BLOCK MANAGEMENT AND LOCAL CONTROL FOR
A COMPUTER OPERATED MODEL RAILROAD

Ned Lecky
Howard Fiderer

ABSTRACT

A system for local train management and identification is presented. It includes a series of bar code readers and isolated track circuits designed to monitor and control the traffic flow between a series of projects and the mainline in a microprocessor controlled model railroad. Hardware and software documentation for this 6502 microprocessor based system are also included.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Figures</td>
<td>3</td>
</tr>
<tr>
<td>Introduction</td>
<td>4</td>
</tr>
<tr>
<td><strong>Section 1: Local Computers</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Section 2: Block Management</strong></td>
<td></td>
</tr>
<tr>
<td>Part 1: Locomotive Recognition</td>
<td></td>
</tr>
<tr>
<td>Part 2: Block Management Hardware</td>
<td>12</td>
</tr>
<tr>
<td>Part 3: Block Management Software</td>
<td>14</td>
</tr>
<tr>
<td>Conclusion</td>
<td>15</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure 0 ---- Standard Project Layout
Figure 1 ---- Bar Code Layout
Figure 2 ---- Basic Computer
Figure 3 ---- System Memory Map
Figure 4 ---- Local Power Supply
Figure 5 ---- Local Data Receiver and Transmitter
Figure 6 ---- Local Computer Layout
Figure 7 ---- Local Computer Discrete Component Socket
Figure 8 ---- Local Computer System Wiring Schedule
Figure 9 ---- Bar Code Sensors
Figure 10 ---- Block Clear Sensors
Figure 11 ---- Bar Code Reader
Figure 12 ---- Track Kill Circuit
Figure 13 ---- Light Controller
Figure 14 ---- Control Data Writer
Figure 15 ---- Block Control Flowchart
Figure 16 ---- Housekeeping Flowchart
The essential requirement in a computer operated railroad is that a state of order is maintained over the system at all times while local projects are given the freedom to control their own operations. This is accomplished through a system of controlling networks arranged in a modular fashion around each block. These networks are comprised of isolated track sections, optical sensors and status indicators. See figure 1.

The basic control hierarchy requires that in general no more than one train be on either the north or south track at any time. Since trains may be switched off the north track, that mainline should be allowed to enter a caution state so that trains may pass through the block unobstructed. A method of assuring a clear track for a fixed time interval is necessary to allow reentry of the train on the siding. Finally, it should be possible to set a similar caution state on the south track to permit a smooth progression of trains on that through mainline.

--Page 4--
This system is implemented through the use of a locomotive recognition system and traffic control circuitry. Locomotives are tagged with a four bit bar code consisting of a row of clock pulses below a row of data. As each clock pulse is received by one optical sensor, data is read in by another sensor directly above it. These clock and data pulses are fed into a shift register which latches the train number for the next read. Train jitter and alignment error are eliminated by constructing the bar code as in figure 1.

Reading is effected through an eight bit parallel to serial shift register that is loaded from the two separate registers that clock in the bar code. Thus the serial information passing by the bar code sensors can be read into the computer at a rate higher than that normally possible. This scheme uses a minimal number of VIA port pins, allowing for greater I/O capabilities. In keeping with this scheme, the status indicator lights and track kill circuits are also controlled by a shift register. In order to save time and more importantly a port pin, the shift clocks for the bar code and control shift registers are tied together. Consequently the bar code data is shifted into the microprocessor as the light and track kill bits are shifted out. Hardware considerations will be discussed further in section 2.

The eventual aim of this system is to allow local projects to gain control and process any locomotives entering the project space.
This will make train processing in the local projects completely transparent to the master controller, freeing it to concentrate on coordination and scheduling problems throughout the system. Currently, our project is capable of maintaining the interface between the local projects and the mainline. At the same time it regulates traffic flow on the mainline to insure adequate spacing between trains, reducing the amount of collision avoidance processing required of the master controller. In short, our local control system enables the local computers to achieve a higher level of independence and utilize a more sophisticated software package.

SECTION 1
LOCAL COMPUTERS

Each local computer is a 6502 microprocessor based system equipped with 2K of EPROM, 1K of RAM, 2 hexadecimal LED displays and a 6522 Versatile Interface Adapter. System configuration is relatively straightforward, with address decoding accomplished by address lines 14 and 15. The EPROM resides in high memory to allow the reset and interrupt vectors to be placed in permanent locations. RAM begins at page 0 in order to simplify programming. A complete system schematic and memory map are given in figures 2 and 3.
It is a requirement of the computer system that a reset signal be generated at the time the computer is activated. The circuit developed for this purpose is basically a delay line. It is composed of an R / C network filtered through two Schmitt triggered inverters already available on the board. The circuit is shown as part of the system schematic.

A specialized +5 Volt power supply was developed to derive power from the + / - 20 Volt AC signal on the tracks. The track power is rectified by a diode bridge to approximately +20 Volts DC, which is then filtered and passed through two voltage dropping resistors to emerge at just less than +10 Volts DC. This is then applied to a positive five volt regulator, whose output is ripple filtered to provide a constant +5 Volt DC source over a load range of 750 mA to 1.2 A. The supply may be used to drive 500 mA to 750 mA load by increasing the value of one of the voltage dropping resistors. A circuit schematic is given in figure 4.

The data on the tracks must be converted to a TTL compatible signal before processing can occur. An LED / Opto-transistor pair has been employed for this purpose. Data transmission from the local computers to the master controller is accomplished through the application of a fixed load across the tracks. Two optically coupled SCR's are used to switch the load. Both the receiver and transmitter circuits are diagrammed in figure 5. Computer communications will
be treated in detail in a later chapter.

The local computers have been constructed on 4 1/2 by 6 inch vectorboards with 44-pin edge connectors, 3M DIP sockets and the Scotchflex wiring system are used. The standard board layout is given in figure 6. As the resistors and voltage regulator dissipate a great deal of heat they should be mounted at least 1/8 inch above the board surface and spaced as shown.

The discrete socket shown holds several resistors and capacitors used in basic computer reset, clock, and control circuits. Component layout and values are given in figure 7. A wiring schedule utilizing the chip shorthand notation of figure 6 is given in figure 8. Implementation of this wiring scheme will produce a basic computer system to be used in conjunction with application hardware.

SECTION 2

BLOCK MANAGEMENT

Part 1: Locomotive Recognition

Each local computer must be able to tell when a locomotive has entered the block or left the block. The computer must also have the ability to differentiate between trains so that it may selectively process them. Recognition of trains is accomplished by using a pair of optical sensors strapped together in a high low fashion. As the locomotive enters the block, it passes by the optical

--Page 8--
sensors of the bar code reader. These sensors consist of an Infrared
Emitting Diode together with a photo transistor. When the light is
reflected back to the receiver, the photo transistor turns on,
driving the conversion circuitry to a logic one.

The conversion circuitry comprises the majority of the hardware
needed for reading bar code. An Op Amp used as a comparator configured
with a variable threshold insures positive triggering on each
clock pulse. This scheme eliminates multiple triggering of the sensors
and insures proper loading of the shift registers. The circuit and sensor
arrangements are documented in figure 9.

It is also essential that the local computers have the ability
to detect a train leaving the block. Another optical sensor has been
placed at the point of exit from the block for this reason. Upon
exit, the caboose passes by a physically higher optical sensor and
triggers it with a strip of reflective tape. This in turn strobos the
clear line on the appropriate shift register, clearing it for the next
read of the bar code. The circuitry used in the clear sensors may be
much simpler since multiple triggering is not a problem. A circuit has
been created without a variable threshold in order to minimize
the complexity of the hardware. In addition to the change just
stated, a need to invert the logic arises since the clear signal
on the shift register is active low. A simple reversal of the leads
on the Op Amp is sufficient to do this. The physical sensor arrangement

--Page 9--
and the associated hardware are shown in figure 10.

The problem remaining is that of coupling the bar code circuitry to the basic computer. Several different methods have been tried in the evolution of the current system.

First a software interface was created that scanned two VIA port pins waiting for clock pulses. When one was found, the data sensor was scanned and read in. The sensor decoding circuits then employed were found to be inherently noisy, so a software filtering method was conceived and implemented. This system, however, would leave the user programs with very little flexibility since all control would have to be programmed into the user's software. Thus, this method was discarded.

The next attempt at handling the block management involved an interrupt driven scheme in which the clock pulse would interrupt the computer. The software system developed kept track of how many pulses had been read in, and after four were available would assemble them into the proper ID code. A problem was encountered that proved insurmountable, however. Track communications, being done on a similar interrupt basis, often take command of the processor for as long as 8.5 ms. At top speed, a 1 / 16 inch wide clock pulse would pass by the sensors in 27 ms. Therefore, to make the time problem tractable a two inch long strip of bar code would have been required, which was deemed an unsatisfactory requirement. Clearly another method of input
was required.

The final and most attractive reading design uses a network of shift registers to record the passage of each train on the two tracks. Two serial to parallel registers read in the code, and a separate parallel to serial register is used to clock the data into the computer. The clock pulse from the optical sensors is used to provide a shift pulse to a serial input register. The state of the data sensor is available at the data input pin of the same register. So, as the train passes by the sensors, its ID number becomes available on the output lines of the serial input register tied to its track.

Three V1A pins are required to read the code in from these registers. One is used to issue a load command to the serial output register, and the other two provide clock and data transmission services. The basic circuit is shown in figure 11.

While this method requires that a controlling routine be run often to monitor the registers, this is easily accomplished by adding the software to the communications routine. The operation of the software then becomes completely transparent to the user, effectively doing nothing but modifying three memory locations in RAM that contain up-to-date information on the train numbers on each of the North track, South track, and siding. User interaction with the program is restricted to operations on the two caution bits. The routine is called by the communications routine 22.5 times each second, so there is negligible data lag, and total execution time

--Page 11--
remains under 10 mS each second. This scheme, while wasting more
time than an interrupt routine, manifests enough benefits to prove
itself as the correct solution.

Part 2: Block Management Hardware

To enable software control of the local project, hardware is
required to allow isolated track control and status indicator light
control. A solid state switch was desired that could pass current in
both directions for controlling power application to the tracks. A
circuit was developed for this purpose using two readily available
opto-SCR's arranged in a back-to-back fashion. To sink the high current
required to drive the LEDs, a simple transistor switch was employed.
The transistor was configured here such that the SCR's are normally
activated, requiring a specific software command to create a break
in the mainline. The track kill circuit is shown in figure 12.

A circuit was also required to allow software control of the
status indicator lights. The lights are designed to operate at
approximately +20 Volts so track power seemed to be an attractive
source. The circuit employed must be deactivated by a floating output
from a tri-stateable buffer, as it is important that the lights are
not connected across the tracks during a data transmission cycle.
Next a high current opto-transistor was selected and equipped with a
diode to prevent false transistor triggering. This enables the lights
only when power of the correct polarity is available on the tracks.

--Page 12--
The modular light control circuit is shown in figure 13.

A shift register method was again chosen to output the control bits to the two track kill circuits and the six indicator lights. This seemed to be the logical choice when port pin conservation was considered. To conserve pins the same clock pulse signal as that used by the bar code reader is tied to this register. The data is placed on a fourth port pin, completing the block management set of I/O pins. A fifth signal is also available on the B port, going high whenever positive power is available on the tracks. Although the lights may be disabled during the negative power and data portions of the track signal, the two track kill circuits must not be effected to allow proper locomotive control. Consequently, the track kill circuits bypass the tri-state buffer and are fed directly from the shift register. Since the total shifting time for the data transfer is only about one percent of the activation time, spurious data passing the control lines may be considered negligible. Therefore, no attempt is made to clear the register before loading new data. The associated hardware and circuit diagram are shown in figure 14.

All control circuitry discussed above is currently constructed on prototyping strips and mounted on the wall beneath each project board. Computer interaction with the hardware is achieved through the five previously mentioned lines. In addition track power must be brought up to the onboard power supply and power and ground

--Page 13--
must go down to the prototyping strip. All of these lines are
c connected to the boards edge connector and run down to a twenty
five pin connector used by the prototyping strip. The pin assignments
for these sockets have been standardized and are pictured in appendix B.

Part 3: Block Management Software

A complete set of software has been developed to do the following
two things: 1) handle input/output between the local computer and the
two shift register systems, and 2) manage locomotive flow as discussed
in the introduction.

Since the input and the output must be done concurrently, we
had to decide where these operations should be placed in the routine.
We chose to handle I/O first in the routine, later calculating the
control bits to be output on the next entry to the routine.

The software is organized as a subroutine that is called by
the communications routine. Basically, the communications software is
entered through an interrupt mechanism each time polarity shifts on
the track. If a data cycle is not pending, the communications routine
soon returns control to the user program. In the interest of speed, a
calling mechanism for our routine is set up so that the management
software is called on only one of the intermediate interrupt times.
Therefore, our software runs at regularly spaced intervals of about
50 mS. Its operations are entirely transparent to the user and appear
as only the RAM location modifications previously discussed. The user
supplies any additional data with the two caution bits.

--Page 14--
Facility flowcharts are given in figures 15 and 16. A fully documented software listing appears in Appendix A.

CONCLUSION

In summary, the problem of block management and locomotive identification has benefited greatly from the bar code reader and train control system. The local computers may now be run independently of the master controller and other existing block machines. The interface between project and mainline is now sufficiently managed to allow a theoretically infinite number of projects on the track to perform their independent functions.

The circuitry developed has been tested extensively and found to be reliable enough to be considered the nucleus of the next generation of control hardware. As of this time the local management system provides an excellent tool for the students, making it possible for the user routine to become much more sophisticated in its train processing approach. The user routines may now be much more selective in their own processes and carefree in their interaction with the main line.

Future additions should include; local locomotive speed control, greater communication between the local computers and the master controller and communication between the local computers. These steps

--Page 15--
are needed to provide a more open communication network around the
tracks. This would make it possible to coordinate activities between
blocks more effectively.

Acknowledgements:

We would like to thank M.G. Littman for conceptual and technical
help, Exxon Office Systems for the use of their equipment, and last but
certainly not least Dallas Brodie for last minute graphics assistance
and moral support i.e., phone calls.

This paper is submitted in partial fulfillment
of M.E. independent work requirements

[Signature]

May 5, 1982

--Page 16--

108
Code 5 (0101) shown.
Data encoded reflective = 1,
matte = 0.

bit: 3 2 0

motion

black matte tape
reflective tape
clock pulse
generations
System Memory Map

0300----RAM Begins
03FF----End of 1K RAM
0400

Ram Expansion

3FFF
4000----LED Display
4001

Also Decoded as LED

7FFF
A000----VIA Decoding Begins
A000

Standard VIA Addresses
A00F

FFFF----VIA Decoding Ends
C000----ROM Decoding Begins

F000----Bottom of 4K ROM
F800----Bottom of 2K ROM
FFFF----Top of ROM and Memory
LOCAL POWER SUPPLY

AC VM 08

470 uf

10A 10W 5A 10

ALLOW FOR LOADS mA

7805

SYSTEM + 5 V

10 V

10
LOCAL DATA RECEIVER AND TRANSMITTER

TRANSMITTER

PB7

DATA TRANSMIT

100

5V

150

10K

6

4

2

1

2 X 4N40

10K

6

4

2

1

5

RECEIVER

1K

5V

4N25

CB1 & 2

DATA RECEIVE

270
LOCAL COMPUTER
DISCRETE COMPONENT SOCKET

FIG 7

5V

IRQ
ROY
&
NMI
NAND
PIN 5 7410

100 pf
3.3 K
1 K
4.7 K

5V

RESET CIRCUIT
PIN 1 7414

10 K
10 uf

Ø1
Ø2
### Local Computer System Wiring Schedule

<table>
<thead>
<tr>
<th>Address</th>
<th>Lines</th>
</tr>
</thead>
<tbody>
<tr>
<td>A80</td>
<td>E5-F5-K9-J8-A38</td>
</tr>
<tr>
<td>A81</td>
<td>E6-F5-K10-J7-A37</td>
</tr>
<tr>
<td>A82</td>
<td>E7-F7-K11-J6-A36</td>
</tr>
<tr>
<td>A83</td>
<td>E4-F4-K12-J5-A35</td>
</tr>
<tr>
<td>A84</td>
<td>E3-F3-K13-J4</td>
</tr>
<tr>
<td>A85</td>
<td>E2-F2-K14-J3</td>
</tr>
<tr>
<td>A86</td>
<td>E1-F1-K15-J2</td>
</tr>
<tr>
<td>A87</td>
<td>E17-F17-K16-J1</td>
</tr>
<tr>
<td>A88</td>
<td>E16-F16-K17-J23</td>
</tr>
<tr>
<td>A89</td>
<td>E15-F15-K18-J22</td>
</tr>
<tr>
<td>A910</td>
<td>K19-J19</td>
</tr>
<tr>
<td>A914</td>
<td>K26-B5-C3-C9-A23</td>
</tr>
<tr>
<td>A915</td>
<td>K25-B3-C4-A26</td>
</tr>
<tr>
<td>A914</td>
<td>B6-C2</td>
</tr>
<tr>
<td>A915</td>
<td>F7-C13-C11</td>
</tr>
</tbody>
</table>

### Data Bus

| DB0     | F14-K33-J9-G3-A33   |
| DB1     | F13-K32-J10-G2-A32  |
| DB2     | F12-K31-J11-G13-A31 |
| DB3     | F11-K30-J12-G12-A30 |
| DB4     | E16-K29-J14-H3-A29  |
Control Lines
ISO
FDY & NMI
NAND PULLUP
R/W
RESET
CLOCK
K2-D1-A21
K2-K6-D2
C5-D3
K3-F10-E10-A22
D9-D9-R1
P2-E13
A34-E13-K10
K13-D11
K30-D10-C1-C10-A25
D7-D5-K37

Enable Lines
LED
RAM
ROM
G8-G5-H5
C12-E9-F9
C5-J10-J20

Power Connections
+5 Volts
Ground
Supply-Edge pin 3-A20-C14-B14-D14-D13-D12-
F19-F18-K9-J21-J27-G11-C17-H1-H1
Supply-Edge pin 4-A1-C7-B7-D7-E9-F9-K1-K21-J12-
C7-G8-H7-H8
Block Control with Data I/O

Enter

Issue Input Lead Pulse

Take Clock Low

Align Read Area and Read a Bit

Align Control Area and Write a Bit

Take Clock High

Write I/C Done

Break Input Into North and South Bar Code Data

Same Read South as Last Time?

SCOUNT = SCOUNT - 1

Load STRING with South Bar Code Contents

SCOUNT = 0*

Do Housekeeping

(figure 16)

Exit

Same Read North as Last Time?

NCOUNT = NCOUNT - 1

NCOUNT = 0*

Load STRING with North Bar Code Contents

We need 8 successive reads (about 1/8 sec) to ensure the train is completely past the bar code sensors and not in transit.
Local Control Software Listing

RAM Address Table

NTRAIN = $03CF ; TRAIN NUMBER ON NORTH TRACK (STUDENT USE)
STRAIN = $03CE ; TRAIN NUMBER ON SOUTH TRACK (STUDENT USE)
STSDTR = $03CD ; STUDENT-USE LOCATION FOR TRAIN NUMBER ON SIDING
CAUT = $03CC ; CAUTION BIT CONTROL BYTE
SIDTR = $03CB ; PROGRAM-USE SIDING TRAIN SAVE AREA
RELDEL = $03CA ; DELAY COUNT SAVE AREA FOR SIDING TRAIN RELEASE
CLRDEL = $03C9 ; DELAY COUNT SAVE AREA FOR CLEAR TRACK CHECK
BCRASM = $03C8 ; LOCATION FOR BAR CODE SHIFT IN
SCEOUNT = $03C7 ; DELAY COUNT FOR READER PASSAGE OF SOUTH TRAIN
NCEOUNT = $03C6 ; SAME FOR NORTH TRAIN
SASEM = $03C5 ; ASSEMBLE AREA FOR SOUTH TRAIN NUMBER
NASSEM = $03C4 ; ASSEMBLE AREA FOR NORTH TRAIN NUMBER
LEAVTR = $03C3 ; SAVE AREA FOR SIDING TRAIN REQUESTING MAINLINE
CONTRL = $03C2 ; SAVE AREA FOR CONTROL BITS -- LIGHTS, TRACK KILLS
VORB = $A000 ; ADDRESS OF PORT B OUTPUT

"CONTRL" Bit Assignments

Bit 7- South Clear Light
Bit 6- South Caution Light
Bit 5- South Stop Light
Bit 4- North Green Light
Bit 3- North Yellow Light
Bit 2- North Red Light
Bit 1- South Isolated Track
Bit 0- North Isolated Track

For all control bits, a ZERO indicates the device is ON and a ONE indicates it is OFF.

"CAUT" Bit Assignments

Bits 2-7- Unused
Bit 1- South Caution Control Bit
Bit 0- North Caution Control Bit

A ONE in a caution bit indicates caution state requested, a ZERO indicates normal action. Caution bit operation is discussed in the text.
Software Listing

LOCCON PHP
  ;SAVE CARRY BIT FOR FIRST ROL OPERATION
  LDA #$49
  ;ISSUE PARALLE NUM LOAD PULSE TO READER
  ;THE HI 4 LEAVES COMMUNICATION PINS ALONE
  STA VORB
  ;PUTS PULSE OUT
  #$41
  ;END PULSE
  STA VORB
  #$00
  ;SET BIT COUNTER TO ZERO

IOLOOP ASL BCRASM
  ;SHIFT BAR CODE IN LEFT
  LDA #$40
  ;TAKE CLOCK LOW
  STA VORB
  LDA VORB
  ;DATA TO BE READ NOW AVAILABLE- READ IT
  AND #$04
  ;ZERO OUT OUTPUT AND COMMUNICATIONS BITS
  CMP #$04
  ;IS IT A ONE?
  BNE ZEROIN
  ;IF NOT, GOTO ZEROIN
  BCRASM
  ;IF SO, ADD IN 1 TO SHIFTED BCRASM

ZEROIN LDY #$40
  ;SET UP Y TO OUTPUT A ZERO
  LDA CONTRL
  ;GET CONTROL BITS
  BPL ZEROUT
  ;IF HI BIT IS A ZERO, GOTO ZEROOUT
  LDY #$42
  ;ELSE SET UP FOR ONE OUTPUT

ZEROUT STY VORB
  ;DUMP OUT Y
  ;RESTORE CARRY BIT
  ROL CONTRL
  ;MOVE NEXT BIT OF CONTRL INTO POSITION
  PHP
  ;SAVE NEW CARRY BIT
  INY
  ;SAME DATA AS ABOVE IN Y, NOW INCR FOR CLOCK
  STY VORB
  ;TAKE CLOCK HI WITH SAME OUTPUT DATA
INX ;INCREMENT BIT COUNTER

$08 ;8 BITS I/O?

IOLOOP ;IF NOT, INPUT AND OUTPUT ANOTHER

;IF SO, GET CARRY BIT (CLEAR STACK)

ROL CONTRL ;CONTRL IS BACK TO ITS ENTRY STATE

CONVRT LDA BCRASM ;PULL NORTH READ OUT OF BCRASM

LSR A ;FOUR RIGHT SHIFTS ALIGNS IT

LSR A

LSR A

LSR A

CMP NASSEM ;IS IT WHAT WE READ LAST TIME?

BEQ SAME1 ;IF SO, GOTO SAME1

STA NASSEM ;ELSE, PUT IT IN NASSEM FOR NEXT COMPARE

LDA #00 ;RESET NCOUNT TO ZERO CONSECUTIVE READS

STA NCOUNT

JMP SOUTH ;AND HEAD FOR SOUTH EQUIVALENT ROUTINE

SAME1 INC NCOUNT ;WE HAVE ONE MORE CONSECUTIVE READ

LDA #06 ;HAVE WE READ IT 06 TIMES (1/4+ SECOND)?

NCOUNT

BNE SOUTH ;IF NOT, DO SOUTH ROUTINE

LDA NASSEM ;OTHERWISE, PUT NASSEM IN NTRAIN

STA NTRAIN

LDA $00 ;AND RESET NCOUNT

STA NCOUNT

SOUTH LDA BCRASM ;EQUIVALENT SOUTH ROUTINE

AND $S0F ;DROP NORTH BITS- SOUTH ALIGNED
CMP SASSEM  ;CONTINUE AS NORTH ROUTINE
BEQ SAME2
    SASSEM
LDA #00
STA SCOUNT
JMP HSKEEP  ;HEAD FOR CONTROLLER ROUTINE
SAME2 INC SCOUNT
    LDA #06
    CMP SCOUNT
    BNE HSKEEP
    LDA SASSEM
    STA STRAIN
    LDA #00
    SCOUNT
HSKEEP LDA SIDTR  ;CONTROL ROUTINE- SIDING CLEAR?
    BEQ NTCK  ;IF SO, GOTO NTCK
        #01  ;ELSE, CHECK NORTH CAUTION BIT
    BIT CAUT
        DECEDEL  ;IF RESET, GOTO DECEDEL
    LDA CONTRL  ;OTHERWISE, SET UP NORTH YELLOW LIGHT, TRACK ON
        #$E2
        #$14
    STA CONTRL
    JMP STCK  ;AND HEAD FOR SOUTH TRACK ROUTINE
( DECEDEL LDA CONTRL  ;SET NORTH TRACK STOP LIGHT, TRACK KILLED
    AND #$E2
ORA #$19
STA CONTRL
DEC CLRDEL ; DECREMENT ASSURED CLEAR TRACK DELAY
BNE STCK ; IF TIMER NOT OUT, GOTO SOUTH TRAIN ROUTINE
LDA SIDTR ; ELSE, GET TRAIN NUMBER ON SIDING
STA LEAVTR ; NOTE IT AS ON THE MAINLINE NORTH
LDA #$00 ; PUT A ZERO IN SIDTR TO SIGNAL STUDENTS
STA SIDTR
LDA #$FF ; SET MAXIMUM KEEP CLEAR DELAY
STA RELDEL
JMP STCK ; AND GOTO SOUTH TRAIN ROUTINE

NTCK LDA NTRAIN ; ANYTHING ON NORTH TRACK?
BNE RESCNT ; IF SO, GOTO RESCNT
LDA LEAVTR ; ELSE, ANYTHING CLEARED TO LEAVE SIDING?
BNE OUTDEL ; IF SO, GOTO OUTDEL
LDA CONTRL ; ELSE, SET NORTH CLEAR LIGHT AND TRACK ON
AND #$E2
ORA #$C0C
STA CONTRL
JMP STCK ; AND GOTO SOUTH TRAIN ROUTINE

RESCNT LDA #44 ; SET UP 2 SECOND INSURE CLEAR DELAY
STA CLRDEL
LDA #01 ; CHECK NORTH CAUTION BIT
BIT CAUT
BEQ NRDOFF ; IF NOT SET, GOTO NRDOFF
LDA NTRAIN ; OTHERWISE, THE TRAIN GOES ON THE SIDING
STA SIDTR

LDA #00 ; AND THE NORTH TRACK MAY BE USED UNDER CAUTION
STA NTRAIN

LDA CONTRL ; SET NORTH CAUTION LIGHT, TRACK ON
AND #$E2
ORA #$14
STA CONTRL

JMP STCK ; AND GOTO SOUTH TRAIN ROUTINE

OUTDEL DEC RELDEL ; DECREMENT DELAY COUNT FOR TRACK HELD CLEAR
BNE NRDOFF ; IF NOT TIMED OUT, GOTO NRDOFF
LDA #00 ; OTHERWISE, ZERO OUT LEAVTR
STA LEAVTR

JMP STCK ; AND HEAD FOR SOUTH TRAIN ROUTINE

NRDOFF LDA CONTRL ; SET NORTH STOP LIGHT, TRACK OFF
AND #$E2
ORA #$19
STA CONTRL

STCK LDA STRAIN ; SOUTH TRAIN ROUTINE- GET SOUTH TRACK TRAIN NO
BEQ GRON ; IF ZERO (TRACK CLEAR), GOTO GRON
LDA #02 ; CHECK SOUTH CAUTION BIT
BIT CAUT

BNE YELON ; IF SET, GOTO YELON
LDA CONTRL ; ELSE, SET SOUTH RED LIGHT, TRACK OFF
AND #$1D
ORA #$C2
STA CONTRL
JMP FINI ;AND GOTO FINI

YELON LDA CONTRL ;SET SOUTH YELLOW LIGHT, TRACK ON
AND #$1D
ORA #$A0
STA CONTRL
JMP FINI ;AND GOTO FINI

GRON LDA CONTRL ;SET SOUTH GREEN LIGHT, TRACK ON
AND #$1D
ORA #$60
STA CONTRL

FINI LDA SIDTR ;MOVE SIDTR TO STSIDTR FOR STUDENTS
STA STSIDTR
RTS ;RETURN TO COMMUNICATIONS FOR SUBSEQUENT RTI
END
## System Connector Schedule

<table>
<thead>
<tr>
<th>Name</th>
<th>J1 (44 pin)</th>
<th>DB25 (25 pin)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>16</td>
</tr>
<tr>
<td>PB1</td>
<td>6</td>
<td>17</td>
</tr>
<tr>
<td>PB2</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>3</td>
</tr>
</tbody>
</table>

Track A  
Track B

All other pins are for user expansion.