PART 1
INTERMEDIATE PROJECT

Build your own WIRELESS DCC SYSTEM

Introduction to this low-cost project

by Mark and Vince Buccini | Allen, Texas | PHOTOS BY MARK BUCCINI

Even beginners in electronics can build this basic, wireless DCC system at low cost. This story describes how.

As I became interested in large-scale railroading, so did my children, who, upon seeing these large and detailed trains rolling along the track, naturally wanted to join in the fun. I wanted my large-scale-model-railroading interest to be a hobby that my entire family and friends alike could enjoy. However, after seeing my prize SD-70 racing along the track at top speed, backward, under the command of my seven-year-old daughter (coached by her brother), using over $300 worth of equipment, I set out to find ways to adapt my railroad to be lower cost and more family friendly. I wanted to keep everybody involved with our railroad, but with fewer dollars at risk.

1. The author's Junior Operator radio controlled, wireless DCC throttle.

The birth of the Junior Operator

We had no problem finding moderately priced large-scale locomotives, such as the Egg Liner and Lil' Critter from Aristo-Craft, that the kids in particular get a kick out of. The price of existing name-brand throttles and decoders is through the roof, though, costing more than the locomotives themselves. These existing throttles also tend to be confusing and difficult to use, especially for younger or casual operators. With this in mind, my son and I decided to build our own throttle and decoder that are more affordable and easy to use. In this series we will describe how we built our easy-to-use Junior Operator radio controlled, wireless DCC throttle, as shown in **photo 1**, which is used with our family-friendly Crazy Garden Railroad, or what we call LocoG.

While I have designed and built several full-featured throttles, with complete keypads, big LCD displays, locomotive recall, service and ops programming, consist support, 28 functions, and so on, these are really for the experienced adult engineer. The Junior Operator throttle described in this series, on the other hand, is designed specifically for ease of use with the casual operator in mind. It boils down to just a single knob for speed, three function-control pushbuttons (F0-F2), and an LED for feedback.

Wireless DCC

The advantages and popularity of digital command control (DCC) for model railroading, which sends digital-command-packet data through the rails to power and control locomotives and accessories, is well documented and a natural draw for an electrical engineer such as me. For those interested, complete technical specifications for DCC are available free at *www.NMRA.org*. What really got my attention, though, was the idea of wireless DCC. I work on similar communication protocols in my professional career. Believe it or not, these date to the late 1990s for automated meter reading (AMR). It's also important to note that wireless DCC technology is exactly the same as wired DCC, except the command data is sent over the air using a handheld, radio-controlled throttle (transmitter) to decoders (receivers) installed in battery-

Garden RAILWAYS MAGAZINE

© 2014 Kalmbach Publishing Co. This material may not be reproduced in any form without permission from the publisher. **www.GardenRailways.com**

powered locomotives and accessories, instead of through the rails.

Our Junior Operator wireless throttle is wireless-DCC compatible. Our throttle is not only easy to use and easy on the wallet, it is designed and tested to be fully compatible with other wireless DCC equipment from QSI-NCE and CVP. On the same railroad where we use the Junior Operator throttle, existing equipment from QSI and CVP work perfectly.

It's not just about the technology. The big advantage of radio control and battery power is that it takes the hassle out of dirty track; you just turn it on and go. For my family and friends, wireless DCC definitely provides a fun and more relaxing experience on our LocoG Railroad.

Radios and microcontrollers

What makes the construction of Junior Operator possible is the Anaren A110LR09A radio module. This is a self-contained radio module on a tiny printed circuit board (PCB) that even includes an embedded antenna (**photo 2**), and it costs around $15. The A110LR09A is flexible technically and is configured for operation using software sent from a separate microcontroller (MCU). What is especially attractive is that, after being properly configured, the A110LR09A is fully compatible with the Linx HP3 module used in other wireless DCC large-scale equipment but, as I said earlier, at half the price and a fraction of the size.

To simplify the construction of our throttle even more, we used the Anaren 430BOOST-AIR booster-pack kit. A booster pack is an add-on to the MSP430 microcontroller Launch Pad development system. The Launch Pad is a low-cost PCB that can stand alone or be used with booster-pack add-ons to quickly prototype MSP430 MCU electronic systems. We will use both the Launch Pad and booster pack in this project. The 430BOOST-AIR booster pack PCB used in this project is shown in **photo 3**.

The complete AIR booster-pack kit actually includes two identical printed circuit boards, each with the tiny radio module already soldered down. It also includes two identical 20-pin dual in-line package (DIP) MSP430G2553 MCUs—all for only $20. For the Junior Operator project we used one circuit board and one MCU for the throttle and saved the others for the follow-on decoder project. Each booster-pack circuit board also has a prototype area of 0.1" on-center, plated-hole pairs that can be used for adding your own components. We used the prototype area to add a potentiometer for speed control, three pushbuttons to send function commands, and an LED for operator feedback.

The AIR booster-pack kit PCB is the foundation from which the throttle is built and it really simplified the construction of our throttle down to just a couple of hours. For completeness, we put the circuit board and a battery pack in a sturdy Hammond plastic enclosure (#1593) that even includes a battery door.

For this project you will not need to write any software or, for that matter, really know anything at all about programming. You will need to get the 1s and 0s into the MCU's memory using a Microsoft Windows personal computer with a USB connection, which will be described later.

The technical stuff

For those interested in the technical alphabet soup, radio controlled, wireless DCC transmitters and receivers in North America pretty much all use a license-free 900MHz ISM (industrial, scientific, and medical) FSK (frequency shift key) format. The A110LR09A radio we used in the Junior Operator throttle is configured with software from the MCU for this exact mode of operation. The A110LR09 is also a true transceiver. This means the same radio module can be used as a transmitter in a throttle, as in this article, or it can also be used as a receiver in a decoder, which we will describe later.

For those not familiar with an MCU, it is actually just a miniature, stand-alone computer that runs small software programs and acts as the brains in a wide variety of embedded systems, such as wall thermostats, blood-glucose meters, and the throttle in this project. During the day, an average person will come in contact with over 100 MCUs, especially if you drive a car. An automobile has more than 30 MCUs, which control everything from your seats to fuel injection.

The software that the MCU needs for this project is provided in what is called "object format;" this is just a bunch of 1s and 0s, and is what is programmed into the MCU Flash memory. The MCU Flash memory is similar to a USB thumb drive, for example, but the MCU Flash memory is self-contained and on-chip.

In the next installment, we'll construct the Junior Operator Throttle.

2. The Anaren A110LR09A radio module, a self-contained radio module on a tiny printed circuit board. It even includes an embedded antenna.

Actual size

3. The 430BOOST-AIR booster pack printed circuit board that will be used in this project.

About the series

April: Building the Junior Throttle
June: Building the decoder

PART 2
INTERMEDIATE
PROJECT

Build your own WIRELESS DCC SYSTEM

Part 2: Building the Junior Operator throttle

by Mark and Vince Buccini
Allen, Texas | PHOTOS BY MARK BUCCINI

The Junior Operator DCC throttle (left) is a basic unit that anyone can build at low cost.

1. The Launch Pad is connected to any Windows computer via a USB cable for programming.

In the April issue, I described my concept for building the Junior Operator, a basic DCC system that's easy to build and simple enough to use that kids take to it right away. In this issue, we'll build the throttle.

In just a few hours, and using readily available supplies, a Junior Operator wireless DCC throttle can be assembled for around $20. This is about one-tenth the price of typical name-brand equipment. At this price, my entire family is now participating and having fun on our LocoG Railroad without me having to worry about cleaning tracks or the possible breakage of expensive equipment.

In addition to the components mentioned in the Throttle materials list, you will need a soldering iron and a drill. None of the soldering or drilling is complicated or difficult. Perhaps the only tricky part of the assembly is soldering down the "805" size, surface-mount components R1, R2, and C1. The 805-size components are tiny parts but there are also pads for the needed components already on the Booster Pack printed circuit board (PCB). Even though they are small components, we were able to solder them down at home with a 30-watt Radio Shack soldering iron.

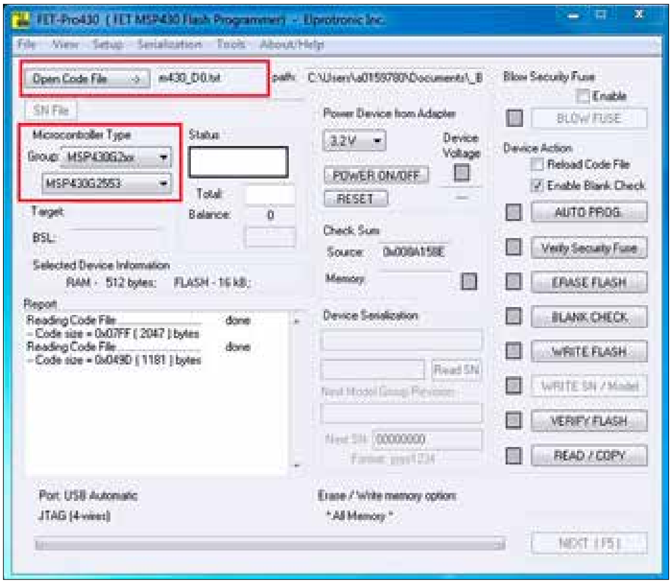
The Booster Pack radio circuit board, MCU launch pad, potentiometer, enclosure, and battery holder are the only critical components. All of the other components you can improvise, if you wish, with whatever is on hand.

Before starting, you should review the electrical schematic showing the needed modifications to the Booster Pack PCB in **figure 1**. The total assembly takes about two to three hours.

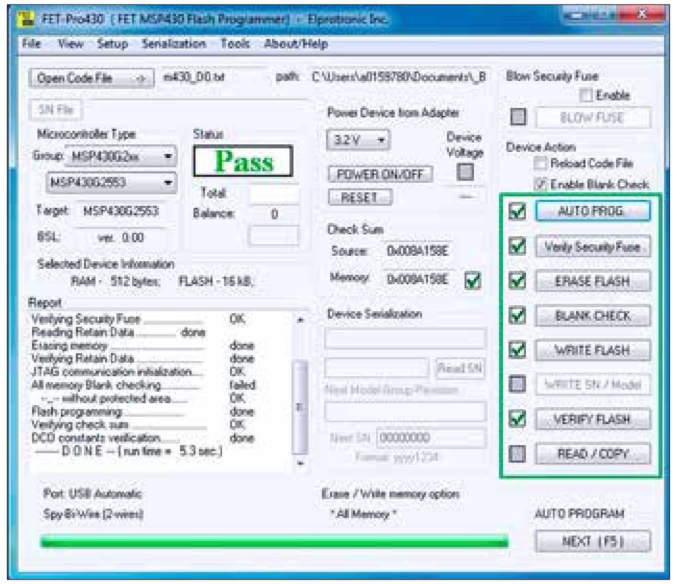
1. Program the MCU. The software needed for the MCU to operate properly must be programmed into its memory. Make sure the MCU object-code software "m430_T0.txt" (the ones and zeros that make everything happen) is saved on your computer—it is available at *www.GardenRailways.com* (see sidebar on p. 34). You will use the free FET-Pro430 Windows utility from *www.elprotronic.com* to program the software into the MCU. From the Elprotronic website, go to the "Download" tab to find the utility (called "Flash Programmers for TI's MSP430 MCU uses TI's FET Adapter" on the download page) then download and install it on your computer.

Make sure there is one of the two 20-pin MSP430G2553 MCUs, included

Garden RAILWAYS MAGAZINE
© 2014 Kalmbach Publishing Co. This material may not be reproduced in any form without permission from the publisher. www.GardenRailways.com



2. Screen showing the code file and microcontroller type for programming.



3. Screen showing check boxes.

with the Booster Pack inserted in the Launch Pad socket, and connect the Launch Pad to a Windows computer using the included USB cable (**photo 1**). A needed Windows USB driver will automatically install on your PC when connected for the first time.

Open the FET-Pro430 program, select "m430_T0.txt" for the object code on your computer, "MSP430G2xx" for the group, and "MSP430G2553" for the device, as shown in and outlined in red in **photo 2**.

To program the m430_T0.txt software into the MCU, click on the "AUTO PROG" button and wait until all of the check boxes are green, as shown in **photo 3** (the boxes are outlined in green). Shut down the software, disconnect the Launch Pad, and remove the MCU from the socket. It is now programmed and ready for the throttle.

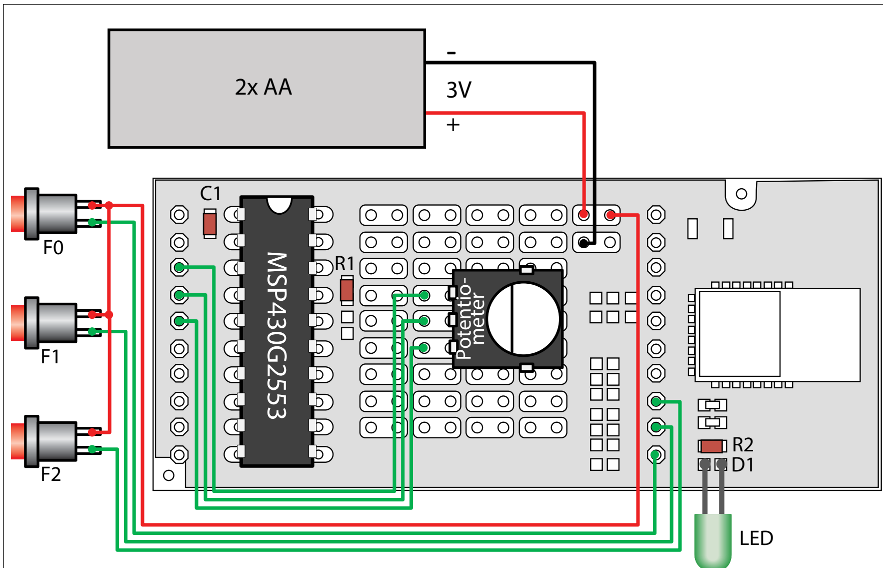


Figure 1: Circuit diagram

MARC HOROWITZ

2. Solder R1, R2, C1, the MCU U2, and LED D1 to a Booster Pack. Two resistors (R1, R2), one capacitor (C1), the already-programmed MCU (U2), and LED D1 need to be soldered onto the Booster Pack PCB (**photo 4**). Make sure to use a soldering iron of 30 watts or less—any more and you risk damaging the components. Surface-mount 805-size pads are already on the PCB for R1, R2, and C1. Neither of the resistors nor the capacitor is polarized so, as long they are soldered to the pads, direction does not matter.

Solder the 47k and 220-ohm resistors, and 0.1µ capacitor to the pads R1, R2, and C1 respectively. Next, solder the already-programmed 20-pin MSP430G2553 MCU to the area on the PCB marked U2. Take special care to orient the MCU correctly, with the pin-1 indicator (a U-shaped divot) facing the edge of the PCB. We used the surface-mount pads already on the circuit board, labeled D1, to attach a standard T1 ¾ LED to the circuit board. The LED leads need to be bent over and trimmed so that the LED is elevated from the circuit board 0.5". The LED will show through the enclosure after final assembly. The LED is polarized, so its orientation is important. The positive lead (the longest one) must be soldered to the pad nearest the "D1" label on the PCB. See the LED attached to the PCB in **photo 5**. After the LED is soldered to the circuit board, place hot glue on top of the LED connection to the PCB for added stability. This step is important, as it is easy to bump the LED and pull the pads off the PCB. The hot glue will not affect any connections.

3. Attach the potentiometer. Cut three pieces of hook-up wire to a length of 2" and strip their ends. These will be used

About the series

April: Building the Junior Throttle
August: Building the decoder

to connect the potentiometer (**photo 6**). Solder the potentiometer to the prototype area of the circuit board exactly as shown in **photo 7**, using the three cut pieces of hook-up wire to connect pins 1, 2, and 3 from the potentiometer to pins 3, 4, and 5 of the header labeled J1, respectively.

4. Wire the three function-select pushbuttons. There are three pushbuttons that are used to select and transmit DCC functions: F0, F1, and F2. Cut five pieces of hook-up wire to a length of 3" and strip their ends. Each of the pushbuttons has two leads; one of the two leads needs a hook-up wire and the other lead needs to be wired as a single common connection with all pushbuttons, as shown in **photo 8**. Solder the single common connection from the pushbuttons to pin 1 of J1, then attach the other of the three individual leads from each of the pushbuttons individually to pins 13, 12, and 11 of J2, as shown in **photo 9**.

5. The battery holder. Solder the red lead from the from the battery holder to one of the two plated holes on the Booster Pack marked "VDD," then solder the black lead to one of the two plated holes marked "GND."

6. Prepare the enclosure. Cut out the provided **drill template** (available at *www.GardenRailways.com*, see sidebar on p. 34) and place it on the top half of the enclosure (the top does not have the battery-compartment door). Punch and drill the five indicated holes, as shown complete in **photo 10**. These holes will let the LED show through, let you access and connect the potentiometer, and allow the pushbuttons to connect to the enclosure. Make sure the template is oriented correctly, with the

Changing the transmitted radio channel and locomotive address

The default object-code file m430_T0.txt is used to program the throttle to transmit commands to locomotive 3 on radio channel 2. You can change both of these at any time, but remember that the locomotive number and radio channel you set in the throttle must match those set in the locomotive decoder.

To change the radio channel, simply add a jumper to Booster Pack J1 that shorts pins 8 and 9 to select channel 3, or shorts pins 9 and 10 for channel 6. Channel 2 is the default and uses no jumper. This means that the transmitter can select one of three radio channels (though there are there are actually eight channels specified in wireless DCC).

If you would like to change the locomotive address, you need to change the first character (these are called data bytes) on the second line of the provided object-code file. For example, if you look at the file "m430_T0.txt," using Microsoft Notepad, you see at the very top of the file the numbers below:

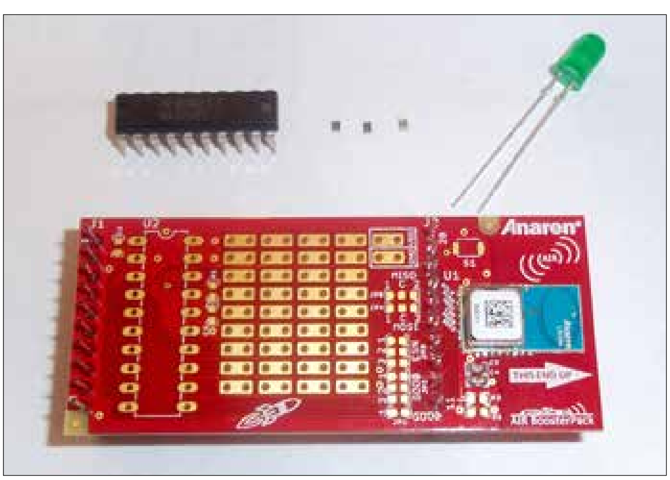
```
@1040
03
```

The "03" is the number of the locomotive address to which the throttle is transmitting; "@1040" is where the code is loaded into the MCU memory—do not change 1040! You may change the 03 locomotive address to anything from 00 to 63—these are hexadecimal numbers that equal 0-99 decimal. If you are not familiar with hexadecimal, you should research this before changing.

For example, if I want to control locomotive 22 (16 hexadecimal) I change the second line of "m430_T0.txt" file as below. Absolutely nothing else is changed.

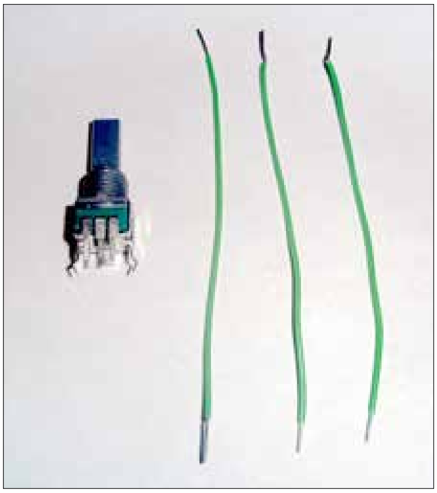
```
@1040
16
```

After the changes to the file "m430_T0.txt" have been made and saved, use FET-Pro430 software and MSP430 Launch Pad to reprogram the MCU on the Booster Pack PCB installed in the throttle. You only need to remove the bottom cover of the throttle enclosure to connect the Booster Pack PCB, as shown in the photo above, inside the enclosure, to Launch Pad during programming.

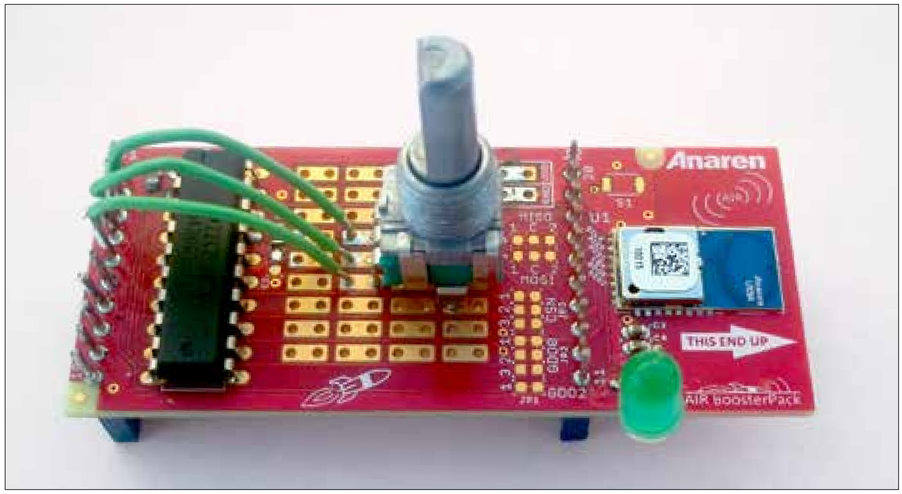


4. The parts that need to be soldered to the circuit board.

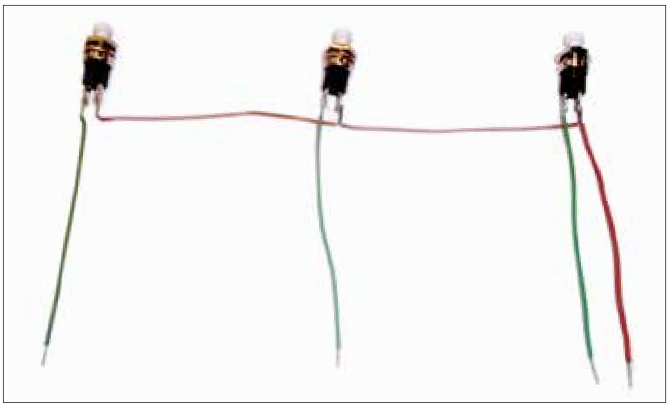
5. The circuit board after the parts have been soldered. Note the green LED. This is before stabilization with hot glue.



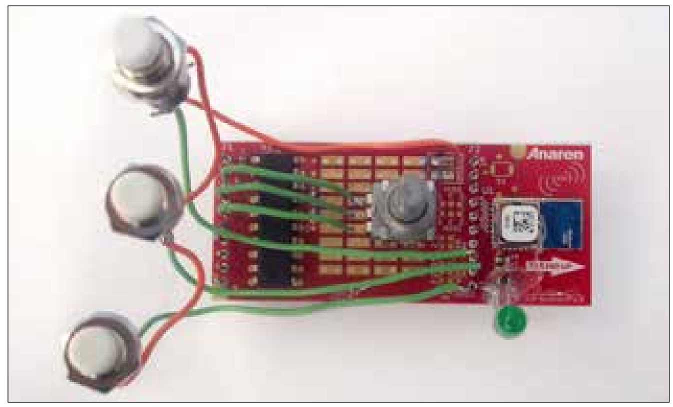
6. The potentiometer with the three new leads.



7. The potentiometer and the new leads installed on the board.



8. One lead of each pushbutton is wired to the others as a common connection.



9. The pushbuttons, properly wired to the board.

words on the template aligned with the openings on the enclosure.

7. Attach the pushbuttons, Booster Pack, and battery box to the top of the enclosure. Attach the three pushbuttons to the top half of the enclosure, inserting them from the bottom through the three holes previously drilled. Secure the pushbuttons from the top by tightening the included nuts. Attach the Booster Pack PCB to the top of the enclosure from the bottom, with the potentiometer and LED coming through the holes drilled earlier. Secure the potentiometer from the top by tightening the included nut. Fasten the battery holder to the inside of the top of the enclosure using the double-sided tape already on the holder. At this point the throttle is fully functional and it would be a good idea to test it before final assembly.

8. Final assembly. Assemble the top and bottom of the enclosure and screw it closed with the four screws included. Add the knob to the potentiometer, install batteries, and the LocoG throttle assembly is complete. As soon as the batteries are added, the status LED will come on. Shown in **photo 11** are the complete Junior Operator wireless DCC throttle and its proud owner.

Throttle materials list

Part	Part Number	Source
RF Booster Pack kit with MCU	595-430BOOST-CC110L	*www.mouser.com*
MSP430 launch-pad kit	595-MSP-EXP430G2	*www.mouser.com*
Hammond enclosure	546-1593YBK	*www.mouser.com*
Hammond battery-holder kit	546-1593YBK	*www.mouser.com*
Potentiometer	688-RK09L114001T	*www.mouser.com*
Panel-mount small pushbutton (3x)	104-0010-EVX	*www.mouser.com*
Green LED T-1 ¾	78-TLHG5400	*www.mouser.com*
Resistor ⅛ watt, 220 ohms SMT	71-CRCW0805220RJNEB	*www.mouser.com*
Resistor ⅛ watt, 47K ohms SMT	71-CRCW080547K0JNEA	*www.mouser.com*
Capacitor 0.1uF SMT	77-VJ0805Y104MXXAC	*www.mouser.com*
Hook-up wire: black, green, red 22AWG	278-1224	Radio Shack
Knob	274-416	Radio Shack

Using the Junior Operator

The Junior Operator throttle runs off of two AA batteries. It is easy to use, with just a single knob to control speed and direction, and three function pushbuttons. By default, the throttle is fixed to control locomotive 3 and it transmits on radio-channel 2. You can change the locomotive and radio channel at any time if you like (see sidebar on p. 32) but always remember the locomotive decoder to be controlled must be programmed with the same locomotive number and radio

10. Holes have been drilled in the enclosure using the drill template as a guide.

channel as the throttle.

The range of the throttle is amazing, at well over 100' outdoors, even with the tiny radio module and no external antenna—plenty for most operations. The lack of an external antenna is actually a benefit because it is usually the external antenna that is the first thing to break.

On power-up, the LED will be fully on at "Stop," which is when the knob is in the center mid-point; it will flash slowly if commanded "Forward" and flash quickly if commanded in "Reverse." The knob controls the speed and commanded direction. To stop, simply return the knob to the middle, which is indicated by a center detent. Forward is to the right—the further to the right, the faster the locomotive will go. Reverse is to the left—the further to the left, the faster the locomotive.

The throttle sends 28-speed-step speed information to the decoder. Pressing and releasing pushbutton 0 or 1 sends function commands 0 or 1 to the decoder. Typically in the decoders, function 0 is used to activate the headlights, while function 1 is used for a bell or other light function. Pushbutton 2 must be held down to send function 2, which is typically used for the horn. The pushbutton functions (left to right) are 0, 1, and 2, with the speed knob on the top when looking at throttle.

To turn on or off the Junior Operator throttle, press and hold down pushbutton 0. To extend the battery life, the throttle also automatically turns off after 15 minutes of inactivity. The throttle will operate for over 100 hours on two AA batteries.

In the next issue we'll build a low-cost wireless DCC motion-and-light decoder, which is a radio receiver tailor-made for our Junior Operator hand-held wireless DCC throttle.

11. The finished Junior Operator and its owner.

Online extras

Cut and paste the code needed to program the MCU, and download a PDF of the drilling template for the throttle enclosure. Visit *www.GardenRailways.com* and click on "Construction & landscaping" under "How to."

PART 3
INTERMEDIATE PROJECT

Build your own WIRELESS DCC SYSTEM

Part 3: Building the wireless motion-and-light decoder

by Mark and Vince Buccini | Allen, Texas | PHOTOS BY MARK BUCCINI

1. The DCC decoder, ready to be installed in your locomotive of choice.

In the last installment in this series we built the Junior Operator DCC throttle. In this article we describe how we built a low-cost wireless DCC motion-and-light decoder, ready for installation, as shown in **photo 1**. This decoder is a radio receiver and a companion to the Junior Operator hand-held wireless DCC throttle. Both the decoder and throttle have been put into service on our family friendly LocoG Railroad.

Our decoder is installed in the locomotive and drives the motor with more than 3 amps of current, based on radio commands received from the throttle. The decoder is also perfect for smaller locomotives, such as the Egg Liner and Lil' Critter; both of these are popular on our railroad. The decoder also outputs DCC functions F0, F1 and F2, which are used to toggle lights and a horn.

Early on, we decided that radio control/battery power was right for LocoG, as this set-up takes all of the hassles out of dirty track; you can just turn it on and go. This is a big plus, especially with the younger Junior Operators who frequently use our railroad. We quickly found that, unlike the wide variety of low-cost HO-scale DCC decoders and throttles available, garden-scale equipment is pricey and definitely not offered with the Junior Operator in mind. Existing equipment is not necessarily easy to use. So we designed and built our own hand-held throttle (June issue) and the locomotive motion decoder that is described here. Both the decoder and throttle are not only low cost, at around $20 each, but they are also easy to use, easy to build, and, by design, completely compatible with existing NMRA wireless DCC equipment from

About the series

April: Introduction and overview
June: Building the Junior Throttle

Garden RAILWAYS MAGAZINE
© 2014 Kalmbach Publishing Co. This material may not be reproduced in any form without permission from the publisher. www.GardenRailways.com

QSI-NCE and CVP. While our system is based on NMRA wireless DCC, and compatible with other wireless DCC equipment, we kept the actual feature list to a minimum so that it is simple and easy to use, with no computer-science course needed. Because of the low cost and ease of use of the equipment on our railroad, both Junior and Senior Engineers can share in the LocoG action.

The LocoG decoder

The decoder is installed in the locomotive and communicates with the operator's hand-held throttle via a 900MHz radio link. The decoder receives and decodes 28-speed-step DCC forward-and-reverse speed commands, allowing precise locomotive motion control. The decoder motor-drive circuit we designed will drive any brushed DC motor forward and backward with 28 speed steps, with up to 3 amps of peak output current, which is plenty for most large-scale locomotives.

We also implemented three DCC command-function commands: F0, F1, and F2. Common with other DCC systems, F0 controls the locomotive's headlights, F1 controls the other lights, and F2 is used for a horn. These are the most-used DCC functions and make for interactive operation without a lot of clutter. Other unmentioned function commands are ignored by our decoder.

For those familiar with DCC, the decoder does actually receive all 29 DCC functions, F0-F28, but we chose to only implement F0-F2, just to make things simple with the hardware. Speed-step 126 commands are also received and decoded properly for motion control. The decoder is designed to support configuration variables CV1-CV4, CV8, CV11, CV17, CV18, and CV29 service programming.

While the companion Junior Operator throttle described in the previous article does not provide the ability to send F3-F28, speed step 126, or service programming, if you are using a full-featured throttle to command the decoder, these features are still available.

The prototype, LEDs, and batteries

On our LocoG Railroad, our go-to locomotive is the Egg Liner, of which we have several. While there is no debate that this little guy is not a prototypical locomotive, it is

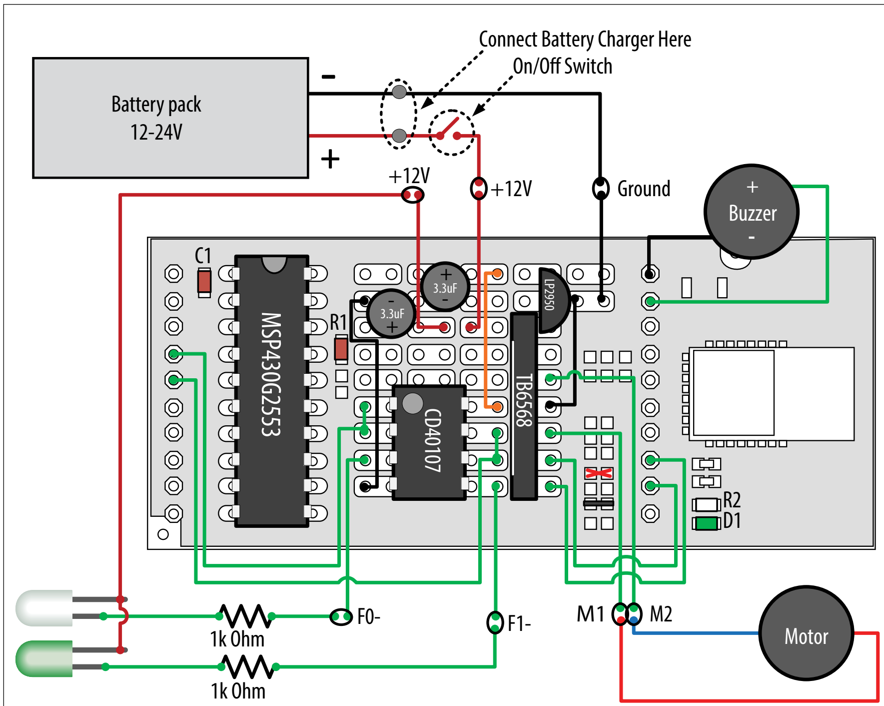


Figure 1: Circuit diagram

ILLUSTRATIONS BY MARC HOROVITZ

cheap and is lots of fun for the kids (and some adults as well!). Its shell can be removed with just four screws, which makes it easy to modify and work on. The decoder has also been put into operation in a Lil' Critter and a few other locomotives on the LocoG Railroad. The decoder can be installed in any locomotive with a single motor.

The two light functions we implemented (output F0 and F1) are designed to drive LEDs. Because there are so many advantages to using LEDs over bulbs, including one-tenth the power consumption, less heat, and over 10 times longer operational life, we have converted all of our trains to LEDs. The function F0 and F1 outputs must be connected to the negative terminal of current-limited LEDs.

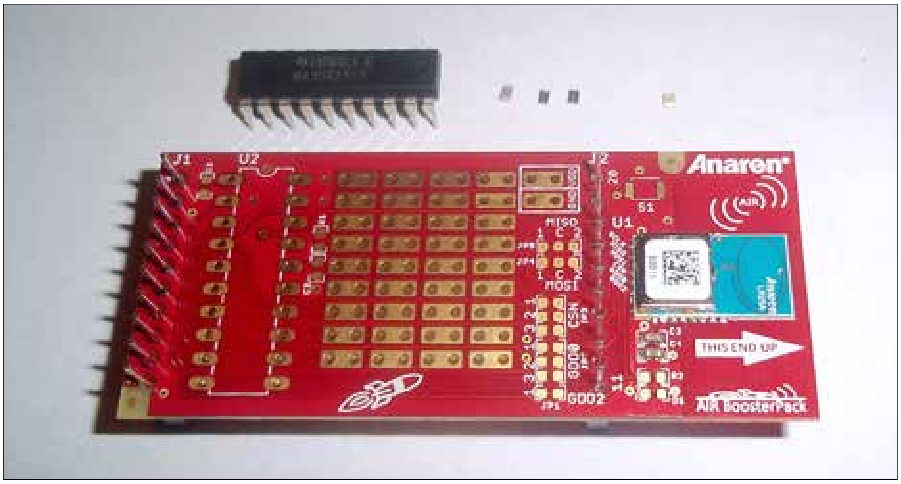
We exclusively use the 14.8V lithium-ion battery packs from *www.all-battery.com*. For a description of this type of battery, please refer to the exceptionally good article by William Canelos in the December 2012 issue of *Garden Railways*. The decoder will operate correctly on any 12-24V battery, but be extremely careful with the positive-negative polarity, as no reverse-polarity connection is built into the decoder. Never connect the battery backward. Also, when you recharge the battery, it should be disconnected from the decoder with a switch or plug.

2. The Anaren "430BOOST-AIR Booster Pack" kit is the heart of this project.

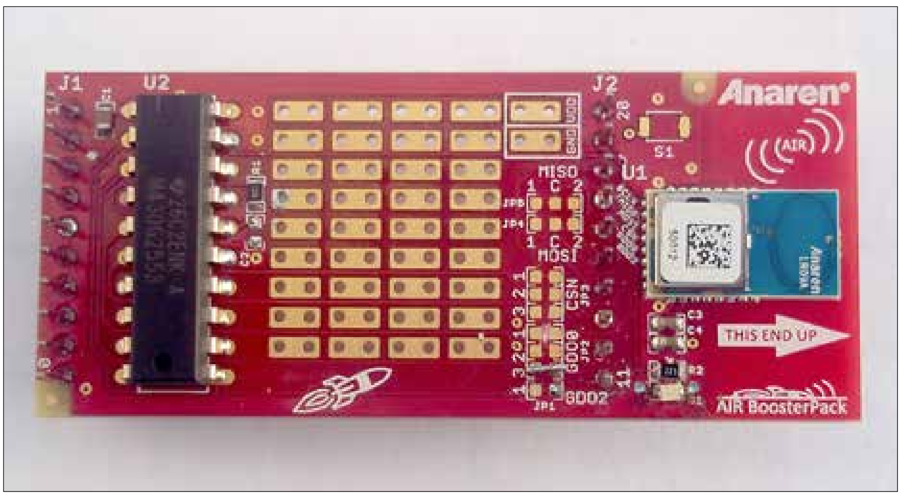
The Anaren Booster Pack

As with the Junior Operator throttle, what makes the construction of the decoder so easy is the $20 Anaren "430BOOST-AIR Booster Pack" kit, as shown in **photo 2**. This kit contains two identical printed-circuit boards (PCBs) that include an already attached, self-contained radio module. It is this radio module that wirelessly sends the DCC commands between the throttle and decoder. The same radio module is also used in our throttle. We have programmed the Anaren radio to be completely compatible with the Linx radio used in other wireless DCC equipment, so our decoder and throttle work perfectly with name-brand equipment from QSI-NCE and CVP that also uses a Linx radio.

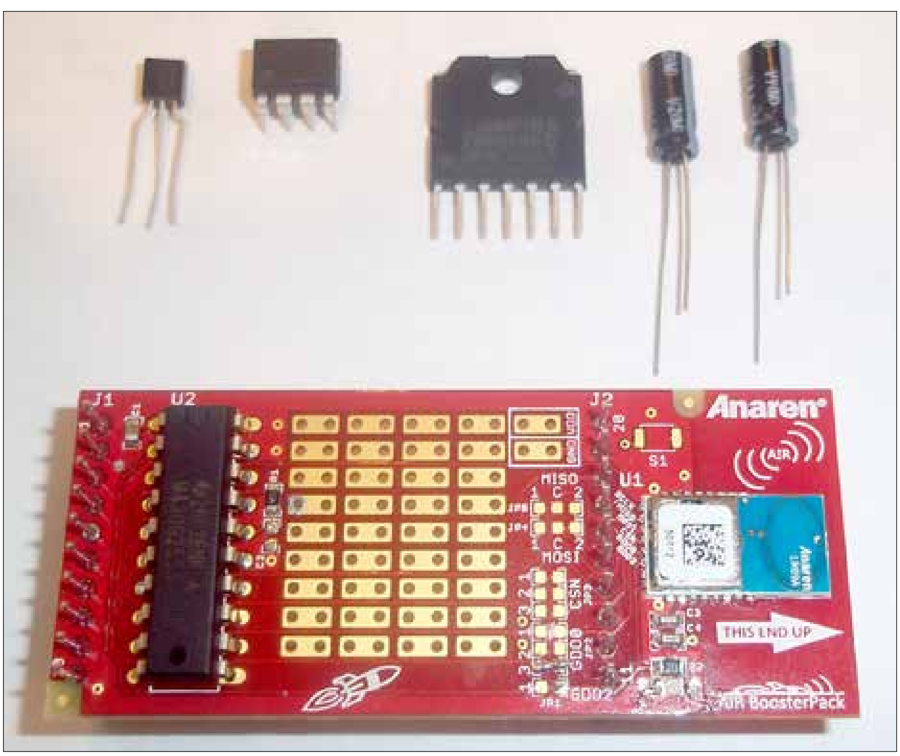
As handy as the tiny Anaren radio



3. Components (top), ready to be soldered to the PCB.



4. The booster pack, modified, with components added.



5. Three ICs and two capacitors are the next components to be added.

module is, it is challenging to solder down as a casual weekend soldering project. However, the best part of the booster pack PCB is that it has this tiny radio module already soldered down. There is also a convenient spot to solder one of the two identical 20-pin MSP430 microcontrollers (MCUs) also included with the kit.

Each Booster Pack PCB also has a prototype area of 0.1" on center, plated-hole pairs that can be used for adding your own components—we used this area for a voltage regulator, motor driver, and external LED driver integrated circuits. To keep assembly as easy as possible, through-hole integrated circuits were selected.

Assembling the decoder for $20

All that is needed for assembly is a soldering iron and the components in the materials list. Total cost is around $20, and the overall assembly of the decoder is similar to the previously described throttle.

The modified Booster Pack PCB is basically the decoder hardware in itself and what is installed in the locomotive. The modification we needed to make to the PCB mainly included the addition of a 7-pin motor driver IC, a 3-pin LP2950-33 voltage regulatior, and 8-pin CD40107 IC, which is used to drive the LED lights in the locomotive. A smaller buzzer, LED, and a few other resistors and capacitors are the only other components needed—everything fits neatly on the Booster Pack. At a size of only 1" x 3", the Booster Pack PCB is small enough to fit in most locomotives.

Before starting, review the electrical schematic shown in **figure 1**, which highlights the needed modification to the Booster Pack PCB. In just two or three hours, the decoder can be assembled and put into operation.

1. Convert locomotive lights to LEDs. The decoder will directly drive two LED groups. With the Egg Liner,the front and rear headlights are wired together in parallel as a group, with a single common ground (cathode) and single 12V supply (anode). Label the common cathode "L0-" and the anode "L0+". Do the same for the four side-lights, grouping them in parallel, labeling the cathode "L1-" and anode "L1+". Each group of LEDs must be current-limited with a single 1k ohm resistor. This is important, as an LED must have a series current-limiting

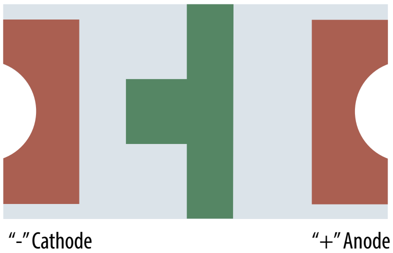


Figure 2—Backside of SMT LED

resistor or it will burn out.

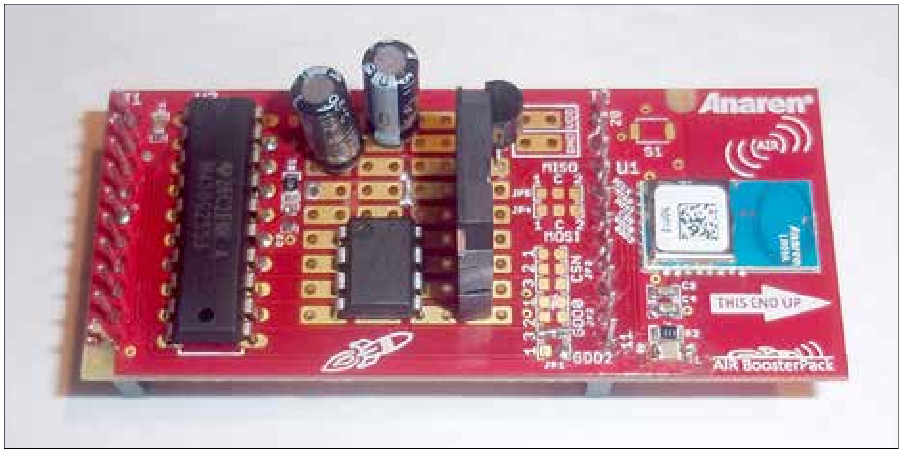
2. Program the MCU. The software needed for the decoder MCU must be programmed into its memory. Make sure the MCU object-code software "m430_D0.txt" is loaded on your computer and follow the directions from the previous article. This code can be found on *Garden Railways'* website (see sidebar on p. 53).

3. Solder R1, R2, C1, the MCU U2, and LED D1 to a modified Booster Pack. Two resistors (R1, R2), one capacitor (C1), the already-programmed MCU (U2), and the LED (D1) need to be soldered onto the Booster Pack PCB. These components are shown prior to assembly in **photo 3**. Surface size mount 805-size pads are already on the PCB for R1, R2, and C1. Neither of the resistors nor the capacitor is polarized so, as long they are soldered to the pads, direction does not matter. Solder the 47k and 220-ohm resistors and 0.1uf capacitor to the pads R1, R2, and C1, respectively.

Next, solder the already programmed 20-pin MSP430G2553 MCU to the area on the PCB marked U2. Take special care to orient the MCU correctly, with the pin-1 indicator (a U-shaped divot) facing the edge of the PCB. Use the surface-mount pads already on the circuit board, labeled D1, to attach the surface-mount LED to the circuit board. The LED is polarized so the orientation is important. The anode (+) of the LED must align with the white "D1"label on the PCB. Unfortunately, you have to look at the back side of the tiny LED to see it's polarity, which looks like a rotated "T"; the vertical line indicates the anode, which is the "+" positive polarity. This can be a little confusing, so see **figure 2**. Carefully note the LED polarity and orient it correctly prior to soldering it down. JP needs to be modified by cutting the trace 1 with a sharp knife, then shorting the pads on trace 3 on JP2 with a small piece of wire. **Photo 4** shows the Booster Pack with R1, R2, C1, U2, and D1 added, and JP2 modified.

4. Solder on the LP2950, TB6568, CD40107, and two 3.3uF capacitors. The three needed ICs and two 3.3uF capacitors are shown next to the Booster Pack in **photo 5**. Solder the three ICs and two capacitors exactly as shown in **photo 6**. The LP2950-33 provides a regulated 3.3V to the MCU, radio, and CD40197. The TB6568 is used by the MCU to drive the loco motor with high current and the CD40107 drives the external LED with high current.

5. Connect the motor-driver IC to the circuit. A black wire from pin 4 of the



6. The PCB with the ICs and capacitors in place.

Changing the received radio channel and locomotive address

The default object-code file "m430_D0.txt" is used to program the decoder to receive commands sent to locomotive address 3 on wireless DCC radio channel 2. You can change both of these at any time, but remember that the locomotive number and radio channel you set in the throttle must match those set in the locomotive decoder being controlled.

To change the radio channel, simply add a jumper on Booster Pack J1. This means that the decoder can select one of 3 radio channels (though there are actually 8 channels specified in wireless DCC). Channel 2 is the default and uses no jumper; channel 3 uses jumper pins 8 and 9 of J1; and channel 6 uses jumper pins 9 and 10 of J1. You can change the radio channel at any time—when you change a radio channel on the decoder you will hear a beep.

If you would like to change the locomotive address, you need to change the first character (these are called data bytes) on the second line of the provided object-code file. For example if you look at the file "m430_D0.txt" using Microsoft Notepad, you see at the very top of the file the numbers below;

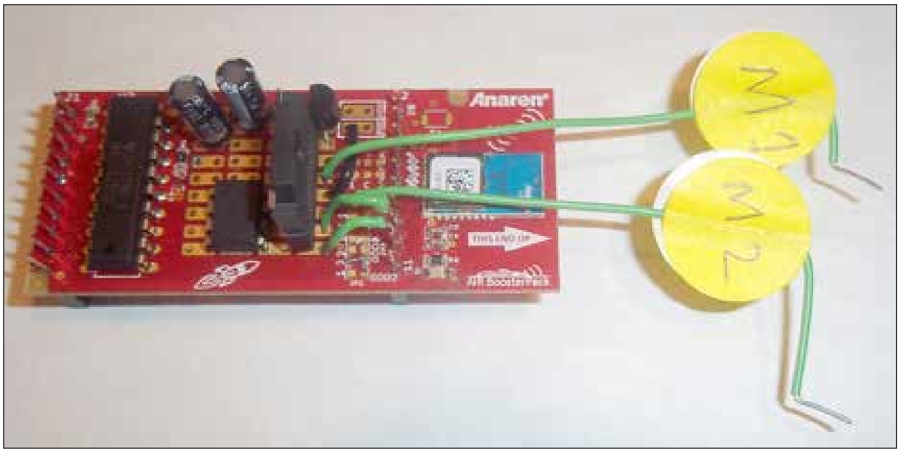
```
@1040
03 09 02 02 05 00 00 02
```

That first "03" is the number of the locomotive address the decoder will receive; @1040 is where the code is loaded into the MCU—do not change 1040 or any other data bytes! You may change the "03" locomotive address to anything from 00 to 63—these are hexadecimal numbers that equal 0-99 decimal. If you are not familiar with hexadecimal, you should research this before changing.

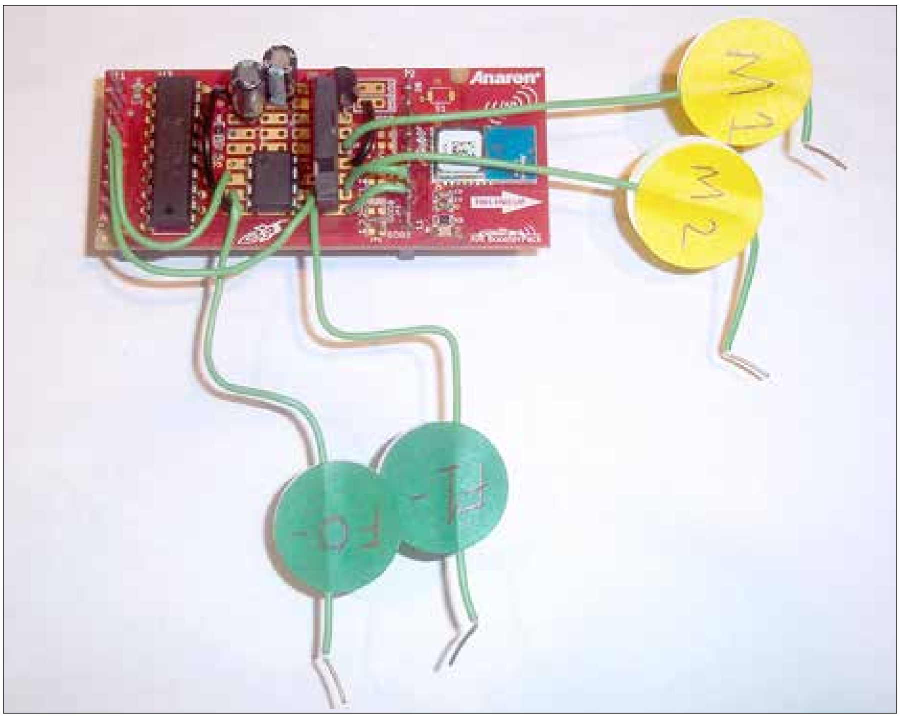
For example if I want to control locomotive 22 (16 hexadecimal), I change the second line of "m430_D0.txt" file as per below. Absolutely nothing else is changed.

```
@1040
16 09 02 02 05 00 00 02
```

After the changes to the file "m430_D0.txt" file have been made and saved, use FET-Pro430 software and MSP430 Launch Pad to reprogram the MCU.



7. The motor driver and assorted wires have been installed.



8. The LED-driver IC has been added.

Decoder Materials List

Parts List	Part Number	Source
RF Booster Pack Kit with MCU	595-430BOOST-CC110L	*www.mouser.com*
MSP430 Launch Pad kit	595-MSP-EXP430G2	*www.mouser.com*
TB6568KQ Motor driver IC	TB6568KQ(08)-ND	*www.digikey.com*
LM2950-33 Voltage regulator IC	595-LP2950-33LPRE3	*www.mouser.com*
CD40107 Open drain buffer IC	595-CD40107BE	*www.mouser.com*
(2) Green LED T 1¾	78-TLHG5400	*www.mouser.com*
(4) Red LED T 1¾	78-TLHR5400	*www.mouser.com*
(2) White LED T 1¾	859-LTW-2R3D7	*www.mouser.com*
Resistor ⅛ watt 220 ohms SMT	71-CRCW0805220RJNEB	*www.mouser.com*
Resistor ⅛ watt 47K ohms SMT	71-CRCW080547K0JNEA	*www.mouser.com*
Capacitor 0.1uF SMT	77-VJ0805Y104MXXAC	*www.mouser.com*
(2) Capacitor 3.3uF	647-UVZ1H3R3MDD	*www.mouser.com*
Hook-up wire, black, green, red 22AWG	278-1224	Radio Shack
Buzzer	273-074	Radio Shack

TB6568 needs to be connected to ground. Next, pins 1 and 2 need to be connected to pins 12 and 13 of J2. Then solder two separate 3" green wires to TB6568 pins 3 and 5; label these M1 and M2, respectively. Pin 2 of the TB6568 does not connect to anything. The complete installation of the motor driver IC is shown in **photo 7**.

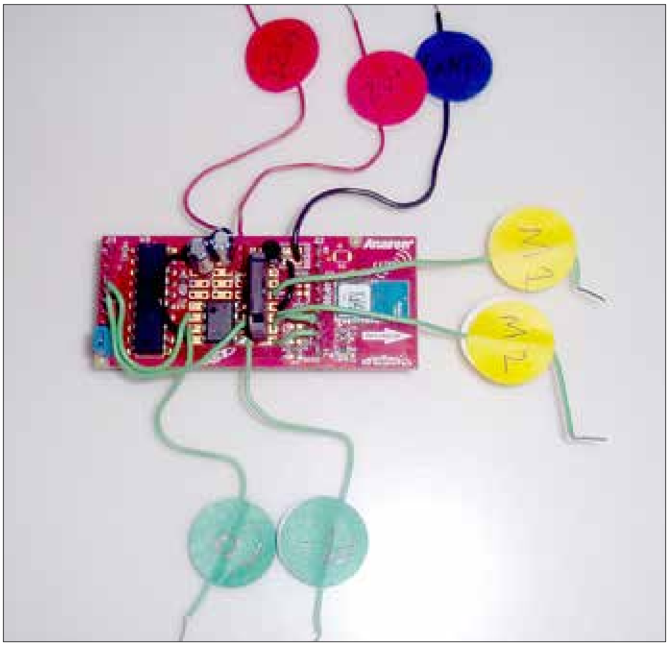
6. Connect the external LED driver IC to the circuit. Two black and red wires from pins 4 and 8 need to be connected to ground and VCC respectively. Then connect pins 2 and 5 from the CD40107 to pins 4 and 5 of J1. Next, short together pins 1 and 2, then separately short pins 6 and 5 of the CD40107. Next, solder two separate 3" green wires to CD40107 pins 3 and 5; label these F0- and F1-, respectively. The complete installation of the LED driver IC is shown in **photo 8**.

7. Create a common ground, 3V+, and 12V+ positive rails. Orient the PCB with the blue radio module on top. Add one 3" piece of black wire to the PCB on any of the plated pairs, below pairs on the top marked "GND"; label this wire "Ground." Two separate 3" pieces of red wire need to be added anywhere on the column to the right of the "ground" column; label these "12V+." **Photo 9** shows all three wires installed.

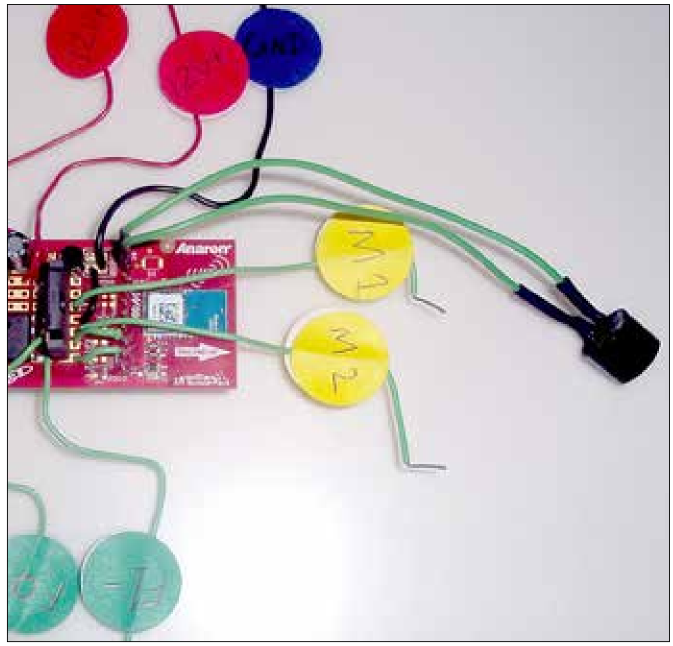
On the PCB, the entire column of prototyping-through-hole pairs below, including the pair labeled "VDD," needs to be connected together as a single common connection; this is 3V+. The same thing must be done with the column of pairs below "GND;" this is ground. Additionally, the third column from the left, next to ground, needs to be connected together; this is 12V+. A good way to connect a column of pairs is with a piece of lead from a resistor or capacitor.

8. Attach the buzzer. Attach the positive and negative leads from the buzzer to pins 19 and 20 of J2, respectively. These can be soldered on or attached remotely with wires as we did, shown in **photo 10**. We attached the buzzer remotely just in case the buzzer is actually installed outside of the locomotive's shell.

9. Install decoder, battery, and on/off power switch. Connect the decoder wires M1 and M2 to the red and black motor wires in the locomotive. Disregard any track pick-up wires inside the locomotive. Connect the wires in the modified locomotive LEDs labeled L0- and L1- to the green wire labeled F0- and F1-. Both of



9. Common ground and positive-voltage wires are soldered in.



10. Finally, the buzzer (horn) has been added.

the wires in the locomotive labeled L0+ and L1+ need to be connected to either one of the two red wires labeled 12V+. Place the battery inside the locomotive; we connected a Tenergy Li-Ion #18650 14.8V 4400mAh battery to the chassis with double-sided tape.

Now, with double-sided tape, attach the decoder to the battery, with the antenna facing up; make sure the locomotive body will close over the battery and decoder. The red (+) positive and black (–) negative wires from the battery connect to one of the red 12V+ and black ground (GND) wires on the decoder. A switch needs to be added in series with the red positive connections to disconnect the decoder when turned off and when the battery is being charged. Some chassis already have a switch (two out of my three Egg Liners do); for others you will have to add one. Also, the battery wires need to be available somewhere on the outside of the locomotive for recharging. Please note that, though the battery terminals of the decoder are labeled "12V+," any battery from 12V-24V can be used with the decoder.

10. Reattach the locomotive shell. Double check your work, comparing it to the schematic in **figure 1**. Reattach the shell to the chassis and you are ready to go.

Using the decoder

Once connected to the motor, lights, and battery, the decoder will receive commands from a throttle that is on the same radio channel and same locomotive address. By default, the decoder is fixed to control locomotive 3 and receive on radio-channel 2. You can change the locomotive and radio channel at any time you like, which is described in the sidebar, but remember that the locomotive decoder must be programmed with the same locomotive number and radio channel that the throttle is transmitting.

On power-up, the LED on the decoder PCB will flash if it is operating but not receiving wireless DCC, and will stay on solid if DCC is being received properly. A solid LED on the decoder PCB is your indicator that throttle commands are being received. On receiving speed commands, the decoder will drive the motor with 28 speed steps, forward or reverse. The decoder has built-in automatic prototypical momentum that delays acceleration and deceleration, similar to that of an actual locomotive. Even a full direction change will automatically decelerate through a full stop and accelerate back up to the target speed in the opposite direction.

When function 0 is received, the decoder will toggle both the front and rear headlights. Function 1 will toggle the side lights. Pushbutton 2 is momentary and, when held down, will send function 2, which enables the horn (we used a simple buzzer) and, when released, is disabled.

At over 100 feet, the range of the decoder is amazing. Just to be safe, we added a timeout feature, which is actually CV11. This provides an automatic motor shut down if no DCC signal is received after a certain amount of time. We have CV11 set to five seconds. After five seconds, if the decoder does not receive a DCC command to its address, it shuts down the motor automatically

The battery mentioned earlier, along with the modern motor-driver circuit we designed, provides plenty of run time between charges and, at 14.8V, has fantastic performance. The average current consumption we measured while operating at full speed is around 200mA, but does depend on speed, load, and grade of track. This means that, with 4400mAh of capacity, a locomotive can run at full speed continuously for more than 20 hours from a single charge. At normal speeds and occasional use, you can expect months between charges.

The power switch will turn the decoder on or off by connecting or disconnecting it from the battery. The decoder needs to be disconnected from the battery while the battery is being charged. Happy railroading!

Online extras

Registered users: Cut and paste the code for the MCU from our website. Visit *www.GardenRailways.com* and click on "Construction & landscaping" under "How to."

Mark Buccini answers a question about radio frequency

Radio Channel	Frequency
0	921.37MHz
1	919.87MHz
2	915.37MHz
3	912.37MHz
4	909.37MHz
5	907.87MHz
6	906.37MHz
7	903.37MHz
8	926.12MHz
9	924.62MHz
10	923.12MHz
11	918.12MHz
12	916.87MHz
13	913.62MHz
14	910.87MHz
15	904.87MHz

Reader-submitted question: I have 2 working throttles and 1 working decoder. I have a pair set for Loco 4 Freq 6 and the second on Loco 5 Freq 2. Not in locos yet just on the bench. I stupidly reversed the input polarity on the second decoder so don't know how much damage I did yet. When you get old you do stupid things! We have an outdoor G scale display here at our retirement community.

At any rate, we have a CVP 5000 throttle and 3 G3 receivers in three of our locos. I have not been able to get any response to or from your designs and CVP.

Can you tell me the exact frequencies that you have used for freqs 2, 3 and 6 ? Would I need to name a loco "4" or "5" plus the correct freq to talk to one of our G3's ?

My PC will not link to the Launch Pad which would likely tell me the freqs.

Response: In terms of the radio frequencies, for the throttle and decoder you built, by design the are compatible with QSI and CVP. And they have been tested for compatibility with both QSI and CVP.

The throttle and loco need to have the same radio channel and loco number.

A table of the radio channel versus frequency is included here. The default for the throttle and decoder are radio channel 2, you can select radio channel 3 or 6 using jumpers are described in the articles.