Developing a Leap-Frog Catch and Release Rail System with Dual Trains

MAE 412

Thomas McBride and Jeb Carter

May 11, 2022

Introduction

The objective of this project was to create an automated computing system which could sense, signal, and actuate to accomplish some task involving a model train system. The resulting system is a catch-and-release system for a dual-train configuration on a closed loop of track. The existing infrastructure of the test stand incorporates a bar code “blocking” system, designating a north block and a south block spanning opposite sides of the loop. To prevent collisions, only one train may be on a block at any time. Once a train has departed the block, another sensor clears the block, allowing the second train to enter. Because of the design of this system, any train which is brought to rest or diverted off the north block will bring the system to a halt, since any other trains waiting at the south-to-north checkpoint will never be allowed onto the block without a clear. This is the inspiration for the project.

The control system developed herein captures a target train (train #1) from the north block onto a separate loop of track, colloquially, putting it ‘in jail’. Once train #1 is brought to rest on the diverted track, the north block is cleared artificially, allowing a second train (train #8) to enter the block, continue past the jail track, and exit the north track back onto the south track on the other side. Once this occurs, train #1 can be confident that train #8 is no longer occupying the north track. As such, the system released train #1 back onto the north track where it then departs the block. At this point, the system has returned to the initial state and can repeat the sequence. In essence, the capture system will always ignore train #8, allowing it to pass jail, and will always capture train #1, then only releasing it once confirmation is given that the north block is clear, preventing all collisions.

In order to accomplish this task, the electronic hardware required includes: 1 vector board computer with a variety of chips, 2 mechanical railway switches, 2 trickle-charge switch control circuits, 2 Hall-Effect sensors, 1 power relay circuit, 1 latch circuit, and 1 secondary power supply circuit.

Overview

To successfully implement the catch and release mechanism, the below track layout (generated by the track layout software, AnyRail) was created, consisting of a large loop of ‘jail’ track branching from the standard North Track via two railroad switches.

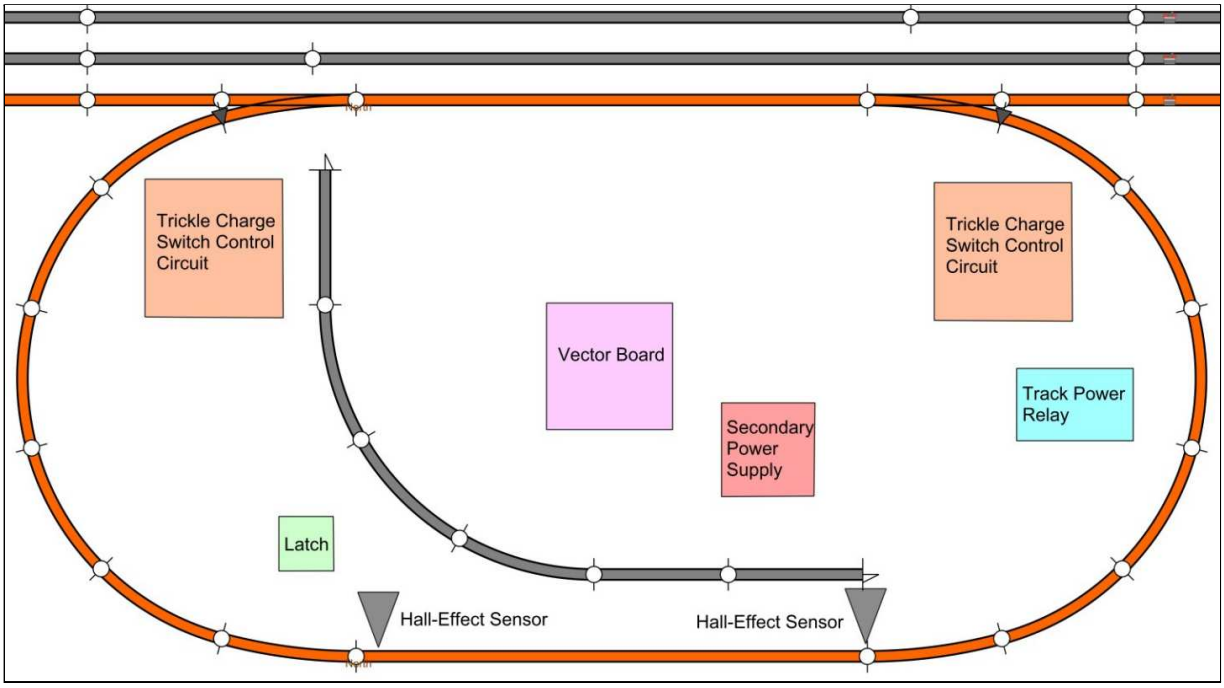


Figure 1: AnyRail Diagram of Catch-and-Release Board

The railroad switches are actuated using the Trickle Charge Switch Control Circuit as required to ‘catch’ or ‘release’ train #1, while allowing train #8 to pass unabated. Immediately when train #1 is detected on the North Track, the railroad switches are repositioned to allow for capture onto the jail loop. A Hall Effect sensor connected to a Latch Circuit is further used to detect the presence of train #1 on the loop and, upon detection of the train, a Track Power Relay Board is used to toggle the power to the loop, successfully capturing the train. The North Block is then

cleared using an open-collector 7405 chip, allowing train #8 to enter the North Track. Once train #8 passes the jail loop and clears the North Track, the Track Power Relay Board is again used to toggle power to the jail track, allowing train #1 to move. Concurrently, the railroad switches are reoriented to allow for the release of train #1 onto the North track. Further along the jail loop but before the exit onto the North Track, train #1 encounters an additional Hall Effect sensor serving as the 'clear' for the Latch Circuit. Train #1 then exits the jail loop at nearly the same time that train #8 enters the North Track and resets the railroad switches to the straight orientation. This 'race' between train #1 exiting the jail loop and train #8 resetting the railroad switches was timed to allow for adequate time for train #1 to escape. The maneuvers then repeat when train #1 is redetected on North Track after train #8 clears the North Track.

Generally, all communications between the daughter boards/sensors and the mother Vector Board computer are accomplished using port A of the VIA chip, buffered by a 74LS244N chip to avoid direct connections between the computer and external devices (Figure 2 maps the VIA pins to their various connections). The only exception to this is pin A0 which is linked to an open collector 7405 chip, connected to the N-Clear line. Additionally, all daughter boards and sensors receive power from an independent power supply (a 'Wall Wart') to avoid excessive current draw from the Vector Board. The track power/ground are also completely isolated from the Vector Board using an opto-isolator to avoid excessive power input to the sensitive computer components. For more specific information regarding the individual daughter board circuits and logic, see the Daughter Board section.

VIA Port	Input / Output	Connection	Purpose
A0	Output	Umbilical - North Clear - Brown	Clear the North Block
A1	Output	Switch 1 Trickle Charge Circuit - Direction	Set direction of switch 1
A2	Output	Switch 1 Trickle Charge Circuit - Trigger	Trigger switch 1

A3	Output	Switch 2 Trickle Charge Circuit - Direction	Set direction of switch 2
A4	Output	Switch 2 Trickle Charge Circuit - Trigger	Trigger switch 2
A5	Output	Track Power Relay Circuit	Set relay state
A6	Input	Hall-Effect Latch Circuit - Data	Sense current state of latch
A7	-	NC	-
B0	Output	Umbilical - White	Run default Echo Program
B1	Output	Umbilical - Yellow	Run default Echo Program
B2	Input	Umbilical - Blue	Run default Echo Program
B3	Output	Umbilical - Orange	Run default Echo Program
B4	Output	Umbilical - Red	Run default Echo Program
B5	Output	Umbilical - Green	Run default Echo Program
B6	-	NC	-
B7	-	NC	-

Figure 2: VIA Port Connection Summary

Vector Board Computer

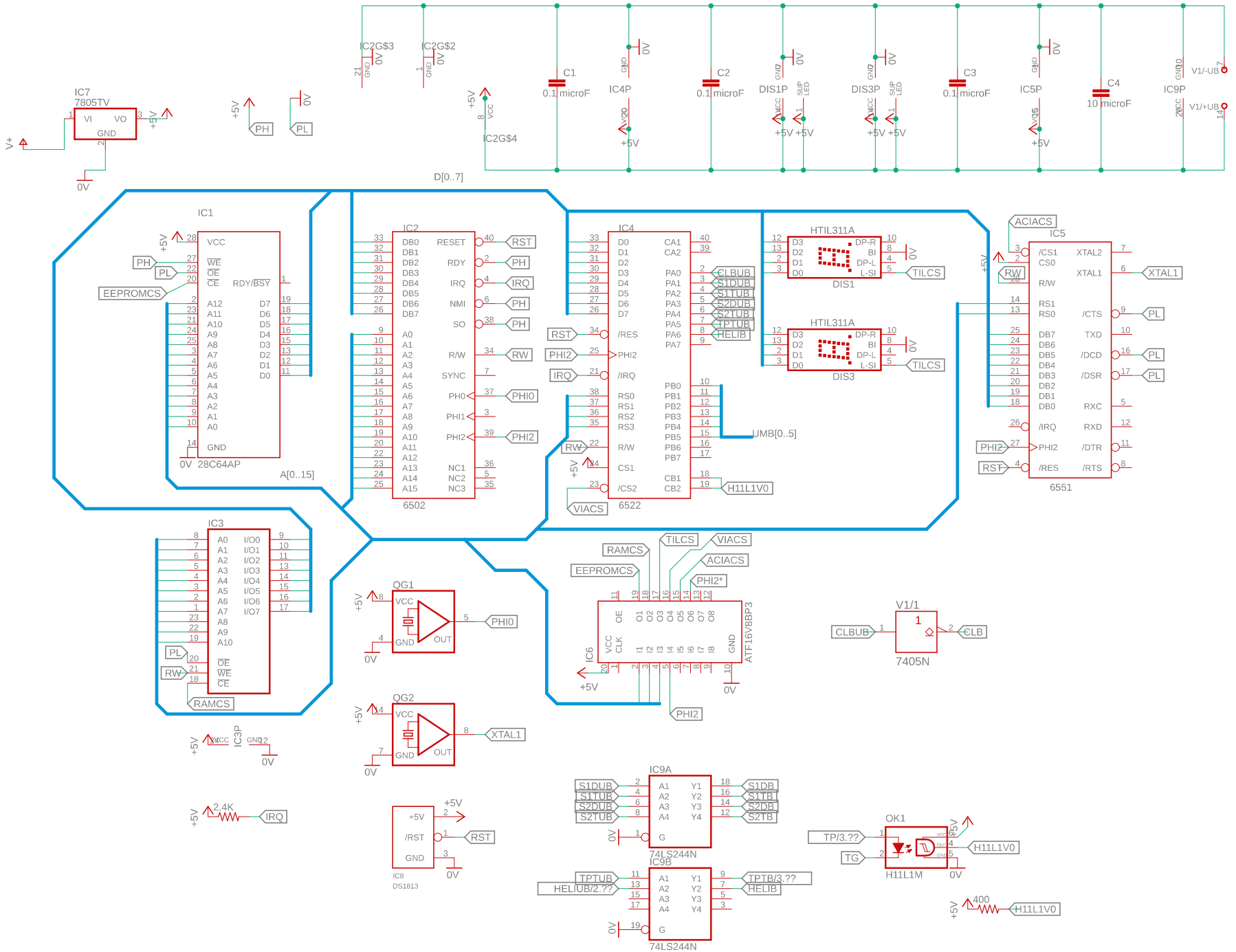
Overview

The Vector Board Computer (Schematic Below) is composed of 15 individual circuit boards connected with 3M's Scotchflex system. The board includes a power adaptor with an accompanying heat sink used to convert the high high voltage from the wall wart supply into a usable 5V supply. Running the length of the vector board are two soldered bus strips which supply the 5V supply and ground to the entire board. These buses are connected by four 0.1 µF de-spiking capacitors. The following table contains an overview of every incorporated chip.

Chip	ID Code	Purpose
EEPROM	28C64B	Read-only memory containing written code
RAM	6116	Temporary read-write memory
MPU	SY6502	Primary computer
VIA	R6522	Interface for external data transfer
Digital Display (x2)	TIL311	Display two 8-bit hex digits
ACIA	6551	N/A - Not Used
GAL	GAL16V8	Stores and executes Address Decode Logic (ADL)
Open Collector Inverter	7405N	Drives N-Clear output pin low
Buffer (x2)	74LS244N	Regulate I/O lines
1 MHz Crystal Clock	14349	Output a 1 MHz Oscillation on PHI0
1.8432 MHz Crystal Clock	27879	Output a 1.8432 MHz Oscillation on XTAL1
Power-On Reset	DS1813	Resets computer to start-state at power-on
Trigger Output Optocoupler	H11L1M	Isolates vector board from track power & ground

Figure 3: Vector Board Computer Chips

VECTOR BOARD



EEPROM (28C64B)

The EEPROM is responsible for storing the written user code that is run by the Vector Computer. It is programmed using a SUPERPRO 280U device program with an inputted HEX file which is coded in the Avocet Wide environment. The pin-out for the chip is represented in Figure 4. Pins 28 and 14 are pulled to +5V and ground, respectively. Pins A0-A12 represent address lines while the I/O pins represent data lines, and these pins are connected to the MPU through a daisy chain connecting all such pins for the MPU, RAM, EEPROM, VIA, ACIA, and the TIL displays. Furthermore, pin 27 is pulled to +5V to disable writing to the chip during operation, while pin 22 is pulled to ground to allow for continuous output. Pin 20, the chip enable, is wired to the GAL to allow for proper address decode logic to operate the chip.

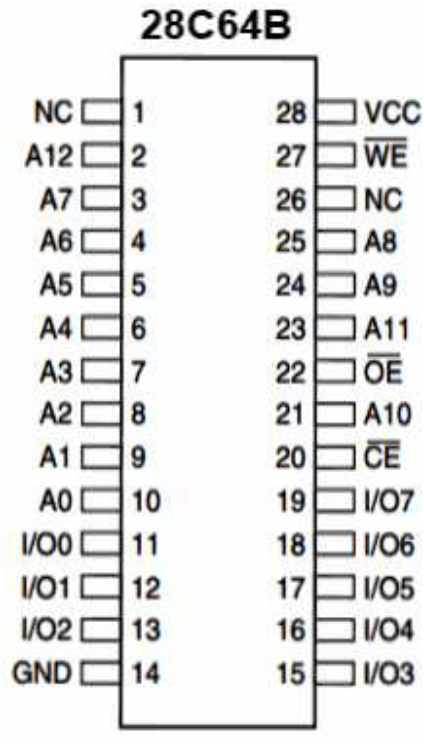


Figure 4: EEPROM Chip Pin-Out

RAM (6116)

The RAM chip holds temporary read-write memory locations and data for the Vector Computer. The pin-out for the chip is shown in Figure 5. Pins 24 (Vcc) and 12 (GND) are connected to +5V and ground, respectively. In a similar manner to the EEPROM chip, pins A0-A10 serve as address lines while the I/O pins are connected to the MPU. Pin 20 (OE bar) is pulled to ground to allow for continuous output, and pin 21 (WE bar) is connected to pin 34 (R/W pin) on the MPU. Finally, pin 18 (CS bar) is wired to the GAL to allow for proper address decode logic to operate the chip.

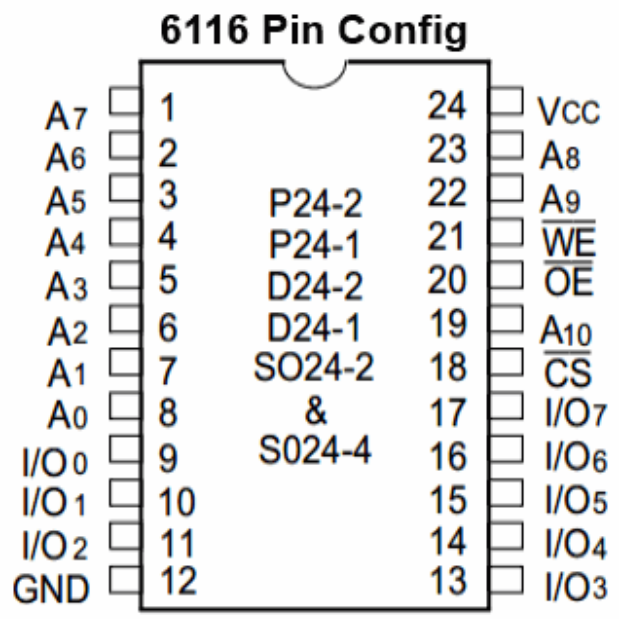


Figure 5: RAM Chip Pin-Out

VIA (6522)

The VIA serves as the Vector Computer's connection to the various external components, after those components are run through a logic buffer for protection. Figure 6 displays the pin-out for the VIA chip. Pins 20 (Vcc) and 1 (Vss) are pulled to +5V and ground, respectively. The pins

labeled RS0-RS3 serve as the address lines for the chip, while pins D0-D7 are the data lines. All data and address pins are daisy-chained to the corresponding pins on the MPU. Pin 34 (RST bar) is connected to the power-on reset chip. Pin 21 (IRQ bar) is connected to the IRQ pin on the MPU and pulled to +5V through a 2.4 kΩ resistor. Pin 22 (R/W bar) is connected to the Read/Write pin of the MPU. Pin 24 (CS1) is wired directly to +5V, while pin 23 (CS2 bar) is connected to the GAL to allow for proper chip selection via the address decode logic. Pin 25 (ϕ_2) is connected to the GAL as well, which provides a buffered form of the ϕ_2 clock. Pins 18 and 19 (CB1 and CB2, respectively) are pulled to +5V through a 400 Ω resistor and connected to the output of the trigger output optoisolator to monitor the state of track power. Pins labeled PA0-PA7 represent the 'A' port, while pins PB0-PB7 represent the 'B' port, and their connections are indicated in Figure 2.

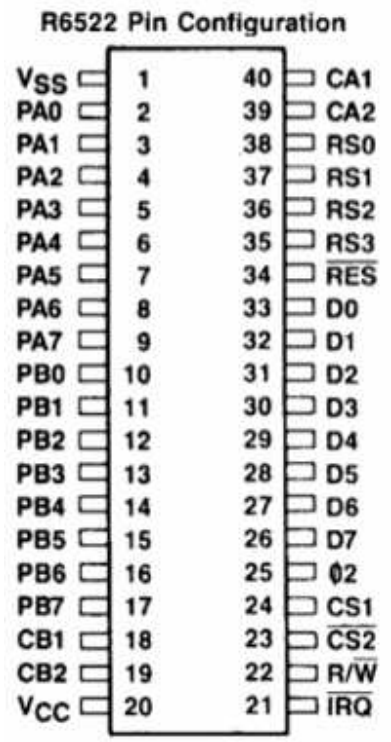


Figure 6: VIA Chip Pin-Out

MPU (6502)

The MPU serves as the ‘brain’ of the Vector Computer, operating all the chips and performing actions based on the code provided in the EEPROM/RAM. The pin-out for this chip is shown in Figure 7. Pin 8 (Vcc) is connected directly to +5V, while pins 1 and 21 (both Vss) are pulled to ground. Pins AB0-AB15 represent the 16 address pins while pins DB0-DB7 represent the 8 data pins. Pin 39 (ϕ_2) serves as the source of ϕ_2 for the GAL. Pin 37 (ϕ_0) serves as the input of the clock signal from the 1 MHz clock chip. Pin 40 (RES bar) is connected to the power-on-reset chip, while pins 2 (RDY), 6 (NMI bar), and 38 (S.O.) are pulled directly to +5V. Pin 4 (IRQ bar) is connected to the VIA IRQ pin and pulled to +5V through a 2.4 kΩ resistor. Pin 34 (R/W bar) is connected to the various R/W pins of other chips, as it drives the R/W process.

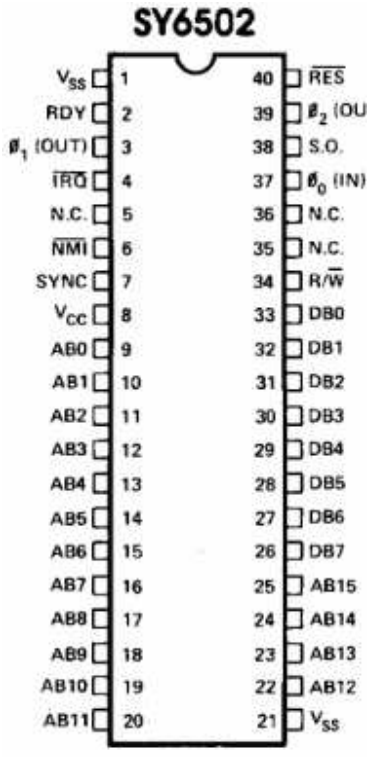


Figure 7: MPU Chip Pin-Out

TIL311 Display

The TIL311 chips display the data present on the data lines connected to them. The first TIL display is connected to data line D0-D3 while the second is connected to D4-D7, in order to capture all the data available. On both chips, pins 1 (LED Supply Voltage) and 14 (Vcc) are connected directly to +5V while pins 7 (Common Ground) and 8 (Blanking Input) are connected to ground. Pin 5 is also connected to the GAL for use in chip selection using the address decode logic.

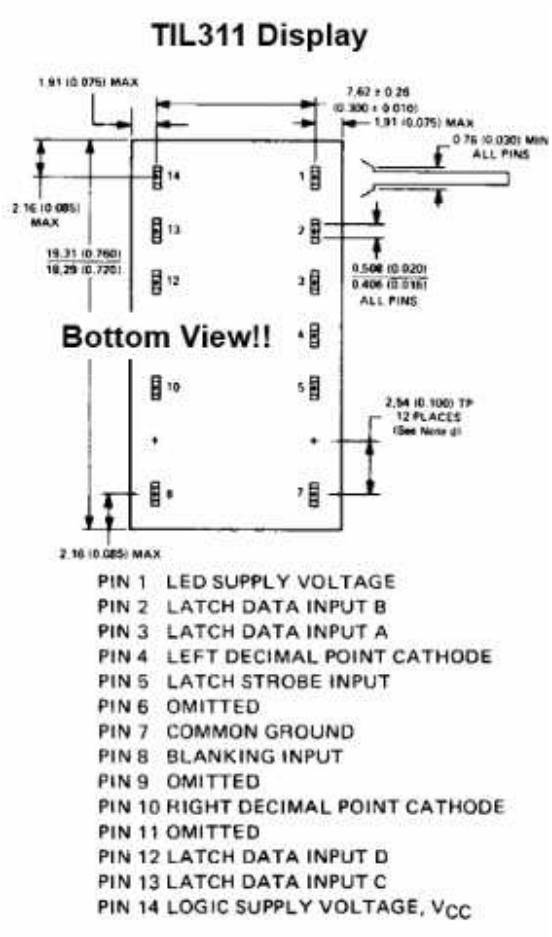


Figure 8: TIL Display Pin-Out

GAL (16V8)

The GAL16V8 handles all Address Decode Logic (ADL) for the computer. This read-only memory contains programmable logic which, based on the value of the 3 highest address lines, A13, A14, and A15, brings a desired chip select (CS) pin low to designate that chip as selected. This system is essential to ensure that only one chip is active at a time, allowing for clear communication between chips and avoiding mixed messages. The format of the programmed GAL is shown on the following page. The pin-out for the GAL is shown in Figure 9 below. Pins 15-19 are wired to the chip select pins of the ACIA, VIA, TIL, RAM, and EEPROM, respectively, and Pin 14, which is the buffered ϕ_2^*, is wired to the corresponding ϕ_2^* pins on the VIA and ACIA. Pins 2-4 correspond to the high address lines, specifically A15, A14, and A13, respectively. Pin 5 is connected to ϕ_2 unbuffered, as received from the MPU. Pin 20 (Vcc) is connected to +5V and pin 10 (GND) is connected to the common ground.

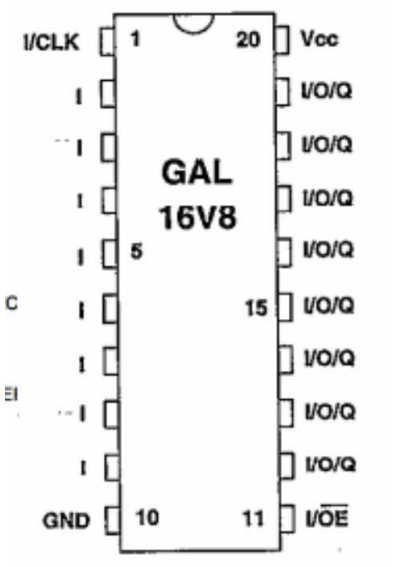
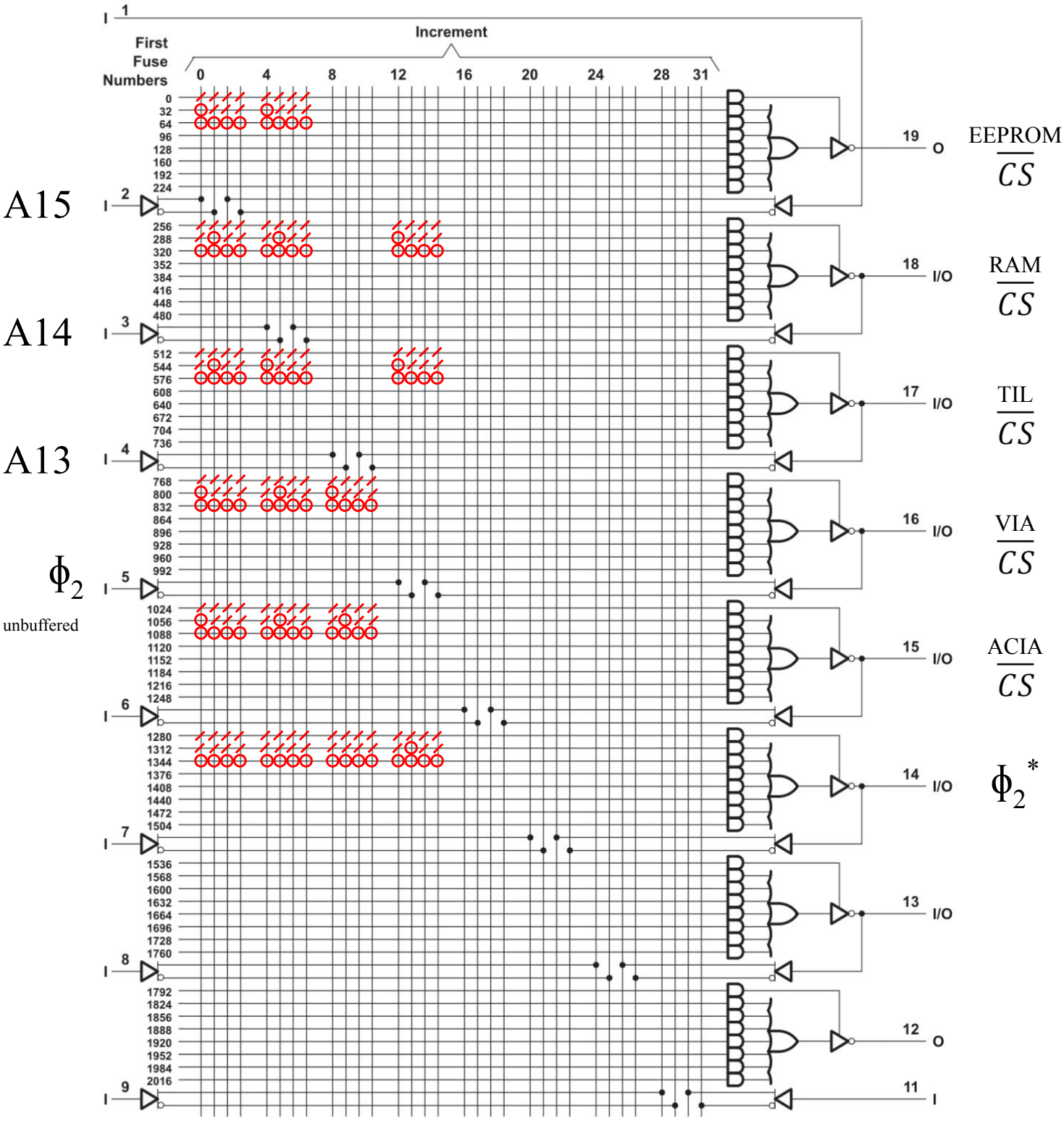


Figure 9: GAL Pin-Out

Thomas McBride & Jeb Carter
MAE 412

logic diagram (positive logic)



Fuse number = First fuse number + Increment

Buffer (74LS244N)

The buffers protect the computer from electrical signals received from external sources as well as standardize “high” and “low” pins to the +5V / Ground conventions. These chips make the system robust and prevent catastrophic problems. Figure 10 shows the pin-out for the buffer. Each input pin on the buffer has a corresponding output which, if enabled by the corresponding enable pin, will output the same state as its connected input. From the VIA, the lines for switch 1 direction, switch 1 trigger, switch 2 direction, switch 2 trigger, and track power toggle are all taken at input pins. Their corresponding buffered versions are connected from their corresponding output pin to the appropriate data output lines to the test stand. The Hall-Effect Sensor Latch unbuffered is used as an input and the corresponding output is connected to the VIA. All of these connections are described in Figure 2. Additionally, all enable pins are wired to common ground along with pin 10 (GND). Pin 20 (Vcc) is connected to +5V.

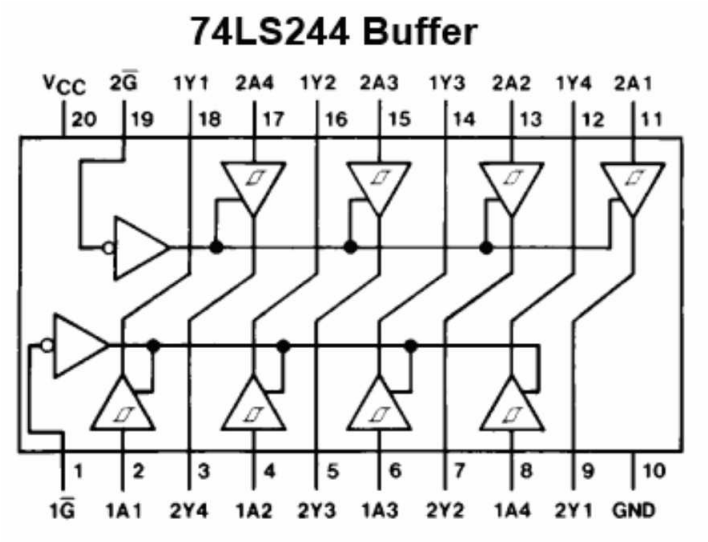


Figure 10: Buffer Pin-Out

Trigger Output Optocoupler (H11L1M)

The Trigger Output Optocoupler shields the vector computer from the track power and ground, which operate at a much higher voltage (+20V) than the vector board (+5V). It also indicates the state of the track power through its sole output pin, V_0. Figure 11 shows the Optocoupler pin-out. Pins 1 and 2 (Anode and Cathode) correspond to the track power and track ground, respectively. Pin 6 (Vcc) is wired to +5V and pin 5 (GND) is wired to common ground. Pin 6 (V_0) represents the output pin of the chip, conveying information about the state of the track power. It is pulled high by a 400 Ω resistor and connected to pins 18 and 19 on the VIA (CB1 and CB2). V_0 floats high by default, and is pulled low in the presence of a sufficient supply voltage.

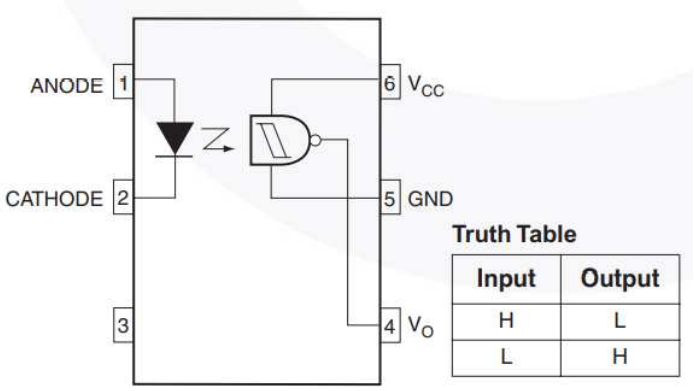


Figure 11: Trigger Output Optocoupler Pin-Out

Power-On-Reset (DS1813)

The Power-On-Reset (POR) chip sends a reset signal to start the computer from start upon power-on. The pin-out for the chip is shown in Figure 12. The operation of the POR chip is simple. Pin 2 is connected to +5V and pin 3 is connected to ground. Pin 1 (RST bar) is the output pin, which floats high when inactive. When the chip detects a power-on, the reset pin pulls low

temporarily, then returns high. This output is connected to the reset pins on the VIA, MPU, and ACIA.

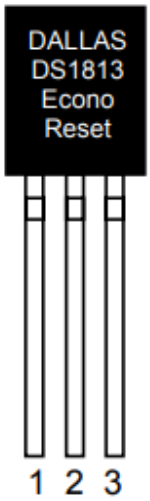


Figure 12: Power-On-Reset Pin-Out

Open Collector Inverter (7405N)

The open collector inverter is used to control the state of a vector board output line which may be influenced by other connections beyond the test stand. In this case, the open collector allows the vector board to pull the N-Clear line low, clearing the north block, even if other power sources pull the same line high by default. It also includes an internal buffer to protect the vector board from external voltages. The inverter functionality is irrelevant, merely reversing the value of the bit which slightly changes the computational logic of the block-clear operation. The pin-out of the open collector is shown in Figure 13. The +5V power and ground are connected to pins 14 (Vcc) and 7 (GND), respectively. The open collector uses one input and one output pin. The unbuffered clear block line from the VIA is connected to pin 1 (1A) and the buffered (and inverted) output in connected to pin 2 (1Y).

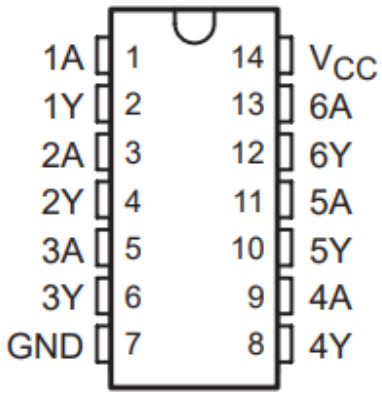


Figure 13: Open Collector Inverter Pin-Out

Clocks (14349 & 27879)

The vector board includes 2 clocks, a 1 MHz clock (14349) which supplies ϕ_0 to the MPU and a 1.8432 MHz clock (27879) which is connected to the ACIA through the XTAL1 line. Pin-outs for both clocks are shown in Figure 14. For the 1 MHz clock, pin 4 (GND) is connected to ground and pin 8 (Vcc) is connected to +5V. Pin 5, the output pin, emits a 1 MHz clock oscillation. For the 1.8432 MHz clock, pin 7 (GND) is connected to ground and pin 14 (Vcc) is connected to +5V. Pin 8, the output pin, emits a 1.8432 MHz clock oscillation.

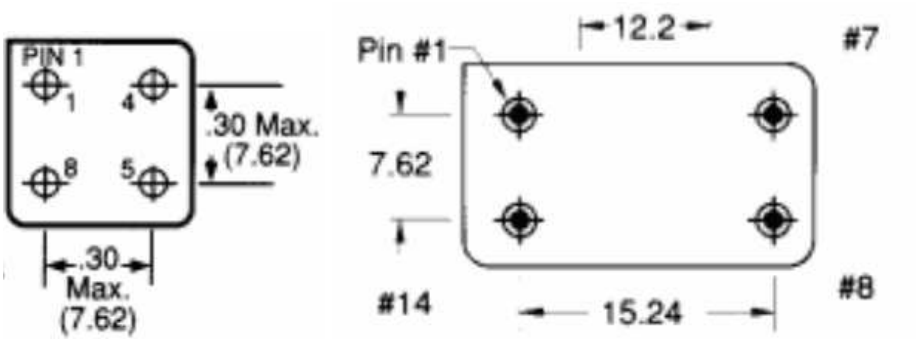


Figure 14: 1 MHz and 1.8432 MHz Clock Pin-Outs

Daughter Boards

Trickle Charge Switch Control Circuit

Two of these boards are used to set the direction of and trigger the railroad switches. Each board uses two output pins of the VIA (pins A1 & A2 or A3 & A4). The first pin is used to drive the directionality of the railroad switches (low meaning the direction is set to turning, high meaning the direction is set to straight). The second pin drives the discharge of two 4700 micro-Farad capacitors, ultimately triggering the physical mechanism (relay) to switch to whichever direction the direction bit is indicating. This pin must only be driven low for a short period (a few milliseconds) before being brought high again. By multiplying the resistance of the resistor in series with the capacitors, it is possible to determine the approximate charging time of the capacitors to be ~9 seconds (for just below ⅔ charging). The power and ground for the components were provided by the separate power supply (‘Wall Wart’), while the capacitors were charged using the track power. Figure 15 shows the circuitry schematic of the trickle charge switch control circuit.

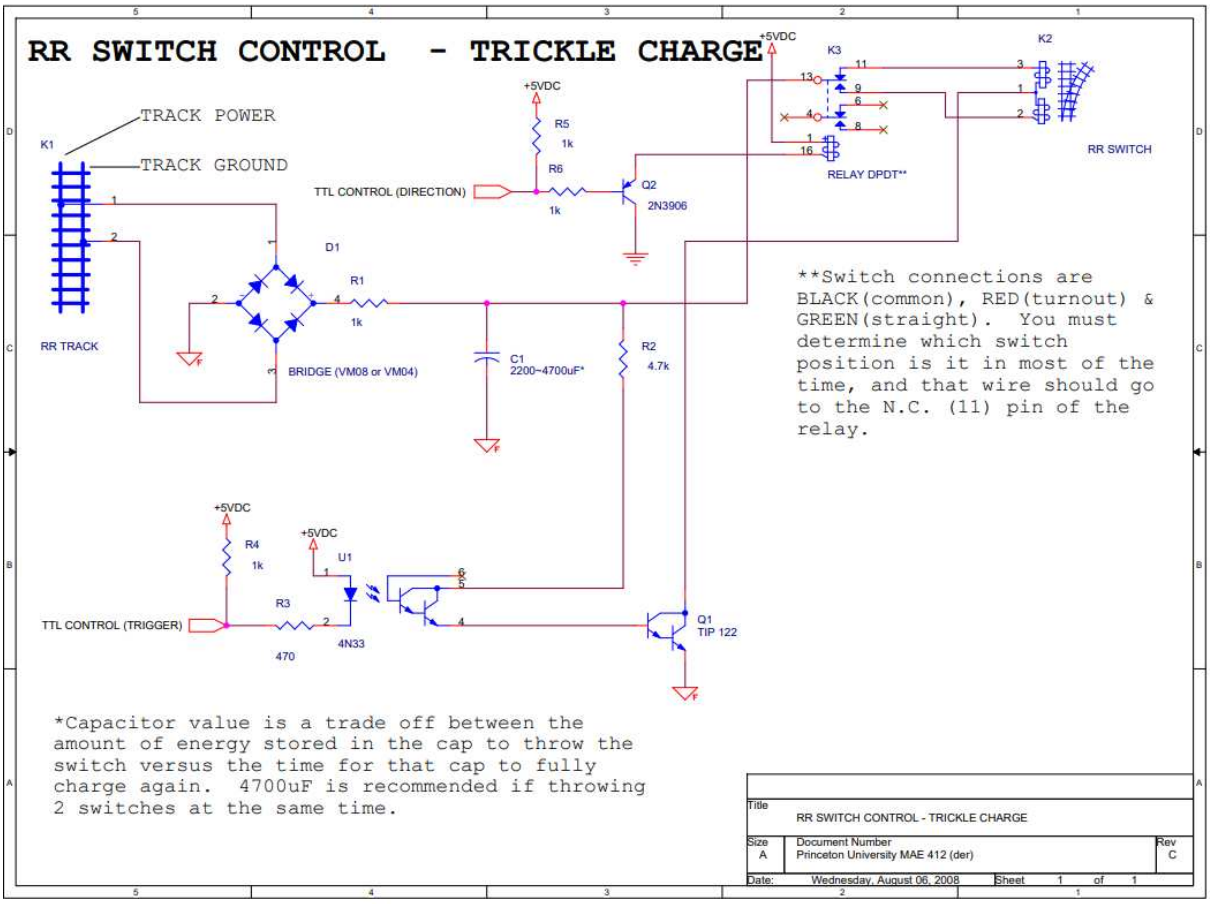


Figure 15: Trickle Charge Circuit Schematic

Track Power Toggle Relay

This circuit is responsible for toggling the power to the jail track in order to capture train #1. It consists of a transistor which is toggled by pin A5 on the Vector Board VIA chip (buffered through the 74LS244N) as well as a relay which connects the main track power to the power of the jail loop. By default, the track power and loop power is open. When power is supplied to the relay, they connect and the jail loop receives power. The output of pin A5 is pulled high to +5V so that when it is low, it serves as a sink, turning off the transistor; whereas, when it is high, it causes the transistor to turn on, thus sending power to the relay. Figure 16 shows the track power toggle relay schematic.

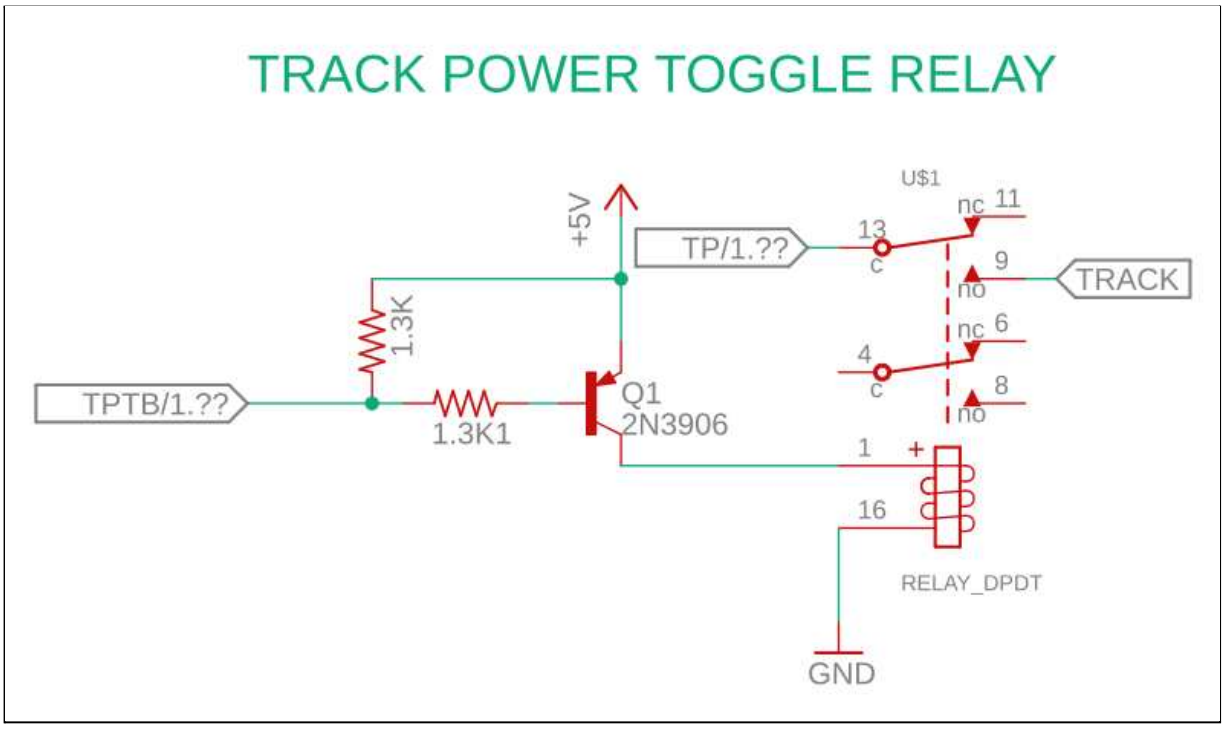


Figure 16: Track Power Toggle Relay Circuit Schematic

Hall-Effect Flip-Flop Board

This board, the schematic for which is shown in Figure 17, is used to sense the presence of a train on the jail track and consists of two Hall-Effect sensors (each with a power, ground, and data-out pin) as well as a 7476N Flip-Flop (consisting of set, reset, power, and ground pins). All powers and grounds are connected to the external power supply (‘wall wart’). Furthermore, the data-out lines for both Hall-Effect sensors are pulled to +5V using a 2.2k Ohm resistor. The data-out of the first Hall-Effect sensor (the sensor first encountered first by a train running along the jail loop) is connected to the ‘set’ pin for the Flip-Flop, while the data-out of the second Hall-Effect sensor is connected to the ‘clear’ pin. Additionally, the data-out of the Flip-Flop chip is fed into the Vector Board through the 74LS244N buffer chip to pin A6 on the VIA.

During nominal operation, train #1 travels along the ‘jail’ loop before encountering the first Hall Effect sensor. The output of the sensor then sets the latch of the flip-flop circuit. The latch is implemented to provide ample time for the Vector computer to identify the signal from the Hall-Effect sensor. Had a latch not been used, it would be possible for the Vector computer to miss the momentary voltage drop from the Hall-Effect sensor (for example, if it is in an interrupt sequence), and thus it would be possible for the computer to never identify the presence of train #1. With the latch, the output of the Hall-Effect sensor is stored until the ‘clear’ pin is activated. In this design, the data-out of the second Hall-Effect sensor is used as the ‘clear’ pin, unlatching the flip flop circuit and making it ready to detect another train.

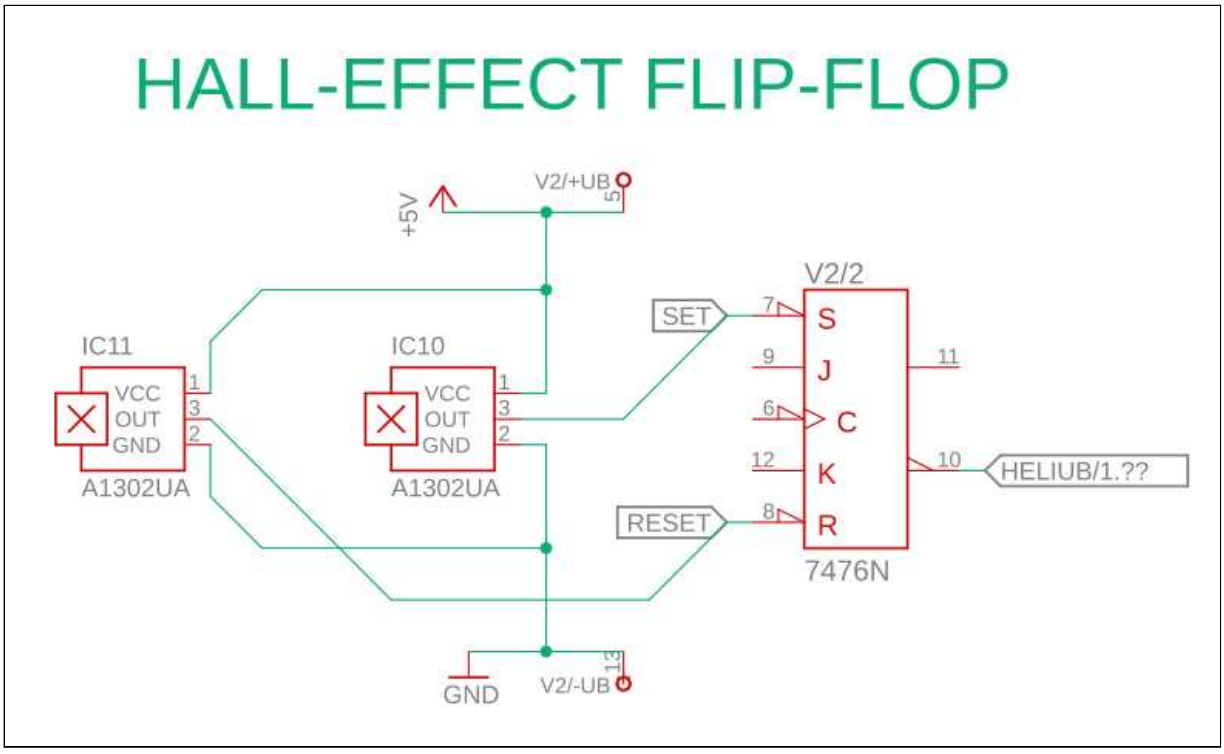


Figure 17: Hall-Effect Flip-Flop Circuit Schematic

Track Hardware

General Construction

All project hardware, including the tracks, Vector Computer, various daughter boards, and associated wires, are housed on a roughly 3’ by 4’ segment of plywood. Tracks consist of N-gauge rails with a turning radius no tighter than 9 ¾”.Tracks are laid according to the AnyRail diagram in Figure 1, and they are mounted using small nails through predrilled holes, set in place with a hammer and finishing tool.

The daughter boards are mounted to the project board using threaded standoffs. Wires are typically twisted together and run in a manner such that no bundles are parallel with another to avoid noise. Staples are used to fix the wires in place as necessary and various holes are drilled in the project board to allow for wires to run underneath tracks where applicable.

Furthermore, power and ground ‘hubs’ are placed in optimal locations to allow for multiple devices to connect to power and ground in a secure fashion. Figure 18 shows the completed project board.

Track Connection to Testing Setup

The three parallel segments of track along the top of the board are laid to allow proper integration with the overall testing setup. Rerailing tracks are placed just inside the project border to allow for rerailing in the case of derailment while traveling between the testing setup and the project board.

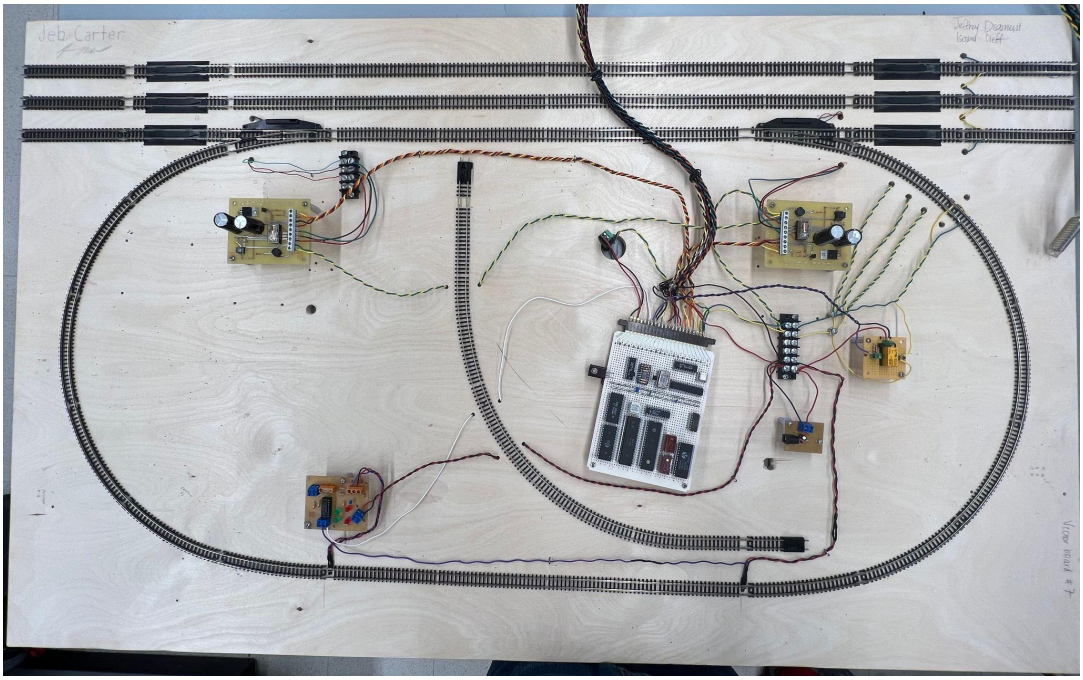


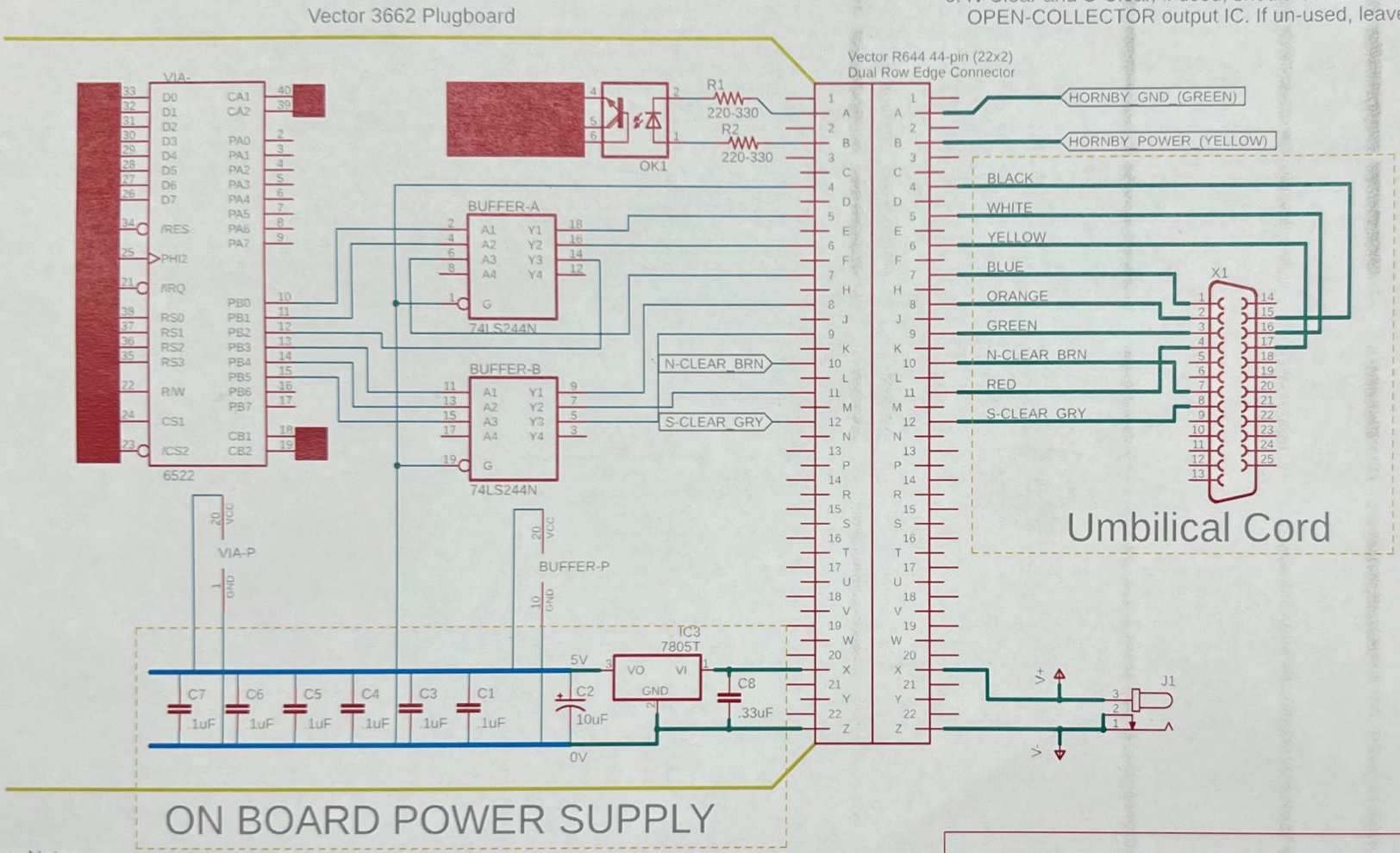
Figure 18: Final Project Board Setup

Vector Computer Connection to Test Setup - Umbilical

The Vector Computer is connected to the overall testing setup using an umbilical cable wired according to the schematic shown on the next page. The wires of the umbilical are connected to the Vector Computer using a 44 pin edge connector. All data lines are wrapped with wires grounded on one end to avoid noise and the grounded wires are connected together to form a grounding star. The wiring of the umbilical are shown in the schematic on the following page.

TEST-STAND CONNECTIONS

Notes:
1. DIN 8-pin , each signal line twist with separate ground.
2. Ground lines connect together at Edge Connector.
3. Twisted grounds not connected (open) at DIN 25-pin.
4. Don't forget to power 74LS244 buffer.
5. If computer is working but block lights do not work, connect Pin 15 of buffer from PB5 to PB6
6. N-Clear and S-Clear, if used, should be connected to an OPEN-COLLECTOR output IC. If un-used, leave disconnected.



Notes:
1. A heat sink should be used with the regulator.
2. Heat sink thermal compound needs to be used between heat sink and regulator.
3. Use a small machine screw and nut to clamp the heat sink to the regulator.
4. Heat sink should float above board. Stagering regulator leads into a triangle footprint help stability and make soldering easier.
5. Use of 30awg on Vector Board, 22awg off-board. The only exception being the voltage regulator to the power bus bars.

TITLE: Test-Stand connections_DB25	
Document Number:	REV: 1.1
By: Jonathan Prévost	
Date: 3/24/22 4:19 PM	Sheet: 1/1

Vector Board Schematic Labels
Table of Contents

Code

Customized Echo 3.1 Program for Autonomous Operation

```
;       Label Declarations
;
DFLAG  EQU $3FF         ; 00 EQU No return data to send
;                       FF EQU Return data ready to send
WDATAH EQU $3FE         ; High byte of return data
WDATAL EQU $3FD         ; Low byte of return data
BLKID  EQU $3FC         ; Track computer block id number
RDATA  EQU $3FB         ; Controller data sent to this computer
FRANUM EQU $3FA         ; Current wave frame number
OKWR   EQU $3F9         ; 00 EQU Controller not ready to accept return data
;                       FF EQU Controller ready to accept return data
INTMPU EQU $3F8         ; FF EQU MPU has issued a poll request
POS    EQU $3F7         ; Output during positive power cycle
NEG    EQU $3F6         ; Output during negative power cycle
PLRTY  EQU $3F5         ; 00 EQU Negative polarity, FF EQU Positive polarity
LCV    EQU $3F4         ; Local variable
MASK   EQU $3F3         ; Local constant
CNT    EQU $3F2         ; Local variable
TIME   EQU $3F1         ; Median period of a '0' and a '1' bit of
;                           controller data
NRTHGN EQU $32B                 ; Flag to tell if north light is green #$00 off #$FF on
SUTHGN EQU $32A                 ; Flag to tell if south light green - #$00 off #$FF on
;
NOVRD  EQU $32C         ; *** V 3.1 *** OVERRIDE OF NORTH LIGHTS/KILLS
                        ; *** V 3.1 *** $00 -> NO OVERRIDE
SOVRD  EQU $32D         ; *** V 3.1 *** OVERRIDE OF SOUTH LIGHTS/KILLS
                        ; *** V 3.1 *** $00 -> NO OVERRIDE
```

```
;
WBUF   EQU $3D0         ; Output table - Data to be written to port B
FTAB   EQU $3AC         ; Reconstructed 'nibbles' of control data
LFLAG  EQU $377
STFLG  EQU $389         ; ALL NEW AND IMPORTANT
TABLE  EQU $38A         ; Input table - IFR contents for each input cycle
;
NTRAIN EQU $3CF         ; Train number on north track
STRAIN EQU $3CE         ; Train number on south track
STSDTR EQU $3CD         ; Student-use location for train on siding
;CAUT   EQU $3CC         ; Caution bit control byte - left over V2.0
;SIDTR  EQU $3CB         ; Program-use  siding train save area - left over V2.0
;RELDEL EQU $3CA         ; Delay count save area for siding train release - V2.0
;CLRDEL EQU $3C9         ; Delay count save area for clear track check - V2.0
BCRASM EQU $3C8         ; Location for bar code shift in
SCOUNT EQU $3C7         ; Delay count for reader passage of south train
NCOUNT EQU $3C6         ; Delay count for reader passage of north train
SASSEM EQU $3C5         ; Assemble area for south train number
NASSEM EQU $3C4         ; Assemble area for north train number
;LEAVTR EQU $3C3         ; Save area for siding train requesting mainline - V2.0
CONTRL EQU $3C2         ; Save area for control bits-lights, track kills
CBITS  EQU $3C1         ; Save area for communications bits during LOCCON
YLTOGN EQU $3C0                 ; Toggle for which light to turn from yellow to green
                                        ;    01 -> south light, 02 -> north light
;
;
;       Versatile Interace Adapter  (VIA)  Addresses
;
VDDRB  EQU $A002        ;  Data direction register B
VORB   EQU $A000        ; Output register B
VT2L   EQU $A008        ; Timer 2 low byte
VT2H   EQU $A009        ; Timer 2 high byte
VACR   EQU $A00B        ; Auxilliary control register
VPCR   EQU $A00C        ; Peripheral control register
```

```
VIFR   EQU $A00D        ; Interrupt flag register
VIER   EQU $A00E        ; Interrupt enable register
;
;
;   **** BEGINNING OF USER SECTION ****
;VIA $8000 ADDRESSES
V8BI    EQU $8000
V8BD    EQU $8002
V8AO    EQU $8001
V8AD    EQU $8003

;LOCAL VARIABLES
DISP    EQU $4000

;INITIALIZATION
        ORG $E000
        SEI
        CLD
        LDX #$FF
        TXS
        LDA #$BF; 1011 1111
        STA $A003       ; Make A-Port Output but A06 input
        LDA #$34
        STA $A001       ; Drive A port to 0011 0100. This sets switch triggers to high.
        LDA #$03; THIS IS COMPUTER No. 3
        STA BLKID
        LDA #$00
        STA NOVRD       ; NO OVERRIDE ON NORTH TRACK
        STA SOVRD       ; NO OVERRIDE ON SOUTH TRACK
        STA DFLAG       ; NO DATA TO SEND
        JSR INIT        ; INITIALIZATION
;       ----------------
START   JMP PB1
;       ----------------
```

```
FOUND   LDA $A001               ; Bring switch 2 low
                AND     #$F7    ; 1111 0111
                STA $A001       ; Bring A3 bit low
                LDA $A001               ; Bring switch 1 high
                ORA     #$02    ; 0000 0010
                STA $A001       ; Bring A1 bit high

                LDX     #$00
                LDY     #$00
DELAYL  INX             ; kill time (0.25 second)
                BNE DELAYL
                INY
                BNE DELAYL      ; Delay

                LDA     $A001           ; Trigger
                AND     #$EB    ; 1110 1011
                STA $A001       ; Bring A02 and A04 low
                LDX #$00
DELAYB  INX             ; kill time
                BNE DELAYB
DELAYC  INX             ; kill time
                BNE DELAYC
DELAYD  INX             ; kill time
                BNE DELAYD
                LDA $A001
                ORA     #$14    ; 0001 0100
                STA $A001       ; Bring A02 and A04 high

WAIT    LDA #$02
                STA DISP
                LDA     $A001           ; Loop until hall-effect is triggered
                AND     #$40    ; 0100 0000
                BEQ     CATCH   ; If A07 is low, H-E was triggered, so jump.
                JMP     WAIT    ; Jump to WAIT
;       ----------------
CATCH   LDA     #$03
```

```
                STA DISP
                LDA $A001               ; Toggle track power
                EOR     #$20    ; 0010 0000
                STA $A001       ; Reverse A05 bit
                LDA     $A001           ; Clear block
                ORA     #$01    ; 0000 0001
                STA $A001       ; Bring A0 high
                LDX #$00
DELAYA  INX             ; kill time
                BNE DELAYA
                LDA $A001
                AND     #$FE    ; 1111 1110
                STA $A001       ; Bring A0 low
        LDA $A001               ; Set switch 2 to high
                ORA     #$08    ; 0000 1000
                STA $A001       ; Bring A3 bit high

                LDX     #$00
                LDY     #$00
DELAYR  INX             ; kill time (0.25 second)
                BNE DELAYR
                INY
                BNE DELAYR      ; Delay

                LDA     $A001           ; Trigger
                AND     #$EB    ; 1110 1011
                STA $A001       ; Bring A02 and A04 low
                LDX #$00
DELAYO  INX             ; kill time
                BNE DELAYO
DELAYP  INX             ; kill time
                BNE DELAYP
DELAYQ  INX             ; kill time
                BNE DELAYQ
```

```
                LDA $A001
                ORA     #$14    ; 0001 0100
                STA $A001       ; Bring A02 and A04 high


WAITA   LDA     STRAIN          ; Loop until STRAIN is 8
                CMP     #$08
                BNE     WAITA
                JMP FREE; Jump to FREE
;       ----------------
PB1             JMP PB2
PB4             JMP FOUND
;       ----------------
FREE    LDA #$04
                STA DISP
                LDA $A001               ; Bring switch 2 high
                ORA     #$08    ; 0000 1000
                STA $A001       ; Bring A3 bit high
                LDA $A001               ; Bring switch 1 low
                AND     #$FD    ; 1111 1101
                STA $A001       ; Bring A1 bit low

                LDX     #$00
                LDY     #$00
DELAYM  INX             ; kill time (0.25 second)
                BNE DELAYM
                INY
                BNE DELAYM      ; Delay
                LDA     $A001           ; Trigger
                AND     #$EB    ; 1110 1011
                STA $A001       ; Bring A02 and A04 low
                LDX #$00
DELAYF  INX             ; kill time
                BNE DELAYF
DELAYG  INX             ; kill time
```

```
                BNE DELAYG
DELAYH  INX             ; kill time
                BNE DELAYH
                LDA $A001
                ORA     #$14    ; 0001 0100
                STA $A001       ; Bring A02 and A04 high
                LDA $A001               ; Toggle track power
                EOR     #$20    ; 0010 0000
                STA $A001       ; Reverse A05 bit
                JMP WAITB
;       --------------
PB3             JMP     PB4
PB2             JMP ARENA
;       --------------
WAITB   LDA #$05
                STA DISP
                LDA     NTRAIN          ; Loop until NTRAIN is 8
                CMP     #$08
                BNE     WAITB
                LDA $A001               ; Bring switch 2 high
                ORA     #$08    ; 0000 1000
                STA $A001       ; Bring A3 bit high
                LDA $A001               ; Bring switch 1 high
                ORA     #$02    ; 0000 0010
                STA $A001       ; Bring A1 bit high

                LDX     #$00
                LDY     #$00
DELAYN  INX             ; kill time (0.25 second)
                BNE DELAYN
                INY
                BNE DELAYN      ; Delay

                LDA     $A001           ; Trigger
                AND     #$EB    ; 1110 1011
```

```
                STA $A001       ; Bring A02 and A04 low
                LDX #$00
DELAYI  INX             ; kill time
                BNE DELAYI
DELAYJ  INX             ; kill time
                BNE DELAYJ
DELAYK  INX             ; kill time
                BNE DELAYK
                LDA $A001
                ORA     #$14    ; 0001 0100
                STA $A001       ; Bring A02 and A04 high
                JMP ARENA       ; Jump to ARENA
;       ----------------

ARENA   LDA #$01
                STA DISP
                LDA     NTRAIN
                CMP     #$01
                BEQ     PB3             ; FOUND

                JMP ARENA
;       -------------------

;
;   **** END OF USER SECTION ****
```