

# COMPUTERS

## MICRO-HELL

From Electron, July 1980

By K. H. J. Roberts, PA0KLS, VALKENWAARD

Translated and adapted by [Andrew Lovell](#) G6BZS, SM6MOJ

Click [here](#) for the initial page on HELL schreibers

### Introduction

The HELL printer was invented by Doktoringenieur Rudolf Hell in Germany in about 1930. It was first produced commercially in 1932 and both civil (6) and military versions were produced. The military version was described by S.A. Cook G5XB in Rad Com (3).

It is pleasing to note that the Hell system is starting to attract attention again, having fallen into disuse after the war. It has now become possible to imitate the old mechanical system electronically thanks to the availability of microelectronics and microcomputers.

One of the best examples of this is when HELL printing is imitated by a computer which has been programmed for this purpose. It is not the ideal way, but easiest to construct. One can then emulate a Hell printer by loading the program from tape or disk.

Following a note about such a program in a REFLKTIES by PA0SE (1), there was a considerable amount of comment by micro owners. This article will attempt to explain in general how things work since each micro has its own features and limitations. The HELL program for the Apple II is based on the HIRES (high resolution) graphics mode. This has a raster of 280 (=40x7) dots horizontally and 192 dots (=24x8) dots vertically, in which each dot can be individually switched from black to white. We will see that such fine graphics are needed to emulate a HELL printer.

### The HELL Feldfernschreiber (Army Hell printer)

Hell printing is a sort of facsimile or picture printing transmission system. The letters are made up of vertical lines which are drawn on paper tape by means of

a hammer tapping on the inky threads of a quick threaded screw. The lines are arranged from left to right by moving the paper tape, which is transported to the left.

The pictures of the letters are transmitted in the form of on-off modulation of a tone or carrier, which is converted in the receiving apparatus to recognizable figures or letters. Interference can distort the signal, but it is printed on the tape so that the recipient can use the vastly superior pattern recognition powers of his eye to work out what the message was. The receiving apparatus does not have to make a decision in real time, as is the case with a Murray coded teleprinter.

Receiving and transmitting are a-synchronous. The received text is printed twice on the paper, one above the other, which is necessary if the text is always to be legible. If the transmitter and receiver do not run at exactly the same speed, the text is printed at an angle on the paper. It climbs up or down, but is always legible thanks to the double row of letters. If the receiver speed is adjusted, one can get the text neatly arranged on the tape, but this is not necessary for legibility. Further information can be found in articles by Hans Everg DJ0SA (2) and others (3-7).

## Micro-Hell Receiver

Micro-Hell works the same way in principle as on the original Hell of 28 dots in height. The Apple II has six strips vertically, five could be arranged on a Spectrum. A black line divides off each strip. Diagram 1 shows what an Apple II screen looks like. On each strip, which you should think of as following each other in a line, the computer puts in black or white dots from the bottom of the letter and upwards, in accordance with the incoming signal. Two dots are updated simultaneously, one on the lower half of the strip and the other on the upper half, exactly 14 dots higher. Once the two sets of 14 dots are in place, the computer starts to update the next vertical line to the right (Figure 2). When the end of the first strip is reached, we go to the beginning of the next strip and start again.

If we ensure that each dot is updated at intervals of exactly 4.08 milliseconds the HELL letters will be neatly regenerated. It is not difficult to calculate this timing if you know that a HELL character consists of 7 picture columns and 14 picture elements per line and that 2.5 characters per second are transmitted on the military version. By putting a feature into the program which permits the updating timing to be varied it is possible to receive text from any machine and get the text neatly on one line, as on the original equipment, even if the speed of the transmitting HELL printer is not correct.

## RECEIVING PROGRAM

Since exact timing of the program is necessary, and the time base is the program loop time, it is only possible to program in Machine Code or Assembler. It is unfortunately not possible to program in BASIC since the computer has to have good graphics which are accessible from the machine code. The task of writing such a program is not something which the novice programmer should attempt

## MAIN PROGRAM

1. Check whether there is a tone (1) or not (0)
2. Plot a dot in the lower half (Rout)
3. Plot a dot in the upper half (Rout)
4. Change the waiting loop time if required
5. Wait in the waiting loop
6. Go back to 1

These are two plot routines, one for the lower and one for the upper half of the strip.

PLOT DOT (Subroutine)

1. Calculate the address of the next dot (next line, next strip?)
2. Plot the dot white for (0) or black (1)
3. Ready

Some notes:

1. Make sure that each branch of the program uses the same number of program cycles irrespective of which branch is chosen. This can be difficult to reconcile with "smart" programming.
2. The plot point on the upper half of the screen must be one picture line later than the equivalent plot point on the lower half in order to make the letters line up neatly. The updating point must be able to move from one side to the other in the middle of the strip.
3. In my program, it starts to update the new strip before the old one is completely finished to avoid cutting letters in half. This is done by having two extra plotting points updated on the new strip before the old one is finished.

4. Additionally. I have put in a black cursor which moves in front of the printed text. so that one can see at once where the new text is being printed. This means another doubling of the number of plot points.
5. It is probably possible to plot routine which works with different addresses.
6. Changes to the wait loop can be made by pressing one of the keys. This key lengthens or shortens the wait time in small increments.

## **TRANSMIT PROGRAM**

It is fairly easy to program a computer to send HELL characters, basically the same routine as for a morse code generator but with different timing and character set. The heart of the sending program is a look-up table. HELL characters have a different appearance to the computer's own character set, so a conversion table has been put into the memory, comparable to the encoder drum of the original apparatus. The contents of the table is shown in diagram (3) in hexadecimal form. Two immediately adjacent bytes combine to form one picture line. The bits must be read with the highest value (MSB) first, which gives a picture line composed of 14 bits. The two lowest bits of the second byte are not used. A HELL character is composed of 7 picture lines, of which 5 are used for the actual letter and 2 for the space between the letters. For the Army Hell Printer, columns 2 to 15 are used. Columns 2 to 11 contain character information, and columns 12 to 15 contain the space between the letters.

All the characters are arranged in the ASCII sequence. The letter E on the sixth row now consists of 3ff0, 3330. 3330. 3030. 3030. When converted to ones and zeroes, this becomes 00111111111100, 00110011001100, 00110011001100, 00110000001100, 00110000001100.

There is a start pulse in columns 0 and 1. These are used in the GL stop-start system. This system only uses columns 0 to 13. By this means. the same character generator can be used for two different HELL systems.

## **TEXT BUFFER**

When text is transmitted with an Army HELL printer, the keys must be pressed in at fixed intervals determined by the rotation speed of the encoder drum. This takes a bit of practice to get used to. However in a microcomputer we can put in a text buffer. Then any operator can type text in the text buffer at his (her) own speed whilst the sending program transmits the text in the text buffer with perfect regularity. It is even easier for the operator if the contents of the text buffer can be shown on the monitor screen (split screen operation).

### **TEXT BUFFER PROGRAM CONSTRUCTION**

1. Look to see if there is a character in the text buffer. If not, use a space. If yes. take a letter out of the buffer.
2. Take the first byte from the conversion table.
3. Send out 8 bits with the MSB first.
4. Take the next byte out of the conversion table.
5. Send out 6 bits with the MSB first.
6. Look to see if any key has been pressed.
7. Take the next byte out of the conversion table.
8. Go six times to 3.
9. Go to 1.

Some notes;

1. Make sure that there is an interval of 4.08 milliseconds between sending out each bit, this can be done by a waiting loop just in front of the sending out instruction, controlled by the preceeding program. One could also use a programmable counter timer (not in the APPLE) which produces an interrupt every 4.08 milliseconds. For the GL system this time should be 1.66 milliseconds.
2. It is very convenient to have a FIFO text buffer. Alength of 128 or 256 is more than sufficient.
3. It is possible to correct typing errors by using the left arrow (<) if the character has not been sent. This does not of course work if the character has been sent.
4. Make sure that the sending routine can not go further than the start of the text buffer, not even if the timing is critical and the left arrow (<) is used.

5. It is a nice touch if the sending program automatically switches over to receive if the text buffer empty.

## **THE ORIGINAL APPLE PROGRAM**

The program was developed for use with an APPLE-II or APPLE plus computer. It is written in machine code but can be fed in as Applesoft. However an Applesoft ROM or language board must then be installed in the apparatus. The quantity of memory needed is 16k RAM. Any seriously interested person is welcome to contact the translator.

## **OTHER COMPUTERS**

It is possible that no other computers have yet been programmed to emulate a HELL printer. Perhaps After reading this article there will be a few people who want to have a go at it. The hi-res graphics on the APPLE are not strictly necessary. The only difference is that the text would be larger. A Spectrum would give five strips, which is perfectly sufficient.

One could then try to do some sort of bulletin board, which the HELL printer is very suitable for. Hell printing is a very pleasant branch of our hobby, which has many unexplored possibilities.

If you succeed with a program, please let the author or translator know so that others may enjoy the fruits of your labours.

## **HELL TRANSMISSIONS**

It sounds not unlike an East European military station which often intrudes on 40 metres, but identifications are normally given in CW to advise other Amateurs that these are in fact Amateur stations on the air.

There is a regular weekly European Hell net.

Contests are arranged every year, with both short wave and VHF sections.

## References

1. REFLEKIES 1979, Technical reflections over ideas and new designs in Elektron, the Journal of the Dutch Radio Society (VERON)
  2. Electron June 1977, Hans Evers DJ0SA
  3. Hellscreiber, what it is and how it works. Radio Communication April 1981, S.A.G. Cook G5XB
  4. Hellskrift, Historisk Återblick, QTC 9:1980, Karl-Gustav Strid SM6FJB
  5. Faltofjarrskrivare T.typ 58, System Siemens-Hell, SARTG News No 36, Karl-Gustav Strid SM6FJB
  6. Hell op de twee-meter band in alle modes. Electron, H.G. Dikker PE0HGD
  7. [The Hellschreiber, a rediscovery](#), Ham Radio Dec 1979, Hans Evers (DJ0SA)
- 

This article has been prepared for the net by [LA9IHA](#).