

## HAL ID-1A REPEATER IDENTIFIER

### IV. PROGRAMMING THE MEMORY

The HAL ID-1A read-only memory (ROM), which stores the identification message to be transmitted, is of the diode matrix type. It is designed so that a minimum number of diodes are required to store a given message.

Although the HAL ID-1A is normally supplied with the desired message precoded into the memory, the user can change the stored message by altering the arrangement of the diodes in the matrices. The process, a relatively simple one, will be explained in the following paragraphs.

#### **Memory Format**

The memory is capable of storing forty code "characters" (dots, dashes, or spaces). Each character is composed of two logic "bits". The memory is made up of two separate matrices, one called the "dash" matrix and the other the "space" matrix. Each stores one of the two bits that make up a given character.

As the message is transmitted, the contents of the memory are read out sequentially, one character (two bits) at a time. The output signals are fed to the keyer circuitry, where they cause the keyer to reproduce the stored character.

Since each character consists of two bits, anyone of four possible combinations (and therefore any of four different characters) may be stored at each memory location:

**TABLE 1: Character Codes**

Character	Space Matrix	Dash Matrix	
Dot	0	0	(1 indicates the presence of a diode 0 indicates the absence of a diode)
Dash	0	1	
Space	1	0	
End	1	1	

Note that the keyer automatically produces a short space, equal to the length of a dot, after each dot or dash. No separate code character is needed to generate this intercharacter space. The "space" character therefore is used only to generate the longer spaces required between complete letters or between words. It is equal in length in length to two dots. The "end" character is always used as the last character in the coded message; it tells the keyer circuitry that the message is completed.

#### **Coding the Memory**

To code the memory, all you need to do is connect diodes at the appropriate spaces in the matrices. For example, if the first character to be transmitted is a dash, a diode is connected at location 1 in the dash matrix, but location 1 of the space matrix is left vacant.

Figure 2 shows the physical arrangement of the matrix circuit board, viewed from the top, with the memory coded to transmit the message "DE WB9XYZ". To illustrate how the coding is done, we will follow the process used to store this message.

The first step is to break the message down into a series of dots, dashes and spaces, as shown in Table 2 in the following page. The dash and space matrix columns are then filled in by using the code patterns given in Table 1. One space character is used between each letter or number in the message; three are used to form the longer spaces between words. **An end character must be included as the last character in the message.**

**TABLE 2: Coding the Memory for “DE WB9XYZ”**

Memory Location	Letter	Character	Space Matrix	Dash Matrix
1		Space	1	0
2		Space	1	0
3	D	Dash	0	1
4		Dot	0	0
5		Dot	0	0
6		Space	1	0
7	E	Dot	0	0
8		Space	1	0
9		Space	1	0
10		Space	1	0
11	W	Dot	0	0
12		Dash	0	1
13		Dash	0	1
14		Space	1	0
15	B	Dash	0	1
16		Dot	0	0
17		Dot	0	0
18		Dot	0	0
19		Space	1	0
20	9	Dash	0	1
21		Dash	0	1
22		Dash	0	1
23		Dash	0	1
24		Dot	0	0
25		Space	1	0
26	X	Dash	0	1
27		Dot	0	0
28		Dot	0	0
29		Dash	0	1
30		Space	1	0
31	Y	Dash	0	1
32		Dot	0	0
33		Dash	0	1
34		Dash	0	1
35		Space	1	0
36	Z	Dash	0	1
37		Dash	0	1
38		Dot	0	0
39		Dot	0	0
40		End	1	1

If the message to be coded (including the end character) does not require the use of all 40 memory locations, it is well to insert extra space characters at the beginning of the message. These will delay the beginning of the message, allowing time for control relays to activate the transmitter before the first character is sent. In our example, the message plus the end character require only 38 memory locations, so we insert two space characters ahead of the first message character.

Once the table has been constructed, we may use the coding chart in Figure 2 to mark the locations in the matrix where diodes must be inserted. In the drawing, the diode locations in both the dash and the space matrices are numbered from one to 40. Starting with the space-matrix column in the table, we mark a diode at locations one and two of the space matrix, since the space-matrix column contains ones for these locations. We leave locations three through five empty, however, since the space-matrix column contains zeroes. The process is continued for all forty locations in the space matrix. The diodes in the dash matrix are then marked in the same way, using the contents in the dash-matrix column in the table.

Figure 3 provides a spare coding chart to use in programming your own message. Once the table of characters has been constructed and the chart properly marked, diodes are inserted in the board at the required locations and are soldered. Be sure that the diode cathodes connect to the printed conductors on the bottom of the circuit board. It pays to carefully recheck your work at each stage of the process to guard against wiring errors.

When soldering the diodes in place, use a low-voltage, pencil-tip soldering iron. Use only enough heat to obtain a good connection. Overheating the joints may damage the circuit board or the diodes. Use a limited amount of solder and check to make sure that excess solder does not bridge across adjacent conductors.

### **Changing the Stored Message**

If you are *recoding* a memory to change the stored message, it often saves time to construct a table for both the old and new messages. By comparing the tables you may find it is not necessary to remove all of the existing diodes—some may already be in the correct position for the new message.

Remove unneeded diodes from the board by snipping the diode leads with a pair of wire cutters. It is safer to snip the diodes out than to risk damage to the circuit board by trying to unsolder them. Carefully remove the leads from the holes with a soldering iron and needle-nosed pliers. Clean each hole to ease the reassembly process.

Germanium signal diodes such as the 1N270 make suitable replacement diodes. To save the trouble of locating and replacing defective diodes after the board has been completed and tested, check each diode (whether new or used) with an ohmmeter before installing it.

**UPDATE:** The **Central Semiconductor CDSH270** is a Silicon Schottky diode specifically designed to replace Germanium diodes manufactured with 1950s technology like the 1N270, 1N277, etc. with 1990s technology. Advantages of this new technology are lower forward voltage, lower leakage, faster switching speed, and a more robust package.

*Mouser part # 610-CDSH270 was a reliable source for this diode in 2008 at about 25 cents each.*

### **Factory Recoding**

HAL Communications Corp. will recode your identifier's memory for a \$5.00 service charge plus shipping costs. Please notify the factory before returning your keyer for recoding.

### **Manual**

A full manual (18 pages) is available from HAL Communications, Box 365, Urbana, IL 61801 for \$25.

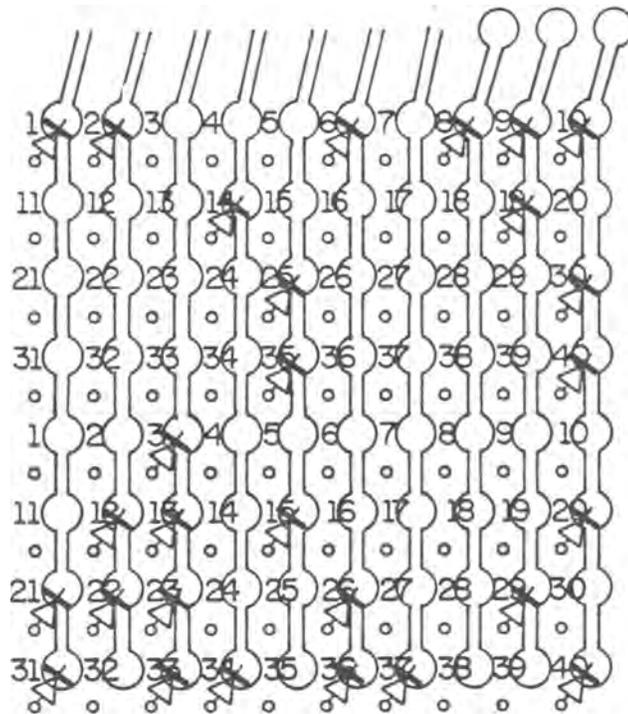


Figure 2: Coding Chart for DE WB9XYZ

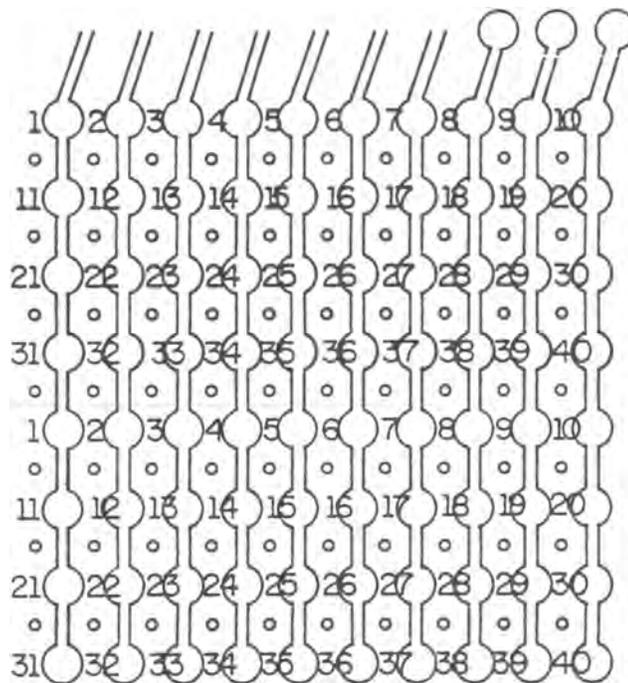
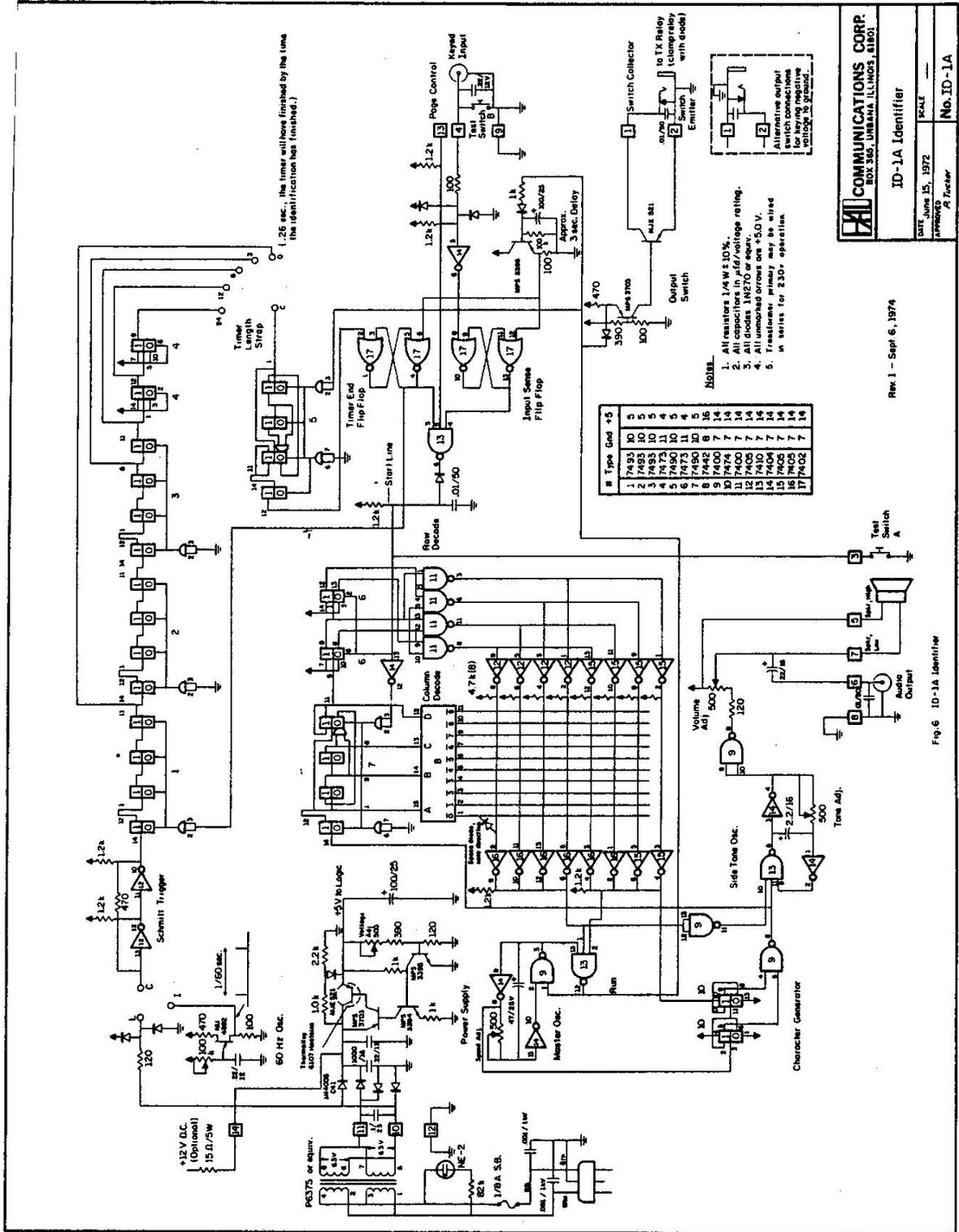


Figure 3: Blank Coding Chart



**HAL COMMUNICATIONS CORP.**  
 BOX 365, URBANA, ILLINOIS, 61801.

**ID-1A Identifier**

DATE: June 15, 1972  
 SCALE: —  
 APPROVED: A. Tricker

**No. ID-1A**

Rev. 1 - Sept 6, 1974