectric RAM: a new, non-volatile data storage chip, like the flash drive for a camera. It is dynamically fast and incredibly stable.) I needed another library that would allow the Arduino to read the keypad I would be using to operate the turntable. Lastly, I created a library which enabled me to blink the LEDs without using delays.

Once I had selected my stepper library, it took some time to learn how to use it. I searched through forums for insight on how to operate it effectively. The library I chose was written for operating a CNC (Computer Numeric Control) printer, router or plotter, so the biggest challenge was taking a library designed for driving things in a straight line, and making it think in a circle. I had to convince it that step points 0 and 6,400 (in this case) were the same point. I did this by writing my own function to help it figure out "which way?" was the shortest way to go the distance. I saved the last position by writing it to the FRAM so it always knows from where to start.

To learn and practice with my library, I needed a way to interface with it. I didn't want to use my final turntable, as I needed something I could tweak and mark up while I worked out any kinks. So, I created a quick mock-up using tongue depressors my wife had painted orange for a kids program she leads at work. I attached one to the shaft of the motor to represent the bridge of the turntable. I screwed four more to each corner of the motor to represent four radiating "tracks." On one of these, I located an optical sensor that allows the Arduino to always find the location of "0" stop point (see Figures 3 and 4).

I was able to map where tracks 1, 2, and 3 are located down to within 0.05625 degrees, or 1/6400th of a 360° radius. That's a fine enough increment that each position can be tweaked to meet the spurs that radiate out from the turntable, well within the tolerances of any locomotive. The stops on any D&RG narrow gauge turntable work out to be 8°38'15.6" apart – or 153.5 steps. To stay "on track" with my final turntable installation, I will have stops at about every 153-154 steps. Taken to an extreme conclusion, this would allow for up to one incoming track and 40 spurs. When complete, mine will have only 11.

To communicate with my Arduino, I chose a membrane 3x4 matrix keypad, because it was what I could find locally. Although I'm using this for now, in the final application I'm going to switch to an old-fashioned telephone pad, simply for the tactical feedback. This is all the control I need to operate my turntable. The Arduino needs two pieces of information before turning the table; the track ID and which end of the bridge needs to be there. Since the bridge on my turntable has an air motor deck on one end, I will use this to identify which end is which. The **#** symbol on the keypad identifies the back end of the bridge, and the **∗** symbol on the keypad, the front. To select the track I want, I simply type **4#** to turn the back end of the engine so it can be backed into stall 4. **11∗** will turn the front of the engine to continue down spur number 11. There's no need to have a separate control button for each track and another for each end. It's intuitive and beautiful in its simplicity.

Note: While writing locations into my FRAM library, I was getting some strange numbers coming back. After combing through the code line by line, I discovered a flaw in the library and was able to correct it. I have since notified the library owner, so (hopefully) it won't be a problem for others in the future.

The steps of code for my library look a little like this: Define the libraries used, declare and initialize the variables, set up parameters for reading the FRAM, set up the keypad, set up the LED blinker's parameters, configure the stepper motor, check for FRAM, read the FRAM, set up the keyboard parameters, set up the stepper motor parameters, find home position for the turntable, the main program loop, and define special functions (such as keypad event handler, get stub data, find home position, etc.). (See Figures 5 – 7.)

Before installing my turntable onto

TurnTable8 §

```
//********************************//
// FRAM Memory Layout...
// Data is stored as a word byte
// Address0 # of Stubs stored
// Address1 Last Posistion of deck
// Address2^n Stub Data stored format (location, front, back)
// if Address0 = 99

// Pre-set memory
memset(stubLocation, 9999, sizeof(stubLocation));
memset(frontCount, 9999, sizeof(frontCount));
memset(backCount, 9999, sizeof(backCount));

readFram();
```

Figure 5. Code sample for managing FRAM memory.

our layout, I will create a wiring diagram so I can have a custom circuit board made to replace the project breadboard I am currently using. This would also minimize the amount of loose wiring needed and make for a much neater installation. In order to add additional functionality (such as more than 25 stops), I would need to go to a larger Arduino, as this application is already coming close to maxing out the memory limits of the UNO. Any enhancement, such as an LCD readout panel, would require another library, which would also push beyond the memory capacity for this unit. I would like to build in a way that allows erasing of the stops so a new set of stops can be entered. Currently, the reset is a separate program that gets loaded into the Arduino UNO to reset the FRAM.

While this project has taken a lot of research, along with some troubleshooting, it should be much easier for the second turntable on our layout, which has only three stops. I am finding the challenge fun and interesting as I am learning how powerful an Arduino can be. I am already thinking of other places on our

Figure 6. Function for finding the home position on the turntable.

TurnTable8 §

```
void findHomePosition()
{
  //Defines the actual position to be the new zero position
  // May not be called while stepper is still moving
  stepper.moveTo(maxStepcount);

  while (digitalRead(homePositionpin) == LOW)
  {
    stepper.run();
  }
  Serial.print("CurrentPosition just Before Home Reset: "); Serial.println(stepper.currentPosition());
  stepper.setCurrentPosition(0);  //Must be done together to set Zero Correctly
  stepper.moveTo(0);              // "" "" "" "" "" "" "" "" "" ""
  stepper.run();                  // "" "" "" "" "" "" "" "" "" ""
```

Figure 7. Function for turning the bridge during programming of stub locations.

TurnTable8 §

```
void moveBridge(char keypressed)
{
  // This Function is used to move the Bridge
  // During the Stub setup process

  switch (keypressed) {
  case '1':
    nextStep = -1;
    stepper.move(nextStep);
    printFlag = false;
    break;
  case '3':
    nextStep = 1;
    stepper.move(nextStep);
    printFlag = false;
    break;
  case '4':
    nextStep = -100;
    stepper.move(nextStep);
    printFlag = false;
    break;
  case '6':
    nextStep = 100;
    stepper.move(nextStep);
    printFlag = false;
    break;
  case '7':
    nextStep = -1000;
    stepper.move(nextStep);
    printFlag = false;
    break;
  case '9':
    nextStep = 1000;
    stepper.move(nextStep);
    printFlag = false;
    break;
  }
  stepper.run();
}
```

layout where an Arduino can add realistic animation and functionality, such as the shaft and tunnels of the cut-away mine my wife is now planning. I would encourage anyone to look into Arduinos as a made-to-order microcontroller for any railroad animation project.

About the Authors

Greg, a senior network administrator for a large corporation, is a computer geek from the '70s. He fell in love with model railroading as a boy, watching his father build a small basement layout, and then later, fell in love with Donna – his wife.

Donna grew up hearing stories of the railroad from her grandfather and father. The former was a 40-year conductor and yardmaster for the Milwaukee RR, and the latter, a Gandy Dancer on the section.

The couple both had other hobbies when they met but model railroading has been the perfect shared hobby for them both.

Donna, a residential architect for 30 years, handles the design, structures, and landscaping, whereas Greg does all of the operations, track work, lighting, locomotion, and rolling stock. Thus, it has been the perfect joint hobby for the narrow-minded pair who grew up and now live in a suburb south of Minneapolis, MN. They have two children ages 23 and 25.

A SPROG Solution

Operating Trains with Raspberry Pi, JMRI, and SPROG DCC

"Once upon a time in a land far, far away called Australia, there was someone who thought he could write an article for Americans. ☺"

By Martyn Jenkins

All photographs by Martyn Jenkins (unless indicated).

Ever since I got involved with DCC in the late 1990s, I have wanted a fully-functional, wireless solution to control my layout. Most of the DCC manufacturers offer a range of radio-based systems, some with more functions than others; but the cost of these in Australia makes them an uneconomical option for me. None of these systems offer the ability to display information in the same manner as the Wi-Fi throttle application found in *Java Model Railroad Interface (JMRI)*, which can be used in conjunction with a mobile device.

The Miniature Train Club – Gold Coast (our local model railway club here on the Gold Coast, Queensland, Australia) exclusively uses Wi-Fi throttles to operate the club layouts at exhibitions. (See Figure 1.) We decided to use this system as we have 11 different DCC systems used by members on their home layouts. So, if we decided to pick a brand of DCC system from our members' preferences, we would be opening a Pandora's box as to which brand we use.

All of our members have either Android or Apple smartphones, so a Wi-Fi throttle was a no-brainer and stopped any arguments over which system we should purchase. The club decided to use a Wi-Fi throttle and a Digitrax Command Station, as we already had a Digitrax DCC system.

What is JMRI?

JMRI is a software suite consisting of several programs. The most popular programs in this suite are *DecoderPro*® and *PanelPro*®. One of the strong points of this software is that it is free to download and use from the JMRI page at the Source-Forge website. JMRI has been developed by a worldwide team of volunteers over several years who themselves are model railroaders.

DecoderPro was the first program developed by the team. It was designed to allow easy programming of DCC decoders via a *Graphical User Interface (GUI)*. This makes it easier to set up and fine-tune DCC decoders without having to know the specifics of configuration variable (CV) numbers and their numeric values. If you are setting up decoders in several of the same types of locomotive, DecoderPro allows you to set up

Figure 1. A Miniature Train Club member using a Wi-Fi throttle.

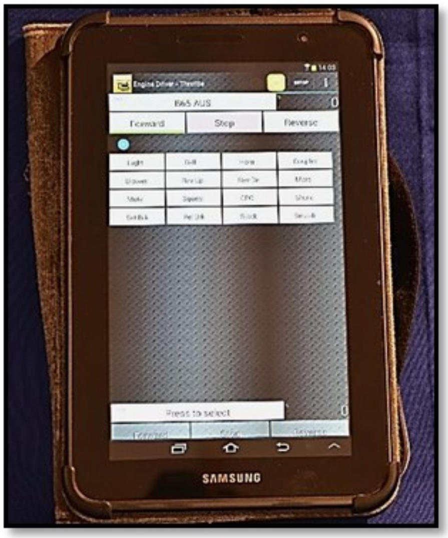


Figure 2. The Engine Drive app running on an Android smartphone.

one locomotive and then save the settings so you can load them into the other locomotives without re-entering them. Many people using JMRI use only DecoderPro, making it the preferred computer software for setting up decoders amongst the model railway community in Australia.

PanelPro was developed to allow for the creation of *Centralized Train Control (CTC)* panels and track plans. PanelPro also allows you to control lights, turnouts, sensors, signals, and more. When using occupancy detection with sensors built into the layout, you can track the location of a locomotive around the layout on the CTC panel or on your track plan. Furthermore, since you can control turnouts from the CTC panel, you can then create track warrants.

JMRI also contains several other programs and utilities that are very useful in the operation and maintenance of a layout that has a DCC system installed. One of these features is the JMRI WiFi Throttle; this can be run directly within JMRI on your computer, or it can be controlled through a Wi-Fi throttle app.

What is the Wi-Fi Throttle App?

A Wi-Fi throttle app is a program that runs on a smartphone, iPod, or tablet, and communicates – through a computer running JMRI – with a locomotive on the layout. There are apps available for both iOS (Apple) and Android. For Apple devices, the app is named "WiThrottle", and for Android devices, you can use either the "Engine Driver" or the "DigiTrains" app.

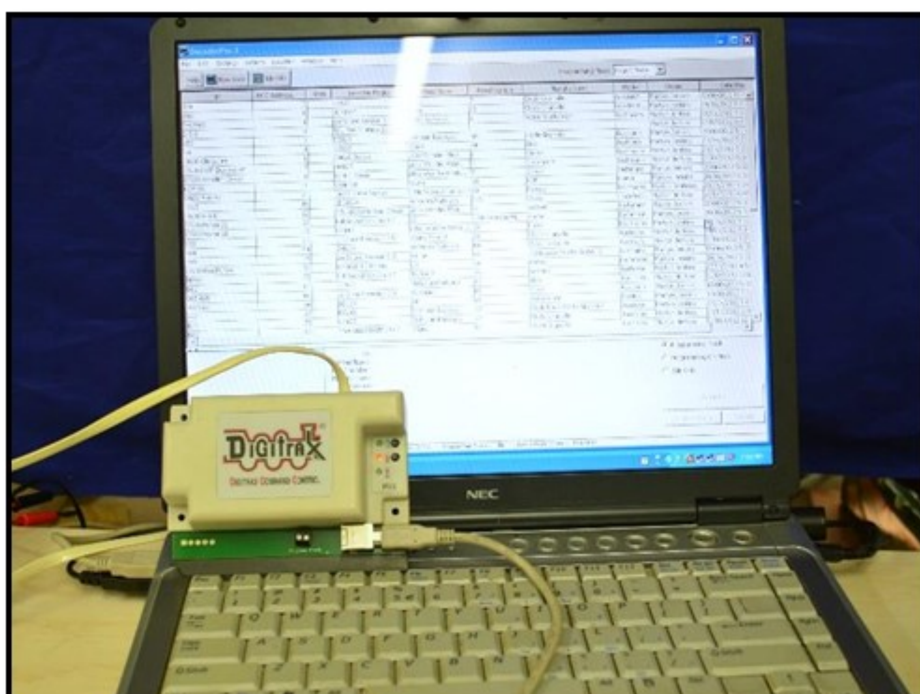
At our club, we have a mixture of all three apps running on members' devices. I personally prefer the Engine Driver app, as I have an Android device and have been using it for several years and it is completely free. (See Figure 2.) But lately, I have been using the DigiTrains app, since it is easier to move between the different functions of the throttle, such as going to the track plan to change a point and then returning to locomotive control and driving the train. It also appears to allow you to control more locomotives simultaneously than the Engine Driver App.

The WiThrottle app (Apple iOS) is available in two versions. The free version, called WiThrottle LITE, allows you to control up to two locomotives and that is about it. The full version is $12.00 AUD from the Apple App Store and gives you access to the stationary decoders, route control track, CTC panels, and much more.

Using the Wi-Fi Throttle App

If you want to use a Wi-Fi throttle app, the first thing you will need is a model railway connected to a JMRI-compatible DCC system. JMRI sup-

Figure 3. The Digitrax PR3 acts as a bridge between the layout LocoNet and a USB connection to the computer.



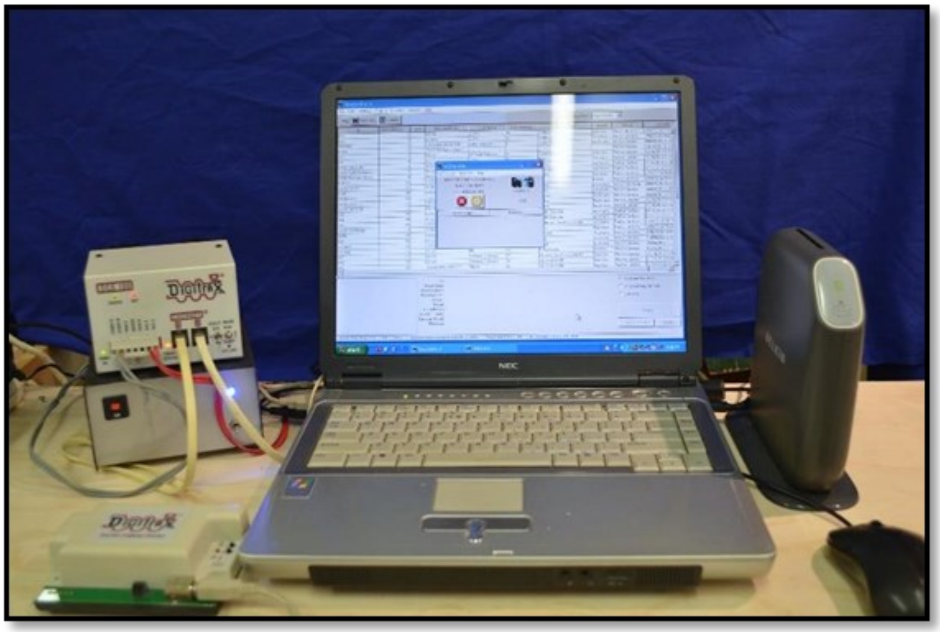


Figure 4. The complete initial setup!

ports nearly every brand of DCC systems from the obscure ZTC system to the popular systems such as Digitrax, NCE, and Easy DCC; support also includes do-it-yourself (DIY) systems such as DCC++ as well as the SPROG DCC product line.

Next, you need a computer running JMRI; this computer should be connected to your DCC system via a manufacturer-approved interface. For example, the LocoNet of a Digitrax system can be connected to a "LocoBuffer" or a "PR3" interface unit (see Figure 3), both of which provide a USB port that can be connected to your computer. On some systems, such as NCE, there is a serial port located on the DCC command station. The specifics will depend on the options your DCC system manufacturer has for this function.

Now you need to set up a wireless router to make WI-Fi throttle work. You could use your home network, but I have found that – with other people in the home streaming video from websites, such as Netflix and YouTube, and making other demands on the home network – a cheap, dedicated router for the model railway layout works best. Now, how do we connect the mobile devices that serve as Wi-Fi throttles through the router to the PC controlling the layout with JMRI?

First, you need to connect the PC to the router; this is done using the wireless network connection settings, which are dependent on the type of computer operating system you are using. Then, you connect your mobile device to the router in the same manner as you would use to connect it to your home network for internet access. (Make sure your wireless router is password-protected so other people cannot connect to the PC and, therefore, your model railway.)

We are nearly there! All we have to do is make sure the PC is running JMRI, start the Wi-Fi throttle app on our mobile device, and connect to our model railway. (See Figure 4.)

Too Much Technical Stuff?

Our model railway club has an average age of 65 years, so we do have quite a few members who cannot set up the system due to their limited computer knowledge. For example, the old way we set up our exhibition layouts was so complex for some of our members that certain members had to be in attendance to set up the system and monitor it over the weekend or we could have problems. What we wanted was a method of using a Wi-Fi throttle app that was simple: just plug in the power, start the app on your smartphone, connect and – *hey presto* – you are driving trains without:

- setting up a PC,
- setting up JMRI,
- setting up a wireless router, or
- setting up a DCC system.

A Better Way

Well there now is a solution out of the UK that uses the Raspberry Pi *Single Board Computer (SBC)* and a new DCC system from SPROG.

The SPROG II (see Figure 5) allows

Figure 5. The SPROG II.

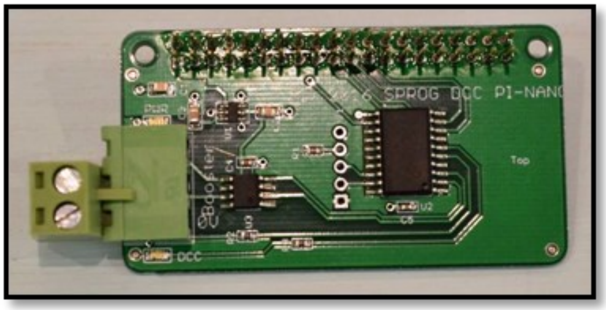


Figure 6. The Pi SPROG Nano board.

you to operate a layout directly from JMRI, using either the computer-based throttle found in JMRI or a Wi-Fi throttle app; this device has been available for a number of years. It has a USB port that will connect to a PC running JMRI, and it also has connections that go to your track. After connecting the system, you can program any type of DCC decoder and even run a small layout. This little device will provide 1A of track power, while its big brother (the SPROG III) provides the same features, but with 2.5A of track power.

The people who developed the SPROG family of USB-to-JMRI interfaces now have designed a board – named the Pi SPROG Nano (see Figure 6) – that plugs directly onto the general purpose input/output (GPIO) pins of a Raspberry Pi 3.

What is a Raspberry Pi?

The Raspberry Pi family is a series of small, *Single Board Computers (SBCs)* developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. (See Figure 7.) The original model became far more popular than anticipated, selling outside of its target market for uses such as robotics and model trains.

Since the introduction of the Raspberry Pi computer in 2012, the price has dropped substantially, and a number of add-on devices – such as cameras – have been developed. The price of the current version, known as the Raspberry Pi 3, is only $65.00 AUD. This now makes it cost effective to install a dedicated computer (a Raspberry Pi) running JMRI – along with an attached Pi SPROG Nano DCC controller (see Figure 8) and a [Tam Valley Depot 5A booster]() – in each of our exhibition layouts! Another added benefit is that the club can leave its primary layout DCC systems on the club layouts and not disconnect them just so we can take them to a model train show.

I got my first Raspberry Pi about three years ago when they first released the Raspberry Pi B model, and I have been using one along with a SPROG II DCC unit. This was better than the old method with a complete DCC system, but it still required some knowledge to set up.

The newer Raspberry Pi 3 has 1GB of RAM and can run an operating system similar to Microsoft Windows. It can run JMRI, and furthermore is Wi-Fi and Bluetooth capable. Now, we can do away with the wireless router, instead, we can connect the Wi-Fi throttle apps directly to the Raspberry Pi 3. That's one less thing to set up at a show!

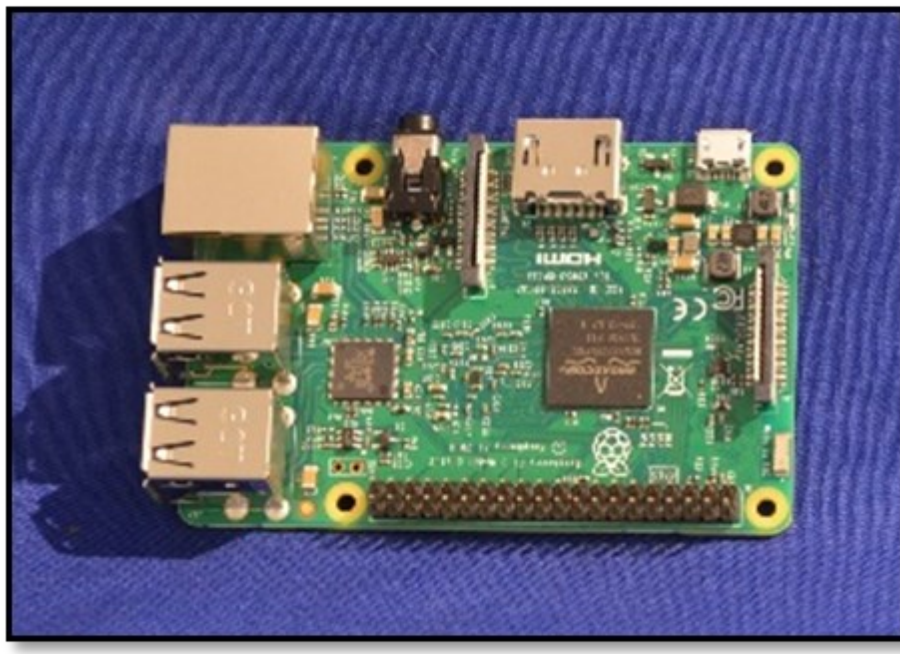


Figure 7. A Raspberry Pi Single Board Computer (SBC).

A New Way of Using Wi-Fi Throttle in JMRI

SPROG DCC in the UK actually has two recent boards that plug directly onto the Raspberry Pi 3 GPIO pins. The first is the Pi SPROG Nano (described above), which provides a DCC signal that is ready for a booster. This is the board our club now uses along with a Tam Valley boost-

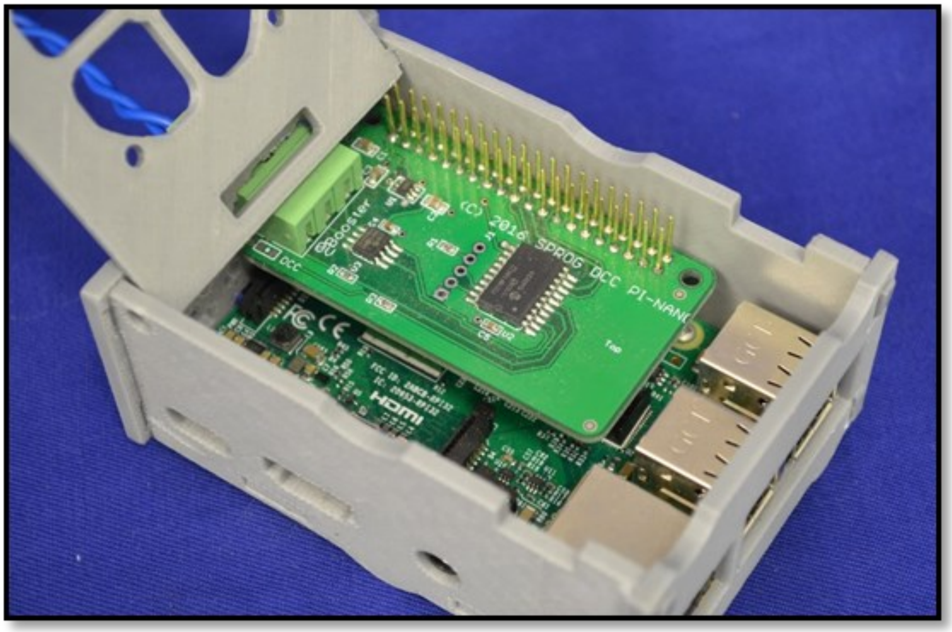


Figure 8. A Raspberry Pi wearing a Pi SPROG Nano, all within a 3D-printed case.

er to power the rails. It has run at the last three shows we have attended without requiring any attention. The other product is the Pi SPROG One (see Figure 9); it also plugs onto the GPIO pins of the Raspberry Pi 3, and it provides a 2.5A DCC track output, which should be sufficient to run smaller layouts without the need of an additional booster.

Finally, we now have – in a package (See Figure 10) no bigger than a deck of cards – a PC running JMRI, a wireless router, and a 2.5A DCC system. And we can set up this system to start the Raspberry Pi 3 and JMRI automatically; all we have to do is configure (only once) the software loaded on the Raspberry Pi 3.

Now some people's eyes will start to glaze over when I start to talk about setting up software, especially if they are not familiar with it. So, the good folks at SPROG DCC will provide (for a minimal cost) a preformatted micro SD card with the operating system for the Raspberry Pi 3 and JMRI already installed and pre-configured. Just insert the micro SD card into the card reader slot of the Raspberry Pi 3, power up the computer, start the Wi-Fi throttle app on your mobile device, connect wirelessly, and you are now driving trains!

If you are computer savvy, you can, of course, set up the Raspberry Pi 3 and JMRI yourself. SPROG DCC has provided a detailed description of what is required to set up the system on their website, along with details on how to set up the system with previous versions of the Raspberry Pi and a SPROG II or III.

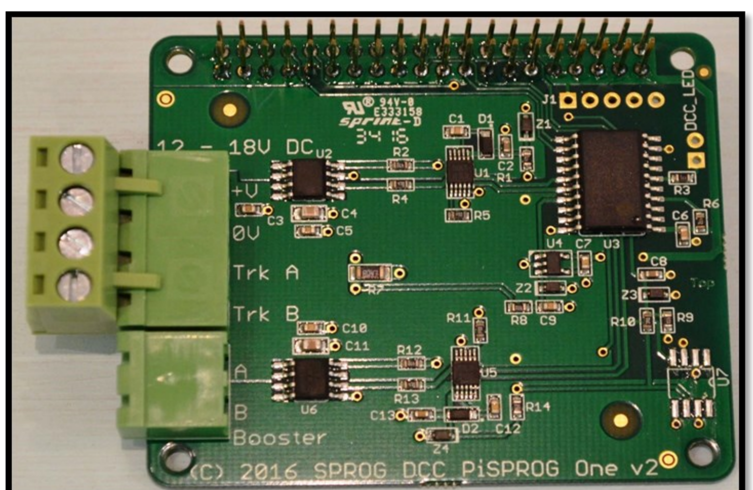
Being as this is such a small system, we could mount this into one of our club's layout modules. Then, all you would need to do is assemble the modules, connect two power supplies to the layout module that has the Raspberry Pi 3 mounted within it, start your mobile Wi-Fi throttle app, connect to the system, and start enjoying driving trains. This is what we have done in our club.

Since the beginning of this year, our club has been using the Raspberry Pi 3 / SPROG DCC system, and we have found the wireless range to be over 500M (1,640 feet). So, unless you have a layout that is over half-a-kilometer in size, you should not have a problem with the Wi-Fi signal. If you do find that you need additional coverage – say for a large G-scale garden railway – you can use a commercial wireless repeater/booster available at any PC or electronics store, or online.

What about the case?

Now some people will want a case to hold all this technology in one neat, protective package. There are several cases available on eBay and

Figure 9. The Pi SPROG One board.



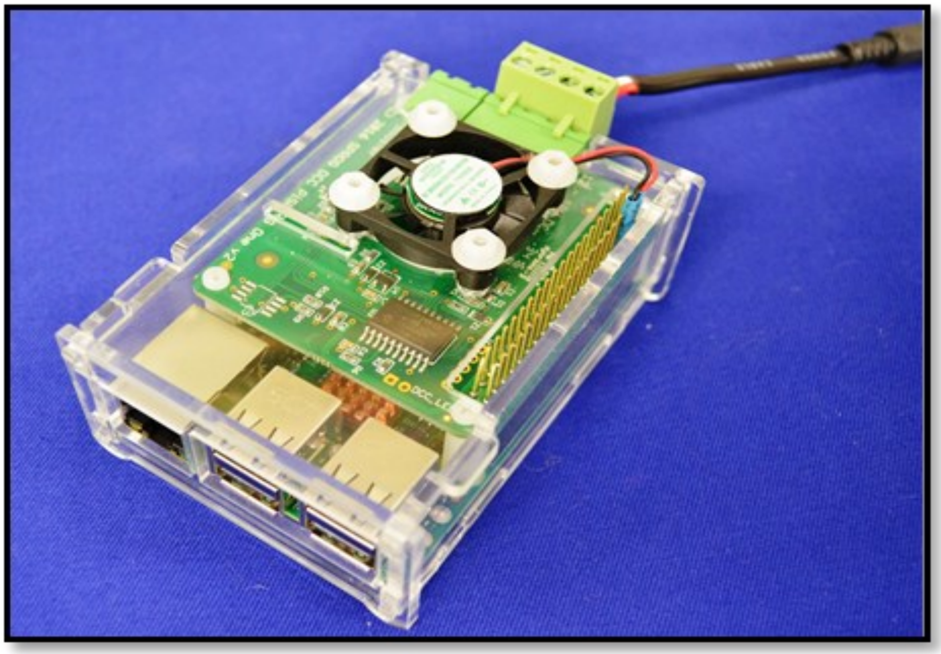


Figure 10. A Raspberry Pi 3 with a Pi SPROG One board attached, all mounted inside a clear laser-cut case I purchased on eBay. The whole unit is Wi-Fi ready and just needs to be plugged in to a power supply.

other sites online. When shopping for a case it is important to ensure it is large enough to hold both the Pi SPROG board as well as the Raspberry Pi 3. (See the case in Figure 10, above.)

Since I have a 3D printer, I looked up a design on the Thingiverse website for a Raspberry Pi 3 case and modified it to suit the Pi SPROG boards. (See the case in Figures 8, above.) Now I can print off as many as I require.

Conclusion

I hope this article has inspired you to consider trying this DCC system option. As you can see, it is relatively inexpensive as compared to branded DCC systems. Plus, since mobile device manufacturers bring out new models of their products every few months, there are a plethora of spare, inexpensive mobile devices to be had from eBay and other sources.

Next year, I plan to attend the NMRA convention in Kansas City, MO USA. As these devices are so small (thereby allowing me to bring them from the land down under on the big silver bird), I will try and do a couple of clinics at the convention on this technology.

Useful Links

- SPROG DCC Ltd
- Pi-SPROG
- JMRI SPROG Set Up
- JMRI WiFi Throttle
- WiThrottle App
- Engine Driver App
- DigiTrains App
- Raspberry Pi Set Up
- Raspberry Pi Case
- Tam Valley Depot DCC Booster

About the Author

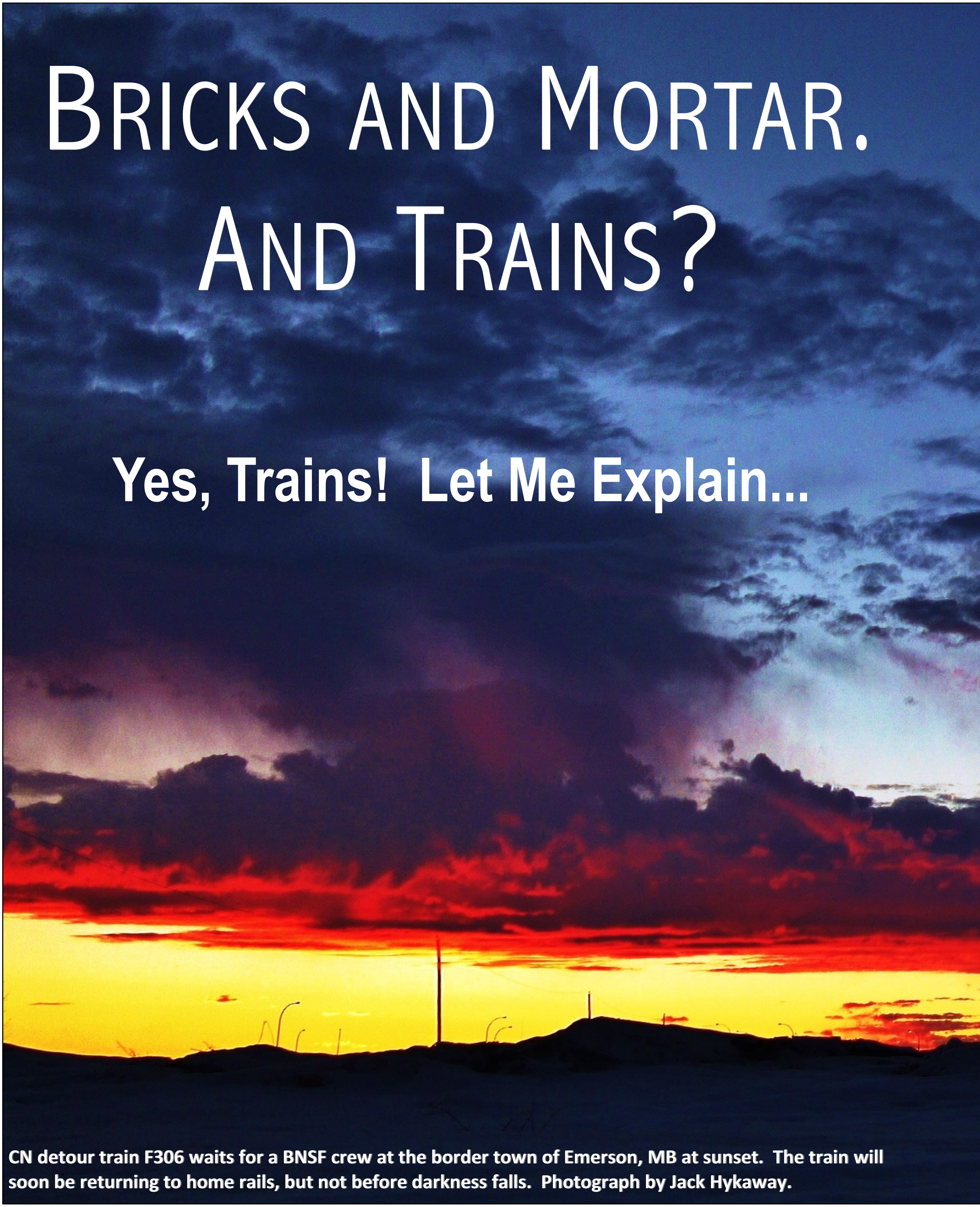
Martyn Jenkins is located on the Gold Coast in Sunny Queensland, Australia with his wife of 23 years, Mary. He is an avid model railroader, modeling the BNSF modern era in N Scale, Victorian Railways, or V/Line (an Australian prototype) in HO scale, Puffing Billy narrow gauge (another Australian prototype) in On30 Scale, and a G scale garden railway. As he says, "it's not a hobby it's an addiction." Martyn also is the divisional superintendent for the NMRA Australasian Region looking after division 1 Queensland, which is the largest division in Australia with over 200 members.

With a background in the electrical industry, Martyn has worked for Honeywell, Landis & Gyr, Westinghouse, Nissan, and others. He now teaches electro-technology subjects at Gold Coast Institute of Technical and Further Education, as well as works as an electrical inspector with the Queensland government.

Martyn's YouTube channel has additional information on the Raspberry Pi and a host of other subjects; check it out and subscribe.

Bricks and Mortar. And Trains?

Yes, Trains! Let Me Explain...



CN detour train F306 waits for a BNSF crew at the border town of Emerson, MB at sunset. The train will soon be returning to home rails, but not before darkness falls. Photograph by Jack Hykaway.

Jack's Junction

By Jack Hykaway

In the classic fable *The Three Little Pigs*, a big bad wolf blows down the first two pigs' homes – made of straw and sticks respectively – but is unable to topple the solid walls of the third pig's home, which were constructed of brick and mortar. The bricks the mortar work together to create a sturdy product; build a brick wall without mortar and it would topple in the wind, and we all know it would be a waste of time to build a brick wall without the building blocks themselves. Therefore, if either element was excluded, there would be no wall at all.

Broaden your perspective from our modest brick wall to the entire North American railway network, and you'll find that the two are oddly comparable. In this analogy, the bricks represent the cities and the customers served by the railroad across the continent – forming the main structure of the network – while the mortar represents the trains themselves – the "glue" that connects the system together. Just like with our brick wall, the cities depend on the trains for transportation and the trains depend on the cities for traffic.

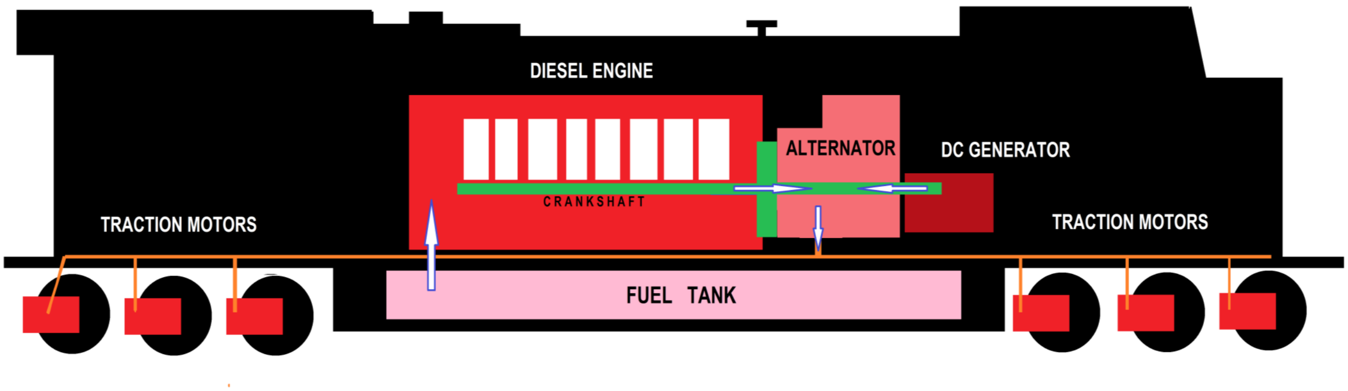
I want to focus on this "glue" in this edition of Jack's Junction. Specifically, how the rail network keeps the country and economy connected, moving goods and people from city to city every day.

To keep the trains moving, North American railroad companies depend on the power of internal combustion. The diesel engine at the heart of every locomotive is the workhorse of the continent's railroad industry; ferrying goods and people from coast to coast. This simple – yet essential – piece of technology affects each and every one of our lives on many levels; from the food we eat to the people we are connected with.

To understand this incredible piece of engineering, we must turn back the clock to 1897, when Rudolf Diesel built the first prototype. The basics of the diesel engine have stayed nearly the same since the day Rudolf Diesel completed the prototype.

Four-stroke diesel engines used in locomotives use a combination of air, heat, and fuel oil to create combustion. Following ignition of the previous cycle, the piston descends

CN detour train F306 pushes east in Winnipeg, MB. F306 was a detoured BNSF grain train, running through Canada to avoid the snowed-in BNSF line over Marias Pass in Montana. Photograph by Jack Hykaway.



A simple diagram of a train diesel engine. Diagram by Jack Hykaway.

and starts the combustion cycle anew in four unique strokes. Fresh air, forced in to the engine via a turbocharger, improves the efficiency of the combustion process.

Starting with what's known as the "suction stroke," the piston sucks fresh air into the cylinder via a set of inlet valves on the top of the cylinder. As the crankshaft rotates, the piston is forced back to the top of the cylinder, compressing and superheating the fresh air trapped inside. This is known as the "compression stroke." A mist of diesel fuel is then injected into the compressed air, and is ignited by the heat (and not by a spark plug). This explosion forces the piston to the bottom of the cylinder – this stroke, known as the "power stroke," is the only stroke where the piston is generating mechanical power. The fourth – and final – stroke is known as the "exhaust stroke," where – due to inertia from the fast downward motion in the "power stroke" – the piston moves upwards again. The exhaust valves are opened to release the exhaust gasses and the cycle starts again.

The inlet and exhaust valves are opened and closed at precisely the right moment using a pair of cam shafts – one shaft for the inlet valves and one set for the exhaust valves. Cam shafts run the length of the engine, with a teardrop-shaped attachment positioned over every valve. The shafts are driven by the main crankshaft via a timing belt, which interlocks with gears on the ends of the cam shafts. The camshafts are driven at ½ of the speed of the crankshaft to ensure the inlets are operating at the perfect time.

This ignition process repeats itself rapidly to produce a steady stream of mechanical power output. In four-cylinder four-stroke engines, one of the four cylinders is always in the "power stroke" phase. This means that the engine is always producing power at any given moment, and that the engine operates very smoothly. This is not the case for a four-stroke one-cylinder diesel engine: it is only firing ¼ of the time, and therefore only producing power ¼ of the time.

In the 4,400-horsepower, 12-cylinder, four-stroke EVO engine used in General Electric's new ET44AC locomotive, three cylinders are always firing to ensure plenty of power output at any given moment and a smooth-running locomotive. (See Jack's Junction, January 2017)

As large as it may be, the diesel engine in a railway locomotive is only half of the story. The other half involves electricity – and lots of it. A massive alternator just in front of the diesel engine on board generates the power necessary to feed the four to six electric traction motors, which turn the wheels of the locomotive.

Alternators use electromagnetic fields to generate AC current for the traction motors. In English, that means that a rotating set of coiled wires (spun by the crankshaft of the diesel prime-mover) creates a magnetic field powerful enough to produce the electrical output necessary to move the wheels of the locomotive.

A typical alternator includes two sets of coils, known respectively as rotor and armature coils. Direct current (DC) runs through the wires surrounding each of the rotor coil's poles, thus creating a rotating magnetic field around the rotor. The initial supply of DC current is supplied to the rotor coils by a small DC generator (mounted next to the alternator), which also generates its power from the mechanical rotation of the crankshaft of the diesel prime-mover.

As the spinning rotor coils pass within close proximity of the stationary armature coils, an AC current is induced by the magnetic field. These rotors – known as Salient Pole Rotors – can have several dozen poles (depending on their diameter) to generate huge quantities of electricity while rotating at a relatively low RPM. Obviously, the faster the rotor is spun, the more electricity is produced and the more power the locomotive will put through the wheels.

A new General Electric ET44AC locomotive can produce up to 3,280kW of AC power from the unit's 4,400-horsepower prime mover. This power plant on wheels sends the huge amount of power to the locomotive's six traction motors, where the electricity is used to drive the wheels.

The traction motors on modern diesel-electric locomotives are mounted between the wheels around each axle. Traditionally, 600-volt DC traction motors were used. Thanks to power semiconductors, railroads began to experiment with AC-driven traction motors. Variable-frequency drives are used on AC-powered locomotives to allow for a wide range of motor speeds, a robust induction motor with less wear and tear than its DC counterpart, and a much simpler motor to build and maintain. For these reasons, AC motors are now favoured by North American railroad companies, and are widely-used on modern diesel-electric locomotives that polish the rails today.

In a nutshell; fuel is burned in a diesel engine to generate a DC power supply and to spin the rotor coil of an alternator. The alternator produces a large amount of AC power, which is fed to the locomotive's AC traction motors and instruments. The traction motors – mounted on the axles – use the electricity to turn the wheels, propelling the locomotive forward or backward.

Next time you see a train roar by on the mainline, listen to the deep roar of the locomotive and think of the incredible combination of technology that makes this huge machine hum. Appreciate the huge power generated by the diesel engine and the alternator by observing the thousands of tons of freight being moved across the country at 60 MPH. Just like our brick wall, the different elements work together to create a strong, reliable product.

About the Author

Jack Hykaway is a student, currently attending a post-secondary institution in his hometown of Winnipeg, Canada. He is an amateur videographer and writer and enjoys exploring and documenting nearby railroads and railroad operations in both written and visual formats of his work.

Jack joined the YouTube Model Builders e-Mag team as a content editor in 2013; his main focus of late has been producing his bi-monthly column Jack's Junction for the eMag.

Follow along with Jack's videography on his YouTube channel at https://www.youtube.com/user/WinnipegRailfanner1.

YouTube Model Builders LIVE! Want to see live shows discussing modeling techniques, YouTube, and Web resources?

Check out the LIVE show that airs monthly .

— Free to you!

For more information on diesel engines and their components, please consult these links:

- **Inside Track:** http://www.up.com/aboutup/community/inside_track/ge-tier-4-11-17-2016.htm
- **Cummins Engines:** https://cumminsengines.com/how-a-diesel-engine-works.aspx
- **Learn Engineering - Alternator:** https://www.youtube.com/watch?v=tiKH48EMgKE
- **Smarter Every Day:** https://www.youtube.com/watch?v=xflY5uS-nnw
- **GE Transportation:** https://www.getransportation.com/locomotive-and-services/evolution-series-locomotive
- **Wikipedia:** https://en.wikipedia.org/wiki/Diesel_engine
- **How Stuff Works:** http://auto.howstuffworks.com/diesel.htm
- **Diesel Trains | How Diesel Locomotives Work? | Locomotive engine production:** https://www.youtube.com/watch?v=JU6zpBT5Sh8
- **Detailed Diagrams and Schematics:** https://www.myodesie.com/wiki/index/returnEntry/id/3030%23Major%20Components%20of%20a%20Diesel%20Engine#Major Components of a Diesel Engine

The Trackside Photographer

If you are looking for a wealth of interesting railroad lore along the tracks: depots, freight houses, signals, interlocking towers, bridges, trestles, shops, turntables and other trackside structures and equipment, then look no further than *The Trackside Photographer*.

Visit *The Trackside Photographer* and explore the visual and cultural landscape that the railroad moves through, with photographs of trackside structures and scenery and writing about the history of sites that are rapidly changing, or have already disappeared.

Food For Thought...

DIY vs. DIBuy

By Andy Crawford

As modelers, there often is a tension in our decision-making process regarding the best way to reach some particular end.

There can be many variables that affect our decisions, but they often distil down to this: do I build it, or do I buy it? While there are other choices – such as what to buy or what to build – this seems to be a common and relevant starting point. It also is good (in my humble opinion) to establish early on your personal "threshold to building." That is, how far you are willing to go with a commitment to educating yourself or investing time, versus investing your hard-earned cash.

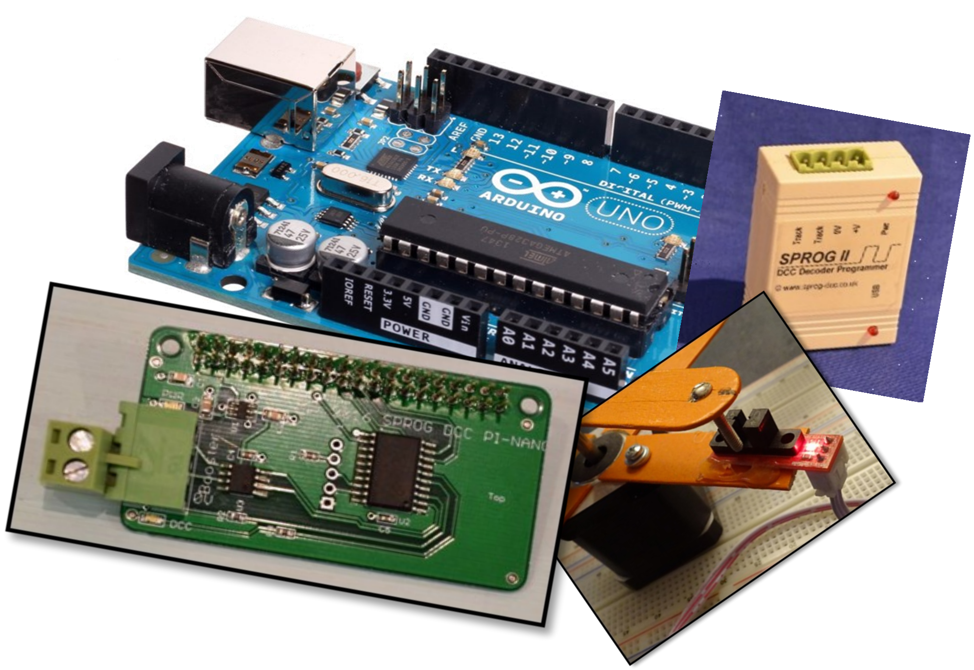
My friend Miles Hale has described his solution to this dilemma as, "If it's available, buy it; if not DIY" (my paraphrase). I agree, generally speaking. This nearly always is true of structures or components for your layout. It's difficult to justify to yourself the investment in casting a bridge abutment when many manufacturers make abutments of many types and of various materials.

There are exceptions to this rule, and we prototypers tend to encounter those quite often. For example, you might be in the pursuit of the perfect bridge pier, which more accurately reproduces the particular variances of how your chosen prototype did it; given the specific nuances, there may be justification for casting or carving this yourself. (This is a vast simplification of this decision threshold, but I believe it makes the point.)

However, the "DIY vs. DIBuy" argu-

Choosing to build a solution ourselves can lead to significant cost savings and a solution that is custom-built for the layout resulting in a raised level of self-confidence and personal satisfaction.

Image of Arduino UNO used under CCA-SA 2.0 Generic license. Other images courtesy of Martyn Jenkins and Greg Heinz.



ment really comes to a head when selecting the electronics and control systems that govern the operation of our railroads.

Many would conclude that ready-to-run solutions are available from Digitrax, NCE, and so forth to facilitate nearly any control requirement one might have.

Others may advocate a "roll-your-own" approach that integrates a variety of custom-wired, custom-programmed electronic components. I can't completely disagree with either conclusion. However, the two factors that affect these overly-simplistic views are flexibility and cost.

> ***"Two factors that affect our decision to 'DIY or DIBuy' are flexibility and cost."***

Flexibility: It's quite hard to beat the flexibility and degree of freedom that comes from building or assembling your own electronics solution.

You may select among Raspberry Pi, Arduino, PicAxe, Launchpad, or ESP chips. You get to choose how those components communicate with each other, be that via the RS485 protocol common in C/MRI deployments, or the IP (Internet Protocol) of Ethernet and Wi-Fi that we may use with ESP chips, Raspberry Pi boards, and – with less popularity – Arduino boards. Or one may settle on a more simplified connection, using I2c or SPI with Arduino, Raspberry Pi, etc. There even is the option of direct signaling between PicAxe and others.

An entire article could be dedicated just to communication protocols and methodologies. Personally, I've come to lean further towards Wi-Fi, which is infinitely flexible, reduces the infrastructure required, increases the distributed nature of modern microelectronics, and provides a cost benefit.

Cost: This point is slightly more difficult to argue, but must be a consideration. Very few of us have unlimited funds to invest in our modeling, nor infinite time, so we must argue with ourselves to resist running down rabbit holes so that we can get the most bang-for-the-buck.

The goal is to get to the result we desire, while still leaving funds and time set aside to invest in locomotives, structures, and the layout at large. So, I might suggest you keep this ever present in your mind as a sliding scale as you work through the decision process. It is important to remember that commercial products always will have decision points such as support, R&D, profit, etc. that are considered when they are priced. This may tilt the scale towards DIY as being more cost-effective.

Few of us treat our time as money, but it is. We have a finite amount of both and have to decide whether investing money or investing time is most appropriate in reaching our goals.

With Arduino Nano boards in the couple-of-bucks price range and ESP chips in the sub-two-dollar range – including Wi-Fi – DIY solutions that involve these products are hard to ignore. As a personal example, I had priced an NCE DCC solution (and I'm a pretty big fan of that product) at something in the $1,400-$1,600 neighborhood for a few wireless throttles, a command station, some boosters, etc. However, I ended up building a DCC++ system (https://github.com/DccPlusPlus) at a cost of around $15 (closer to $50 if you choose the official Arduino components rather than Chinese knockoffs).

I'm very happy with this solution; it provides all the capacity that all off-the-shelf DCC systems provide, plus many of the capabilities of such "enhanced" systems as ESU, ECoS, and native coupling with JMRI including WiThrottle – hence the "++" of DCC++. (I absolutely love the ECoS system and believe it is the direction of the future in the control systems space.) But, putting together such a DIY DCC++ system can require an investment of time and learning.

However, with the use of some DIY thinking, you can get some quite advanced capabilities that are some years away from becoming commonly available and at a reasonable cost. For example, if you're interested in ambient sound around your layout, or would like to bring life to your layout beyond the movement of trains, DIY is very nearly a requirement. But look what you can get for your efforts; a layout can really come to life with animated lighting systems, signal systems, and ambient audio for a river, wildlife, and industrial sound effects.

Most of what I have presented here is overly simplified, and it should be assumed that it will be a more complex internal debate regarding the right direction FOR YOU. It's also important to clarify – as I've stated in prior editions of this column and on many live shows – that I am of the opinion that there is NOT just one "right" way to do something; while I'll support the reality that there are "wrong" ways, there are usually more than one – and often many – right ways do a particular thing.

> ***"Few of us treat our time as money, but it is. The goal is to get to the result we desire, while still leaving funds available for other parts of the layout."***

The topic of this edition of the column is definitively one of those discussions. There are a lot of choices here, a lot of components to choose or to integrate, several good communication methodologies between components, and several overall control system architectures. Let me suggest you revisit one of my earlier recommendations: where is your "threshold to building?"

Your background, aptitude, and time vs. the money you have available to invest likely will have an impact on your choices, as they should. There simply is no one-size-fits-all solution available, in my mind, just as there is not just one type of modeler.

The vast selection of available components may have the disadvantage of making your choices more challenging, but with that comes the ability to further refine your choices to better fit your personality and priorities.

The future of our hobby is bright and the opportunity to seize the advancements in detailed structure kits and new technology for your layout is at hand.

Commitment to making some decisions, spending some time to educate yourself, and putting some technology in your layout can add a tremendous amount of flexibility and enjoyment to your modeling and operations. And much of it can all be had for quite a low cost. But the debate goes on, will you DIY or DIBuy?

About the Author

Andy Crawford, 38 years old, is a technology provider to mid-sized businesses and financial institutions, and a lifetime model railroader. Starting young in the hobby with a train set, like many others, and after spending a decade as an armchair modeler, he returned to the hobby a few years ago, in full force. He models a very exacting replica of a 15 mile section of the Clinch Valley District of the famous Pocahontas Division of the Norfolk & Western Railway in 1952.

His interest in exacting replication of the prototype, fine scale craftsmanship, weathering, and prototypical operation can all be seen in his work. For him, recreating the experience of being a railroad professional, 1/87th the size, in the 50's is all the focus that is needed. You can check out Andy's YouTube channel here: https://www.youtube.com/channel/UC8l2bTYfzVY37328sGPD9Bw.

This free YouTube Model Builders e-Magazine is produced by the dedicated YouTube Model Builders team on a volunteer basis. Please visit their channels and show your support!

YouTube Model Builders Team

www.YouTubeModelBuilders.com

Team Role	YouTube Channel/Name
Team Lead	ModelerManMike
Secretary	JD - Loggin' Locos (Acting)
Technical/IT	Barry Rosier
Outside Business & Giveaways Coordinator	William "Big Bill" Graham
Promotion and Distribution	ModelerManMike (Acting)
Graphic Artist	Position is Open
YouTube Model Builders eMag	
Editor-in-Chief	JD - Loggin' Locos
Content Editor	Blayne Mayfield
Content Editor	Jack Hykaway
Content Editor	Jen Hanson
eMag Articles Ambassador	Harry M. Haythorn - UPHS #4043
eMag Photographer	Jack Hykaway
YouTube Model Builders LIVE!	
Host and Production Manager	William "Big Bill" Graham
Director	Barry Rosier
YouTube Model Builders Hangouts	
Hangouts Lead	Johnny Reb "Southeast Rails"
Hangouts Moderator	Barry Rosier
Hangouts Moderator	Andy Crawford

If you would like to submit an article for the YouTube Model Builders eMag or pictures for the Community Collage section, please contact us at: YTMBeMag@gmail.com. Submission guidelines can be found at www.YouTubeModelBuilders.com.

Disclaimer:
This is a free, community based, electronic magazine and as such all articles, photos, diagrams, and illustrations included within the articles and in the Community Collage section, as well as any opinions expressed within the articles are those of the individual author/writer and do not necessarily reflect the opinions of YouTube Model Builders Team or staff. We are in no way associated with, sponsored by, or compensated by YouTube, and/or Google, or any other entity.