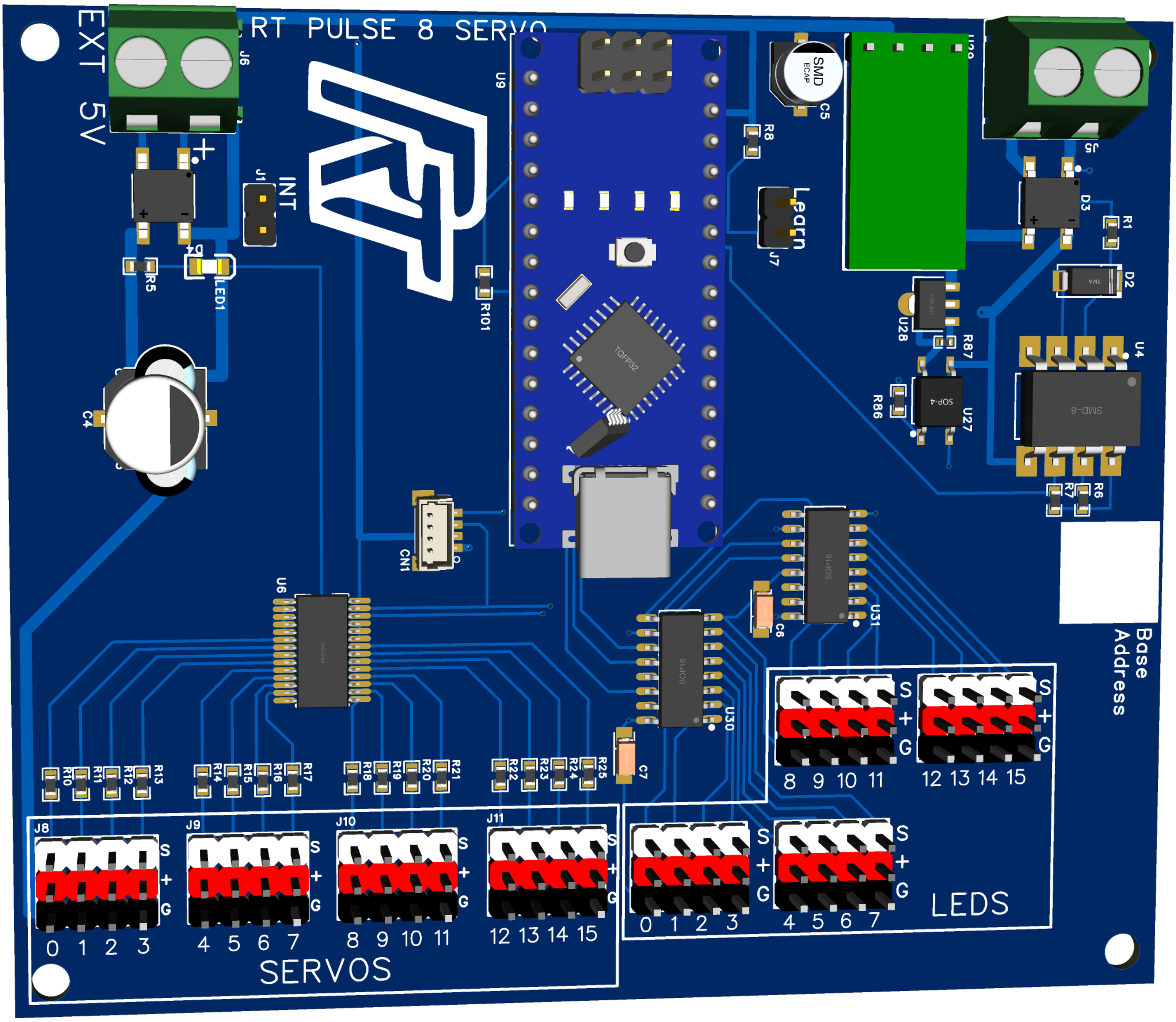
OPERATING MANUAL

Model Railroad DCC accessory decoder.

This board is a DCC accessory decoder for servos.

This document describes the operation of the board.

Please read through all of this document before use.



In use.

If you have purchased a complete/tested RT_Pulse_8_Servo continue on the next page.

Using this firmware on github:

https://github.com/Rosscoetrain/DCC-Turnout-Decoder-Servo

The board will control 16 servos.

The firmware currently needs to be uploaded twice to the Arduino Nano to ensure the eeprom on the board is setup correctly.

Please read the instructions in the defines.h file.

Open the firmware in the Arduino IDE.

Un-comment the line in the defines.h file as described there. (Line 30 - 34)

Upload the firmware to the Arduino Nano.

On the serial monitor you should see: 11:48:31.374 -> Resetting CVs to Factory Defaults

Comment out the line in the defines.h file as described there. (Line 30 - 34)

Upload the firmware again to the Arduino Nano.

Using the serial monitor enter the following command.

```
<>
```

You will then see a response like this:

```
17:40:32.025 -> CVs are:
17:40:32.025 -> CV1 = 1
17:40:32.025 -> CV9 = 0
...
```

All is now ready.

Connection to the layout.

How you connect to your layout is really dependent on your setup.

This is a how to connect to a DCC-EX command station with separate power supply for the servos.

The DCC track is connected to the DCC Input connector on the decoder.

The power supply can be 5 - 6V DC or 5 AC and is connected to the PWR IN connector on the decoder.

In most cases you do not need an external power supply and the servos can be powered by the DCC input. Place a jumper on the INT header. See image on page 1.

WARNING

If you are using different power supplies for your command station and a DC power supply for this decoder. The power supplies must have a common ground.

Do not have an external power supply connected if you have a jumper on INT.

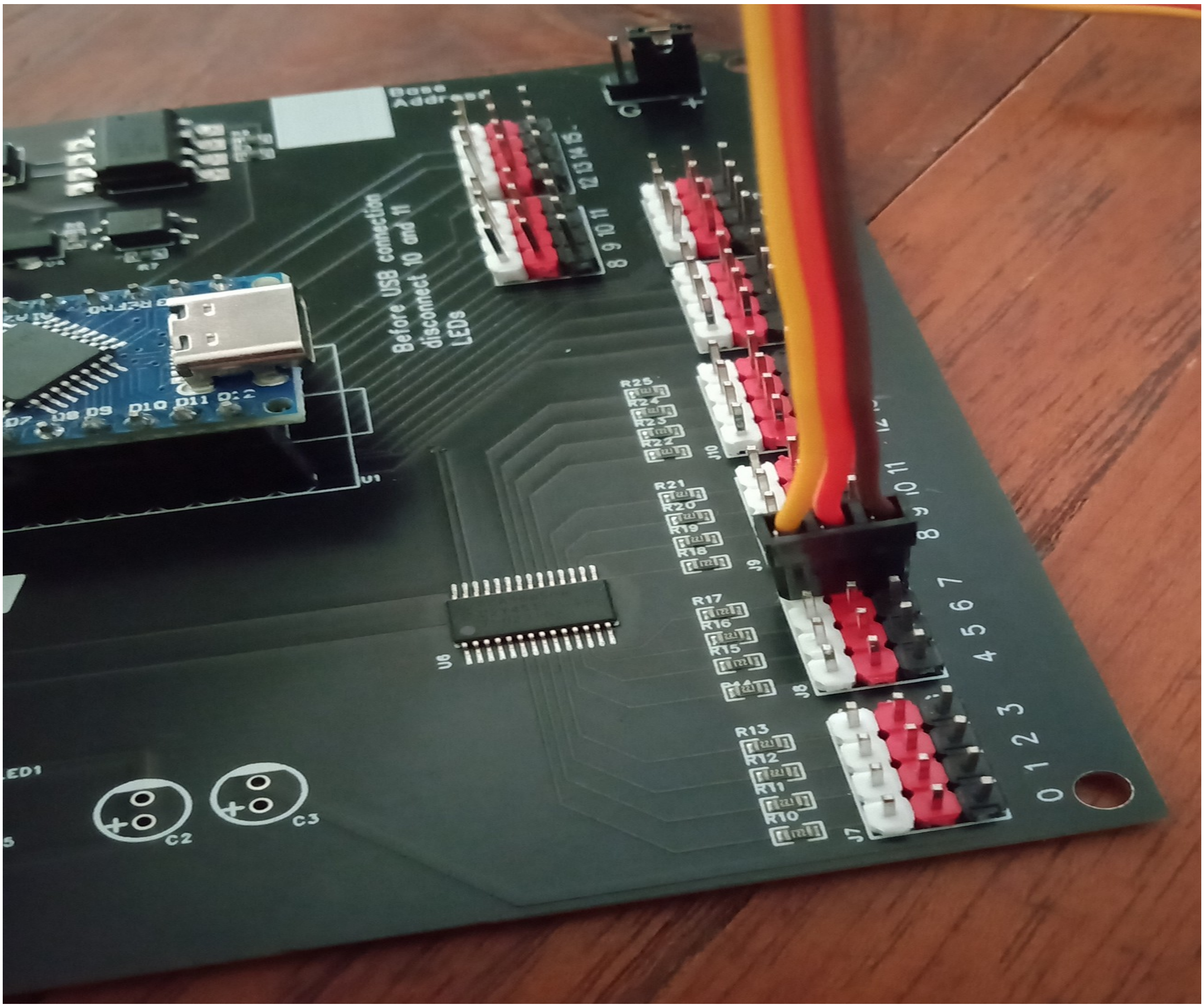
Connecting Servos

The servos are connected as below.

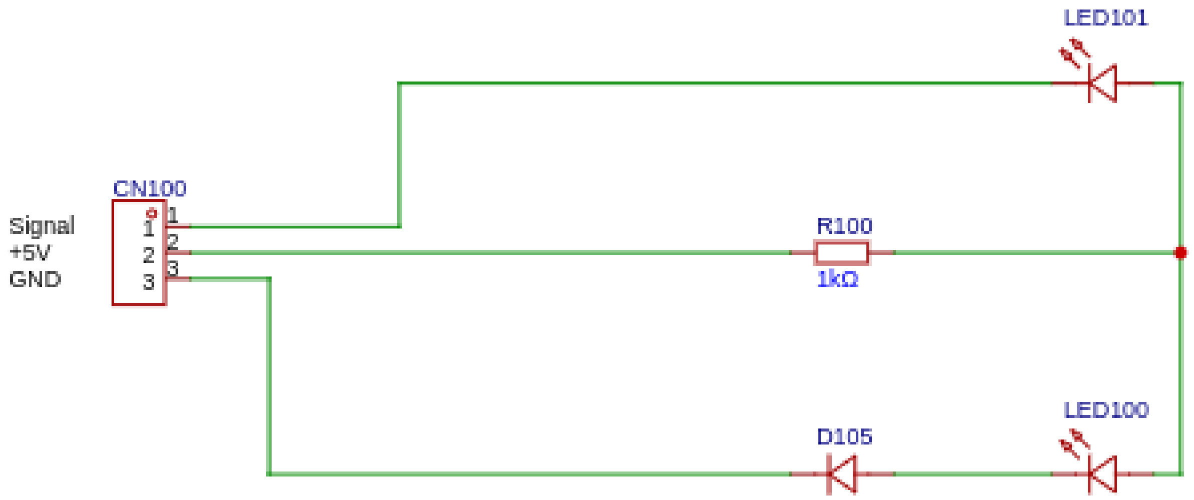
The wiring to the PCB is connected thus:

S = Servo control (white on decoder) yellow or white on servo
+ = Servo +V (red on decoder) red on servo
G = Servo Ground (black on decoder) black or brown on servo

The connectors are pin headers and designed to be used with servos directly or via servo extension cables. The servo connectors are colour coded to the servos.



Connecting Leds



The diagram above is one way to wire up the leds for connection to the RT_Pulse_8_Servo decoder. LED2 will be the active led when a throw command is issued for the relevant output. This method provides an indication if in the close or thrown position.

The 1kΩ resistor allows a reasonable brightness of the leds. If you need brighter then reduce the value of this. There is a limitation of how much current can be provide on each pin (7.5mA) and in total 75mA.

With a 1kΩ the current is 3mA per pin. With a 680Ω resistor the current is 4.5mA per pin.

If you have 16 leds the maximum for each pin is 4.5mA which can be done with a 680Ω resistor.

The connections are:
Signal - the closest pin of the three on the led connectors to the Arduino nano. (white)
+5V - +5V supply from the decoder. (red)
GND - Ground connection to the decoder. (black)

D105 is a 1N4148 or 1N4007 diode.

LED100 will be the led lit when a throw command has been sent for the relevant address.

Power Supply

The power supply for the servos can be either from the DCC input or a separate 5-6V DC supply.

If using a separate supply it is recommended to use 6V DC as there is some voltage drop on the input protection.

To use the supply from the DCC input place a shorting jumper on the INT pins.

WARNING

If you are using different power supplies for your command station and a DC power supply for this decoder. The power supplies must have a common ground.

Do not have an external power supply connected if you have a jumper on INT.

Serial Commands

Several commands are available via the Arduino serial monitor for configuring or displaying information on the decoder.

<?>	Show available commands
<>	Show current Control Variables
<A address>	Change decoder base linear address
<C address>	Close a turnout at address
<T address>	Throw a turnout at address
<M output mS / 10>	Set decoder output closed position
<N output mS / 10>	Set decoder output thrown position
<O output S / 10>	Set decoder output move time
<P output [0:1:2:3]>	Set decoder output configuration
<Z>	Soft Reset

Where output is 0 - 15 as on the decoder pcb

The M and N position commands are in mS / 10 eg. if the servo is at 110 closed then the value here is 11.

The O move time is the time it takes the servo to move in 10th of a second, eg. if the servo need 1 second to move then this is 10.

P configuration is as below, the time is how time it will take to move from closed to thrown and vv:
0 - move servo at maximum speed (this is the default)
1 - move servo fast (0.5 second)
2 - move servo medium (1 second)
3 - move servo slow (2 seconds)

The <A address> is the decoder linear address to use within the DCC command station. When you set an address it will display the correct base address to use for the decoder at the serial monitor. Eg will give a base address of 1 and the turnouts will be assigned addresses 1 – 16.

The default address is 1, you need to change this if using more than one stationary decoder on your layout. Once the address is set, this address and the next 16 are the addresses you use to control your turnouts. Eg, 1-16, 5-20 ...

Some examples using the serial monitor are:

<C 1>	Close turnout at address 1
<T 2>	Throw turnout at address 2

How you add them to your DCC Command Station will depend on the command station.

Base addresses are multiples of 4 + 1 eg, 1, 5, 9, 13, 17, …

The address can be between 1 and 2037

In all cases the 16 servos will be addressed from the base address for the next 16 addresses eg, base address 1, addresses are 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16. base address 5 addresses are 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20.

Learning Mode.

To set the address on the decoder in learning mode.

Connect the decoder to your DCC track via the DCC input connector. It’s best not to have any servos connected at this point.

Put a jumper on the Learn header next to the Arduino nano.

Turn power on to the track.

The LED on the nano will flash three times to show it is in learning mode.

Send a throw or close command to the base address you want for the decoder.

Base addresses are multiples of 4 + 1. eg, 1, 5, 9, 13, 17, …

The address can be between 1 and 2037.

Once the address is learnt, power off the track and remove the jumper from the Learn header. Your decoder will now respond at the new addresses.

Programming Track Setup.

The CV’s can be set with the decoder connected to a programming track.

Connect the decoder DCC IN to the programming track of your command station.

How you send a write command to the decoder CV will depend on your command station.

Eg Using a DCC-EX command station connected to an Arduino IDE serial monitor send the following command to change the address:

<W 1 address>

Use the table on pages 12-14 to determine the correct value to use for address. The value in the column CV1 is the value to use in the above command. The value in the column base address will then be the base address for the decoder.

The following CV's hold the information for each servo output. They can be set by <W [CV] value>

Output	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CV																
close	33	38	43	48	53	58	63	68	73	78	83	88	93	98	103	108
throw	34	39	44	49	54	59	64	69	74	79	84	89	94	99	104	109
time	35	40	45	50	55	60	65	70	75	80	85	90	95	100	105	110
config	36	41	46	51	56	61	66	71	76	81	86	91	96	101	106	111
position	37	42	47	52	57	62	67	72	77	82	87	92	97	102	107	112

close - the servo position at close divided by 10

throw - the servo position at throw divided by 10

time - the time the servo takes to move from close to throw of throw to close

config - 0 = max speed movement
- 1 = fast 0.5 seconds to move from one position to the other
- 2 = medium 1.0 seconds to move from one position to the other
- 3 = slow 2.0 seconds to move from one position to the other
- 4 = extra slow 5.0 seconds to move from one position to the other

position - the last position that the servo was in

The decoder automatically updates the position CV for the output after each move and will restore the servo to that position on restart.

The close and throw CVs need to be determined for each servo.

The default values for these are 11 and 25 when multiplied by 10 give the setting that is sent to the servo.   This equates to a approximately 90deg movement of the servo.

Address Table (CV9 = 0)

CV1	Base Address	CV1	Base Address	CV1	Base Address	CV1	Base Address
1	1	31	121	61	241	91	361
2	5	32	125	62	245	92	365
3	9	33	129	63	249	93	369
4	13	34	133	64	253	94	373
5	17	35	137	65	257	95	377
6	21	36	141	66	261	96	381
7	25	37	145	67	265	97	385
8	29	38	149	68	269	98	389
9	33	39	153	69	273	99	393
10	37	40	157	70	277	100	397
11	41	41	161	71	281	101	401
12	45	42	165	72	285	102	405
13	49	43	169	73	289	103	409
14	53	44	173	74	293	104	413
15	57	45	177	75	297	105	417
16	61	46	181	76	301	106	421
17	65	47	185	77	305	107	425
18	69	48	189	78	309	108	429
19	73	49	193	79	313	109	433
20	77	50	197	80	317	110	437
21	81	51	201	81	321	111	441
22	85	52	205	82	325	112	445
23	89	53	209	83	329	113	449
24	93	54	213	84	333	114	453
25	97	55	217	85	337	115	457
26	101	56	221	86	341	116	461
27	105	57	225	87	345	117	465
28	109	58	229	88	349	118	469
29	113	59	233	89	353	119	473
30	117	60	237	90	357	120	477

CV1	Base Address	CV1	Base Address	CV1	Base Address	CV1	Base Address
121	481	151	601	181	721	211	841
122	485	152	605	182	725	212	845
123	489	153	609	183	729	213	849
124	493	154	613	184	733	214	853
125	497	155	617	185	737	215	857
126	501	156	621	186	741	216	861
127	505	157	625	187	745	217	865
128	509	158	629	188	749	218	869
129	513	159	633	189	753	219	873
130	517	160	637	190	757	220	877
131	521	161	641	191	761	221	881
132	525	162	645	192	765	222	885
133	529	163	649	193	769	223	889
134	533	164	653	194	773	224	893
135	537	165	657	195	777	225	897
136	541	166	661	196	781	226	901
137	545	167	665	197	785	227	905
138	549	168	669	198	789	228	909
139	553	169	673	199	793	229	913
140	557	170	677	200	797	230	917
141	561	171	681	201	801	231	921
142	565	172	685	202	805	232	925
143	569	173	689	203	809	233	929
144	573	174	693	204	813	234	933
145	577	175	697	205	817	235	937
146	581	176	701	206	821	236	941
147	585	177	705	207	825	237	945
148	589	178	709	208	829	238	949
149	593	179	713	209	833	239	953
150	597	180	717	210	837	240	957

CV1	Base Address	CV1	Base Address	CV1	Base Address	CV1	Base Address
241	961	246	981	251	1001		
242	965	247	985	252	1005		
243	969	248	989	253	1009		
244	973	249	993	254	1013		
245	977	250	997	255	1017		

For addresses above 1017 set CV9 = 1 and CV1 = 0 to 255 and add 1024 to the base address above.

Eg. for base address 1021 - CV9 = 1 and CV1 = 0, for base address 1024 CV9 = 1 and CV1 = 1

For CV9 = 0, the base address can be calculated by the following:

base address = (CV1 - 1) * 4 + 1

The CV1 value can be calculated by the following:

CV1 = (base address - 1) / 4 + 1

Addendum

References.

Servo accessory decoder firmware:

https://github.com/Rosscoetrain/DCC-Turnout-Decoder-Servo