Kathleen Peel takes the wraps off the COMX 35. With a built-in joystick and 35k memory the new arrival is full of Eastern promise.

THE COMX 35 is well made and nicely packaged, supplied complete with cassette leads, a cable to connect to your domestic TV, a power supply with integral three pin plug and a Basic Manual.

The keyboard houses plastic keys with a calculator-type feel but without any loud speaker feedback. Because of its size it is unsuitable for touch typing. The joystick is centre-biased and produces non-printing codes when activated; its use is mainly for self-written and commercial games.

The cassette interface at 600 baud seems relatively trouble free and actually requires that both earphone and microphone leads are left connected. This allows a spoken header for each tape track which is heard through the computer loudspeaker on playback. Both program and data files may be saved.

Expansion is through a double sided 44-way socket, at present no information exists as to the connections of the socket.

There are plans for an RS-232 — Centronics interface to take a suitably badge-engineered version of the extremely popular Sharp four-colour printer plotter. Floppy disc drive, speech synthesiser and memory expansion to 67K are all as they say "to be available shortly".

The machine is one of the coolest running micros I have seen, it barely heats up which is a good indicator as to the reliability of the electronics.

The Computer is based on the RCA 1802 microprocessor — an 8 bit register-orientated central processing unit, CPU. Its main features are low power consumption, a register array, R0-RF, consisting of sixteen 16-bit scratchpad registers and 91 easy-to-use instructions. A summary of the registers is given in table 4.

Switching on repeats a display routine waiting for any key other than space to be pressed. It then prints on the screen
COMX Basic V1.00
READY.

As you enter data it is printed in white with the computer response which can be selected, in cyan and the cursor in pink, an interesting use of colours for data presentation.

Program errors are denoted by error codes which although easily referenced in the manual, will be of little use to the inexperienced beginner without further explanation. For instance "unacceptable character in number field" is the explanation for error code 45.

Editing a line is simple. The line is called into the editor and displayed at the bottom of the screen, the cursor is spaced along underneath the line to the desired position and either I,C or D is typed — Insert, Change, Delete — and the amendment made on the cursor line. Typing Control S puts the correction into the edit line and allows further changes. A second Control S puts the corrected line back into the

program. The maximum line length is 95 characters and the cursor line commences below the end of the line to be edited, so it may be displayed by up to three lines which is a bit confusing. Not the best editor I have seen, but by no means the worst.

A novel command, Control R, recalls the text prior to the last press of the return key with any data typed after the return superimposed at the beginning of the line. Very useful for changing line numbers and minor changes at the end of often repeated commands.

At switch on there are 30934 Bytes available to the user according to Print Mem. It prints 256 less than actually available to allow for stack growth. The Basic implementation on this micro has one major drawback — it is incredibly slow, taking approximately four times as long to complete the timing tests used for the Spectrum — Oric evaluation presented in the March 1983 issue of Your Computer.

The usual selection of Basic commands are available, but disturbingly the manual has no references to any printer command.

This indicates the level of standardisation of COMX Basic. Table 1A lists commands which just vary in the keyword and Table 1B lists commands that are either not defined in the dictionary or have a non-standard meaning.

Redefinable character set

There are no simple structures such as If — Then — Else and On — Gosub but their relevance is debatable. The Basic interpreter inserts and deletes spaces as it merrily tidies up your data entry. As you may type PR — a shortened form of PRINT, there is a slight problem with lines such as PR INT (A /256) which the interpreter resolves as PRINT (A /256).

The whole of the character set is redefinable by using the Shape command, each character being formed within a 8 × 9 character cell. The two most significant bits are used to define the colours and the remaining six the pixel content of the relevant row of the character. This theoretically is capable of giving high resolution and the program below does that. As you can see, not all the character codes are usable within the display and the user is left with about 112 definable characters. The characters are duplicated in the top and bottom half of the character set, each half able to use a set of four colours — Black, Blue, Green and Cyan or Red, Magenta, Yellow and White.

Unfortunately, there is a slight problem, whenever the shape command is used, the screen nearly always blinks — it seems to be a timing problem. If the programmer does not redefine characters whilst a program is running then this will not be a problem.

There are no Draw, Plot and Circle commands which is not surprising as the display does not appear to be memory mapped. This is the hi-res program:

```
1 B=0:CPOS (0,0):CLS
2 FOR A=32 TO 127:GOSUB 7:NEXT
3 FOR A=144 TO 256:GOSUB 7:NEXT
4 GOTO 2
7 PRINT CHR$ (A);B=B+1:IF B>96
   EXIT 10
8 RETURN
```

This is the low-res program:

```
10 Z$ = "0123456789ABCDEF":FOR A = 144
   TO 256:GOSUB 50:NEXT A
12 WAIT (500):GOTO 12
50 A$ = "":FOR B = 0 TO B1:C$ = MID$
   (Z$, 1 + INT(RND(16)), 1):A$ = A$ + C$:
   NEXT B:SHAPE (A,A$):RETURN
```

I mentioned earlier that data entry from the keyboard is white and the computer response cyan, which can be changed to a number of other permutations. Unfortunately there is a

side effect, coloured graphics entered via the keyboard in a program change colour when printed by the computer.

Larger picture

The machine has the usual six colours plus black and white. The paper colour covers the whole screen for all colours and does not leave a border like on the Oric or Spectrum. The display which is 24 × 40 characters covers

virtually the whole of the TV screen and gives a picture almost 20 percent larger than the Spectrum with the same character definition. Spectrum definition is 24 × 32. Colours are good, stable with very little noticeable dot crawl. The colour commands are in table 3.

The sound commands in table 2 are also good and capable of giving a wide range of realistic noises from lasers to explosions, and could even provide a fair imitation of speech

which would give an extra dimension to games. Although the machine does not suffer from the dreaded Sinclair power supply hum, there is a hum coming from the loudspeaker which is annoying.

The joystick is effectively four additional keys which are activated by pushing the joystick in an appropriate direction. These keys produce non-display character codes

---

Kathleen Peel takes the wraps off the COMX 35. With a built-in joystick and 35k memory the new arrival is full of Eastern promise.

The COMX 35 is well made and nicely packaged, supplied complete with cassette leads, a cable to connect to your domestic TV, a power supply with integral three pin plug and a Basic Manual.

The keyboard houses plastic keys with a calculator-type feel but without any loud speaker feedback. Because of its size it is unsuitable for touch-typing. The joystick is centre-biased and produces non-printing codes when activated; its use is mainly for self-written and commercial games.

The cassette interface at 600 baud seems relatively trouble free and actually requires that both earphone and microphone leads are left connected. This allows a spoken header for each tape track which is heard through the computer loudspeaker on playback. Both program and data files may be saved.

Expansion is through a double sided 4~way socket, at present no information exists as to the connections of the socket. There are plans for an RS232 - Centronics interface to take a suitably badge-engineered version of the extremely popular Sharp four-colour printer plotter. Floppy disc drive, speech synthesiser and memory expansion to 67K are all as they say "to be available shortly".

The machine is one of the coolest running micros I have seen, it barely heats up which is a good indicator as to the reliability of the electronics.

The Computer is based on the RCA 1802 microprocessor - an 8 bit register-orientated central processing unit, CPU. Its main features are low power consumption, a register array, R0-RF, consisting of sixteen 16-bit scratchpad registers and 91 easy-to-use instructions. A summary of the registers is given in table 4.

Switching on repeats a display routine waiting for any key other than space to be pressed. It then prints on the screen

```
COMX Basic V1.00

READY
```

As you enter data it is printed in white with the computer response which can be selected, in cyan and the cursor in pink, an interesting use of colours for data presentation.

Program errors are denoted by error codes which although easily referenced in the manual, will be of little use to the inexperienced beginner without further explanation. For instance "unacceptable character in number fold" is the explanation for error code 45.

Editing a line is simple. The line is called into the editor and displayed at the bottom of the screen, the cursor is spaced along underneath the line to the desired position and either I,C or D is typed - Insert, Change, Delete - and the amendment made on the cursor line. Typing Control S puts the correction into the edit line and allows further changes. A second Control S puts the corrected line back into the program. The maximum line length is 95 characters and the cursor line commences below the end of the line to be edited, so it may be displaced by up to three lines which is a bit confusing. Not the best editor I have seen, but by no means the worst.

A novel command, Control R, recalls the text prior to the last press of the return key with any data typed after the return super- imposed at the beginning of the line. Very useful for changing line numbers and minor changes at the end of often repeated commands.

At switch on there are 30934 Bytes available to the user according to Print Mem. It prints 256 less than actually available to allow for stack growth. The Basic implementation on this micro has one major drawback - it is incredibly slow, taking approximately four times as long to complete the timing tests used for the Spectrum - Oric evaluation presented in the March 1983 issue of Your Computer.

The usual selection of Basic commands are available, but disturbingly the manual has no references to any printer command.

This indicates the level of standardisation of COMX Basic. Table lA lists commands which just vary in the keyword and Table lB lists commands that are either not defined in the dictionary or have a non-standard meaning

Redefinable character set

There are no simple structures such as If - Then - Else and On - Gosub but their relevance is debatable. The Basic interpreter inserts and deletes spaces as it merrily tidies up your data entry. As you may type PR - a shortened form of PRINT, there is a slight problem with lines such as `PR INT (A /256)` which the interpreter resolves as `PRINT (A/256)`.

The whole of the character set is redefinable by using the Shape command, each character being formed within a 8 x 9 character cell. The two most significant bits are used to define the colours and the remaining six the pixel content of the relevant row of the character. This theoretically is capable of giving high resolution and the program below does that. As you can see, not all the character codes are usable within the display and the user is left with about 112 definable characters. The characters are duplicated in the top and bottom half of the character set, each half able to use a set of four colours - Black, Blue, Green and Cyan or Red, Magenta, Yellow and White.

Unfortunately, there is a slight problem, whenever the shape command is used, the screen nearly always blinks - it seems to be a timing problem. If the programmer does not redefine characters whilst a program is running then this will not be a problem.

There are no Draw, Plot and Circle commands which is not surprising as the display does not appear to be memory mapped. This is the hi-res program:

```
1 B=O:CPOS (0,0):CLS
2 FOR A=32 T0 127:GOSUB 7:NEXT
3 FOR A=144 T0 255:GOSUB 7:NEXT
4 GOTO 2
7 PRINT CHR$ (A);:B=B+l:IF B>958 EXIT 10
8 RETURN
10 Z$="0123456789ABCDEF":FOR A=144 TO 255:GOSUB 50:NEXT A
12 WAIT (500):GOTO 12
50 A$="":FOR B=0 TO 18:C$=MID$ (Z$,1+INT(RND(15)),1):A$=A$+C$: NEXT B:SHAPE
(A,A$):RETURN
```

I mentioned earlier that data entry from the keyboard is white and the computer response cyan, which can be changed to a number of other permutations. Unfortunately there is a side effect, coloured graphics entered via the keyboard in a program change colour when printed by the computer.

Larger picture

The machine has the usual six colours plus black and white. The paper colour covers the whole screen for all colours and does not leave a border like on the Oric or Spectrum. The display which is 24 x 40 characters covers virtually the whole of the TV screen and gives a picture almost 20 percent larger than the Spectrum with the same character definition. Spectrum definition is 24 x 32. Colours are good, stable with very little noticeable dot crawl. The colour commands are in table 3.

The sound commands in table 2 are also good and capable of giving a wide range of realistic noises from lasers to explosions, and could even provide a fair imitation of speech which would give an extra dimension to games. Although the machine does not suffer from the dreaded Sinclair power supply hum, there is a hum coming from the loudspeaker which is annoying.

The joystick is effectively four additional keys which are activated by pushing the joystick in an appropriate direction. These keys produce non-display character codes which auto-repeat and can be read by the key function. This returns the ASCII code of the current key being depressed. So If Key = 136 Then. . . Because of the nature of the keyboard scanning routine, the use of the key facility in a program such as figure 2 creates problems and does not smoothly move the character around the screen. It is necessary to make the variable the value of the last key pressed and integrate the variable as in figure 2. This point is omitted from the manual.

The content of the manual for the beginner is good and quite clear but spoilt by rather too many silly errors. There is virtually no technical information in the manual. It really should include 1802 assembler instructions, memory maps, system variables and I/O socket connections. They may not be required initially, but you will not get very far without them.

Glancing through the ROM reveals three commands not in the manual. Dos Pout and Tout. Dos and Pout give error code 62, "ROM or ROM card not present" and Tout reruns Ready.

There appear to be ROM routines for double and quadruple-size printing available which should be very useful.

CONCLUSIONS

- The Comx 35 cannot be recommended for business use as it lacks speed and a suitable keyboard.
- If an adequate library of high-quality machine-code arcade games can be assembled then the Comx 35 could be treated as a games machine with computing capabilities, where the integration of a joystick with the keyboard will make for a more realistic arcade-style presentation than a normal keyboard. The demonstration tape contains some good machine-code arcade-type games.
- For a 1983 computer it has an extremely slow version of Basic and it uses an outmoded CPU which is unknown in today's popular micros.
- It has an incomplete version of Basic and some commands have non-standard meanings
- Error codes need to be more explicit about the nature of the error and could be greatly enhanced by the use of examples in the manual
- It has a non-standard size keyboard.

Table 1. Comx 35 standard Basic commands.

```
ABS ASC ATN CHR COS DATA DEFINT DEG END EXP FOR-NEXT GOSUB GOTO IF-THEN
INPUT INT LEN LET LIST LOG MEM MID MOD NEW PEEK POKE PI PRINT RAD READ REM
RENUMBER RESTORE RETURN RND SGN SIN SQR TAB WAIT
```

Table lA.

```
FVAL=VAL KEY=INKEY
```

Table lB. Non-standard Basic commands.

`CALL (NN, a, b)`	Transfer execution to machine code routine at address NN passing data a and b into R8 and RA.
`CLD`	Clear all strings and arrays.
`CLS`	Clear screen from current cursor position
`CPOS (Y, X)`	Place the cursor at position Y,X.
`DEFUS NN`	Creates a space between variables and the start of the program (NN) to be used for machine language routines
`DIM`	A maximum of 26 arrays limited to 255 in any dimension, also 26 strings maximum. Destroyed by Run, New, CLD and perhaps Editing.
`DLOAD`	Load data stored by a previous DSave.
`DSAVE`	Save data for subsequent retrieval by DLoad.
`EOD`	Prints hex address of end of data.
`EOP`	Prints hex address of end of program.
`EXIT N`	Unconditional branch to line N. If used as an escape from a For/Next loop it should jump to the next level down of nesting if applicable.
`FIXED N`	Formats the printing of all numbers. N is number of digits to the right

	of the decimal point.
`FNUM (exp)`	Round to nearest whole and convert to floating point.
`FORMAT N`	Specifies field size N for printed numeric data until turned off.
`INUM (exp)`	Round to nearest whole number and convert to integers.
`PLOAD`	Load program stored by a previous Psave
`PSAVE`	Save program for subsequent retrieved by PLoad.
`RND (N)`	Returns a random floating point number from 0 to less than N. A=RND (15) will not work. A=INT (RND (15)) is acceptable.
`RUN N`	Runs program from line N but does not clear the data space.
`RUN +`	The computer replaces all interpretive branches such as Goto 150 with absolute addresses and then Runs the program.
`SHAPE (A, "18 HEX numbers")`	Redefines character code A - see text.
`TIME (T)`	Time to elapse before jumping to subroutine
`TIMEOUT (N)`	Address N - 50 units per second for T.
`TRACE (A)`	A30 Trace off A != 0 Trace on.
`USR (NN, a, b)`	As call but used as part of an expression returning a 32-bit binary integer number constructed from RB (low) and RA (high).

Table 2. Sound commands.

`MUSIC`	Note,octave,amplitude
`NOISE`	Frequency, amplitude
`TONE`	Frequency, octave, amplitude
`VOLUME`	Level

Table 3. Colour commands.

`COLOUR (N)`	Where N = 1 to 12 and represents the colour combination used for keyboard input and computer response - initially set at 12.
`SCREEN (N)`	Where N = 1 to 8 and represents the background colour - default set to 1, black.
`CTONE (N)`	N != 0: computer response a shade brighter than the screen colour. N =0: effect turned off. Colour and screen are global and affect the whole of the display.

Table 4. 1802 Register summary.

D	8 Bits	Data Register (Accumulator)
DF	1 Bit	Data Flag (ALU Carry
R	16 Bits	1 to 16 scratchpad Registers
P	4 Bits	Designates which register is program counter
X	4 Bits	Designates which register is data pointer
N	4 Bits	Holds low-order Instruction digit
I	4 Bits	Holds high-order Instruction digit
T	8 Bits	Holds Old X, P after Interrupt
IE	1 Bit	Interrupt Enable
Q	1 Bit	Output Flip-Flop.

Figure 1. Program to print out character sat and pause while space key depressed.

```
1 FOR A=1 TO 255
2 IF KEY=32 THEN GOTO 3
5 PRINT A, CHR $ (A);
8 NEXT A
10 END
```

This modification is required to enable the above program to work.

```
3 IF KEY=32 THEN WAIT (20): GOTO
```

Figure 2. Operating joystick.

```
10 CPOS (0, 0): CLS
20 A=12: B=20
30 IF K=136 THEN A=A-1     Up
40 IF K =139 THEN B=B-1    Left
50 IF K =138 THEN A=A+1    Down
60 IF K =137 THEN B=B+1    Right
70 CPOS (A, B) PR "*"
80 GOTO 30
```

Figure 3

Comx 35	Spectrum
| ```
1  B=0
10 FOR A=1 TO 10,000
15 GOTO 20
20 B = B+1
25 LET K=KEY
30 NEXT A
40 PRINT A,B
50 END
80 GOTO 25
``` | ```
1  LET B=0
10 FOR A=1 TO 10,000
15 GOTO 20
20 LET B = B+1
25 LET K=KEY
30 NEXT A
40 PRINT A,B
50 STOP
80 GOTO 25
``` |
| RUN 400 secs.
RUN+ 350 secs. | 100 secs. |