www.Commodore.ca
May Not Reprint Without Permission

commodore

comments and bulletins concerning your COMMODORE PET™

The Transactor

BULLETIN #11
April 30, 1979

PET™ is a registered trademark of Commodore Inc.

This months Transactor contains an excellent article by Brad Templeton on Pet interrupts. Also information on a PET to IBM selectric interface.

This is also the final issue of Volume 1. A subscription form for Volume 2 follows. When submitting the form, please address it to Commodore's Agincourt address followed by 'Attn. The Transactor'.

Transactor Volume 2

Yes it's renewal time. Although your Volume 1 subscription covers one more issue, here is the Volume 2 subscription form. The Transactor operates on a break-even basis. Therefore the cost of "The Transactor Volume 2" will be $15.00.

Recently I have received various requests regarding article subjects. If you have a subject idea you and other PET users would like to see discussed in future Transactors, please include it at the bottom of the order form.

....

To receive Transactor Volume 2 bulletins, please return this form with your cheque for $15.00 annually renewable, to CBM 3370 Pharmacy Avenue, Agincourt, Ontario, M1W 2K4. Volume 1 back issues will be available at 10 dollars for a limited time only (while supplies last).

VOL 2

NAME...

COMPANY (if applicable)..

ADDRESS..

....POSTAL CODE.........................

RECEIPT REQUIRED? YES ☐ NO ☐
(Invoices cannot be issued for the $15.00 annual fee)

IDEAS & COMMENTS...

....

www.Commodore.ca
May Not Reprint Without Permission

Convert Upper to Lower Case

The following is a program that will convert all upper case text to lower case. However, keep in mind that any graphics above the alphabetic keys will now be unusable if they are to appear simultaneously with lower case letters.

```
59030 FOR T=1024 TO 8006-FRE(0):A=PEEK(T)
59031 ON Z GOTO 59034,59037
59032 IF A=153 OR A=178 THEN Z=1
59033 NEXT
59034 IF A=34 THEN Z=2:NEXT
59035 IF A=58 OR A=0 THEN Z=0
59036 NEXT
59037 IF A 64 AND A 91 THEN POKE T,A+128
59038 B=PEEK(T+1):IF B=34 OR B=0 THEN Z=0
59039 NEXT:END
```

When writing the program use no spaces. The program will convert strings and PRINT statements but will not affect DATA statements. Also, it may terminate with a '?NEXT WITHOUT FOR ERROR IN 59036' but that's OK.

Of course you need not use the same line numbers. They were chosen due to their unusualness. The program was then recorded using the UNLIST routine in Transactor #7. It can then be merged with other programs with a good chance of <u>not</u> interfering with other program lines.

Karl J.

www.Commodore.ca
May Not Reprint Without Permission

INTERUPTS ON THE COMMODORE PET

(c) 1979 Brad Templeton

One of the most important features of the COMMODORE PET operating system is the use of interrupts. They are used to reset the PET, and they handle most of the tape and all of the keyboard i/o. This article will provide an introduction to interrupts on the 6502 (The PET's cpu) and a description of how the PET handles them. For your information, pseudo source listing is provided for the interrupt software of the PET, as produced by my disassembler.

Under normal conditions, a processor executes machine code in a linear fashion. It moves through memory, obtaining instructions (which can be one, two or three bytes long) and executing them. Sometimes, certain programmed instructions cause jumps to other places, just like GOTO and GOSUB of basic. To make a machine more flexible, however, interrupts are provided to do jobs that would be very expensive to do in software.

Essentially, an interrupt is controlled by a line right into the processor. When the processor detects the correct voltage on this line, an interrupt may be generated. First, in order to simplify matters, the processor finishes the instruction it is presently carrying out. Then, if the in-

www.Commodore.ca
May Not Reprint Without Permission

terrupt is ok (interrupts can be masked). The processor saves the program location it was at, and the contents of its flags onto the stack. It then goes to a special reserved area of memory (in ROM on the PET) and pulls out two bytes indicating what location it should start executing from. It then goes there and executes machine code until the instruction RTI (Return from Interupt $40) is encountered. It then goes back to the stack and restores its flags, and loads the location it saved to the instruction counter. It then goes and executes the code after where it stopped as though nothing had occured. (If the interrupt program was correctly written)

On the 6502, three types of hardware interrupts can occur, as well as a fourth special type. The locations they branch to are kept in byte pairs called vectors at the end of memory. One of these interrupts, NMI or Non Maskable Interrupt, can not be used on the PET. Its vector, $FFFA-B, points to $CA60, which is the middle of a subroutine. The line for this is also fixed off by a resistor on the pc board. Later PETs may plan to include this.

The interupt called for power up is named RES. It branches to a routine which sets up basic and the operating system. It also, through what I consider to be one of the PET's worst design flaws, branches to the routine to destructively test how much memory is in the machine. At the very start, it also tests the condition of the diag-

www.Commodore.ca
May Not Reprint Without Permission

nostic sense (MSB of $E810), and goes to the diagnostic routine if this is set. RES is fired by power up, or by grounding pin 27 on the bottom of your memory expansion bus. If you set it by touching that pin, it does not clear memory below $400, so programs there (the tape buffers) are safe. This is, unfortunately, a very small area. It vectors through $FFFC-D.

The general use, hardware inte upt is the IRQ. IRQ vectors through $FFFE-F, as does BRK. This points to location $E66B in the PET. It is generated every 60th of a second by the tv hardware, and can also be generated from the memory expansion bus, on pin 28. It is also connected to the 6522 versatile interface adaptor. I will discuss the 60 per second interrupts here in detail. For information of generation by the 6522 (there is another whole article's worth of material in there) you can write MOS[1] for the manual on it. Interrupts can be generated from it at exactly timed intervals, and by certain i/o conditions on the user port and IEEE bus. The exactly timed intervals are used to send precise frequency signals to the tape. (In fact, the 6522 is the PET's tape interface!)

The 60 per second interrupts do the following:

Scan the keyboard, checking for new keys and decoding them.

Increment the real time clock, and check for midnight

Flash the cursor if it is on. ($0224 = 0)

Test tape recorder status for stop-start

www.Commodore.ca
May Not Reprint Without Permission

Copy a byte for the break key test.

Whatever else you want them to do.

When the IRQ occurs, the code at $E66B (see source) saves the processor register A, X and Y on the stack. It then checks, by loading back from the stack, the flags, to see if the BRK flag was set. The BRK, a software IRQ, vectors through the same place, but sets the BRK flag. This is handy to test what type of interrupt occured. It then does a jump indirect to one of two places in RAM ($219 or $21B) depending on the type of interrupt.

Normally, the RAM IRQ vector is set to $E685, which is the standard IRQ code. BRK has no default setting. The small piece of code you see after the JMP indirects is the return code, which restores the registers and does the RTI. The first thing INT_CODE does is the JSR INCR_CLOCK which increments the clock and copies the PIA register the break key test uses. When Steve Punter of Mississauga saw this with the disassembler, he devised an ingenious way to disable the BREAK key of the PET. By telling the PET to branch to $E688 instead of $E685 by means of a POKE 537, 136 statement, the PET bypasses the INCR_CLOCK subroutine, and does not test the break key. (Note INCR_CLOCK passes through a JMP vector table in high ROM at $FFEA) This has the side effect of turning off the real time clock. When this statement is not used the clock proceeds normally. After it is updated, it is compared with a three byte table that con-

www.Commodore.ca
May Not Reprint Without Permission

tains the value for midnight. If it is midnight on the clock, it is zeroed. The PET also keeps a secondary clock just after the main one. This is used for calibrating the real time clock. About every 6 seconds, this clock reaches a special limit, and when it does, it is zeroed, and the main clock is not incremented on this cycle. This is becuase the interrupt generator runs slightly faster than exactly 60 times per second. Even with this compensation, you may have noticed the clock is a few seconds off after several hours of PET operation. If they had used the 60 hz ac power line for the interupt, it would have been more accurate, but that would have caused problems for PETs sold abroad.

After doing the clock, it proceeds to flash the cursor, once every third of a second, if the location FLASHING ($224) is set to zero. (POKE 548,0 in a program turns the cursor on, but with some bugs - try it and see.) It does it with a very silly method that has no apparent purpose, instead of the standard method, a EOR $80. It then sets up two keyboard test locations.

In using your PET, you may have noticed that if the tape drive is stopped by the machine itself, that you can push stop and play and the motor will run again. This is handled by the section of code at $E6CD. After this comes the keyboard interpretation routines. The method of decoding the keyboard PIA has already been published in your

www.Commodore.ca
May Not Reprint Without Permission

PET manual, and in PET user notes, so I will not dwell on it here. Once it has the matrix coordinate of the key, it waits for it to stabilize, to avoid bounce and repeating letters. (The TRS-80 does this poorly). It then converts the matrix number to an ascii character through the table at $E75C. (You can use this table in your programs, if you want to account for how long a key is held down - a great real time feature!) It then puts the key in the correct place in the keyboard buffer starting at $20F. Finally it goes back.

www.Commodore.ca
May Not Reprint Without Permission

WHAT YOU CAN DO

Because the PET IRQ goes through RAM, it is one of the main links you have that can give you operating system control. You can insert your own programs before and after the interupt code to have your PET do two jobs at once, like handle i/o while running basic. I have used interupts to write programs to:

Interpret the PET keyboard and the full sized keyboard I attached to the PET like a regular keyboard.

Provide functions like repeat after a certain period of time and shift lock.

Turn the ! key to a statement number key, so that it would provide a line number 10 higher with every push.

Have upper case letter keys print out as full basic keywords.

Display whole pages of PET memory constantly on the screen.

Provide a non-destructive reset that works in special cases.

Much more is possible.

To use your own programs, you merely set them up in some convenient location (machine code only), preferably starting at location that ends in $85, such as $385 in the second tape buffer. Something located there can then be started with a POKE538,3 and stopped with POKE 538,230, rather than having to write a special machine language program that disables the interrupt with SEI, changes the locations, and enables the interrupt with CLI. You do not

www.Commodore.ca
May Not Reprint Without Permission

need to disable if you are only changing one byte of the location. Put some code there and follow it with a JMP $E685. This way it does your code and proceeds on to do its own. If you put in the following series:

```
EE 50 80 4C 85 E6
```

starting at $385 (901 base 10), and initiate it with POKE 538,3 you will see a byte on the screen constantly increasing in "value", once every 60th of a second. The PET will also be doing everything else as usual. The following code:

```
A2 00 BD 00 00 9D 50 80 E8 D0 F7 4C 85 E6
```

will dump a page of memory on the screen constantly. You can poke 905 with the page you wish to examine. Try 0,1,2,4,31,232. It starts with page 0. When scanning page 0, move the cursor and see what happens.

While doing this, you may have noticed that there is no flicker whatsoever on the screen despite the massive amount of writing to it being done. (Far faster than BASIC printing). This is because the interrupt is fired by the screen scan signal, and the screen is doing nothing shortly after the interrupt goes. This is also why the flashing cursor will never "snow" the screen. You can store almost half a screen without "snow" this way.

Sometimes it is important to put code in after the interrupt code of the PET. This can be done by manipulation of the stack, and is necessary for programs like the statement

www.Commodore.ca
May Not Reprint Without Permission

numberer or keyword printer I included in my list above. I have included some code you can put in to allow you to do this. >PROG means the high order byte of where your post interupt code starts and PROG is the low order byte. PCLO and PCHI are two locations for storing the correct pc you can use. The program works by altering the stack, so that the PET goes to your program when it RTIs. The second part of the program, which finishes your routine off (GOBACK) resets the stack and restores the proper program counter and machine registers. You should be able to have a lot of fun with it.

It should be noted that probably the only reason the IRQ vector is in RAM is that the PET does change it for tape i/o routines. There is a table of possible vectors starting at $FD28 in the rom, and the table ends with the standard vector $E685. If you ever change the high order byte of the IRQ RAM vector, you must reset it before tape i/o is done. If you don't, the PET will reset it anyway, but the tape i/o may not be done, and you may crash your PET.

Incidentally, the disassembler was written in the system language B (a very nice, much improved BCPL) here at the University of Waterloo where I go to school and work for the Mathematics Faculty Computing Facility. This article was also prepared and formatted on the same Honeywell 66/60. Many of the labels used in the disassembly were provided through the massive effort of examining the PETs ROMs done

www.Commodore.ca
May Not Reprint Without Permission

by Jim Butterfield of Toronto. My next article for the Transactor will be on programming interactive games for the PET.

① The 6522 Data Sheets (24 pgs.) and other MOS publications are available through dealers.

www.Commodore.ca
May Not Reprint Without Permission

Here is the code for the interrupts on the PET

```
E66B   48          INTERUPT    PHA
E66C   8A                      TXA
E66D   48                      PHA
E66E   98                      TYA
E66F   48                      PHA
E670   BA                      TSX
E671   BD 04 01                LDA   $104,X
E674   29 10                   AND   #$10
E676   F0 03                   BEQ   $E67B
E678   6C 1B 02                JMP   [BRK_LOW]
E67B   6C 19 02                JMP   [IRQ_LOW]
E67E   68          RETURN_INT  PLA
E67F   A8                      TAY
E680   68                      PLA
E681   AA                      TAX
E682   68                      PLA
E683   40                      RTI


E684   60                      RTS


E685   20 EA FF    INT_CODE    JSR   INCR_CLOCK
E688   AD 24 02                LDA   FLASHING
E68B   D0 23                   BNE   $E6B0
E68D   CE 25 02                DEC   C_TIMER
E690   D0 1F                   BNE   $E6B0
E692   A9 14                   LDA   #$14
E694   8D 25 02                STA   C_TIMER
E697   A4 E2                   LDY   C_COLUMN
E699   4E 27 02                LSR   C_STATE
E69C   B1 E0                   LDA   (C_ROWADR),Y
E69E   B0 06                   BCS   $E6A6
E6A0   EE 27 02                INC   C_STATE
E6A3   8D 26 02                STA   CHAR_UND_C
E6A6   0A                      ASL
E6A7   B0 03                   BCS   $E6AC
E6A9   38                      SEC
E6AA   B0 01                   BCS   $E6AD
E6AC   18                      CLC
E6AD   6A                      ROR
E6AE   91 E0                   STA   (C_ROWADR),Y
E6B0   A2 FF                   LDX   #$FF
E6B2   8E 23 02                STX   KEY_IMAGE
E6B5   E8                      INX
E6B6   8E 04 02                STX   SHIFT_FL
E6B9   A2 50                   LDX   #$50
E6BB   AD 10 E8                LDA   PIA1
E6BE   29 F0                   AND   #$F0
```

www.Commodore.ca
May Not Reprint Without Permission

```
E6C0   8D 10 E8        STA   PIA1
E6C3   A0 00           LDY   #$0
E6C5   AD 10 E8        LDA   PIA1
E6C8   0A              ASL
E6C9   0A              ASL
E6CA   0A              ASL
E6CB   10 07           BPL   $E6D4
E6CD   8C 07 02        STY   C1_STAT
E6D0   A9 3D           LDA   #$3D
E6D2   D0 07           BNE   $E6DB
E6D4   AD 07 02        LDA   C1_STAT
E6D7   D0 05           BNE   $E6DE
E6D9   A9 35           LDA   #$35
E6DB   8D 13 E8        STA   PIA1_B4
E6DE   90 0A           BCC   $E6EA
E6E0   8C 08 02        STY   C2_STAT
E6E3   AD 40 E8        LDA   PORT_B
E6E6   09 10           ORA   #$10
E6E8   D0 0A           BNE   $E6F4
E6EA   AD 08 02        LDA   C2_STAT
E6ED   D0 08           BNE   $E6F7
E6EF   AD 40 E8        LDA   PORT_B
E6F2   29 EF           AND   #$EF
E6F4   8D 40 E8        STA   PORT_B
E6F7   A0 08           LDY   #$8
E6F9   AD 12 E8        LDA   KB_ROWIN
E6FC   CD 12 E8        CMP   KB_ROWIN
E6FF   D0 F6           BNE   $E6F7
E701   4A              LSR
E702   B0 05           BCS   $E709
E704   48              PHA
E705   20 3F E7        JSR   DECODE_KBD
E708   68              PLA
E709   CA              DEX
E70A   F0 08           BEQ   $E714
E70C   88              DEY
E70D   D0 F2           BNE   $E701
E70F   EE 10 E8        INC   PIA1
E712   D0 E3           BNE   $E6F7
E714   AD 23 02        LDA   KEY_IMAGE
E717   CD 03 02        CMP   KEY_DOWN
E71A   F0 20           BEQ   $E73C
E71C   8D 03 02        STA   KEY_DOWN
E71F   AA              TAX
E720   30 1A           BMI   $E73C
E722   BD 5B E7        LDA   $E75B,X
E725   4E 04 02        LSR   SHIFT_FL
E728   90 02           BCC   $E72C
E72A   09 80           ORA   #$80
E72C   AE 0D 02        LDX   KEYCOUNT
E72F   9D 0F 02        STA   KEY_BUFF,X
E732   E8              INX
E733   E0 0A           CPX   #$A
E735   D0 02           BNE   $E739
```

www.Commodore.ca
May Not Reprint Without Permission

```
E737   A2 00                    LDX   #$0
E739   8E 0D 02                 STX   KEYCOUNT
E73C   4C 7E E6                 JMP   RETURN_INT
E73F   BD 5B E7    DECODE_KBD   LDA   $E75B,X
E742   D0 07                    BNE   $E74B
E744   A9 01                    LDA   #$1
E746   8D 04 02                 STA   SHIFT_FL
E749   D0 10                    BNE   $E75B
E74B   C9 FF                    CMP   #$FF
E74D   F0 0C                    BEQ   $E75B
E74F   C9 3C                    CMP   #$3C
E751   D0 05                    BNE   $E758
E753   2C 11 E8                 BIT   PIA1 + 1
E756   30 03                    BMI   $E75B
E758   8E 23 02                 STX   KEY_IMAGE
E75B   60                       RTS


F736   AD 05 02    UPDATE_CLK   LDA   CLOCK_2
F739   69 01                    ADC   #$1
F73B   8D 05 02                 STA   CLOCK_2
F73E   90 03                    BCC   $F743
F740   EE 06 02                 INC   CLOCK_2 + 1
F743   C9 6F                    CMP   #$6F
F745   D0 07                    BNE   $F74E
F747   AD 06 02                 LDA   CLOCK_2 + 1
F74A   C9 02                    CMP   #$2
F74C   F0 26                    BEQ   $F774
F74E   EE 02 02                 INC   M_CLOCK + 2
F751   D0 08                    BNE   $F75B
F753   EE 01 02                 INC   M_CLOCK + 1
F756   D0 03                    BNE   $F75B
F758   EE 00 02                 INC   M_CLOCK
F75B   A2 00                    LDX   #$0
F75D   BD 00 02                 LDA   M_CLOCK,X
F760   DD 88 F7                 CMP   $F788,X
F763   90 17                    BCC   $F77C
F765   E8                       INX
F766   E0 03                    CPX   #$3
F768   D0 F3                    BNE   $F75D
F76A   A9 00                    LDA   #$0
F76C   9D FF 01                 STA   $1FF,X
F76F   CA                       DEX
F770   D0 FA                    BNE   $F76C
F772   F0 08                    BEQ   $F77C
F774   A9 00                    LDA   #$0
F776   8D 05 02                 STA   CLOCK_2
F779   8D 06 02                 STA   CLOCK_2 + 1
F77C   AD 12 E8                 LDA   KB_ROWIN
F77F   CD 12 E8                 CMP   KB_ROWIN
F782   D0 F8                    BNE   $F77C
F784   8D 09 02                 STA   PIA_COPY
F787   60                       RTS
```

www.Commodore.ca
May Not Reprint Without Permission

Here is the source for the post interrupt code program

```
START       LDA  $105,X          GET
            STA  PCLO            PROGRAM
            LDA  $106,X          COUNTER AND
            STA  PCHI            STORE IT
            LDA  PROG            PUT IN YOUR
            STA  $105,X          OWN CODE
            LDA  >PROG           LOCATION
            STA  $106,X
            JMP  $E685           DO NORMAL INTERUPT
            REM  THIS CODE GOES AFTER YOUR CODE, TO RETURN
GOBACK      LDA  PCHI            RESTORE
            PHA  OLD
            LDA  PCLO            LOCATION
            PHA
            TSX
            DEX                  RESET
            DEX                  STACK
            DEX
            DEX
            TXS
            JMP  $E65E           DO RTI
```

www.Commodore.ca
May Not Reprint Without Permission

KVENICH & ASSOCIATES

International Trade Brokers

151 Carlingview Drive Unit 5 Rexdale, Ontario M9W 5E7 . Phone 675-3333
Telex 06-989100

ATTENTION: NEW PRODUCTS

KVENICH & ASSOCIATES HAVE BEEN ASKED TO DISTRIBUTE THE WAVECOM INTERFACE. DESIGNED BY THE MICRO-SYSTEMS ENGINEERING GROUP, THIS NEW PRODUCT WILL INTERFACE THE PET COMPUTER AND THE I.B.M. SELECTRIC TYPEWRITER FOR COMPREHENSIVE WORD PROCESSING.

THE WAVECOM INTERFACE IS A STAND ALONE DEVICE WHICH CONTAINS ITS OWN PROCESSOR AND ROM MEMORY. NO PROGRAMMING IS REQUIRED TO RESIDE IN THE PET'S 2nd CASSETTE BUFFER AS IS THE CASE WITH MANY INTERFACES PRESENTLY ON THE MARKET. THIS FEATURE ALLOWS THE USER THE 2nd CASSETTE FOR BUSINESS FILES OR ACCOUNTING PROGRAMS.

FEATURES OF THE WAVECOM INTERFACE

1. A stand alone interface.
2. Plugs into the PET COMPUTER via the IEEE port.
3. All parts are included with the WAVECOM interface (including plugs, and wire connectors).
4. Plugs into the SELECTRIC typewriter.
5. Installation of solenoids and plugs are required for the SELECTRIC by the user or dealer all parts are included. are provided.
6. Can be used with a SELECTRIC terminal. Communication is via a telephone handset through an acoustic coupler. No modifications are necessary.
7. Will also operate with many other computors such as TRS-80, SOCERER, and APPLE.
8. Unplug the SELECTRIC typewriter from the WAVECOM Interface and the typewriter will return to normal manual operation.
9. Allows any computor to completely control all user controlled key including, the TAB FUNCTION and the BACKSPACE key.

NEWS RELEASE

www.Commodore.ca
May Not Reprint Without Permission

HOME COMPUTER CENTRE
Computers for Home & Small Business
(416) 222-1165
222-1166
6101 YONGE STREET, WILLOWDALE, ONTARIO M2M 3W2, CANADA

HOME COMPUTER CENTRE
ANNOUNCES

THE NEW RELEASE OF PET SOFTWARE

The following programs are now officially released with complete documentation.

1. ENTRY - List Price $24.95
Used as a general purpose data entry program for business applications with user definable entry format, the program may be used for a Mail List, Daily Journial, General Ledger, Record Keeping etc. It works with cassette printer, and other IEEE devices.

2. PROCESS - List Price $24.95
General purpose data process program. It is designed for limited data processing power on the PET. Basic operation includes SORT, EDIT, DELETE, INSERT, and MACRO. The program is particularly useful for merging large amounts of data from different input sources.

3. INVENTORY - List price $24.95
Inventory control program on the PET Data includes,item #, description, quantity on hand, reorder limit and prices. It generates inventory report and low inventory report. Handle up to 60 items on the 8K PET. Data may be insert, delete, change, on the memory instantly.

All the HCC offically released programs come with complete documentation. The programs are intended for practical business applications, and special techniques are used to insure easy operation and data reliability. Special features include interactive message, error-free operation, recoverable operator errors, general I/O etc. The released programs have been tested for an extended period of time.

www.Commodore.ca
May Not Reprint Without Permission

COMPUTER COURSES FOR EVERYONE

Human Computing Resources Corporation presents an ongoing program of courses on computers.

The courses have been created in response to the growing need for an objective, non sales oriented, viewpoint on how to evaluate personal computers, microcomputers and minicomputers. They will be attractive to people from many walks of life – business people, professionals, artists, engineers, enthusiastic new users. They will be doubly attractive to people who have researched the computer market and find they lack the expertise to choose one system over another.

Being offered in spring/summer 1979 are:

- Introduction to Computing and Personal Computers (bimonthly; 9 hours)
- How to Buy a Computer for Small Business (23 May and 18 July; one day)
- Introduction to Microprocessors (23 and 30 June; 14 and 15 August; 2 days)
- Introductory Programming in BASIC (monthly; 18 hours)
- Programming in PASCAL (bimonthly, beginning in June; 18 hours)

In the works are courses on word processing, computers in the law office, and computers in medicine and in the medical office (two courses).

Our instructors are skilled educators, business people and creative computer professionals. They have had broad experience with all types of computers and computer applications.

All courses are held at HCR's offices, 10 St Mary Street, Suite 401, Toronto (near Yonge and Bloor), or in downtown Toronto hotel suites. Courses are priced at from $55 to $115. Fees for all courses are income tax deductible.

For more information, mail in the form below, or call us at 922-1937.

Please send me information about these courses:

( ) Introduction to Computing and Personal Computers
( ) How to Buy a Computer for Small Business
( ) Introduction to Microprocessors
( ) Introductory Programming in BASIC
( ) Programming in PASCAL
( ) Introduction to Electronic Troubleshooting
( ) Introduction to Computer Graphics
( ) Introduction to Word Processing
( ) Computers in the Legal Office
( ) Computers in the Medical Office
( ) Frontiers of Medical Computing

NAME ______________________________

COMPANY & TITLE ______________________________

ADDRESS ______________________________

____

____

PHONE ____________ ______

www.Commodore.ca
May Not Reprint Without Permission

NAKCOMM
SYSTEMS INC. 80 HALE ROAD, UNIT 7, BRAMPTON, ONT. L6W 3M1 • (416) 459 7616

Nakcomm Systems Inc. wishes to extend thanks to Commodore for the opportunity to offer you several new PET compatable items.

You may find these units an economical and easy way to expand the capabilities of your PET:

Mini Printer Model TC-100
40 Character Per Second
96 Character set
5 X 7 dot matrix
List Price $499.95

Full Size Keyboard Model 74-KB
List Price $199.95

32 K Byte Expansion Board
Model PME-32 List Price $912.95

SPECIAL NOTE: We also have available, an Interface Model PTP-10, allowing the PET to operate with any Centronics Printer. List Price $69.95

Orders placed directly on Nakcomm Systems Inc. will be dealt with promptly.

Our Terms and Conditions are as follows:

- Payment - Cash (cheque or money order) with order.
- Warranty - 90 Days parts and labour.
- Delivery - 1 to 3 weeks depending on item and stock.
- After Warranty Service - Done on the Nakcomm premises.

If you require any additional information on the above described units or further explanation of our offer, contact us at any time.

Yours very truly,
Nakcomm Systems Inc.

Donald R. Young,
Marketing Co-Ordinator.

Dealer Inquiries Invited