The Very Basics of

^ Controlling Your Layout with Arduino

Steve Sweeney

stevesweeneywisc@hotmail.com

Steve Sweeney – Professional Bio

Education

- Bachelor of Science – Computer Science Engineering
  - University of Nebraska - Lincoln
- Masters Degree – Business Administration
  - Marquette University

Work History

- Microsoft, Cabela's, Harley-Davidson Motor Company, AT&T

Basically….a Computer Nerd

Steve Sweeney – Train Bio

Through The Years

- The first set –the "Christmas HO oval"
- College
  - Joined a club near SEA-TAC airport while at Microsoft
  - "Portable HO" – 4'x8' (in 2 foot sections)
- Recent Reintroduction – Thank You Santo Mortillaro and Al Richter
  - Madison 400 (Operations on Wundrock and Weber layouts)
- Big Al's Layout – HO 6' x 11' HO (dual mainline, yard, turntable, industry spurs)
- Portable Layout - John Allen's Timesaver (N-Scale)
- Permanent Home Layout - Nebraska Central (multi-deck N-Scale)

The VERY Basics of Controlling Your Layout with an Arduino

Model railroaders expect a lot from electronics. They want automation, sound, lighting, signaling, and relay controls to automate and simplify operations and have fine control over their layout. This clinic will cover the basics of the “many hats” an Arduino can wear when it comes to layout control (i.e. besides direct connected sensors, controls and lights, it can be a DCC decoder, LCC node, C/MRI node, Loconet node, WiFi Throttle, Speedometer, etc.)

What's Inside This Clinic

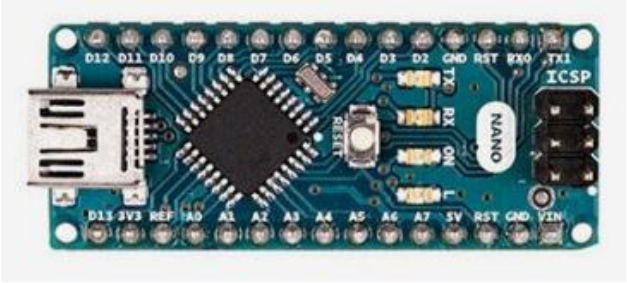
- Hardware – Quick overview of the many different options
  - What is an Arduino (and "other" hardware), Where to get them, How much they cost, etc.
  - Not going to show in detail how to implement specific solutions
- Arduino Solutions –
  - Examples of the many things you can do with an Arduino
- Where to get Inspirations – links provided throughout presentation
  - Model Railroading with Arduino (https://www.mrrwa.org/)
  - Geoff Bunza (https://model-railroad-hobbyist.com/blog/geoff-bunza)
  - Michael Adams (http://www.utrainia.com/65-arduinocmri-and-rs485)

What is an Arduino

- Microcontroller – Very small, runs one program, affordable, “easy” to program
- How does this compare to a PC? –
  - Original IBM PC – 4.77 MHz usually delivered with 16KB of memory (could expand to 256KB)
  - Arduino Nano – 16 MHz with 16KB or 32KB of memory

Uno

Nano



Where to get an Arduino

- Where to buy? –
  - Arduino – https://www.arduino.cc/
    - Nano – Quantity 3 - $28.20
    - Uno – Quantity 1 - $22.00
  - Amazon –
    - Nano – Quantity 3 - $12.99
    - Uno – Quantity 1 - $16.98
- Open-Source Hardware and Software
  - Arduino allows for the study of their hardware to understand how it works, make changes to it, with expectation any changes are shared back with the community.
  - Arduino releases all of the original design files (Eagle CAD) and allows for both personal and commercial derivative works, as long as they credit Arduino and release their designs under the same license.
  - The Arduino software is also open-source.
- Many Different Models
  - Uno, Nano, Mega, Micro, Portenta, MKR, Yún, Due, Leonardo, Zero

What Makes The Arduino Versatile

- Multiple Input/Output Pins
  - Example Interfaces
    - Push Buttons
    - Photocells
    - Current Detectors
    - Servo Motors
    - Switches (rocker, toggle)
    - Stepper Motors
    - MP3 Playback Chips
    - RFID Readers
- Network Capable –
  - Transmit (TX) - Receive (RX) pins
  - SPI (Serial Peripheral Interface)
    - Other Arduinos and/or Computers
    - WiFi
    - Ethernet
    - Bluetooth

Layout Control

- DCC – Direct Command Control (NMRA Standard)
- LCC – Layout Command Control (NMRA Standard)
- C/MRI – Computer (Chubb) Model Railroad Interface (NMRA Standard)
- LocoNet – Digitrax's commercial offering
- NCE – Well…it's NCE (artist formerly known as North Coast Engineering)
- WiFi Throttle – Connects directly to JMRI WiFi server.

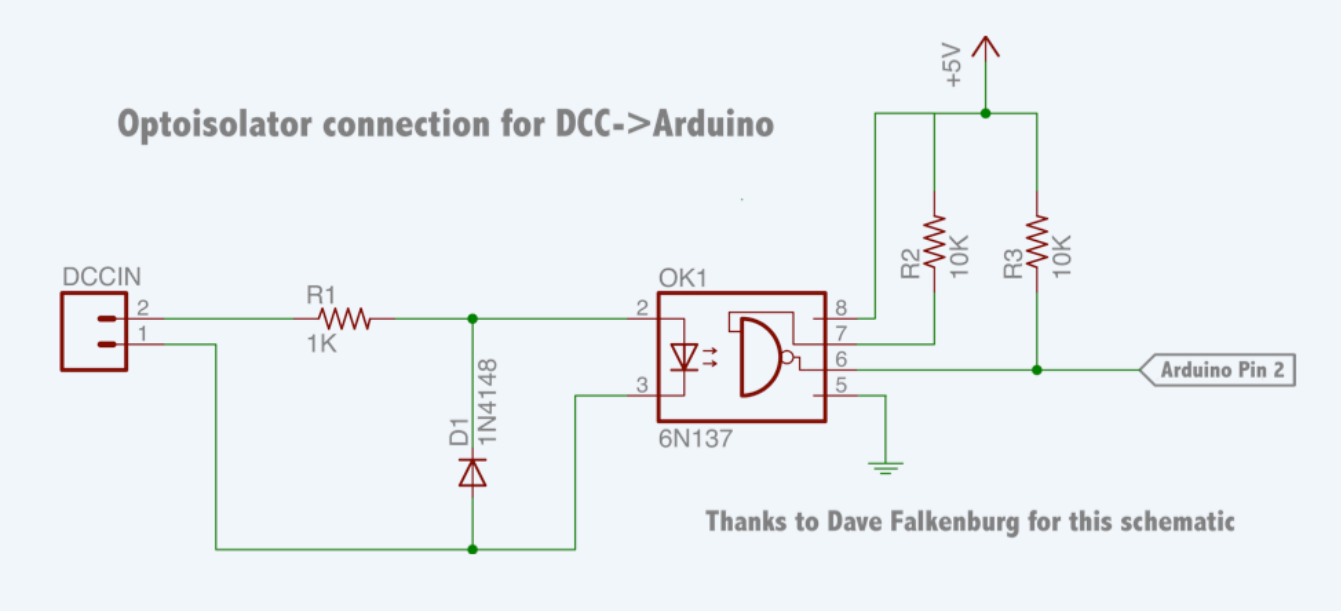
  Not dependent on the DCC base station connected to JMRI.
- DCC Base Station – DCC Command Station

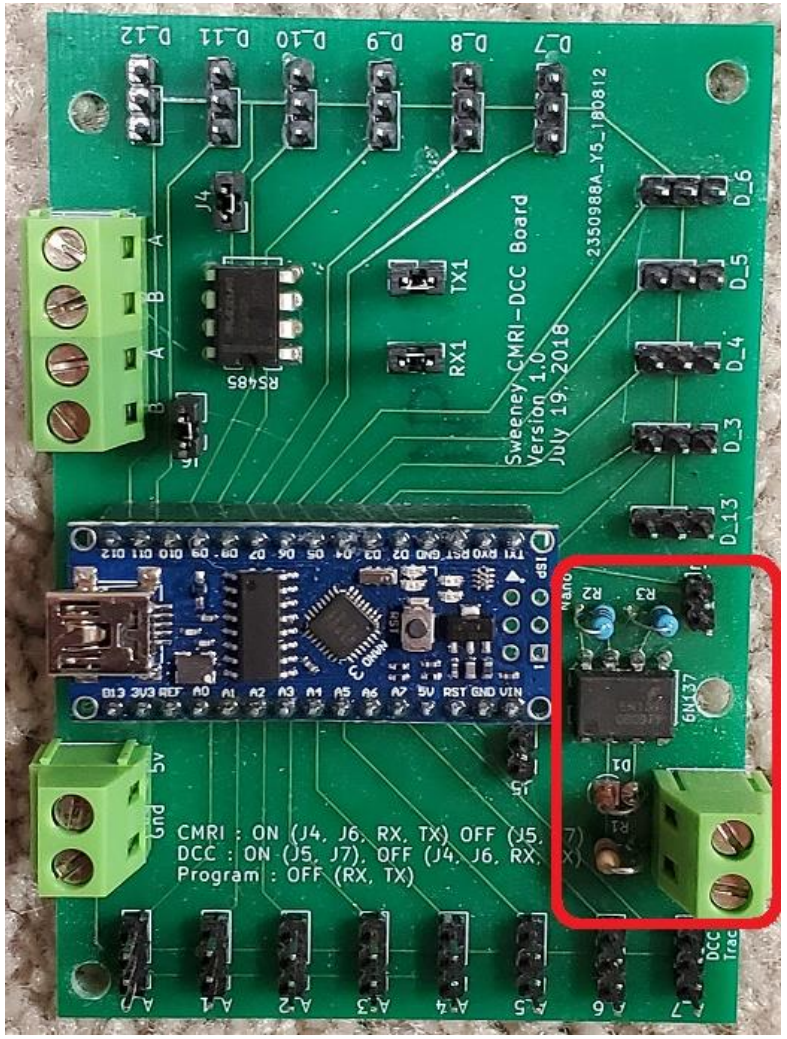
DCC (Direct Command Control)

- NMRA – https://www.nmra.org/index-nmra-standards-and-recommended-practices
  - S-9.2 DCC Communications Standard (Adopted July 2004)
  - This standard covers the format of the information sent via Digital Command Stations to Digital Decoders. A Digital Command Station transmits this information to Digital Decoders by sending a series of bits using the NMRA digital signal described in S-9.1. This sequence of bits, termed a packet, is used to encode one of a set of instructions that the Digital Decoder operates upon.
- MynaBay – http://www.mynabay.com/dcc_monitor/
  - Software development company (Apple, Android, Windows, Hardware)
  - Open Source DCC Library: https://github.com/MynaBay/DCC_Decoder
  - Software Examples: Accessory Decoder & Monitor
- Geoff Bunza – https://model-railroad-hobbyist.com/blog/geoff-bunza?page=2
  - SMA10 - Build a 17 Function DCC Decoder for $5
  - SMA12 - 17 Channel Configurable Multifunction $5 DCC Decoder For Servos

DCC (Direct Command Control)

- Sweeney Board – How I did it
  - 6n137 Optocoupler $0.76
  - 1N4148 Diode $0.05
  - 1K Ohm Resistor $0.06
  - 10K Ohm Resistor $0.06





LCC (Layout Command Control)

- NMRA – https://www.nmra.org/lcc (first standards adopted Feb 2016)
  - Goal of LCC is similar to DCC, create a standard for manufacturers to follow and interoperate
  - Take some of the burden off the DCC bus
- OpenLCB – https://openlcb.org/
  - A group that is developing the necessary code, protocols to successfully implement LCC
  - Utilizing Open Source for development
- OpenMRNLite – https://www.arduinolibraries.info/libraries/open-mrn-lite
  - OpenMRN is the extensible implementation of the OpenLCB protocol suite. The Lite version has been adapted to work with the programming model and drivers of the Arduino ecosystem.

CMRI (Computer Model Railroad Interface)

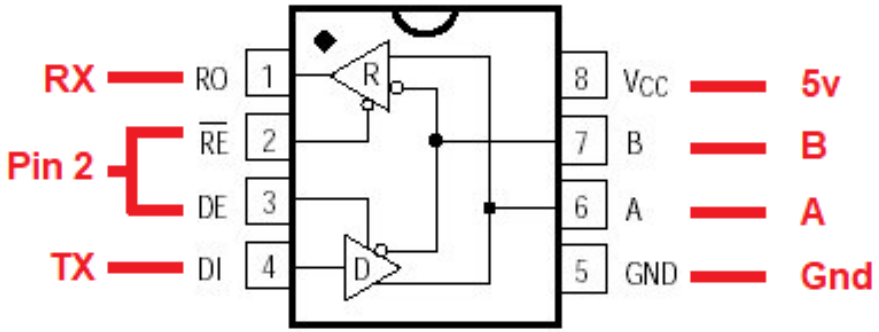
- NMRA – Layout Control Specification
  - https://www.nmra.org/sites/default/files/standards/sandrp/Other_Specifications/lcs-9.10_cmri_intro_v1.0.pdf
  - https://www.nmra.org/sites/default/files/standards/sandrp/Other_Specifications/lcs-9.10.1_cmrinet_v1.1.pdf
- CMRI – https://www.jlcenterprises.net/
  - The C/MRI system was created by Dr. Bruce Chubb in 1985, and introduced to the model railroad community through a 16-part series of articles in Model Railroader magazine.
  - In 2014 the C/MRInet Protocol was adapted as a Group Standard by the NMRA and listed in their standard's section (as listed above)
  - Hardware and software designs are Open Source with excellent documentation.
  - There is an NMRA Special Interest Group, CMRI SIG, providing information and discussion regarding C/MRI.

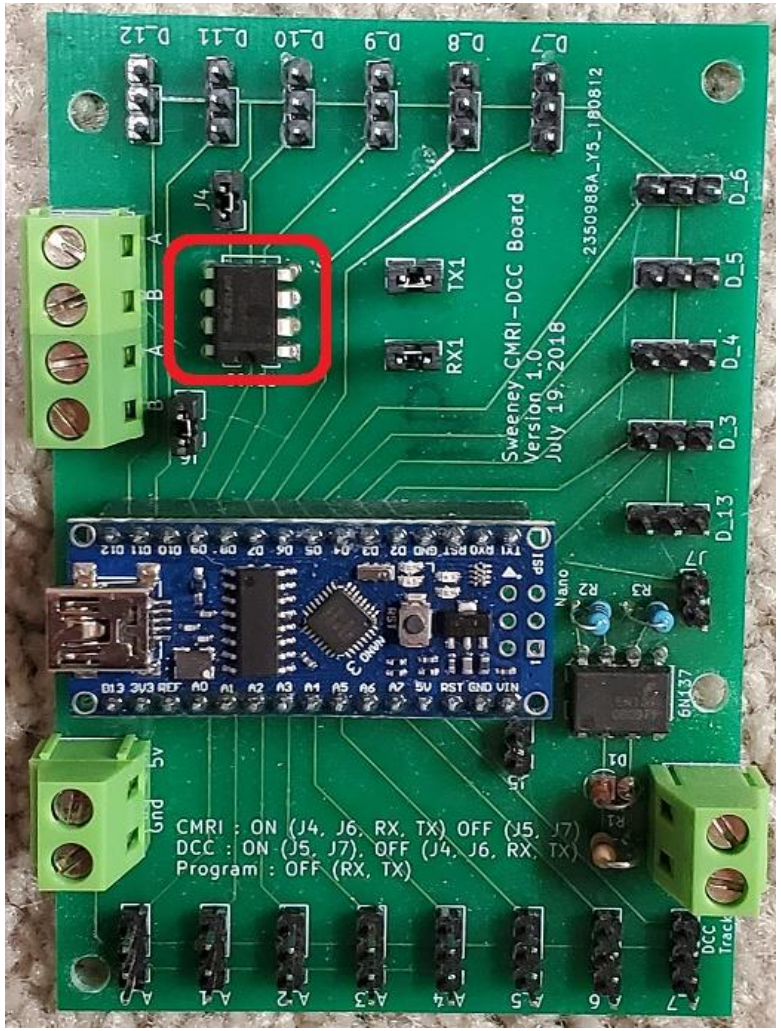
CMRI (Computer Model Railroad Interface)

- ArduinoCMRI – https://github.com/madleech/ArduinoCMRI
  - Library for connecting your Arduino with JMRI by emulating Bruce Chubb's Computer/Model Railroad Interface (C/MRI) System. Provides maximum flexibility to tailor your solution to fit your needs
  - Features:
    - Simple API that handles GET, SET, and POLL requests from JMRI automatically.
    - Easy access to input and output data.
    - Emulates an SMINI up to a SUSIC with up to 2048 digital lines available.
    - Error tolerant.

CMRI (Computer Model Railroad Interface)

- Sweeney Board – How I did it
  - MAX 485 $0.26
  - RS-485 to USB Transceiver $1.10



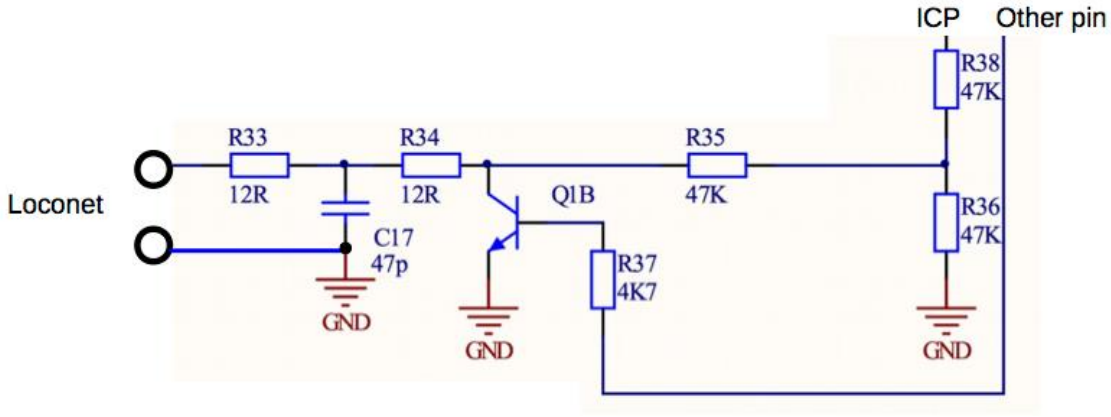


LocoNet

- LocoNet – Open Source Library
  - The LocoNet Arduino library can be found in the Arduino Library Manager or downloaded from: https://github.com/mrrwa/LocoNet/archive/master.zip
  - To interface the Arduino to a LocoNet network, you need to use an interface circuit like the one shown in the schematic below, which came from John Plocher's SPCoast website where he has designed several LocoNet Shields. Alternatively for personal experimenting with LocoNet there is also a Minimal LocoNet Interface shown further down the page.
- Examples in Repository –
  - Block Occupancy Detector
  - Linx (PC interface to LocoNet)
  - Control Panel (buttons, LEDs, throw a turnout)
  - Fast Clock
  - Throttle
  - Packet Monitor

LocoNet

- Connection to Arduino –
  - The RX input to the AVR ICP pin which on an Arduino UNO is digital pin 8.
  - The TX output can be any Arduino pin, but the LocoNet library defaults to digital pin 6 for TX.

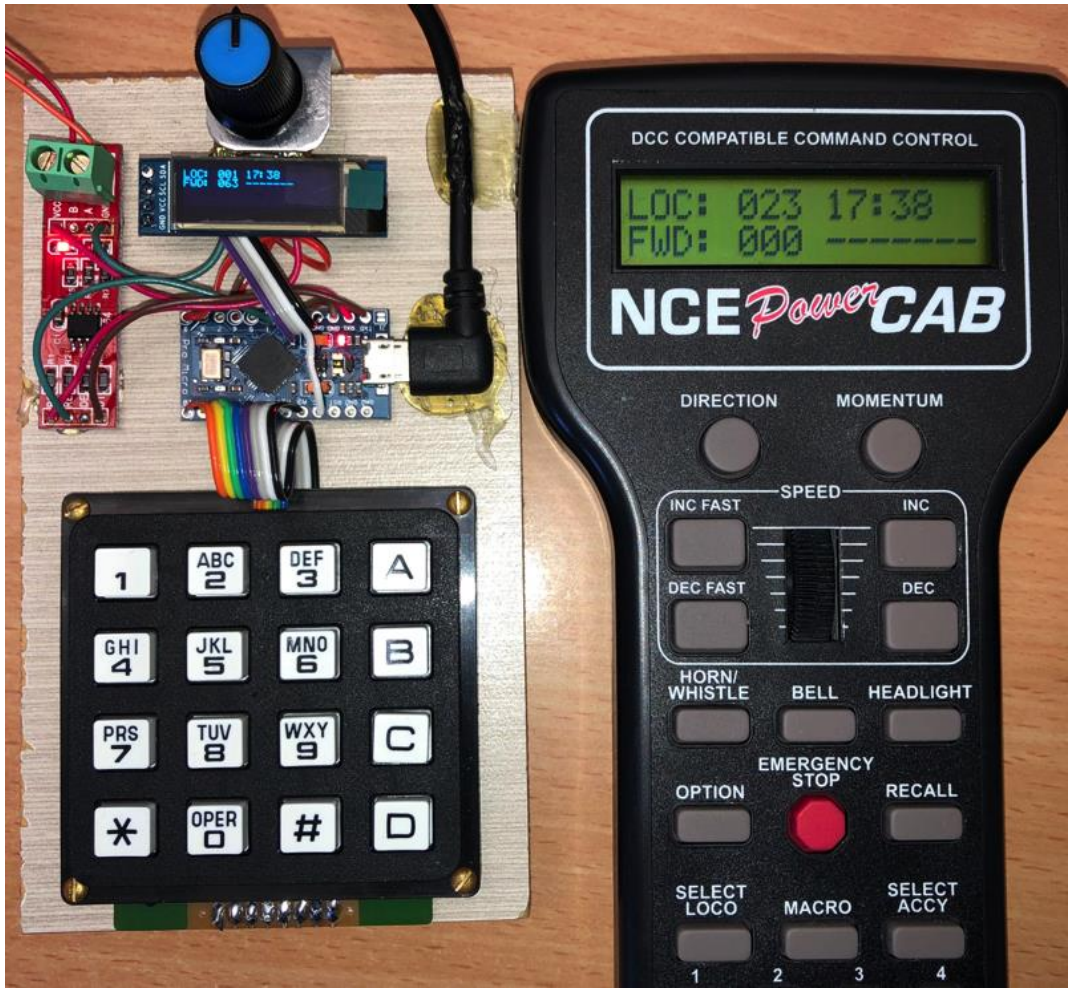


NCE

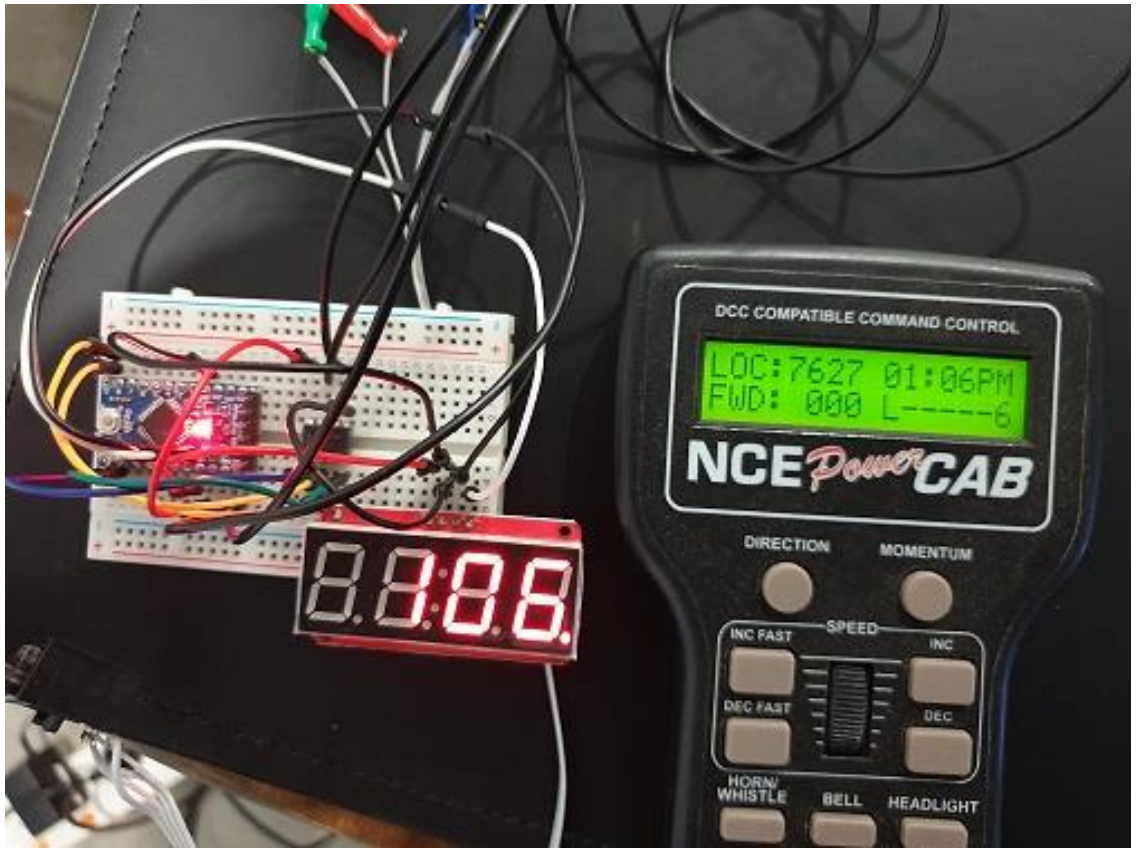
- NCE Cab Bus – Open Source Library
  - The NCE Cab Bus Arduino library can be found in the Arduino Library Manager or downloaded from: https://github.com/mrrwa/NceCabBus
- Examples in Repository –
  - Throttle - Has a 4×4 keypad, a potentiometer to set the speed and a 128 x 32 pixel OLED display to display the 2×16 lines of text that would usually be displayed on a NCE Throttles's LCD display.
  - AIU - Emulates an Auxiliary Input Unit (AIU) using 14 Arduino input pins and an RS485 interface.
  - FastClock - Decodes the FastClock information on the Cab Bus and writes it out as Text on the Arduino Serial port. It would take minimal effort to instead connect it up to one of the many large LED / OLED display modules.

NCE

Throttle



FastClock



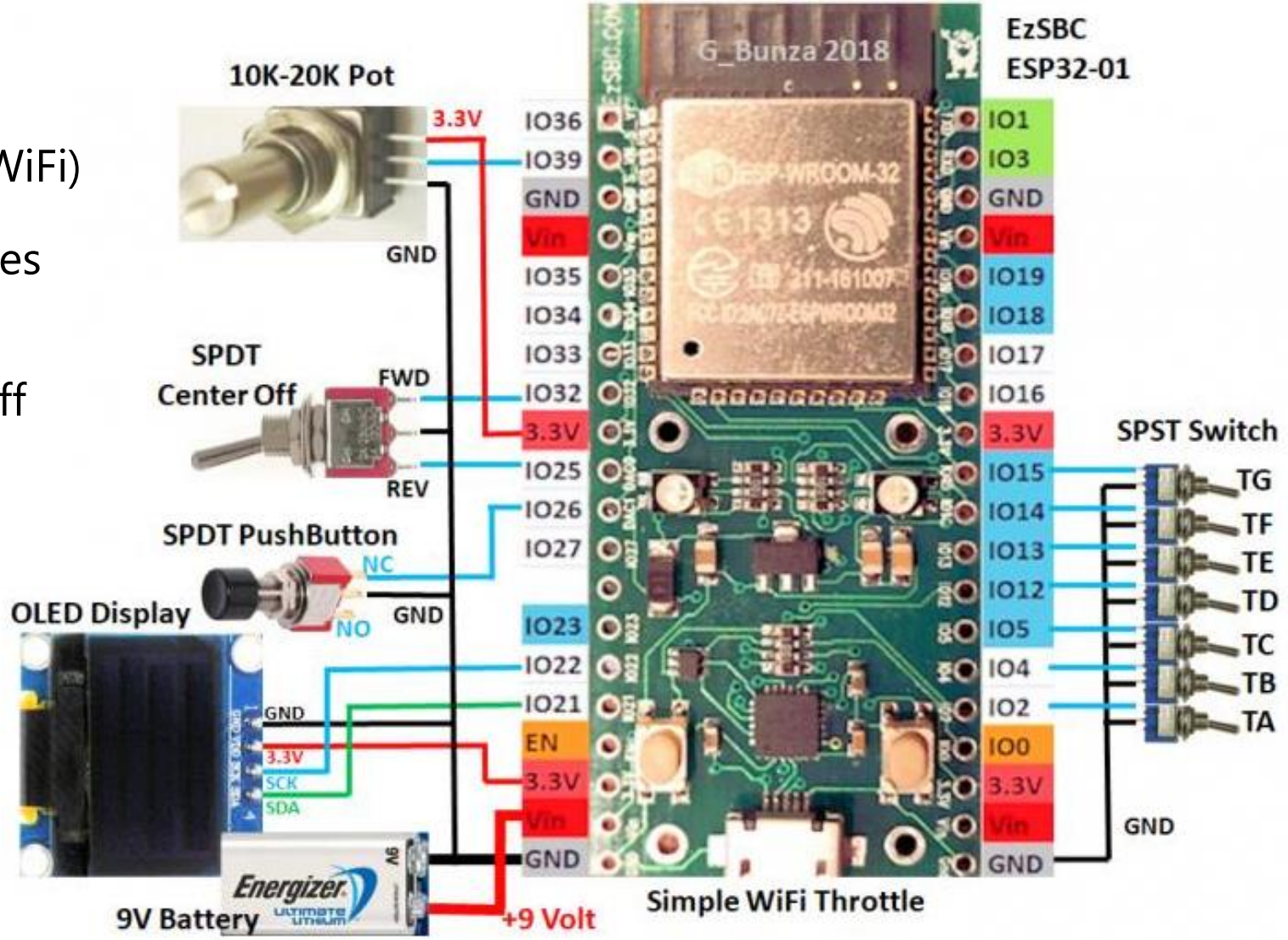
WiFi Throttle (requires JMRI)

- Geoff Bunza – https://model-railroad-hobbyist.com/node/35652
  - Premise – If on your layout, you are running JMRI, then you have the capability of connecting your cell phone or wireless tablets as a remote throttle for DCC control via the JMRI WiFi server.
  - End Goal – Create a simplified throttle, with a speed control knob. Reduces the possibility of inadvertently changing things by a newcomer or a visitor.
  - Features –
    - Connects directly to the JMRI WiFi server, and therefore is NOT dependent on your DCC base station connected to JMRI.
    - Supports loco address selection
    - Six (or more) function switches
    - Speed and direction control
    - Runs on a standard nine volt battery

WiFi Throttle

WiFi Throttle

- Parts List –
  - ESP32-01 ("special" Arduino with WiFi)
  - SPDT Sub Miniature Toggle Switches (Qty 6)
  - SPDT Mini Toggle Switch Center Off
  - 10K Linear Mini Potentiometer
  - Knob for Potentiometer
  - 0.96 Inch OLED Display
  - 9 volt Battery Holder Snap



DCC++ Open Source DCC Base Station

- What is it? https://github.com/DccPlusPlus
  - The DCC++ is an open-source hardware and software system for the operation of DCC-equipped model railroads.
  - The system consists of two parts, the DCC++ Base Station and the DCC++ Controller.
  - The DCC++ Base Station consists of an Arduino micro controller fitted with an Arduino Motor Shield that can be connected directly to the tracks of a model railroad.
  - The DCC++ Controller provides operators with a customizable GUI to control their model railroad. It is written in Java using the Processing graphics library and IDE and communicates with the DCC++ Base Station via a standard serial connection over a USB cable or wireless over BlueTooth.
- TrainElectronics – Dave Bodnar https://model-railroad-hobbyist.com/node/25429
  - DCC++ Open Source DCC Project, DCC++ Infrared Throttle & Point-to-Point Controller, High Power DCC Booster, and Current Sensors

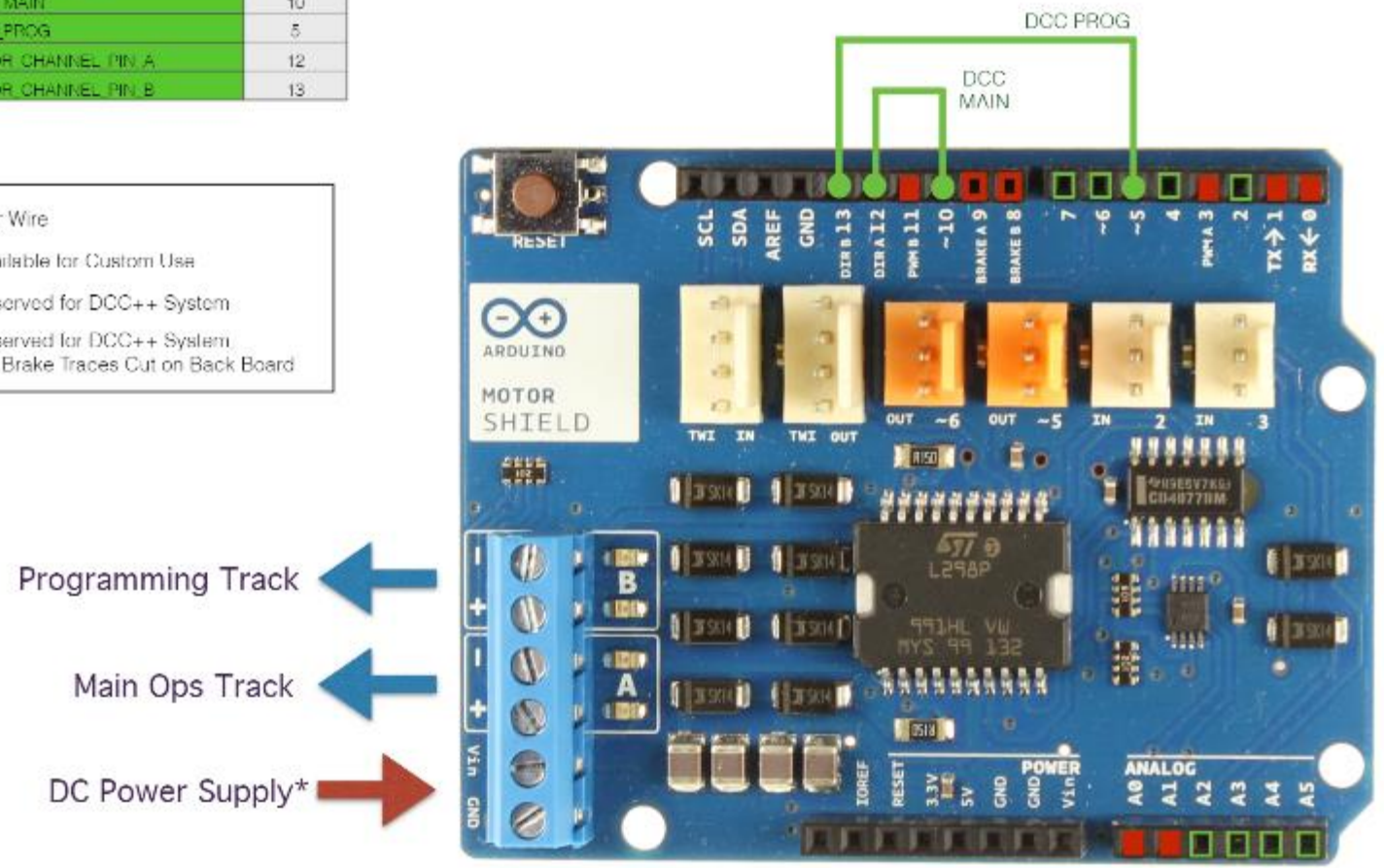
DCC++ Open Source DCC Base Station

- Motor Shield
  - Amazon - $28.89

DCC++ Base Station Signal Name	Arduino Motor Shield
SIGNAL_ENABLE_PIN_MAIN	3
SIGNAL_ENABLE_PIN_PROG	11
CURRENT_MONITOR_PIN_MAIN	A0
CURRENT_MONITOR_PIN_PROG	A1
DCC_SIGNAL_PIN_MAIN	10
DCC_SIGNAL_PIN_PROG	5
DIRECTION_MOTOR_CHANNEL_PIN_A	12
DIRECTION_MOTOR_CHANNEL_PIN_B	13

Pin Mappings for Arduino UNO with Arduino Motor Shield

- Jumper Wire
- Pin Available for Custom Use
- Pin Reserved for DCC++ System
- Pin Reserved for DCC++ System Unless Brake Traces Cut on Back Board



Thank You !!

Questions/Answers

Model Railroading with Arduino

- MMRwA – https://mrrwa.org/
  - Main Menu on Website
    - DCC Interface, Loconet Interface, NCE Interface
  - Articles on Website
    - NCE Cab Bus Supported
    - Locobuffer
    - Loconet Sound Module
    - Loconet Throttle
    - Grade Crossing Control
    - DCC Turntable Stepper Motor Driver