

=====

Excalibur and BASIC

By Bob Green.

The following Commands are in (ROM) but unfortunately were never implemented. LSET,PSET,SPEEK and SPOKE. So, what are you missing out on ?.... Probably one of the most valuable attributes for the Excalibur, would be the ability to SET individual screen pixels. Complete control of colour and high resolution graphics are just not possible without this function. Well all is not lost, Kinetic Systems have had ideas for some time about a re-vamp of BASIC. Avenues explored included exchange of the 16K Basic Eproms to a healthy 32K thereby permitting heaps of additional routines such as a Monitor program, Routines for LSET, PSET, DRAW, CIRCLE, LINE etc, Routines for Cassette users which would enable greater flexibility, the mind boggles !!!!! However Kinetics do not have the time to write these routines, but do have contacts with certain people that can. The question is, how many users want an update and are prepared to pay for it? I for one are 100% in favour. In order to get an idea of the demand, I propose that all those interested drop me a line pronto or ring, and let me know the kinds of things you would like to see included in a new Excalibur Basic. Extensive testing would be a prime requirement, in order to ensure bug-free ROMs, as would be the need to bank-select the additional ROM's to ensure that there is no (RAM MEMORY) lost in the transition. The other possibility is that certain users may have the necessary know-how to write some of the routines needed. If so, PLEASE come forward and reveal your shining talents. My postal address is: 3 Wooton Ct, Melton South, Vic 3338. Phone 743-3149.

Using USR

It is possible to call any of the routines in ROMs, from BASIC by the use of the USR Statement, eg; to call the CLS Routine we must force the CPU to jump to location 58 Hex. OK! type in the following program;

```
10 Poke -958,88:Poke -957,0
20 X= USR (0)
```

and run. You will be rewarded with a Clear Scrn and READY>. How does this work?..Simple, Decimal 88 in line 10 is the equivalent of 58H (your ROM routine location) and is the (LSB) value, Dec 0 is Equiv to 00Hex and is the (MSB) value. Therefore we have poked into locations 957-958 the Hex address of 0058. Line 20 then causes the CPU to jump to line 10 addresses and do a further jump to the CLS routine in Rom. All the other jump table functions should be available in the same way. To return to MEMORY SIZE Message, use LSB=1 : MSB=1

A Brick-bat

I am getting a lot of requests for Cassette Software, but no-one has sent any programmes to assist in building up the Library. Without your help, this service will cease to function. Let's have some programmes to help maintain some perspective. Your cassette will be returned, together with any latest additions - if applicable). New!- 'Hangman' and 'Programming the 6845'

Extended Disc BASIC.

Did anybody notice the article in E.T.I re. an Extended Basic for the TRS-80, SYSTEM-80?..... Many of the commands made available through this disc or cassette programme would greatly enhance EXCALIBUR'S Rom Basic. After contacting the author of this creation (A Queensland kid), the offer has been made to look into adapting EXTENDED BASIC for the Excalibur. What is needed, is an assembly dump of ROMs on disc in TRS-80 disc format before any further progress can be made. If any user has a TRS-80 or has access to same, please contact me on Melb. (03) 743-3149.

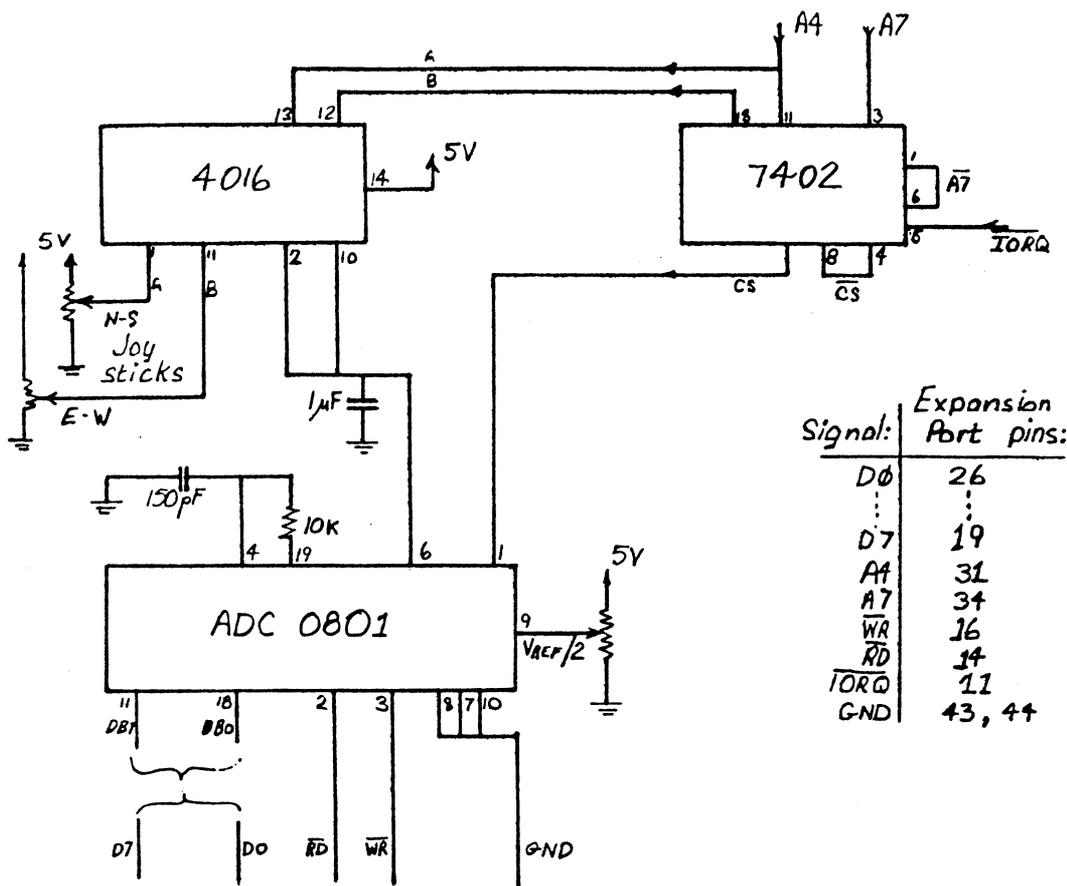
=====

A Proportional Joy-stick for the Excalibur

by J. Deerson

Two broad types of joy-stick controller exist: switched, and proportional. The simple switched type has a set of four switches which are operated by N, S, E, or W movement of the stick. A much better arrangement from the operator's viewpoint is the proportional type, which allows a direct "mapping" of the

monitor screen dot positions on to all the joystick positions - i.e. stick movement causes proportional cursor movement. Thus an analog to digital converter (ADC) interface is required. A joystick is good value in many situations. Imagine adapting Wordstar to joystick control! Or drawing pictures on the screen! In fact, it's useful wherever operator controlled cursor movement is required. I used a special ADC chip and created two extra ports to do the job. However there are many other solutions (e.g. use an op-amp and do the a-d conversion in software). My circuit is shown in Fig. 1. At present, all ports from 80H to FFH are unused on the Excalibur. The circuit shows N-S joystick pot sampled by port 80H, and E-W at 90H. A BASIC listing which will move the cursor around the screen is shown below.



```

5   OUT 48,10:  OUT 49,0      REM MAKE CURSOR STEADY
10  DEFINT A-C:  OUT 128,0    REM START CONVERSION (N-S)
20  B=INP(128)/10:  B=80*B    REM GET ADC VALUE
30  OUT 144,0     REM START CONVERSION (E-W)
40  C=(INP(144)/3.2)+B      REM GET ADC VALUE
50  IF C>1919 THEN C=1919
60  A=C/256           REM "6845" CURSOR REG. VALUE
70  OUT 48,14        REM (HIGH BYTE)
80  OUT 49,A
90  A=((C/256)-A)*256    REM GET "6845" CURSOR
100 OUT 48,15        REM LOW BYTE VALUE
110 OUT 49,A
120 GOTO 10          REM LOOP BACK INDEFINITELY.
    
```

Line 5 puts the cursor into "non-blinking" mode on entry to the program. Line 10 starts conversion of the N-S pot value (a Write to port 80H). The digital result, which can have any value in the range 0 to 256, is then divided by 10 to give a value in range 0 to 25 (24 lines on a screen) then multiplied by 80 (characters per line), and assigned to variable B. Dividing first by 10 and converting immediately to an integer value reduces cursor jitter, as slight variations in voltage and pot position are rounded off. Line 30 gets the E-W digital value; dividing by 3.2 reduces the range to 0 to 80. C is assigned the sum of this value and B. Thus C value now represents the pot position as a screen character position in range 0 to 1919 approximately.

It's all plain sailing now. Line 50 restricts C to the proper range. Line 60 gets the value to be placed in the High byte of the 6845 Cursor Position regis-

ter pair and lines 70, 80 place it there. Line 90 gets the (integer) remainder and puts this in the Low register. Then do the whole thing again. I bought a neat little joy-stick at Dick Smith just before Christmas for \$4.50. The I.C.s may amount to \$10 or so depending on what you have lying around the place. I mounted the whole circuit on an I.C. prototyping board. Be careful when connecting the wires to the expansion bus: bus contention could arise if you connect up the wrong address lines for example.

This project is a sort of follow-up and complement to the light-pen I constructed some time ago. What about a mouse, anybody? Another "urgent" project is expansion RAM - say another 64k - to take further advantage of the Excalibur's excellent graphics capability. Present RAM prices should get you this for about \$50 I would say. Well worth it! Of course, all these projects would be so much easier if there was a plug-in PCB capability on the Excalibur, e.g. STD or S100. Ideas?

=====