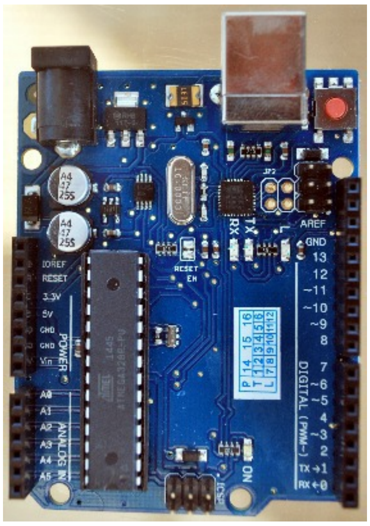
Arduino Turntable MkII

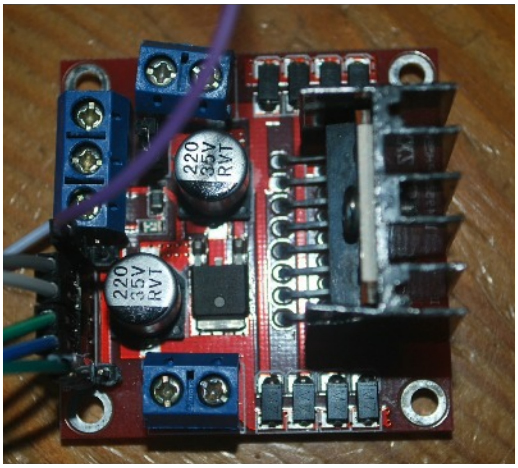
The original turntable design was manually controlled because suitable track registration was found to be difficult to achieve automatically. This was because the stepper motor (28BYJ-48) was geared and the slop in the gears made accuracy to ‘N’ Gauge requirement impossible. Some time was wasted trying Hall Effect transistors, reed relays and photocells to no avail.

The MkII design uses a Nema17 motor type 17HS13-0404S directly driving the turntable bridge. This motor has a step angle of only 1.8 degrees, therefore, getting this down to 0.9 degrees when half-stepping. This approach has proved most effective in practice.

Further development of the basic turntable has allowed control via infra red and even DCC. The reason for the DCC control is because my layout is fully computer controlled and automation of the turntable’s operation was a natural on-development. Button or keyboard control is also an option that can be applied to this design as can an IC2 bus LCD readout be added.

The basic components comprise an Arduino, H Bridge (L298N ZV) and Nema17 stepper motor all applied to the basic Peco turntable.





The wiring connections are straight forward:

Nema17 Green and Black to either of the two-terminal connection block
Nema17 Red and Blue to the other of the two-terminal connection block
In1 on H Bridge to Pin 2 on Arduino
In2 on H Bridge to Pin 3 on Arduino
In3 on H Bridge to Pin 4 on Arduino
In4 on H Bridge to Pin 5 on Arduino

The terminal nearest the edge of the Hbridge’s three-terminal connection block connects to 12v+ regulated power source of at least 1 amp. capacity.

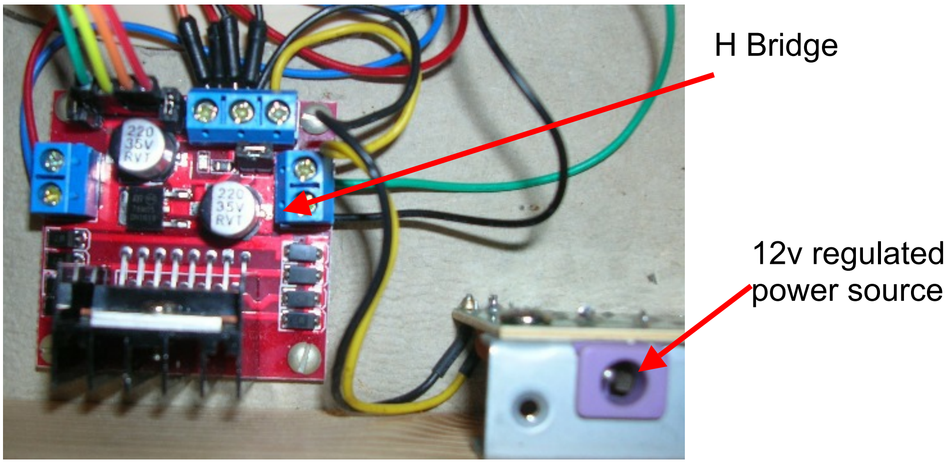
The centre terminal of the Hbridge’s three-terminal connection block connects to 12v- and a GND connection on the Arduino.

The inner-most terminal of the Hbridge’s three-terminal connection block provides a 5V source to the Arduino when disconnected from the USB on a computer.

Pin 6 of the Arduino connects to the ‘S’ (signal) connection of the IR sensor.

The remaining power connections on this sensor will go to the GND and 5V connections on the Arduino.

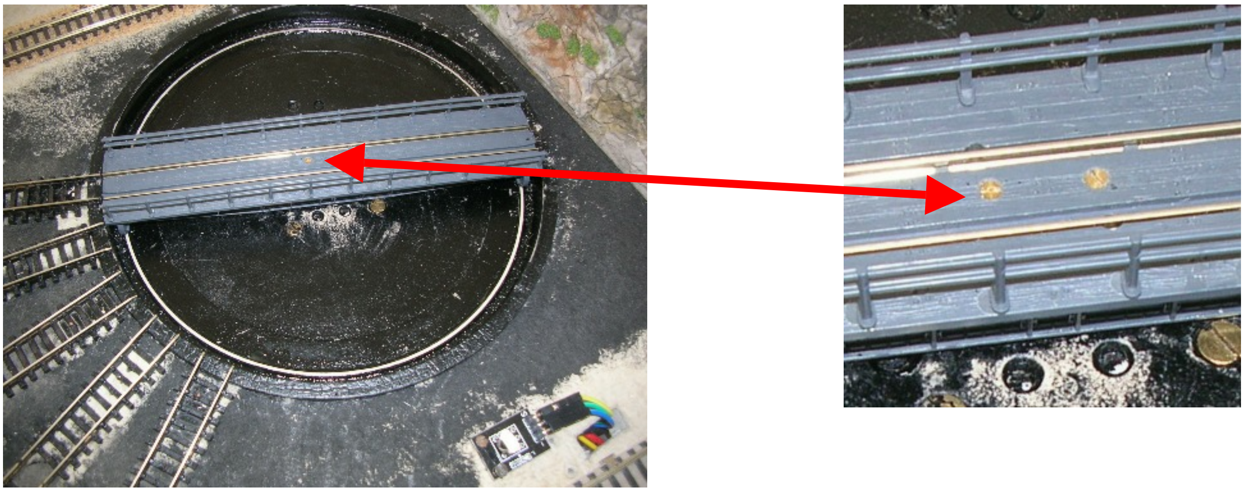
The Arduino connects 7 to 12 provide connections to buttons or flash switches enabling track selection via buttons if required.



Peco Turntable Preparation

The Peco N-55 turntable has a physical limitation encountered during construction of the Bridge. Ensuring that the bridge is absolutely horizontal turned out to be a problem in practice. Gluing the hub to the underside of the bridge proved to be inadequate. Instead, the hub was bolted to the bridge with two 12BA nuts and countersunk bolts. This allows a shim to be placed between one or other side of the hub and the bridge in order to level the track bed. In my case, a small piece of tissue paper on one side was all that was required. The magnification effect at the end of the bridge gave the exact vertical alignment which was, before-hand, almost 1 mm out at the outer edge.

Once assembled, the hole in the hub should be ‘reamed’ out with a standard 5 mm drill. This will give a good interference fit to the stepper motor shaft.



The Peco bearing retainer ring is not used. The centre hole in the turntable is opened up with a circular file to ensure no contact between the hub and the turntable tray. This way the stepper motor bearings take the load without any friction from the turntable tray.

Four holes are drilled in the turntable tray to align with the Stepper motor fixing holes and located to ensure that the motor drive spindle is exactly central. These holes form a 31mm square and a little geometric drawing on the turntable tray is advised to give the exact location for the 4-m3 screws. This is probably the most critical part of the project to get right.

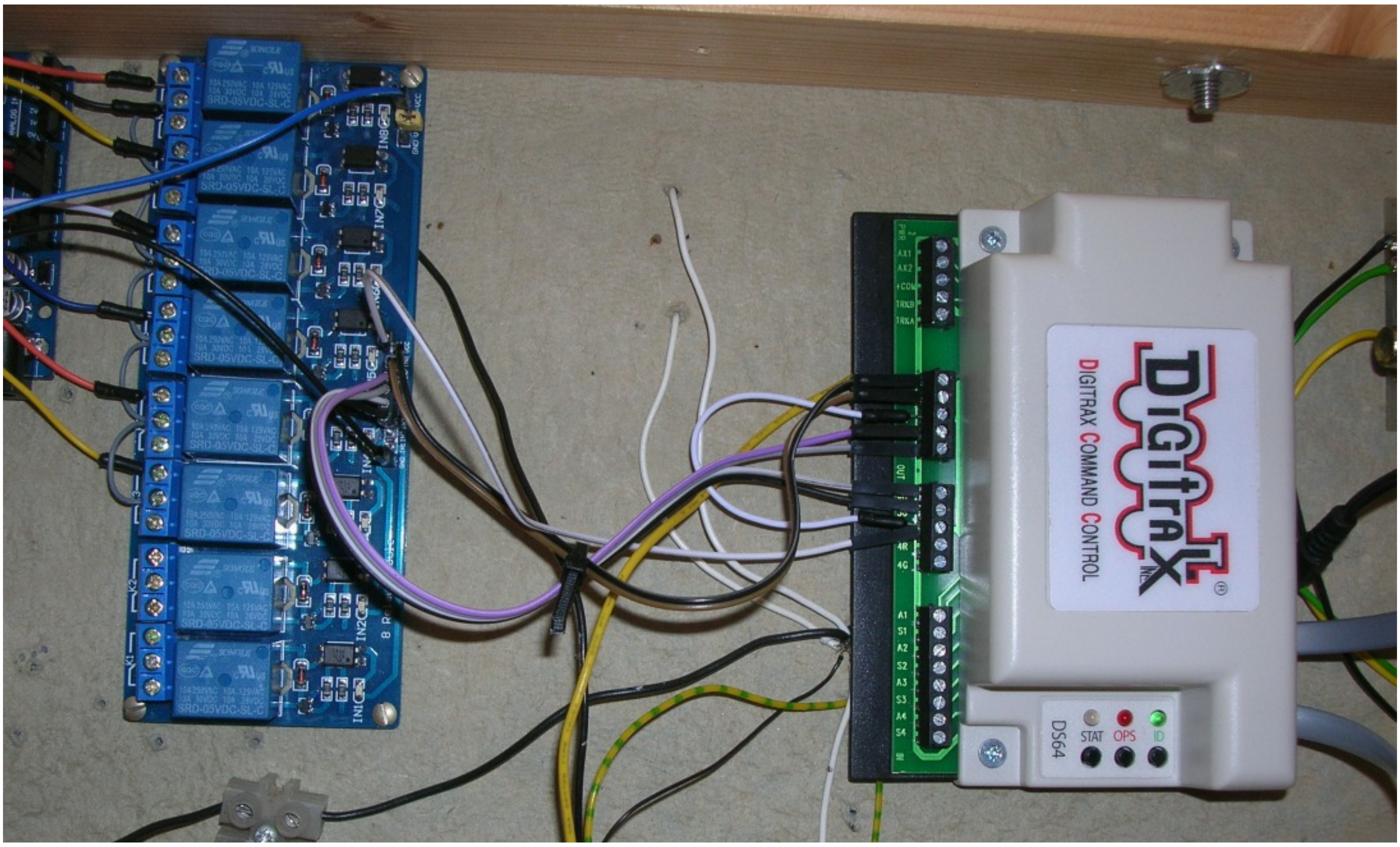
Aligning the Tracks and the Bridge

The software gives turntable bridge positions approximately one-half ‘N’ Gauge track apart. However, for alignment accuracy to be perfect the bridge should be initially ‘zeroed’ using the IR remote ‘OK’ button or whichever button is set to track 0. Rotate the turntable chassis to align the bridge with track zero. Secure the chassis position e.g. drill through the outer lip of the chassis with 1mm drill and use track pins. Using the IR Remote move the bridge to track 1 position and align the track for track 1 to the bridge. Continue this process for each track. Normally one thinks of aligning the bridge to the tracks but in this case it is the other way round and this method preserves the accuracy in the simplest way.

DCC Adaptation

The object of this adaptation is to allow the Turntable to be controlled from a DCC controller or Throttle.

The design uses a bank of relays where the contacts control the voltages on pins 7 to 12 on the Arduino. The relays are controlled by a DCC point controller. In this example, a Digitrax DS64 point controller is used.



The ‘Common’ connections on the DS64 are connected to the GND on the Relay Bank. Each DS64 output is wired to a relay winding connection.. The GND on the Relay Board is connected to the Arduino GND connection - a GND connection terminal will have to be added to accommodate all the GND connections to the Arduino.

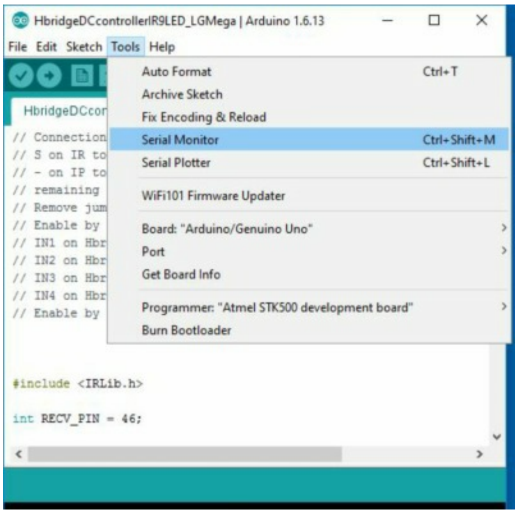
The operational concept is that when a ‘point’ is selected on the DCC Controller, it closes a relay and the relevant relay contacts connect an Arduino pin (pins 7 to 12) to GND or ‘LOW’. This in turn sets the turntable off to seek the requested track.

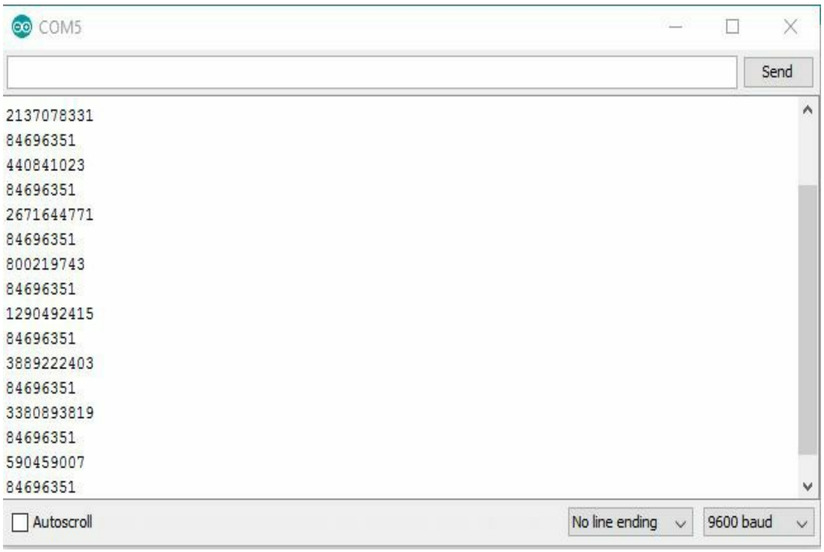
The DS64 is programmed to hold the relay closed for 1.6 seconds in order to avoid contact bounce.

I expect other DCC point controllers could do the same job but being a through-and-through Digitrax fan for over 10 years, I have used a DS64 unit.

IR Remote Codes

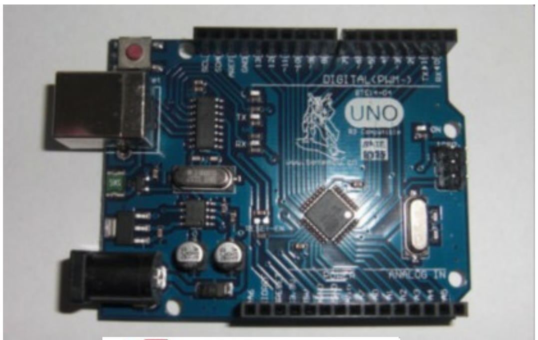
Infra-red remotes output a stream of numbers, different for each button pressed. The first task is to discover what each of these numbers are. To do this, a simple program (or ‘sketch’ to use the Arduino parlance) is required. The Arduino programming software, downloadable free and safely from https://www.arduino.cc/en/Main/Software for Windows, Linux or Mac, has a facility called the ‘Serial Monitor’ that will allow these codes to be viewed. From this a list of what numbers are produced by each button can be prepared. Once these codes are known then it will be possible to proceed to writing the sketch for the IR Train Controller.





There is often a repeating code that outputs even when a button is not being pressed and these should be ignored. It will be obvious which is the correct number at the point of pressing the relevant button.

Parts List



UNO R3 ATmega328P, CH340T USB Arduino Compatible Microcontroller Board

£5.45

Dual Bridge L298N Stepper Motor Driver Controller Board Module

£3.27

Nema17 Stepper Motor 26Ncm/37oz.in 12V 0.4A

£9.99

Remote Control and Sensor

£2.59

10pcs Dupont Cables M-F, M-M, F-F Jumper Breadboard Wire GPIO Ribbon

£1.20 per type

AC 100-240V to DC 12V/2A Converter UK Regulated Power Supply Adapter Transformer = 5.5 mm x 2.1 mm plug

£6.79

The above prices are taken from eBay searches and are quoted to show how inexpensive these parts can be.

I do not expect the magazine staff to deal with queries and advice on the Arduino and its programming. An example sketch (program) for this Arduino application, infrared library and the sketch for reading the Turntable MkII codes are all available for download from http://www.modeltraincatalogue.com/arduinoproject.html by clicking the ‘Turntable MKII’ button. The example sketch will require tailoring to the individual IR Remote’s requirements. I am happy to correspond on this particular project through email: support@modeltraincatalogue.com.