Nottawasaga Model Railway Signal System

There are sets of signals, mounted on a cantilever tower, located over the outer and inner mainline tracks at Easton and Westville. These signals are placed just before the major crossovers and indicate the status of the crossovers and the track ahead when entering either Easton or Westville from the rest of the layout. The Signal heads are Searchlight type, meaning only one light which can display a Red, Yellow/Orange or Green.

The signal display is not necessarily prototypical in the aspects but rather more familiar to everyday users such that Red means Stop, Yellow/Orange means a caution indicator and Green means Proceed, All Clear.

Over each track, the Lower Signal Head will be the most important head to observe when approaching the modules as they will indicate the condition of the crossovers and track immediately behind the signal cantilevers. The Upper Signal Heads indicate the status of the crossovers and track on the opposite module. From these sets of four Signal Heads, an operator will be able to determine how his train will proceed through the crossovers and tunnels to the opposite side and on which track he will emerge on the opposite side module.

Approaching Easton from the layout, the upper and lower signals on the right indicate the status of the outer track and the upper and lower signals on the left indicate the status for the inner track. The approach to Westville is opposite in that the signals on the left indicate the status of the outer track and the signals on the right indicate the status of the inner track. Other than that, the aspects on the signal heads will mean the same thing for each approach.

There is a control box at Easton and Westville that operate the two sets of crossovers at each location. The small panel has two sets of momentary push buttons along with LEDs to indicate the condition of the crossovers. The Westville crossovers can be operated from the Easton panel and vice versa.
There is one rule to follow for the crossovers. Only operate the crossovers from the control panels by pushing the appropriate push button. These are momentary buttons and do not need to be held. Never operate the crossovers manually by finger throwing the tie strip. The signal system and the control panels do not have feedback from the crossovers so if anything is thrown manually then all the indicator lights will be out of sync and will not reflect the true nature of the crossovers.

As the crossovers are operated from the control panels, the LEDs on the panel will change to reflect the new condition of the crossovers and, after a very short delay, the signal heads on the cantilevers will indicate information as to the state of all the crossovers. On the panels, two LEDs for each crossover will indicate Green for Closed (Straight) and Yellow/Orange for Thrown. The signals will inform any train operators that are approaching Easton or Westville from the rest of the layout, what the status of the crossovers are, and whether the track around the loop through the mountain is occupied or clear.

It is important to understand that the signals do not indicate traffic on the layout so it will still be necessary for operators to know where other traffic happens to be located and what could occur as trains progress on their current path. This installation is specific for allowing an operator to understand how the crossovers immediately ahead and on the opposite side of the loop are aligned and whether the crossover alignments will agree with the operator’s desired path.

There are some easy rules to understand.

- The lower signal head for the track that you are on is the one of immediate concern. Green will indicate that the crossovers immediately ahead are closed and you can proceed through them on the same track. If the lower signal is RED, you MUST stop before your engine reaches the cantilever as one or both of the crossovers are thrown and you will derail and or cause a short on the frog.
- The corresponding Upper Signal head for your track will indicate the status of the crossovers on the opposite module and can indicate whether you will continue on the same track Green, or be crossed over to another track, Yellow/Orange.
- If both the upper and lower signal heads indicate RED, then the track ahead is occupied by another train through the tunnel. The occupancy detectors only detect a train on the loop and tunnel sections. The crossover sections, or lead in modules, are not connected to the detector circuits. It is advisable that you be very aware of other trains when you see Red over Red and you may need to be well back from the crossovers to avoid a collision with a train that is head on to your train.

Following are examples of the aspects that will be seen as viewed from the approach to the cantilevers.
### What the Signals Mean

<table>
<thead>
<tr>
<th>Westville</th>
<th>Easton</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Outer Upper</strong></td>
<td><strong>Inner Upper</strong></td>
</tr>
<tr>
<td>Easton outer crossover closed</td>
<td>Easton inner crossover closed</td>
</tr>
<tr>
<td>Westville inner crossover closed.</td>
<td>Westville outer crossover closed.</td>
</tr>
<tr>
<td><strong>Outer Lower</strong></td>
<td><strong>Inner Lower</strong></td>
</tr>
<tr>
<td>Westville outer crossover closed Traverse from Westville outer track to Easton inner track.</td>
<td>Westville inner crossover closed Traverse from Westville track to Easton inner track.</td>
</tr>
<tr>
<td>Easton outer crossover closed. Traverse from Easton Inner Track to Westville inner track</td>
<td>Easton outer crossover closed. Traverse from Easton Outer Track to Westville outer track</td>
</tr>
</tbody>
</table>

**NOTE:** When changing from one track to the other, as at Easton Inner track to Easton outer track as pictured above, pay attention to the proper upper signal for Westville. Here, as you will be on the outer track, the Easton outer upper signal is the important one to observe what is happening to Westville, NOT the Easton inner upper signal.
<table>
<thead>
<tr>
<th>Westville</th>
<th>Easton</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Outer Upper</strong></td>
<td><strong>Inner Upper</strong></td>
</tr>
<tr>
<td>Easton outer crossover closed.</td>
<td>Easton inner crossover thrown to Easton run around</td>
</tr>
<tr>
<td><strong>Outer Lower</strong></td>
<td><strong>Inner Lower</strong></td>
</tr>
<tr>
<td>Westville outer crossover closed. Traverse from Westville outer track to Easton outer track.</td>
<td>Westville inner crossover thrown to run around. Traverse from Westville inner track to Easton run around at Easton</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Westville</th>
<th>Easton</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Outer Upper</strong></td>
<td><strong>Inner Upper</strong></td>
</tr>
<tr>
<td>Easton outer crossover closed.</td>
<td>Easton inner crossover thrown to Easton run around</td>
</tr>
<tr>
<td><strong>Outer Lower</strong></td>
<td><strong>Inner Lower</strong></td>
</tr>
<tr>
<td>Westville outer crossover closed. Traverse from Westville outer track to Easton outer track.</td>
<td>Westville inner crossover thrown to run around</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Westville</th>
<th>Easton</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Outer Upper</strong></td>
<td><strong>Inner Upper</strong></td>
</tr>
<tr>
<td>Easton outer crossover closed. STOP before cantilever at Westville</td>
<td>Inner track occupied at Westville</td>
</tr>
<tr>
<td><strong>Outer Lower</strong></td>
<td><strong>Inner Lower</strong></td>
</tr>
<tr>
<td>Westville outer crossover closed. Traverse from Westville outer track to Easton outer track.</td>
<td>Inner track occupied. STOP before cantilever at Westville</td>
</tr>
<tr>
<td>Westville</td>
<td>Inner Upper</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Outer Upper</td>
<td>Easton inner crossover thrown to Easton run around</td>
</tr>
<tr>
<td>Inner Lower</td>
<td>Westville outer crossover closed Traverse from Westville outer track to Easton inner track at Easton.</td>
</tr>
<tr>
<td>Westville inner crossover thrown to run around. STOP before Easton cantilever. Traverse from Westville run around to Easton run around</td>
<td>Easton outer crossover thrown. Traverse from Easton inner track to Easton outer track then to Westville outer track</td>
</tr>
</tbody>
</table>

**NOTE:** A train on the Easton inner track will traverse to the Easton outer track. The Easton outer upper signal now is the signal that will indicate the route to take at Westville, which indicates that the train will traverse back to the inner track at Westville.

Additional variations may be observed, but they will be indications that are shown on the tables above. When in doubt as to what the signals indicate, best practice is to stop your train, well before the modules, and check the crossover control panels to clarify how the crossovers are actually positioned. Remember, it is now possible to control the crossovers at Easton and Westville from the control panel at Easton, and vice versa.
The Technical Stuff

The electronics for the signal and turnout system comprises of a number of circuit boards. Most of the circuits are based on Electronic Circuits for Model Railroads by Robert Paisley. [http://home.cogeco.ca/~Erpaisley4/CircuitIndex.html](http://home.cogeco.ca/~Erpaisley4/CircuitIndex.html)

Some modifications to these circuits were done by John Houghton and the circuit boards were designed by John Houghton with use of the Eagle Software [http://www.cadsoft.de](http://www.cadsoft.de). A number of the boards have been mass produced by a web service at OSH Park [https://oshpark.com](https://oshpark.com).

The main crossover control boards contain a capacitor discharge circuit by Robert Paisley [http://home.cogeco.ca/~Erpaisley4/CDPSU.html](http://home.cogeco.ca/~Erpaisley4/CDPSU.html) and an Arduino Pro Mini computer. [http://www.arduino.cc](http://www.arduino.cc). The Arduino reads the push button inputs from the control panels at Easton and Westville, toggles the appropriate crossovers through the CD circuit, adjusts the control panel indicator LEDs and alters the cantilever signal head LEDs to match the turnouts. Eight Searchlight electronic searchlight boards based on Robert Paisley circuits [http://home.cogeco.ca/~Erpaisley4/556SearchLightSignal.html](http://home.cogeco.ca/~Erpaisley4/556SearchLightSignal.html) are activated by the Arduino to adjust the Searchlight heads. Four current induction circuits, based on Robert Paisley circuits [http://home.cogeco.ca/~Erpaisley4/DccBodXfmr.html](http://home.cogeco.ca/~Erpaisley4/DccBodXfmr.html) detect a current source, engine, in the loop tracks through the tunnel between Easton and Westville. These detectors are sensed by the Arduino and the Signal heads are adjusted to show the track occupancy conditions. The Capacitor Discharge circuit operates from a 16 Volt DC power Supply donated by Bill Hambly, who also donated the two cantilever tower kits. The rest of the circuits operate from 5 Volts DC. Martin Alborough donated the push buttons and housings and manufactured the panel faces for the two control panels at Easton and Westville.

Depending on the PDF viewer that you have, the links listed above might be live, meaning to visit the mentioned web pages, simply click on the blue underlined link and it will be opened in your web browser. Failing that, simply select the link, copy it and then paste it into your web browser’s address title bar.

For those who are really adventurous, the following pages contain the Arduino program that actually runs the whole system. Learn more about the Arduino and the programming language at the Arduino site linked above. The program is written in a C language and is well documented throughout the sketch. The variables and names are straightforward so non-programmers might be able to follow along with the code to see how things operate.

After the Arduino sketch the schematic for the entire system is published. The schematic is covered in 10 pages so that you can see all the labels etc as clearly as possible. In addition, there is a two page version of the schematic that indicates a more overall view of the schematic.
/*
NMRY Turnout Control

D2 to D3 as Button input for switch change request
D4 to D7 as control output to throw the switch coil circuit.
D8 to D9 as Digital output Pins for switch position indicator.
   Turn associated RED LED on or off.
   GREEN is hardwired to always ON.
A0 Outer This Side Lower Signal RED
A1 Outer Opposite Side Upper Signal YELLOW
A2 Inner This Side Lower Signal RED
A3 Inner Opposite Side Upper Signal YELLOW
*/

#include <Bounce2.h>

// Panel Switch Input
#define OUTERSWITCH 2
#define INNERSWITCH 3

//Control Panel Switch Indicator LEDs
#define OUTERLED 4
#define INNERLED 5

//Signal Output Inner/Outer Upper Lower Red
#define OUTERLOSIGRED A2
#define OUTERUPSIGRED 12
#define INNERLOSIGRED A1
#define INNERUPSIGRED A3
#define INNERLOSIGYEL A0

// Oposite Inner Upper Yellow
#define THATINUPSIGYEL 13
#define THATINNERCROSSOVERSENSE A4
#define THATOUTERCROSSOVERSENSE A5

//Switch Machine Output
#define CTLTHROWNOUT 6
#define CTLCLOSEDOUT 7
#define CTLTHROWNIN 8
#define CTLCLOSEDIN 9

//Occupancy Detector
#define DETECTOUTER 10
#define DETECTINNER 11

//Constants
#define THROWN 1
#define CLOSED 0
#define BIGDELAYTIME 700
#define DELAYTIME 500
#define HALFDELAYTIME 100
#define REDLOW
#define YELLOW
#define GRNHIGH
#define OCCUPIEDLOW
#define CLEARHIGH

//Instantiate a Bouncer2 object for each BUTTON Input
Bounce outerdebounce =Bounce();
Bounce innerdebounce =Bounce();
int outerSigLo = GRN;
int innerSigLo = GRN;
int outerSigUp = GRN;
int innerSigUp = GRN;
int thatInSigUpYel = GRN;
int thatInCrossoverState = HIGH;
int thatOutCrossoverState = HIGH;
int outerThrownSigYel = GRN;
int innerButton = 0;
int innerButtonRead = HIGH;
int innerToggle = CLOSED;
inDetect = CLEAR;
int outerButton = 0;
int outerButtonRead = HIGH;
int outerToggle = CLOSED;
int outDetect = CLEAR;

void setup() {
  //setup outputs for Indicator LEDS on Panels.
  //Green is hard wired to always ON.
  pinMode(OUTERLED, OUTPUT);
  digitalWrite(OUTERLED,LOW);
  pinMode(INNERLED, OUTPUT);
  digitalWrite(INNERLED,LOW);

  //Setup Signal Head Outputs to Green
  //A2
  pinMode(OUTERLOSIGRED,OUTPUT);
  digitalWrite(OUTERLOSIGRED,outerSigLo);
  //A1
  pinMode(INNERLOSIGRED,OUTPUT);
  digitalWrite(INNERLOSIGRED,innerSigLo);
  //D13
  pinMode(OUTERUPSIGRED,OUTPUT);
  digitalWrite(OUTERUPSIGRED,outerSigUp);
  //A3
  pinMode(INNERUPSIGRED,OUTPUT);
  digitalWrite(INNERUPSIGRED,innerSigUp);
  //D12
  pinMode(THATINUPSIGYEL,OUTPUT);
  digitalWrite(THATINUPSIGYEL,thatInSigUpYel);
  //A0
  pinMode(INNERLOSIGYEL,OUTPUT);
  digitalWrite(INNERLOSIGYEL,outerThrownSigYel);
  //A4 as Input
  pinMode(THATINNERCROSSOVERSENSE,INPUT);
  //A5 as Input
  pinMode(THATOUTERCROSSOVERSENSE,INPUT);

  //setup switch coil control circuits. Establish Closed as default start.
  //D4,D5=1
  pinMode(CTLTHROWNOUT,OUTPUT);
  digitalWrite(CTLTHROWNOUT,HIGH);
  pinMode(CTLCLOSEDOUT,OUTPUT);
  digitalWrite(CTLCLOSEDOUT,LOW);
  delay(DELAYTIME);
  digitalWrite(CTLCLOSEDOUT,HIGH);
  delay(BIGDELAYTIME);
  //D6,D7=2
pinMode(CTLTHROWNIN, OUTPUT);
digitalWrite(CTLTHROWNIN, HIGH);
pinMode(CTLCLOSEDIN, OUTPUT);
digitalWrite(CTLCLOSEDIN, LOW);
delay(DELAYTIME);
digitalWrite(CTLCLOSEDIN, HIGH);
delay(BIGDELAYTIME);

pinMode(OUTERSWITCH, INPUT_PULLUP); // switch 1
outerdebounce.attach(OUTERSWITCH);
outerdebounce.interval(5);

pinMode(INNERSWITCH, INPUT_PULLUP); // switch 2
innerdebounce.attach(INNERSWITCH);
innerdebounce.interval(5);

// D10 D11 as Detector Inputs
pinMode(DETECTOUTER, INPUT_PULLUP);
pinMode(DETECTINNER, INPUT_PULLUP);

}

void loop(){
// Read outer panel button for is it pressed
outerButton = outerdebounce.update(); // 1 if state changed
outerButtonRead = outerdebounce.read();

//button pressed switch from closed to thrown
if ((outerButton == 1) && (outerButtonRead == LOW)){
  if(outerToggle == CLOSED){
    digitalWrite(CTLTHROWNOUT, LOW);
    delay(HALFDELAYTIME);
    digitalWrite(OUTERLED, THROWN);
    outerSigLo = RED;
    outerThrownSigYel = YEL;
    delay(DELAYTIME);
    outerButton = outerdebounce.update();
    digitalWrite(CTLTHROWNOUT, HIGH);
    delay(BIGDELAYTIME);
  }
}

//button pressed switch from thrown to closed
if(outerToggle == THROWN){
  digitalWrite(CTLCLOSEDOUT, LOW);
  delay(HALFDELAYTIME);
  digitalWrite(OUTERLED, CLOSED);
  outerSigLo = GRN;
  outerThrownSigYel = GRN;
  delay(DELAYTIME);
  outerButton = outerdebounce.update();
  digitalWrite(CTLCLOSEDOUT, HIGH);
  delay(BIGDELAYTIME);
}

outerToggle = 1 - outerToggle;
outerButtonRead = outerdebounce.read();
while(outerButtonRead == 0){//Do nothing until button is released
  outerButton = outerdebounce.update();
  outerButtonRead = outerdebounce.read();
}

// Read Inner panel button for is it pressed
innerButton = innerdebounce.update(); // 1 if state changed
innerButtonRead = innerdebounce.read();

//button pressed switch from closed to thrown
if ((innerButton == 1) && (innerButtonRead == LOW)){
  if(innerToggle == CLOSED){
    digitalWrite(CTLTHROWNIN, LOW);
    delay(HALFDELAYTIME);
    digitalWrite(INNERLED,THROWN);
    innerSigLo = RED;
    thatInSigUpYel = YEL;
    delay(DELAYTIME);
    innerButton = innerdebounce.update();
    digitalWrite(CTLTHROWNIN, HIGH);
    delay(BIGDELAYTIME);
  }
  innerToggle = 1-innerToggle;
  innerButtonRead = innerdebounce.read();
  while(innerButtonRead == 0){//Do nothing until button is released
    innerButton = innerdebounce.update();
    innerButtonRead = innerdebounce.read();
  }
}

//check for inner and outer track occupied
inDetect = digitalRead(DETECTINNER);
outDetect = digitalRead(DETECTOUTER);
thatInCrossoverState = digitalRead(THATINNERCROSSOVERSENSE); //A4
thatOutCrossoverState = digitalRead(THATOUTERCROSSOVERSENSE); //A5

// Establish Red Signals for variations of the track being occupied and status of crossovers

//Outer Track Signal conditions
//This Outer Low signal RED is hardwired to that Outer Upper Signal Yellow

if(outDetect == OCCUPIED){
  outerSigUp = RED;
} else{
  outerSigUp = GRN;
}

if (outerToggle == THROWN){
  outerSigLo = RED;
}

if ((outerToggle == CLOSED) && (outDetect == OCCUPIED)){
  outerSigLo = RED;
}

if((outerToggle == CLOSED) && (outDetect == CLEAR)){

outerSigLo = GRN;
}

// Inner Track Signal conditions
// This Inner Low Signal Red is hardwired to That Inner Upper Signal Yellow

if((thatOutCrossoverState == LOW && thatInCrossoverState == HIGH) || (inDetect == OCCUPIED)){
  innerSigUp = RED;
} else{
  innerSigUp = GRN;
}

if((outerToggle == CLOSED) && (innerToggle == CLOSED) && (inDetect == OCCUPIED)) {
  innerSigLo = RED;
}

if((outerToggle == CLOSED) && (innerToggle == THROWN)){
  innerSigLo = RED;
}

if((outerToggle == THROWN) && (outDetect == OCCUPIED)){
  innerSigLo = RED;
}

if((outerToggle == THROWN) && (outDetect == CLEAR)){
  innerSigLo = GRN;
}

if((innerToggle == CLOSED) && (outerToggle == CLOSED) && (inDetect == CLEAR)){
  innerSigLo = GRN;
}

// Write all values to the Signal Heads
  digitalWrite(INNERLOSIGRED,innerSigLo); // A1
  digitalWrite(TATIHINUPSIGYEL,thatInSigUpYel); // D13
  digitalWrite(INNERUPSIGRED,innerSigUp); // A3
  digitalWrite(OUTERLOSIGRED,outerSigLo); // A2
  digitalWrite(OUTERUPSIGRED,outerSigUp); // D12
  digitalWrite(INNERLOSIGYEL,outerThrownSigYel); // A0
}