ABSTRACT

An interface has been designed to enable any device with a 300 baud RS-232 serial port to communicate with the commercially available HORNBY Zero-1 model train controller. One can send commands through the interface to the Hornby device which perform the same function as depressing a button on the keypad or manipulating the slide switch on the front of the unit. Since most microcomputers have a serial port, they can be connected to the interface and can be programmed to perform many of the functions that would have been done painstakingly with the Hornbay alone; hence, the interface has enabled a much more "user friendly" environment.

INTRODUCTION

To attempt to control sixteen locomotives at once with the Hornby unit alone is extremely difficult and burdensome. Although the company offers "slave" modules which can give the user real time control over four locomotives at once, and an extension is available which extends to sixteen separate slide controls, one for each loco, it is very costly to do so, and really does not take full advantage of the microprocessor controlled Hornby unit. By enabling any device with a 300 baud RS-232 port to communicate with the Hornby, we can use the Hornby much effectively.

Originally, I set out to improve the user friendliness of the Hornby unit. As mentioned above, simultaneous control of many locomotives is very cumbersome, if not just because of the number of keystrokes necessary to change control from one locomotive to another. By attaching a microcomputer to the Hornby, the method of user interaction is limited only by the speed at which the Hornby can detect keystrokes the
maximum communication rate through the interface—about 10 characters per second) and the imagination of the programmer. I have written a set of subroutines in MBASIC (microsoft, inc.) which make the method of communication transparent to the user; he need only know which parameters are necessary to control any given locomotive or send messages to an accessory. It is quite simple for even a novice programmer to use these routines to achieve a much more efficient and friendly interface to the Hornby unit and his model train layout.

I have included three sample programs which use the general subroutines; one enables the user to control any one of ten trains at once (speed, direction), another sends messages to the accessories, and the third combines these two to demonstrate the "user friendly" nature of this interface configuration. The third program takes advantage of the many keys of the computer keyboard and gives the user real time control over six trains while also enabling him to send messages to the accessories. Also, an important aspect of any user interface is introduced: feedback—a visual display of the respective speeds of the locomotives and the message pending or sent to the accessories.

The initial philosophy that I adopted was to let the Hornby continue to do the microprocessor control; the interface would only communicate with the Hornby in some way without attempting any direct locomotive or accessory con-
trol. The interface does not conflict with the normal operation of the Hornby; it is also easily installed without having to "cut traces" or really disturb the existing controller much. The circuit designed can be easily put onto a printed circuit board and installed inobtrusively underneath the Hornby unit.

The interface that has been designed should be rather inexpensive to produce, especially compared with the alternative purchase of sixteen slave units. But, more importantly, the interface enables the user to take full advantage of the power of a separate microcomputer and the inherent power of the Hornby.

THEORY OF OPERATION

HARDWARE

General

Refer to the appendices for schematics and drawings. The interface operates by receiving control characters via a standard, non hand-shaked, 300 baud RS-232 line, decoding the characters, and acting appropriately. All commands to the interface consist of a single legal control character (see figures 1 and 2) corresponding to a single action on the Hornby (e.g. depressing the 'LOCO' key on the Hornby is equivalent to sending the character '<' to the interface when the interface is enabled). All illegal characters are
decoded out and do not effect the interface (although some dummy characters must be sent to separate each legal control character--see keyboard control below).

The hardware does not interfere with the operation of the Hornby; rather, it simulates the keyboard switch closures, or it couples the appropriate signal to the TMS1000 micro-computer within the Hornby according to the character sent to the interface by the user (see figures 1 and 2 for character codes). Most of the interface consists of random logic CMOS integrated circuits, powered by the Hornby's internal 15 volt power supply (a higher power 7815 voltage regulator is installed on the Hornby PC board instead of the existing 78L15); the remainder of the interface circuitry is dedicated to the communications interface--an RS-232 line receiver, a baud rate generator, and a UART--powered by a 78L05 regulator also attached to the 15 volt supply.

It seems that, where there would normally be "contact closures" (e.g. keypad switches), a capacitive coupling is used by the Hornby. The interface employs CMOS 4066 analog switches to simulate these closures. The TMS1000 has been used to control a microwave oven with a capacitive keypad; this is probably why the Hornby uses a capacitive coupling system. The TMS1000 accepts the 4066 closure as equivalent to the capacitive couple without any trouble.
<table>
<thead>
<tr>
<th>Speed</th>
<th>Control Character</th>
<th>ASCII Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>&quot;A&quot;</td>
<td>100 0001</td>
</tr>
<tr>
<td>1</td>
<td>&quot;C&quot;</td>
<td>100 0011</td>
</tr>
<tr>
<td>2</td>
<td>&quot;B&quot;</td>
<td>100 0010</td>
</tr>
<tr>
<td>3</td>
<td>&quot;F&quot;</td>
<td>100 0110</td>
</tr>
<tr>
<td>4</td>
<td>&quot;G&quot;</td>
<td>100 0111</td>
</tr>
<tr>
<td>5</td>
<td>&quot;E&quot;</td>
<td>100 0101</td>
</tr>
<tr>
<td>6</td>
<td>&quot;D&quot;</td>
<td>100 0100</td>
</tr>
<tr>
<td>7</td>
<td>&quot;L&quot;</td>
<td>100 1100</td>
</tr>
<tr>
<td>8</td>
<td>&quot;M&quot;</td>
<td>100 1101</td>
</tr>
<tr>
<td>9</td>
<td>&quot;O&quot;</td>
<td>100 1111</td>
</tr>
<tr>
<td>10</td>
<td>&quot;N&quot;</td>
<td>100 1110</td>
</tr>
<tr>
<td>11</td>
<td>&quot;J&quot;</td>
<td>100 1010</td>
</tr>
<tr>
<td>12</td>
<td>&quot;K&quot;</td>
<td>100 1011</td>
</tr>
<tr>
<td>13</td>
<td>&quot;I&quot;</td>
<td>100 1001</td>
</tr>
<tr>
<td>14</td>
<td>&quot;H&quot;</td>
<td>100 1000</td>
</tr>
</tbody>
</table>

**Figure 1 - Speed Control Characters**

<table>
<thead>
<tr>
<th>Hornby Key</th>
<th>Control Character</th>
<th>ASCII Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ø</td>
<td>&quot;Ø&quot;</td>
<td>011 0000</td>
</tr>
<tr>
<td>1</td>
<td>&quot;1&quot;</td>
<td>011 0001</td>
</tr>
<tr>
<td>2</td>
<td>&quot;2&quot;</td>
<td>011 0010</td>
</tr>
<tr>
<td>3</td>
<td>&quot;3&quot;</td>
<td>011 0011</td>
</tr>
<tr>
<td>4</td>
<td>&quot;4&quot;</td>
<td>011 0100</td>
</tr>
<tr>
<td>5</td>
<td>&quot;5&quot;</td>
<td>011 0101</td>
</tr>
<tr>
<td>6</td>
<td>&quot;6&quot;</td>
<td>011 0110</td>
</tr>
<tr>
<td>7</td>
<td>&quot;7&quot;</td>
<td>011 0111</td>
</tr>
<tr>
<td>8</td>
<td>&quot;8&quot;</td>
<td>011 1000</td>
</tr>
<tr>
<td>9</td>
<td>&quot;9&quot;</td>
<td>011 1001</td>
</tr>
<tr>
<td>INERTIA</td>
<td>&quot;;&quot;</td>
<td>011 1010</td>
</tr>
<tr>
<td>REVERSE</td>
<td>&quot;;&quot;</td>
<td>011 1011</td>
</tr>
<tr>
<td>LOCO</td>
<td>&quot;&lt;&quot;</td>
<td>011 1100</td>
</tr>
<tr>
<td>←</td>
<td>&quot;=&quot;</td>
<td>011 1101</td>
</tr>
<tr>
<td>→</td>
<td>&quot;&gt;&quot;</td>
<td>011 1110</td>
</tr>
<tr>
<td>FORWARD</td>
<td>&quot;?&quot;</td>
<td>011 1111</td>
</tr>
</tbody>
</table>

**Figure 2 - Keyboard Control Characters**
Speed control

For normal operation, the user controls the speed of the locomotives with the slide switch on the front panel of the Hornby. The slide switch selectively couples together a common signal (Css) to one or more of the four inputs to the TMS1000 microcomputer within the Hornby (K1, K2, K4, and K8—not equivalent to the keyboard lines labelled identically. I label these lines K1ss, K2ss... on my schematics to differentiate them from the keyboard lines which I label K1kbd, K2kbd...). When using the interface, the Hornby's front panel slide switch is disabled; the interface is then free to simulate the switch's encoding by selectively coupling one or more of the Css signal to the four Kss lines with a CMOS 4066 analog switch.

The UART receives a speed control character (see figure 2 for codes) and decodes it for a bank of four analog switches (one 4066) to couple the Css signal to the appropriate Kss lines.

Direction control

One controls the direction of the locomotives by pressing either the forward or reverse switch on the front panel of the Hornby. Internally, this just couples or uncouples a signal from one of the two lines labelled "Sw" to the other. The interface uses two control characters to determine the direction setting. A "?" is interpreted as a command to
forward, a "." to go in reverse. Once a direction control is decoded, the interface sets an RS flip-flop (two cross-coupled NAND gates) which either closes or opens a 4066 switch across the two "SW" lines

Keyboard control

In normal operation, the Hornby's keypad is encoded to couple one of the five lines labelled R5 (not used by the interface), R6, R7, R8, and R9 to one of the lines labelled K1, K2, K4, and K8 on the keypad (I label them Klkbd, K2kbd... because these are not equivalent to the lines labelled the same for the slide switch). The interface simply decodes the control character sent to it by the user, and couples the appropriate signal to the Kkbd lines for a period of time determined by a one-shot (50mS). This one-shot operation is necessary to simulate the "key-push" of the user on the normal keypad. Since the TMS1000 scans the keypad assuming that a human is depressing the keys, the rate of key depression is limited (hence the 50mS key press). Also, the TMS1000 expects to see no key pressed for a time after a key press; at 300 baud, this means that two dummy (blank) characters must be sent after each legal control character. For this reason, a legal control character is latched until the next legal one arrives. The UART can be configured to receive characters at 1200 baud, but this will not speed the rate of communication between the user and the
Hornby. At 1200 baud, eleven dummy characters must be sent between each control character. All of this means that we are simulating a key press of 50mS, and a key release of 49mS (three 300 baud, or twelve 1200 baud characters=99mS). Hence, we actually have an effective communication rate with the Hornby of 100 baud (ten characters per second).

PRACTICAL OPERATION

To use the interface, simply connect a cable between the RS-232 port of the device you are using to the DB-25 plug on the back of the Hornby. Pin 7 is the signal ground, and pin 2 is the signal (the interface is treated exactly as a line printer). To use the Hornby without the interface, put the "Hornby/interface" switch in the Hornby position--the interface will be locked out (no characters will be accepted). To use the interface, just put the switch in the interface position. In this mode, the Hornby keypad will continue to function (except the reverse and forward buttons) so that you can still hit the 'PANIC' or 'CLEAR' keys (or any others if you want); but, the speed control slide switch will have no effect while the switch is in the interface position.

One need not be concerned with the complexities of the Hornby nor with the intricacies of the interface to take advantage of this new tool. Its operation is quite straightforward; in fact, a computer is not even necessary to use the interface. Any device with a 300 baud RS-232 port will
work to control the Hornby with the interface enabled. Using the control characters given in figures 1 and 2, one can simulate a key press on the Hornby, or set the speed for a locomotive (NOTE: it is wise to insert dummy blanks between each control character, even when just typing the characters into the interface). For example, one could type ": 3 < 5 = D ;" which would mean to set locomotive number 5 to inertia 3, speed 8, in reverse; this is equivalent to pressing the keys 'INERTIA' '3' 'LOCO' '5' '<-->', pressing the reverse switch, and setting the speed control to speed 8.

So far, this does not seem like much of an accomplishment; but, when we attach the power of a microcomputer to the interface, we can program entire scenarios which will be sent over the RS-232 port much faster than we can even think. Or, the programmer can set up the computer's keys to correspond to various functions such as raising or lowering the speed of any of a few locomotives (see example program 3). If the user has a real time clock, he could set up a scenario of events which the computer could initiate at the correct time by sending the appropriate characters to the Hornby.

By using the statement LPRINT ": 3 < 5 =", the user would have control over locomotive 5, and the inertia will be set to 3. The user can then use the statement LPRINT "?" to set the direction to forward and the statement LPRINT "C" to start the locomotive at the lowest speed. One must fol-
low all conventions of the Hornby, of course; if one were to follow the previous statements with LPRINT "<  1  >  G", locomotive five would continue as set before, and locomotive one would go to speed 6, in the same direction as loco 5 (forward). Any further speed commands would not affect loco 5, but would control loco 1, just as if we had pressed the keys on the Hornby itself.

SOFTWARE

GENERAL SUBROUTINES -- LOCOMOTIVE CONTROL AND ACCESSORY MESSAGE SENDING

A general interface subroutine section and three example programs which use these subroutines are provided with this report. A program is also provided for the Osborne 1 which is specific to this microcomputer. The general routines make the method of communication with the Hornby transparent to the user; he need only keep track of the appropriate parameters and call the necessary subroutines which take care of all of the ugly work.

NOTE 1: To change the baud rate to 1200 baud, the variable on line number 37 must be set to nine blanks. This should only be necessary if your RS-232 device has no 300 baud setting. You must also alter the hardware to do so, though, by resetting the baud rate generator select lines.

NOTE 2: Some of the code is language dependent. It is all MBASIC, an extremely standard language, but many of te
computers available do not have it (e.g. Apple with its equivalent Applesoft). The statements LPRINT, INSTR, and MID$ might differ from BASIC to BASIC; hence, these routines are meant as examples of how to use the interface, but are not really intended as "end user" equipment. The routines are written in such a manner, though, that one can just adapt them to their computer in MBASIC; for example, one can change the output routine to suit their fancy in EXAMPLE PROGRAM 3 according to the capability of their specific computer.

The first routine is the initialization section. This must be called by stating GOSUB 5 at the beginning of any program which is to use these general routines.

The second of these routines accepts the parameters LOCO (the number of the locomotive to be controlled), SPD (the speed - negative speeds mean reverse, positive mean forward), and INERT (the inertia to be set). The values of LOCO can range from 1 to 16, SPD from -14 to 14, and INERT from 0 to 4 (0 means don't bother to fool with the inertia). To set loco 6 to speed 3, in reverse, the following code would be appropriate:

```
1000 GOSUB 5
1010 LOCO = 6
1020 SPD = -3
1030 INERT = 0
1040 GOSUB 100
```

The subroutine will not send out unnecessary characters, though; it keeps track of the present inertia, direction,
and locomotive. For example, it is not necessary to hit the key sequence 'LOCO' '5' '=' to set the speed of loco 5 with a 'G' if you are already controlling loco 5. Similarly it is not necessary to send out the direction control character if the direction is not to change, even if you are going to control a different locomotive. Also, carriage returns aren't unnecessarily printed out; they are used sparingly, so as to make the interface respond as fast as possible. Of course, the user need not concern himself with this, unless he wishes to write a similar routine for himself.

The third, and last routine is used to send messages to the accessories. The parameters NUM1 and NUM2 are the two numbers to be sent, and ENTR determines whether a left arrow or right arrow is used to enter the numbers (ENTR=0 for left, 1 for right).

EXAMPLE PROGRAM 1

To run this program you must type LOAD "HORNEX1.BAS" and then RUN 1000 (because the program itself begins at 1000--the general subroutines are between 1 and 999). This program prompts the user for the necessary information and uses the general routines to control any one locomotive at a time. The routine also keeps track of the individual locomotive speeds and prints them out after each control command. This is a very simple demonstration of the use of the locomotive control routines.
EXAMPLE PROGRAM 2

Type LOAD "HORNEX2.BAS" and RUN 1000 to use this program. This example demonstrates the method of sending messages to the accessories using the general subroutines.

EXAMPLE PROGRAM 3

Type LOAD "HORNEX3.BAS" and RUN 1000 to run this example. The most complicated of the three examples, this program combines the first two and adds a little in the way of a nicer display. An explanation of the use of this program is really necessary, though. The keyboard functions as a control over locomotives 1 through 6. Loco 1 is controlled by the keys Q, A, and Z, loco 2 with W, S, and X, and so on for the other four locos (uppercase characters only). The upper of the three keys on the keyboard for each loco accelerates the appropriate loco one speed, the middle stops the loco, and the bottom decelerates the loco one speed. For example, if you were to press "RRRZYF", you would accelerate loco 4 to speed 3, set loco 1 to speed 1 in reverse, set loco 6 to speed 1, and then stop loco 4; you must of course wait for the computer to respond or it might miss some of the characters. To send a message, you merely have to type the number keys corresponding to the message you wish to send (e.g. "74") and then either the "-" key or the "=" key for the left or right arrow, respectively. If you make a mistake with the message before you hit the respective arrow key,
you can just keep entering the numbers until you are satisfied. For example, if you were to enter "973474-" in sequence, the message "7 4 <--" would be sent through the Hornby to the accessory. The display also tells you the present speed of all of the locomotives and the present message sent or pending (there will be an appropriate arrow following the message in parenthseses if it has already been sent).

EXAMPLE FOR THE OSBORNE 1

This was written before the general subroutines, hence the somewhat sloppy code. The program can be run by typing RUN "HORN1.BAS." It is very Osborne specific and is not intended for any other system. It operates exactly as Example 3, but the control is extended for nine locomotives ("O", "L", and "." control loco 9, for example -- see EXAMPLE PROGRAM 3 above). The main difference is the elegance of the display, though.

CONCLUSION

At the outset, this project was an attempt to improve the user friendliness of the Hornby Zero-1 model train controller; it quickly developed into a research project dealing with the workings of the Hornby unit -- much time was spent in making a reliable, compact, general, and efficient interface. The interface achieves all of these goals and enables
the ambitious programmer to take full advantage of the Hornby controller. I was pleased to demonstrate to myself, as well as others, the improvements that can be made in the user interface with the Hornby, as well as the ease with which these improvements can be made, even with the simplest of programs written in MBASIC. The interface should be of interest to both computer and model train enthusiasts alike. It draws the computer owner to the world of the model train by enabling him to exercise real time control with the powerful Hornby unit, and it awakens the model railroader to the powerful world of the computer.
REN ********************
REM *** HORNY/RS-232 INTERFACE SUBROUTINES ***
REM ********************
REM Copyright 1983 Phil Dworsky
REM ********************
REM *** INITIALIZATION SUBROUTINE ***
**IM INERTCH$(1);RARRCH$(1);LARRCH$(1);FORWRCH$(1);REVERSCH$(1);LOCOCH$(1)
**IM SPEED$(10)
**INERTCH$="*****
**RARRCH$="*
**LARRCH$="-
**FORWRCH$="*
**REVERSCH$="*
**LOCOCH$="<
SPEED$="ACBFEDLMONJKIH"
REN ***FILL=0 BLANKS FOR 300 BAUD, 9 BLANKS FOR 1200 BAUD ***
FILL$=
FRESLOC = 0
FRESINERT = 0
FRESDIR = 1234
INERT = 0
LOCO = 0
SFD = 0
RETURN
0 REM ********************
0 REM *** SET LOCOMOTIVE (LOCO) TO VELOCITY (SFD), INERTIA (INERT) ***
1 REM ********************
0 IF (LOCO<1) OR (LOCO>16) OR (ABS(SFD)>14) OR (INERT<0) OR (INERT>4) THEN
10 IF (LOCO<INERT) OR ((INERT<0) AND (INERT>0)) THEN GOSUB 200
10 IF (FRESDIR<ABS(SFD)) OR (SFD<0) THEN 100
100 FRESDIR = SIG(SFD)
10 IF SFD>0 THEN 120
120 LPRINT REVERSCH$;"*";FILL$;
130 GOTO 155
130 LPRINT FORWRCH$;"*";FILL$;
130 LPRINT MID$(SPEED$;1;ABS(SFD)+1);"*";FILL$;
130 RETURN
20 REM *** NEW LOCO OR INERTIA... ***
20 IF ((INERT<0) AND (LOCO<0) AND (FRESLOC<0)) THEN LPRINT (F
200 LPRINT LOCOCH$;"*";FILL$;
200 IF LOCO<0 THEN 260
200 LPRINT LOCO;FILL$;
200 GOTO 270
200 LPRINT "1";FILL$;LOCO-10;FILL$;
200 LPRINT "1";FILL$;
200 FRESINERT-INERT
200 FRESLOC-LOC0
200 RETURN
29 REM ********************
00 REM *** SEND A MESSAGE(NUM1, NUM2=0 TO 9) WITH ENTER KEY ***
01 REM *** DETERMINED BY ENTR (O=LEFT ARROW, NOT O=RIGHT) ***
02 REM ********************
00 IF (NUM1<0) OR (NUM1>9) OR (NUM2<0) OR (NUM2>9) THEN RETURN
000 LPRINT NUM1;FILL$;NUM2;"*";FILL$;
000 IF ENTR <> 0 THEN LPRINT RARRCH$;"*";FILL$;
000 IF ENTR = 0 THEN LPRINT RARRCH$;"*";FILL$;
000 RETURN
REM *** EXAMPLE PROGRAM #1 ***
REM *** CONTROL ONE LOCOMOTIVE AT A TIME ***
REM COPYRIGHT 1/83 PHIL DWORSKY
CLS
1000 FOR I = 1 TO 16
1001 SPDTABLE(I) = 0
1002 NEXT I
1003 INPUT "Which locomotive to control (0 to QUIT)"$LOC0
1004 IF LOC0 = 0 THEN 1060
1005 INPUT "What speed (-14 to 14)"$SPD
1006 INPUT "What inertia (0 = same, 1 thru 4 to set)"$INERT
1007 GOSUB 100
1008 SPDTABLE(LOC0) = SPD
1009 GOSUB 1090
1010 GOTO 1010
1011 REM *** QUIT ***
1012 PRINT "Done."
1013 END
1014 REM *** PRINT OUT RESPECTIVE SPEEDS ***
1015 FOR I = 1 TO 16
1016 PRINT SPDTABLE(I)$;
1017 NEXT I
1018 PRINT
1019 RETURN

REM *** EXAMPLE PROGRAM #2 ***
REM *** ACCESSORY CONTROL ***
REM COPYRIGHT 1/83 PHIL DWORSKY
CLS
1020 PRINT "Input the two digits separated by a"$L
1021 INPUT "comma (0,0 to quit)"$NUM1$NUM2
1022 IF (NUM1 = 0) AND (NUM2 = 0) THEN 1030
1023 IF (NUM1 = 0) OR (NUM2 = 0) THEN 1030
1024 PRINT "Left arrow (0) or right (1)"$ENTR
1025 GOSUB 400
1026 PRINT "Message: "$NUM1$"$NUM2$"$
1027 IF ENTR = 0 THEN PRINT "--> Sent."
1028 IF ENTR = 0 THEN PRINT "<-- Sent."
1029 GOTO 1010
1030 REM *** QUIT ***
1031 PRINT "Done."
1032 END
1000 REM ***********************************************
1001 REM *** EXAMPLE PROGRAM #3
1002 REM *** CONTROL LOC0'S 1 THRU 6 AND SEND MESSAGES ***
1003 REM ***********************************************
1004 REM COPYRIGHT 1/63 PHIL DWORSKY
1005 BUBS: 0
1010 DIM SPITABLE(6)
1015 FOR I = 1 TO 6
1020 SPITABLE(I) = 0
1025 NEXT I
1030 TRUE = -1
1035 FALSE = 0
1040 SENT = FALSE
1045 Kbfl = "GAWSXEDCRFVTBHYN"
1050 NERS = 0
1055 DIM IN$(1)
1060 KBLARR$ = "="
1065 KBRARR$ = "="
1100 REM OUTPUT STATUS TO SCREEN
1110 GOSUB 2000
1120 REM SET CHARACTER (IN$) FROM TERMINAL
1125 GOSUB 2100
1130 REM DETERMINE IF SPEED OR MESSAGE CHARACTER
1135 IF (VAL(IN$) = 0) AND (IN$ = "0") THEN 1210
1140 REM A NUMBER-MESSAGE ONLY LAST TWO ENTERED
1145 IF SENT THEN NERS = 0
1150 NERS = 10 * (NERS - 10 * (INT(NERS / 10))) + VAL(IN$)
1155 SENT = FALSE
1160 GOTO 1100
1165 IF (IN$ = KBLARR$) OR (IN$ = KBRARR$) THEN 1500
1170 REM SPEED CONTROL CHARACTER ENTERED
1175 REM FIGURE OUT WHICH LOC0, WHICH FUNCTION (ACCEL, DECEL, STOP)
1180 KEYS= IN$(IN$)
1185 IF KEYS = 0 THEN 1100
1190 LOC0 = 1 + INT((KEYS - 1) / 3)
1200 IF LOC0 = 0 THEN 1100
1205 FUNC = 1 + ((KEYS - 1) MOD 3)
1210 ON FUNC GOTO 1220, 1310, 1340
1230 REM INCREASE SPEED OF LOC0
1235 SPD = SPITABLE(LOC0) + 1
1250 GOTO 1500
1260 REM STOP LOC0
1270 SPD = 0
1280 GOTO 1500
1290 REM DECREASE SPEED OF LOC0
1300 SPD = SPITABLE(LOC0) - 1
1310 REM REFLECT IN SPITABLE AND SEND TO HORNBY
1315 IF SPD = 14 THEN SPD = -14
1320 IF SPD > 14 THEN SPD = 14
1330 SPITABLE(LOC0) = SPD
1400 GOSUB 100
1410 GOTO 1100
1500 REM ** SEND COMPLETE MESSAGE **
1510 SENT = TRUE
1520 NUM1 = INT(NERS / 10)
1530 NUM2 = NERS - 10 * NUM1
1540 IF IN$ = KBLARR$ THEN ENTR = 0
1550 IF IN$ = KBRARR$ THEN ENTR = 1
1560 GOSUB 400
1570 GOTO 1100
1580 REM *** OUTPUT STATUS AND MESSAGE TO SCREEN ***
2010 FOR I = 1 TO 8
2020 PRINT SPDTABLE(I);
2025 IF ABS(SPDTABLE(I))<10 THEN PRINT " ";
2030 NEXT I
2040 DIG1 = INT(MESS/10);
2045 DIG2 = MESS-10*DIG1
2050 PRINT "({}*DIG1*DIG2"
2060 IF NOT SENT THEN 2090
2070 IF ENTR = 0 THEN PRINT "<-- ")
2080 IF ENTR <> 0 THEN PRINT "--> ")
2085 RETURN
2090 PRINT " ");
2095 RETURN
2100 REM *** INPUT A CHARACTER (IN$) SUBROUTINE
2105 REM ADAPT THIS ROUTINE TO SUIT YOUR PREFERENCE
2110 IN$ = INKEY$
2120 IF LEN(IN$) = 0 THEN 2110
2130 RETURN
10 REM FAUL DWORSKY 12/14/82 EEDS 497
20 REM HOMEBY CONTROL PROGRAM
30 REM CONTROLS NINE TRAINS AT ONCE
40 REM THE TRAINS THAT ARE CONTROLLED ARE IN THE
50 REM ARRAY CALLED TRAINS...
60 REM USE THE VERTICAL TRIPLETs
70 REM OF CHARACTERS TO CONTROL EACH TRAIN
80 REM E.G. Q TO RAISE THE SPEED OF TRAIN 1
90 REM Z TO LOWER IT; AND A TO STOP IT.
100 REM
110 REM *** CURSOR CONTROL CHARACTERS ***
120 REM
130 LHR#=CHR$(8)
140 RTR#=CHR$(12)
150 UPR#=CHR$(11)
160 DLR#=CHR$(10)
170 LHR#=CHR$(26)
180 RTR#=CHR$(27)+""
190 UPR#=CHR$(27)+""
200 REM *** DECLARATIONS ***
210 FOR XX=1 TO 9: TRNSM$(XX)="" NEXT XX
220 LARR#="": IARR#="": KARR#="":
230 KK=":"
240 MD#="" MD#="" MD#="":
250 KE#="" KE#="" KE#="":
260 MB#="" MB#="" MB#="":
270 REVERSE#="":
280 MAXPFX-14
290 TRUE-1
300 FALSE-0
310 PRINT "LITTMAN/DWORSKY Multi-Train Control"
320 PRINT PRINT PRINT PRINT
330 PRINT "TRAIN: 1 2 3 4 5 6 7 8 9"
340 PRINT "" PRINT PRINT PRINT PRINT PRINT PRINT PRINT PRINT
350 PRINT PRINT PRINT PRINT PRINT PRINT PRINT PRINT PRINT
360 REM *** SET UP THE GRAPHICS ON THE SCREEN ***
370 REM PRINT HOME
380 PRINT PRINT PRINT PRINT PRINT PRINT PRINT PRINT PRINT
390 PRINT "LITTMAN/DWORSKY Multi-Train Control"
400 PRINT PRINT PRINT PRINT
410 PRINT "TRAIN: 1 2 3 4 5 6 7 8 9"
420 PRINT "" PRINT PRINT PRINT PRINT PRINT PRINT PRINT PRINT PRINT
430 PRINT PRINT PRINT PRINT PRINT PRINT PRINT PRINT PRINT
440 REM *** GET TERMINAL INPUT ***
450 PRINT "Speed: "
460 PRINT DLR#
470 FOR XX=1 TO 9: PRINT TRNSM$(XX); "" NEXT XX
480 PRINT PRINT PRINT PRINT PRINT PRINT PRINT PRINT PRINT
490 REM *** A NUMBER HAS BEEN INPUT ***
500 REM
510 IF LKEYS=0 THEN 2010
520 IF (IN$="0") AND IN$="0" THEN 2100
530 REM
540 IF SENT THEN 2040
550 MSG#=MSG#+IN$+""
560 IF LKEYS=0 THEN MORE=MORE+1 MORE=MORE+1
REM *** A KEYBOARD CHARACTER--TRAIN CONTROL ***
KEYNUM=INSTR(KBD$)$IN$)
IF KEYNUM = 0 THEN 2000
IND$=INT((KEYNUM-1)/3)
IF IND$=0 THEN 2000
TRINPUT$=TRANS$(IND$)
IF TRINPUT$ = PRESTR$ THEN 2200
REM
REM *** A NEW TRAIN TO CONTROL ***
PRESTR$=TRINPUT$;
LPRINT "LOCOS";PRESTR$;" ";LARR$;" ";
LPRINT MIDS$(SPEED$+1)+ABS(TRANS$(PRESTR$));1);"
IF TRSFON$<PRESTR$>0 THEN LPRINT FORWARDS
IF TRSFON$(PRESTR$)<0 THEN LPRINT REVERSE
REM
REM *** PROCESS THE CHARACTER ***
IF FUNC$=2210<2400<2300
REM *** INCREASE SPEED OF PRESTR$ ***
NEWSPD$=TRANS$(PRESTR$)+1
IF NEWSPD$=MAXISP$ THEN 2000
TRANS$(PRESTR$)=NEWSPD$
IF NEWSPD$<1 THEN LPRINT FORWARDS
LPRINT MIDS$(SPEED$+1)+ABS(NEWSPD$);1);"
SOTO 2000
REM
REM *** DECREASE SPEED OF PRESTR$ ***
NEWSPD$=TRANS$(PRESTR$)-1
IF ABS(NEWSPD$)<MAXISP$ THEN 2000
TRANS$(PRESTR$)=NEWSPD$
IF NEWSPD$<1 THEN LPRINT REVERSE
LPRINT MIDS$(SPEED$+1)+ABS(NEWSPD$);1);"
SOTO 2000
REM
REM *** STOP PRESTR$ ***
LPRINT MIDS$(SPEED$+1);"
SOTO 2000
REM
REM *** SEND COMPLETE MESSAGE TO ACCESSORY ***
IF LEN(MSG$)<6 THEN 2000
SENT$=TRUE
LPRINT MSG$;
IF IN$=KBDARR$ THEN LPRINT LARR$;
IF IN$=KBDARR$ THEN LPRINT RARR$;
FOR XX=1 TO 6 ;PRINT DOWN$;NEXT XX
PRINT " Message: ";MSG$;
IF IN$=KBDARR$ THEN PRINT " <--
IF IN$=KBDARR$ THEN PRINT " -->
FOR XX=1 TO 7 ;PRINT UPS$;NEXT XX
SOTO 2000
PRINT HOME;
PRINT " Program exited..."
PRINT " Control relinquished to basic monitor."
Panel "Key-Push" Clock Generation Circuit
Speed Control Circuit

Manual/Interface Operation
Signal Generator Circuit
(Hornby: Manual)
Communication Interface (300 Baud)

Connector 2, pin 16
8k

System Ground

U12
1489
6WD, Vcc
+5

Connector 2, pin 13

+15

Data Available

U10 pin 3

Circuit Diagram

AY-3-1015D UART

Note: For 110 Baud, RSB=+15, RSA=6WD, use F18 (pin 13) for the output pin.

RS-232 Interface to HORNBY ZERO-1 Model Train Controller

Walter P. A. and Andrew D. Shanks
Connector 1 - to Keyboard

Connector 2 - to Slide Switch Solder points and +15v all on Hornby PC Board, RS-232 Input, and External Switch Connections.
<table>
<thead>
<tr>
<th>QTY</th>
<th>PART</th>
<th>NUMBER(S)</th>
<th>+15 CONNECTIONS</th>
<th>GND CONNECTIONS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4076</td>
<td>U1, U16</td>
<td>pin 16</td>
<td>pin 8</td>
<td>CMOS QUAD LATCH</td>
</tr>
<tr>
<td>5</td>
<td>4066</td>
<td>U2, U4, U5, U6, U7</td>
<td>14</td>
<td>7</td>
<td>CMOS QUAD BILATERAL SWITCH</td>
</tr>
<tr>
<td>1</td>
<td>4514</td>
<td>U3</td>
<td>24</td>
<td>12</td>
<td>CMOS 1016 NUX, OUTPUTS NORMALLY LOW</td>
</tr>
<tr>
<td>2</td>
<td>4011</td>
<td>U8, U15</td>
<td>14</td>
<td>7</td>
<td>CMOS QUAD 2-INPUT NAND</td>
</tr>
<tr>
<td>1</td>
<td>4049</td>
<td>U9</td>
<td>1</td>
<td>8</td>
<td>CMOS HEX INVERTER</td>
</tr>
<tr>
<td>1</td>
<td>4013</td>
<td>U10</td>
<td>14</td>
<td>7</td>
<td>CMOS DUAL DFF</td>
</tr>
<tr>
<td>1</td>
<td>4023</td>
<td>U11</td>
<td>14</td>
<td>7</td>
<td>CMOS TRIPLE 3-INPUT NAND</td>
</tr>
<tr>
<td>1</td>
<td>1489</td>
<td>U12</td>
<td>14 (+5)</td>
<td>7</td>
<td>QUAD HDBT LINE RECEIVER</td>
</tr>
<tr>
<td>1</td>
<td>AY-3-1015D</td>
<td>U13</td>
<td>1</td>
<td>3</td>
<td>UART</td>
</tr>
<tr>
<td>1</td>
<td>HC4411</td>
<td>U4</td>
<td>24 (+5)</td>
<td>12</td>
<td>BIT RATE GENERATOR</td>
</tr>
<tr>
<td>1</td>
<td>78L05</td>
<td>Z1</td>
<td>NA</td>
<td>NA</td>
<td>+5V LOW POWER VOLTAGE REGULATOR</td>
</tr>
<tr>
<td>1</td>
<td>7815</td>
<td>Z2</td>
<td>NA</td>
<td>NA</td>
<td>+15V VOLTAGE REGULATOR</td>
</tr>
<tr>
<td>1</td>
<td>81F212 (Series)</td>
<td>S1</td>
<td>NA</td>
<td>NA</td>
<td>8PDT ALTERNATING ACTION CENTRALAB SWITCH</td>
</tr>
</tbody>
</table>

Notes: 2 +15, gnd, except where marked otherwise.

G# 008 Rev A Parts List
RS-232 Interface to HORNBY ZERO-1
**Connector 1** - to Keyboard

<table>
<thead>
<tr>
<th>Pin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SW (B)</td>
</tr>
<tr>
<td>2</td>
<td>NC</td>
</tr>
<tr>
<td>3</td>
<td>R9</td>
</tr>
<tr>
<td>4</td>
<td>R6</td>
</tr>
<tr>
<td>5</td>
<td>R8</td>
</tr>
<tr>
<td>6</td>
<td>KE (M)</td>
</tr>
<tr>
<td>7</td>
<td>NC</td>
</tr>
<tr>
<td>8</td>
<td>NC</td>
</tr>
<tr>
<td>9</td>
<td>NC</td>
</tr>
<tr>
<td>10</td>
<td>CP (System Ground)</td>
</tr>
<tr>
<td>11</td>
<td>K1 (n)</td>
</tr>
<tr>
<td>12</td>
<td>K2 (n)</td>
</tr>
<tr>
<td>13</td>
<td>K3 (n)</td>
</tr>
<tr>
<td>14</td>
<td>K4 (n)</td>
</tr>
</tbody>
</table>

**Connector 2** - to Slide Switch, solder points and +15V all on Hornby PC Board, RS-232 input, and external switch connections.

<table>
<thead>
<tr>
<th>Pin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GND (Pin 7)</td>
</tr>
<tr>
<td>2</td>
<td>RS-232 pin 7</td>
</tr>
<tr>
<td>3</td>
<td>+15V Pin 1</td>
</tr>
<tr>
<td>4</td>
<td>NC</td>
</tr>
<tr>
<td>5</td>
<td>Manual</td>
</tr>
<tr>
<td>6</td>
<td>NC</td>
</tr>
<tr>
<td>7</td>
<td>NC</td>
</tr>
<tr>
<td>8</td>
<td>NC</td>
</tr>
<tr>
<td>9</td>
<td>NC</td>
</tr>
<tr>
<td>10</td>
<td>NC</td>
</tr>
<tr>
<td>11</td>
<td>NC</td>
</tr>
<tr>
<td>12</td>
<td>RS-232 pin 3 (Serial Input)</td>
</tr>
<tr>
<td>13</td>
<td>K1 (s)</td>
</tr>
<tr>
<td>14</td>
<td>K2 (s)</td>
</tr>
<tr>
<td>15</td>
<td>K3 (s)</td>
</tr>
<tr>
<td>16</td>
<td>K4 (s)</td>
</tr>
<tr>
<td>17</td>
<td>GND out</td>
</tr>
<tr>
<td>18</td>
<td>-15V out</td>
</tr>
</tbody>
</table>

**RS-232 Interface to Hornby 2nd Model Train Controller**
component side — circuit prototype sent to Hornby / TrainTronics for evaluation

20 Jan 93

MGL
Component side -- circuit prototype sent to Hornby / TrainTronics for evaluation

20 Jan 83

MGL