

A simple IBM I/O Selectric typewriter controller

JULIO ORTEGA and WILLIAM L. PALYA
Jacksonville State University, Jacksonville, Alabama 36265

A simple inexpensive IBM I/O Selectric typewriter controller is described. The board plugs into an RS-232 port and enables letter-quality output. Computers with editing capabilities can therefore be used as word processors for as little as \$35, about 8 h of labor, and the cost of the typewriter. The controller is a firmware-governed microprocessor, and as a result, it is easily tailored for special applications. Hardware and software are discussed.

A word processor can substantially reduce the time and effort involved in producing a final manuscript. If entry is made through a CRT terminal, most typing errors can be eliminated by simply backing up. Material can be added, deleted, or moved with a few simple commands rather than retyping the entire paper. Spelling correction, proper hyphenization, and even the justification of both margins can be done by the word processor. If a full-function commercial word processor cannot be obtained, a computer that runs an editor or a word processing program like DEC's Runoff can be used as a reasonable substitute.

Unfortunately, most computer printers are optimized for speed or cost and do not produce highly readable copy with a pleasing appearance. "Letter-quality" printers are available but are frequently too expensive to justify for a small laboratory. If a letter-quality printer cannot be obtained, an acceptable, albeit slow, substitute can be assembled from an IBM I/O Selectric typewriter. These are solenoid-actuated Selectric typewriters and can be obtained for about \$100 from your state agency for federal surplus property. An I/O Selectric is the basic typing unit in several printers and can be easily identified by the typing ball and the solenoids mounted under the typewriter mechanism. The presence of a keyboard or input part of the unit is irrelevant and the absence of an ancillary control unit should be viewed as a stroke of good luck.

We designed a small 11-chip circuit board that drives an I/O Selectric as a simple RS-232 terminal. Letter-quality printing can then be obtained through a standard serial port of virtually any computer. Our approach to interfacing the Selectric to the computer was simple. We did not attempt to provide keyboard input from the Selectric, in that we could not envision a case in which CRT terminal input would not be preferable. We did not attempt to provide feedback from the Selectric to

control the timing of the solenoid actuation, in that the difference in speed did not seem to warrant the additional effort.

COMMERCIAL HARDWARE

I/O Selectric printers come in a wide variety of configurations. Most have the character solenoids underneath the left rear, the function solenoids underneath the front right, and the shift/unshift solenoids on the lower right side. A variety of switches are located in various places and sense the shift status, completion of printing, and other states. The first task is to strip all the wiring, with the exception of the wires between the solenoids and their adjacent terminal strips and the electric motor wires up to the first wire nut. We cut all the wires at the switches and pulled the taper pins from the interconnecting taper pin blocks. We then removed the unused electrical and mechanical components, leaving the switches. One side (what will become the plus side) of all the solenoids should then be bussed together. A wire from each solenoid operate position and the common should then be cabled to wherever the control board is to be mounted. A line cord must also be added, with a conveniently mounted fuse and toggle switch. The Selectric can be stripped and prepared in about 3 h.

The Selectric types various characters by rotating a print ball to 1 of 22 horizontal positions and tilting it to one of four rows. The rotation and tilt to a particular character are accomplished by the activation of an appropriate pattern of solenoids. A single- (T1) and double- (T2) tilt solenoid provide for the four tilt positions. For Tilt 0, both are off, and for Tilt 4 both are on. Rotation is somewhat more complex. The upper-/lowercase function solenoid rotates the ball 180 deg, or 11 positions. The character solenoids provide for 1 (R1), 2 (R2), and 5 (R5) units of rotation. An additional solenoid, R2A, doubles the rotation provided by Solenoid R2. A special solenoid (check) provides for activation in the case in which there is no tilt and no rotation. Figure 1 documents the characters obtained with each combination of solenoids; Figure 2 depicts

The authors gratefully acknowledge the contributions of Elizabeth Palya in all phases of producing this system and manuscript. Requests for reprints or software should be sent to William L. Palya, Department of Psychology, Jacksonville State University, Jacksonville, Alabama 36265.

| | | | | | | | | | | | | | | |
|---------|-----|----------------|-------|---|---|-----|------|---|-----|-----|-------|--------|----------|---|
| | | ROTATE TILT | CHECK | 1 | 2 | 1+2 | 2+2A | 5 | 1+5 | 2+5 | 1+2+5 | 2+2A+5 | 1+2+2A+5 | |
| UNSHIFT | 2+1 | g | o | 6 | 5 | 2 | z | 4 | 8 | 7 | 3 | € | | 3 |
| | 2 | b | h | k | e | n | t | l | c | d | u | x | | 2 |
| | 1 | w | s | i | ' | . | ! | o | a | r | v | m | | 1 |
| | 0 | - | y | q | p | = | j | / | , | : | f | g | | 0 |
| SHIFT | 2+1 | (|) | € | % | @ | Z | S | w | g | # |] | | 3 |
| | 2 | B | H | K | E | N | T | L | C | D | U | X | | 2 |
| | 1 | W | S | I | " | . | O | A | R | V | M | | | 1 |
| | 0 | - | Y | Q | P | + | J | ? | , | : | F | G | | 0 |
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | |

Figure 1. Characters obtained with each combination of shift, tilt, and rotate solenoids. The solenoid pattern is given to the left and above the matrix. The actual number of each position is given to the right and below the matrix.

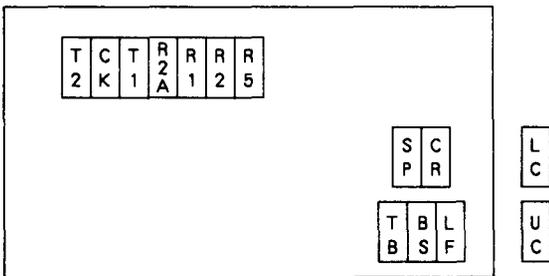


Figure 2. The location of each solenoid as viewed from the top of a Selectric typewriter. The front is depicted in the lower portion of the figure. The solenoids located in the left rear of the typewriter from left to right are: tilt 2, check, tilt 1, rotate 2 doubler, rotate 1, rotate 2, and rotate 5. The functions that solenoids located in the right front provide, from left to right, are: tab, space, backspace, carriage return, and line feed. The upper- and lowercase solenoids are located to the right side of the Selectric.

the location of these solenoids when viewed from the top of the typewriter. The identification of particular solenoids can be confirmed by providing power to the motor, depressing each solenoid arm, and noting what happens.

CUSTOM HARDWARE

The control board requires +5, -5, +12, and unregulated 30-48 V dc. All but the 48 V for the solenoids can be provided from the main computer through the same cable that is used for the RS-232 data path. A somewhat larger than usual capacitor can be added at the power input point of the board to compensate for the long power lines.

A block diagram of the controller board is presented in Figure 3. The board is built around a 6802 microprocessor. The 6802 contains 128 bytes of integral random-access memory (RAM), which is sufficient for the stack and scratch-pad memory required by the controller. The board also uses a 1 KB X 8 programmable read-only memory (PROM) to contain the firmware monitor, a parallel port, or PIA, to interface the micro-computer to the solenoids, and a serial port, or ACIA, to interface the controller to the host computer.

Figure 4 provides a schematic diagram of the controller. Very few support parts in addition to the block diagram are necessary to implement the controller. A 2.4576-MHz crystal and three gates of a 74LS04 hex inverter provide the CPU clock. The clock divided by four, or "enable," is used as the baud-rate clock. UHP407, 80-V, 500-mA drivers are used to switch the current needed to drive the solenoids. The RS-232 signals are buffered through 1488 and 1489 RS-232 drivers and receivers, and one of the remaining inverters in the 74LS04 is used to enable the firmware controller program.

The parts can be purchased for about \$35 if the major chips are obtained from the Motorola University Support Program. The controller can be assembled in about 6 h.

CUSTOM SOFTWARE

As can be seen in the block diagram of the software in Figure 5, the firmware monitor is very simple. The microprocessor simply reads an ASCII character from the RS-232 port, disables further transmission while the character is being printed, reenables transmission, and then waits for another character. A conversion table is used to determine the output pattern for each ASCII character and could easily be changed for some other typewriter or typeball code.

At present, the controller is a passive RS-232 terminal. However, adding intelligence is simply a matter of adding to the monitor. For example, a page that requires two different typeballs could be printed in two passes. During the first pass, all characters could be typed normally except those between the occurrences of a special character. Those characters between the special character would be replaced with blanks. On the second pass, with a different typeball, the reverse would be true. All characters would be replaced with blanks except those between the special characters, which would be printed.

COMMERCIAL SOFTWARE

We presently use a Selectric printer with DEC's Runoff for word processing. Figure 6 illustrates the process and the notable features of the editor and

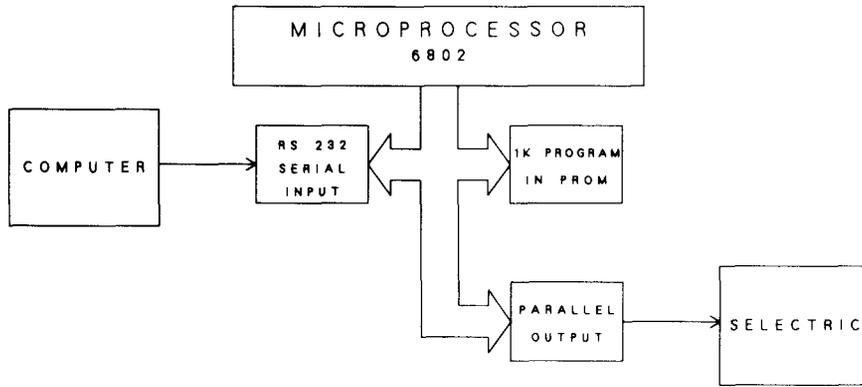


Figure 3. A block diagram of the Selectric controller board. A 6802 microprocessor reads input from a computer through a standard RS-232 serial port and outputs the solenoid patterns to the Selectric through a parallel port. A firmware program in PROM can provide whatever degree of intelligence is desired.

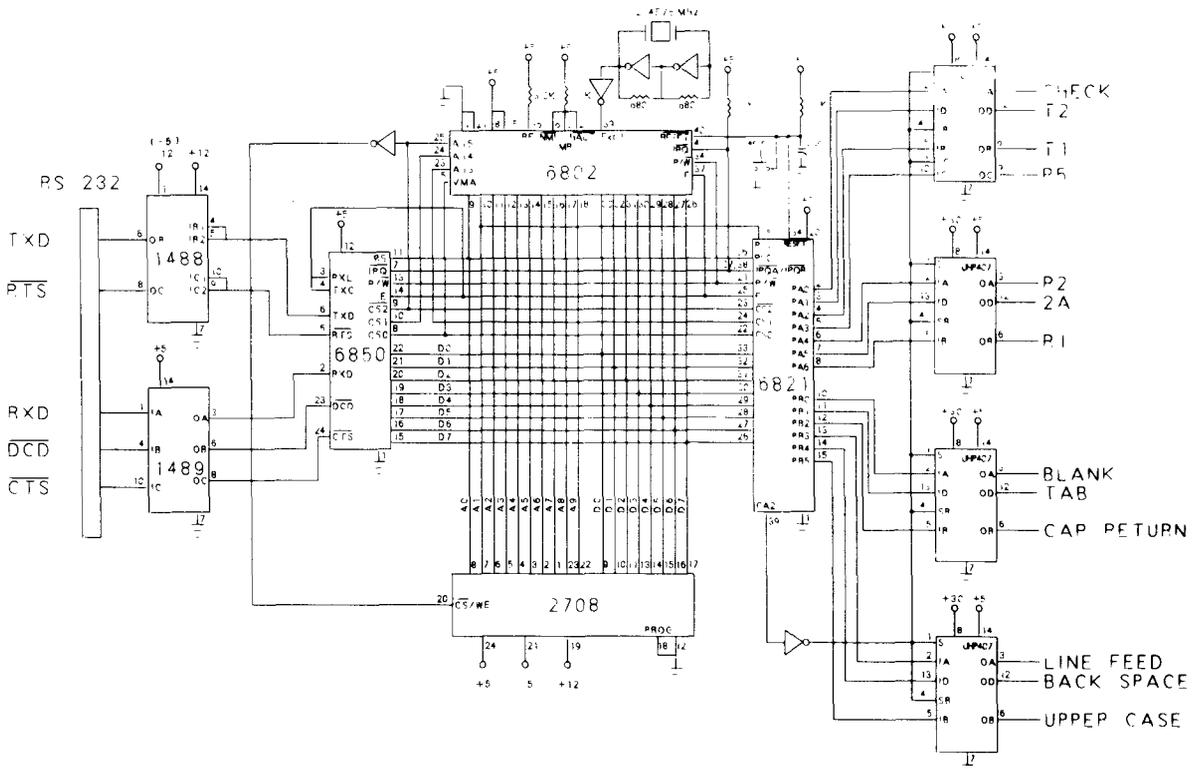


Figure 4. The 6802 reads data from the computer through a 6850 ACIA serial port. The 6850 is interfaced to the RS-232 lines through 1488 and 1489 buffers. A 2708 UVPROM provides program storage. A 6821 parallel controller drives four UHP407 high-current drivers, which in turn activate the solenoids in the Selectric.

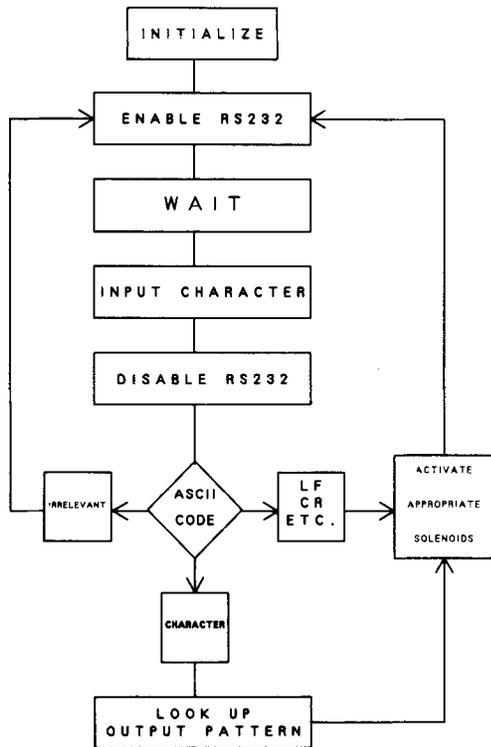


Figure 5. A block diagram of the Selectric controller firmware. When a character is received, the controller disables further input while outputting appropriate solenoid patterns to the Selectric. It then enables input and waits for the next character.

Runoff. Text is entered and manipulated with DEC's editor following the conventions required by Runoff. Subsequently, the file can be output to the Selectric through the Runoff program that formats the output.

The editor is a text processor based on a movable pointer. The "pointer" designates where text is to be inserted, changed, or deleted. The pointer functions like a nondestructive cursor that can be moved forward or backward some number of characters or lines. The "search" command even allows the pointer to be moved to the nth occurrence of some specified text. The editor also provides utilities that allow the movement of a block of text from one place to another and the ability to execute an entire command string some number of times.

Runoff is a text processor optimized for output to hard copy. Runoff will justify the right-hand margin, and it allows the specification of characters per line and lines per page. Text can be added or deleted without regard to its effects on the output, in that all carriage returns, line feeds, and form feeds are determined at the time of printing. Runoff can place the appropriate page number, data, title, and subtitle at the top of each page, and if desired, it will pause after each page is completed to allow individual sheets such as letterhead to be used. Terms can be flagged as they are entered and Runoff will create an alphabetical index providing the page number on which each term appears. In addition, space can be allocated for figures, and footnotes are properly inserted at the bottom of the appropriate page.

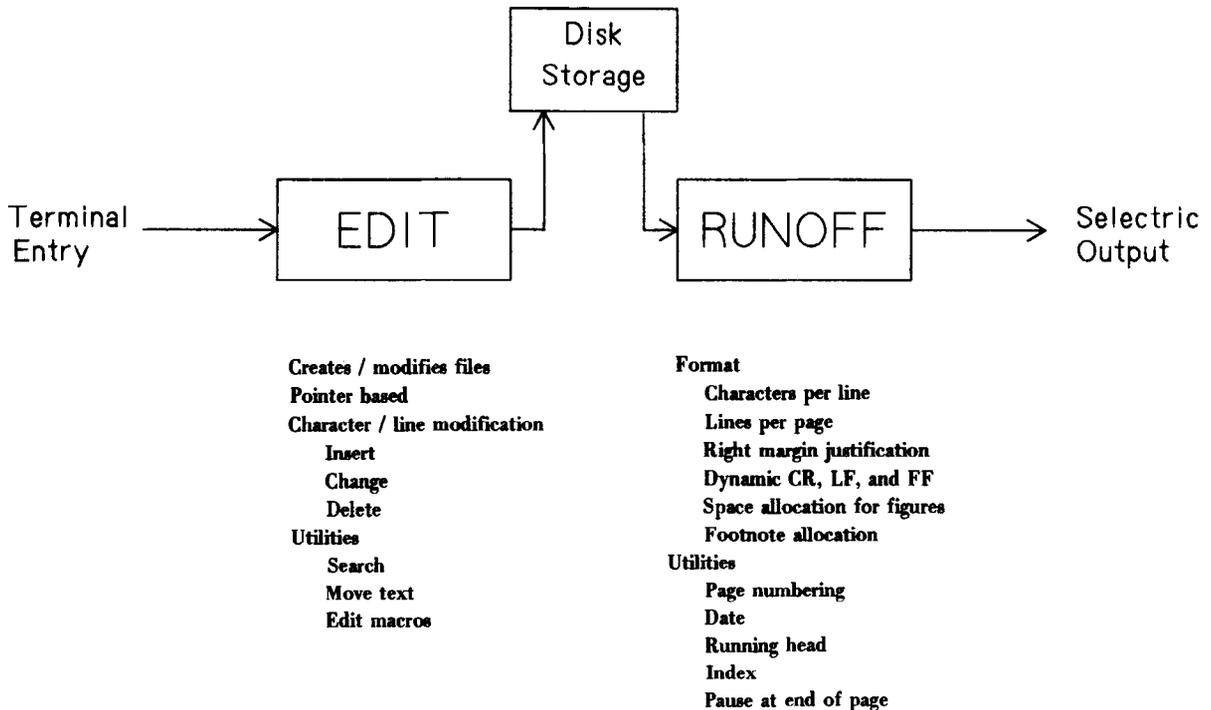


Figure 6. The editor inputs terminal entry, provides for text manipulation, and produces a storable text file. Runoff inputs a disk file, formats it, and generates RS-232 output.