PERCOM Peripheral

Garland, Texas No. 5

FLEX, UNIFLEX, OS-9

by Harold Mauch
President, Percom Data Company

The effort we expended trying to develop 6800 software for the LFD-400 disk system rather painfully taught us that software development requires an enormous amount of time and manpower. Even the software vendors whose sole business is the creation of software do well to develop and document a single significant software package in one year! Consequently, when we began to develop 6809-based products, we decided to use outside vendor-developed system software when possible and reserve the in-house software effort for utilities, special drivers, application programs and customer support.

Since a good operating system is the foundation for both system and application software, we contracted the various software vendors. FLEX/9 by TSC was developed first but was little more than a reassembly of the 6800 FLEX. It was definitely not "FLEXible" enough for the I/0 intensive applications we envisioned for the 6809. UNIFLEX was still being developed but a review of the design specification revealed it was not practical for control applications because the memory requirement was too large and the I/0 restrictions were very "unFLEXible".

The preliminary description of OS-9 was interesting. It had the device-independent I/0 and UNIX-like features that we had come to appreciate in INDEX, plus the capability of supporting a broad range of hardware configurations ranging from simple ROM-based controllers to multi-tasking multi-user data processing systems. Although it took longer to develop than predicted, the result is well worth the wait. *OS-9 is everything we wanted in a 6809 operating system.* It was easily configured to run on a PERCOM 6809 adapter-modified 6800 computer as well as with the PERCOM SBC/9 MPU in a SYSTEM-50 configuration. Tim McKee of our technical staff performed the implementation and has provided several configurations which are set forth below.

Although most of our current and planned 6809 software development is designed for use with OS-9, the PSYMON ROM operating system and the MPX/9 disk system are still available for users with inexpensive or (uncomplex) 6809 system requirements.

CONTENTS

CONFIGURATION 1:

PERCOM SBC/9 with 2716 EPROM, LFD-400 Mini-Disk System, 48K byte memory (contiguous), SYSTEM-50 motherboard and power supply.

	Memory Map
FFFF--F800	OS-9 EPROM P1
F7FF--F7FE	Console ACIA
F7FD--F400	Room for I/O
F3FF--D000	Unused
CFFF--CC00	LFD-400 Drive I/O
CBFF--C000	OS-9 EPROMs PCM1,PCM2, & PCM3
BFFF--0000	RAM

CONFIGURATION 2:

MP-A2 CPU card with PERCOM 6809 Adapter, LFD-400 Mini-Disk System, MP-S serial I/O card, 32K bytes RAM, MP-B or MP-B2 motherboard and power supply.

	Memory Map
FFFF--E800	OS-9 All EPROMS
E7FF--D000	Unused
CFFF--CC00	LFD-400 Drive I/O
CBFF--8020	Unused
801F--8008	Free for I/O
8007--8004	Console ACIA
8003--8000	Free for I/O
7FFF--0000	RAM

CONFIGURATION 3:

MP-A2 CPU card (see note) with PERCOM 6809 Adapter, LFD-400 Mini-Disk System, MP-S serial I/O card, 48K byte RAM, MP-B or MP-B2 motherboard (see note) and p. supply.

	Memory Map
FFFF--F800	OS-9 EPROM P1
F7FF--E020	Unused
E01F--E008	Room for I/O
E007 --E004	Console ACIA
E003--E000	Room for I/O
DFFF--D000	Unused
CFFF--CC00	LFD-400 Drive I/O
CBFF--C000	OS-9 EPROMS PCM1, PCM2, PCM3
BFFF--0000	RAM

Note: The MP-A2 and MP-B/B2 boards require slight modifications that are switch selectable. They may be returned to their original configuration by switching them off.

OS-9 ON THE PERCOM LFD-400

by Tim Mckee

Microwave is delivering their LFD-400 (tm) version of OS-9! It seems to work really well.

My first and most important recommendation, however, is: *Get the DEBUG package.* OS-9 has none of the debug features we have come to know and love. All that OS-9 does is execute a program. You cannot even modify memory without using 'DEBUG'.

DEBUG is perhaps the most extensive debug package that I have seen. It allows the use of complex numerical expressions, including register values and indirect addressing to several levels.

There are many valuable machine language utilities included on the diskette. Using 'DEBUG' and these utilities, you can create diskettes that will 'boot' with the I/O addresses set to whatever value you want.

The first machine that we successfully brought OS-9 up on was -- believe it or not -- a SWTPC 6800 system with a Percom 6809 adapter installed. This was an ordinary 6800 system: MP-A2 CPU card, 32-Kbyte memory from 0-$7FFF, 8-Kbyte memory from $A000-$BFFF, MP-S card at $8004 and a DC-2 disk controller.

Incidentally, adapting an MP-A2 card with our 6809 conversion card is the least expensive way that I know of to make the transition from a 6800 computer to a 6809. The 6809 Adapter is just $69.95.

The only limitation of this system is lack of memory for large programs such as BASIC09. Even this limitation can be done away with by remapping the I/O to another address, e.g., $EXXX, and modifying the processor card slightly.

Helpful Hints:

1. To change the number of tracks that will be formatted on a diskette:

SYSTEM	USER RESPONSE	COMMENT
OS9:	load format [CR]	[CR] = carriage return
OS9:	debug [CR]	--
DB:	l␣format [CR]	␣ = space bar "character"
$XXXX 87		--
DB:	.␣.+39 [CR]	--
$XXXX 23		--
DB:	=↑TT [CR]	TT is (decimal) number of last track to be formatted.
DB:	q [CR]	q - quit DEBUG
OS9:	save <name> format [CR]	omit angle brackets
OS9:	unlink format [CR]	--

The new format utility may be renamed FORMAT.

2. The device descriptor blocks come with devices set to specific addresses. As long as you can map your terminal ACIA to its proper address once, you may change them as needed:

SYSTEM	USER RESPONSE	COMMENT
OS9:	debug [CR]	--
DB:	l␣term [CR]	␣ space bar "character"
$BC09 87		
DB:	.␣.+f [CR]	+f adds hex f to address
$BC18 F7		
DB:	= XX [CR]	XX most significant byte of new console address
$BC19 FE		
DB:	= YY [CR]	YY least significant byte of new console address
$BC1A 13		
DB:	l␣t1 [CR]	t1 is an auxillary terminal
$BC3C 87		
DB:	.␣.+f [CR]	--
$BC4B E0		
DB:	= XX [CR]	See note above for XX.
$BC4C 00		
DB:	= YY [CR]	See note above for YY.
$BC4D 13		

Changes for a parallel printer (p) or serial printer (p1) are made similarly.

```
DB:
$BDAJ 87
DB:
$BDAC E0
DB:
$BDAD 50
DB:
$BDAE 01

DB:

059:
059:
059:
059:
059:
059:
```

```
]pclock [CR]

-.¥.+b [CR]

 =XX [CR]

 =YY [CR]


q [CR]

verify term [CR]
verify t1 [CR]
verify p [CR]
verify p1 [CR]
verify clock [CR]
cobbler <drive> [CR]
```

```
        --

adds hex b to address

XX   most significant
byte of MP-T address.
YY   least significant
byte of MP-T address.

Finished changes

Primary terminal
Auxillary terinal
Parallel printer
Serial printer
MP-T
<drive>   /D0, /D1, etc.
```

When OS-9 is "booted" using the diskette you have just created, all I/O will be done through the new device addresses.

Data recovery while using the DC-2 disk controller card was very bad. In one instance, I was unable to even format a diskette. Using the same diskette and the same system I had no problems after making a modification to the DC-2 card.

It seems that SWTP used the internal data separator circuit on the Western Digital 1771 floppy disk controller chip. Even Western Digital recommends against this. The modification to correct this consists of installing a Percom SEPARATOR(tm). This inexpensive module ($29.95) will work wonders on eliminating disk Read errors. Be sure to specify that it is to go on a SWTP disk controller.

INEXPENSIVE SPEECH FOR YOUR SYSTEM-50

by Sam Campbell

Many techniques now exist for speech processing.

Cost constraints, more than any other consideration, often will determine the type of speech process that is used with a system.

This article will introduce the concept of artificial speech and describe a way to add *low-cost, high-quality* speech to your System-50 computer.

PERCOM's SPEAK-2-ME-2(tm) interface module enables a host computer to control a Texas Instruments' SPEAK & SPELL by disabling the SPEAK & SPELL controller chip and taking control of its data lines.

Handshake between units is accomplished through an 8-bit port such as the modified PIA (Parallel Interface Adapter) port on your 680X computer.

Conversion of a SPEAK & SPELL to accomodate a SPEAK-2-ME-2(tm) is semi-permanent.

Because there is a fair amount of hardware modification required, you may want to take advantage of the installation service PERCOM offers, although most customers make the installation themselves.

An article by Tim McKee elsewhere in this issue of the Peripheral describes how to connect a SPEAK-2-ME-2-modified SPEAK & SPELL to a 6800 computer.

The production of speech in the human vocal system is a matter of the vocal tract being exercised by air from the lungs. There are two kinds of excitation, periodic and random. Periodic excitation is caused by the vocal folds blowing apart and collapsing under lung pressure. Random excitation is created by air rushing over the articulators -- lips, tongue and teeth -- with the vocal folds open.

The vocal tract, which is composed of the cavities of the head (nasal and mouth), filters the input. The placement of the articulators at any instant will determine the frequency of the sound produced, as filtered by the vocal tract. This complex time-variant filter network produces human speech.

To produce speech artifically requires an analog electrical signal that can approximate the human vocal process. Using filter oscillators, and noice generators this can be done fairly well. The complication is control of the static and dynamic parameters of speech.

Texas Instruments' SPEAK & SPELL learning aid uses a process based on a voice-compression technique called linear predictive coding (LPC). This technique can generate high quality speech at a low data rate -- less than 2400 bits per second. Linear predictive coding is so called because coefficients that characterize the digital filter may be predicted from a linear combination of the previous coefficients.

SPEAK & SPELL uses a modified calculator processor (TMC 0271) as its controller. This controller, a member of the TMS 1000 family, has been modified to enhance its binary-coded decimal arithmetic and expanded instruction set. The function of the controller is to pass data to the voice synthesis processor (VSP) and to the display, and to perform the number crunching mathematics required.

The VSP enables communication with the microprocessor and voice-memory. Speech data that has been compressed is supplied to the voice-processor either from the CPU or the voice-memory. The voice-processor decodes this data to produce a time-varying digital filter model of the vocal tract. This model is excited by a representation of either the glottal air impulse (voiced sounds) or the rush of air

(unvoiced sounds). The output of this model is passed through an eight-stage digital-to-analog converter to produce a synthetic speech waveform.

The voice synthesis memory (TMS 6100) stores the speech data as read-only memory. These ROM sets are 16,384 by 8-bit devices (128 Kbits) having an internal 18-bit address counter/register and two 8-bit output buffers. Fourteen of the address bits go to the memory array directly, while the four most significant bits are used in a 1 of 16 chip select. Data is fed serially out to the voice-processor using the ADD 8 line.

IMPLEMENTING SPEECH ON YOUR SYSTEM-50

By Tim McKee

This article discusses the hardware and software needed to interface a Texas Instruments' SPEAK & SPELL, which has been modified with a PERCOM SPEAK-2-ME-2 as discussed in Sam Campbell's article, to your 6800 computer.

There are two quick and easy ways to interface your speech unit using an M6821 parallel interface adapter (PIA). One, you can use a SWTP MP-LA card set for output on channel A and input on channel B, or, two, you can use a Percom 30-pin prototyping card with voltage regulation and a PIA, and work from there.

When you order a SPEAK-2-ME-2, you will be asked which of two cables you need. Either one works. The cable is shown in Figure 1.

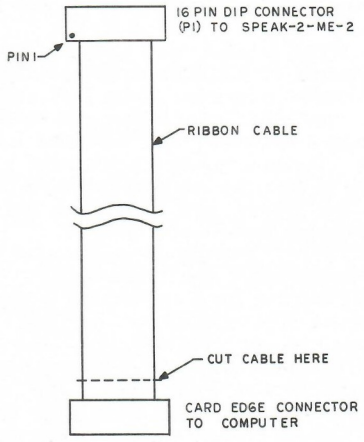


FIGURE 1

Cut off the connector at the computer end. You do not need this connector for your System-50 computer. Separate the wires of the cable to about two inches. Strip about 1/4 -inch of insulation from each wire. Position the cable assembly as shown in Figure 1. The wires numbers, from left to right, are #1 to #16. Cut off #11 and #15. Connect all of the even-numbered wires to ground. The remaining wires are connected as shown in Figure 2.

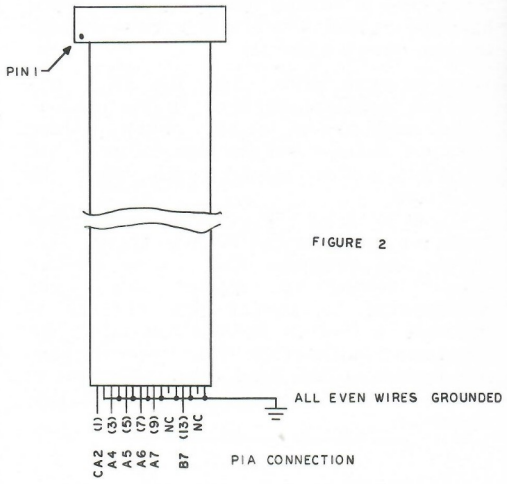


FIGURE 2

Speech driver programs are included at the end of this article. If space permits, we will include an *advanced* speech driver in the next issue of the Peripheral.

Listing # 1 is a position-independent driver for SPEAK-2-ME-2. It assumes the PIA is located at $8000 (port 0. Prior to calling the driver, you must store the word value in VALUE, which is located at the starting address 79. The code could be reassembled to accept the value passed in the Index register, on the Stack, etc. You should do a 'JSR' to the starting address. The driver will speak the word, wait for it to finish, then return. The contents of VALUE are destroyed.

Listing # 2 is a BASIC program that will allow you to install the above driver and initialize the I/0 port. There must be an area of memory inaccessible to BASIC and your DOS where the driver will be stored. This address must be given in line #170. The BASIC must have a USER function, which allows you to use machine language routines from BASIC. The address of the 16-bit value that allows you to

point the USER function at your machine language routine must be specified in line #200. Lines #330 and #340 initialize the port at $8000. If your PIA is at a different location, you must change these lines. Now, any word in the SPEAK & SPELL vocabulary may be spoken. All that you must do is load variable A with the value of the word to be spoken and GOSUB 2000. AO must point to the first byte of the routine.

SPEECH DRIVER for 6800

```
                * THIS DRIVER IS AN ADAPTATION OF THAT WRITTEN BY
                * JAMES STUTSMAN FOR THE TRS-80 MODEL I.

                * THIS DRIVER IS WRITTEN IN POSITION INDEPENDENT 6800
                * CODE.  IT REQUIRES THAT THE CALLING PROGRAM STORE
                * THE VALUE OF THE WORD TO BE SPOKEN IN THE VARIABLE
                * 'VALUE'.  THIS VARIABLE IS DESTROYED BY THE ROUTINE.

                * WRITTEN BY TIM MCKEE
                * COPYRIGHT PERCOM DATA CO.   1981


    (8000)      PIA     EQU   $8000
    (8000)      PIADTA  EQU   PIA
    (8001)      PIACTA  EQU   PIA+1
    (8002)      PIADTB  EQU   PIA+2
    (8003)      PIACTB  EQU   PIA+3


0000 8D 00      ENTRY   BSR   HERE
0002 30         HERE    TSX
0003 EE 00              LDX   0,X         GET ADDRESS OF 'HERE'
0005 31                 INS
0006 31                 INS               FIX STACK
0007 20 02              BRA   *+4         SKIP AROUND VALUE
0009 00 00      VALUE   FDB   0
000B A6 07              LDA A VALUE-HERE,X
000D E6 08              LDA B VALUE-HERE+1,X
000F A7 08              STA A VALUE-HERE+1,X
0011 E7 07              STA B VALUE-HERE,X DRIVER REQUIRES REVERSED SEQUENCE
0013 AA 07              ORA A VALUE-HERE,X OR THE TWO HALVES
0015 26 0C              BNE   SPEAK       SPEAK WORD IF NON-ZERO
0017 C6 05              LDA B #5
0019 4F         RESET   CLR A
001A 8D 2F              BSR   STROBE      SEND A ZERO
001C 5A                 DEC B
001D 26 FA              BNE   RESET       MUST SEND 5 TIMES
001F 86 80              LDA A #$80
0021 20 28              BRA   STROBE


                * THE FOLLOWING CODE WILL 'SPEAK'

0023 4F         SPEAK   CLR A
0024 8D 25              BSR   STROBE
0026 4F                 CLR A
0027 C6 05              LDA B #5
0029 8D 37      OUTADD  BSR   SHFTX4      SHIFT LEAST SIGNIFICANT NIBBLE IN
002B 8D 18              BSR   OUTCHR      SEND NIBBLE
002D 5A                 DEC B
002E 26 F9              BNE   OUTADD
```

```
0030 86 80           LDA A #$80
0032 8D 17           BSR   STROBE
0034 86 A0           LDA A #$A0
0036 8D 13           BSR   STROBE
0038 86 E0           LDA A #$E0
003A 8D 0F           BSR   STROBE
003C 4F              CLR A
003D 8D 0C           BSR   STROBE

             * WAIT FOR COMPLETION OF WORD

003F B6 8002 WAIT    LDA A PIADTB    GET STATUS WORD
0042 2A FB           BPL   WAIT
0044 39              RTS


0045 36      OUTCHR  PSH A
0046 86 20           LDA A #$20
0048 8D 01           BSR   STROBE
004A 32              PUL A
004B 43      STROBE  COM A
004C B7 8000         STA A PIADTA    PUT DATA ON LINES
004F B6 8001         LDA A PIACTA    GET CONTROL CONTENTS
0052 84 F7           AND A #$F7
0054 B7 8001         STA A PIACTA    SET STROBE LINE LOW
0057 8A 08           ORA A #8
0059 36              PSH A
005A 32              PUL A
005B 36              PSH A
005C 32              PUL A
005D 01              NOP             THIS IS A DELAY SEQUENCE
005E B7 8001         STA A PIACTA    STROBE LINE HIGH
0061 39              RTS

             * THIS ROUTINE WILL SHIFT THE LEAST SIGNIFICANT
             * NIBBLE (4 BITS) FROM 'VALUE' INTO ACC A.

0062 37      SHFTX4  PSH B
0063 C6 04           LDA B #4
0065 8D 05   SHFT4   BSR   SHFTR     SHIFT ONCE
0067 5A              DEC B
0068 26 FB           BNE   SHFT4     LOOP 4 TIMES
006A 33              PUL B           RECOVER VALUE
006B 39              RTS


006C 67 07   SHFTR   ASR   VALUE-HERE,X SHIFT MOST SIGNIFICANT BYTE
006E 66 08           ROR   VALUE-HERE+1,X SHIFT LEAST SIGNIFICANT BYTE
0070 46              ROR A           SHIFT INTO ACC A
0071 39              RTS
                     END
     00  ERROR(S) DETECTED
```

SPEECH DRIVER INSTALLER

```
0010 LINE= 0
0100 PRINT CHR$(26);:REM CLEAR SCREEN
0110 REM YOU MUST SELECT AN AREA OF MEMORY INTO WHICH
0120 REM THE MACHINE LANGUAGE CODE FOR THE SPEECH DRIVER
0130 REM WILL BE PLACED.  THIS AREA MUST REMAIN UNTOUCHED
0140 REM BY BASIC AND THE DOS/OPERATING SYSTEM.  THIS
0150 REM AREA MUST BE 114 BYTES LONG.  THIS PROGRAM AND
```

```
0160 REM THE OTHER PROGRAMS ASSUMES A LOCATION OF $A800
0170 A0=43008 : REM REPLACE THIS VALUE WITH YOUR ADDRESS
0180 REM YOU MUST KNOW THE LOCATION THAT CONTAINS THE USER
0190 REM VECTOR.
0200 A9=99 : REM ADDRESS OF USER VECTOR FOR SUPER BASIC
0210 PRINT "INSTALLING BASIC SPEECH DRIVER, PLEASE STAND-BYE"
0220 A=INT(A0/256)
0230 POKE(A9,A) : REM INSTALL MSB OF USER VECTOR
0240 A=A0-(256*A)
0250 POKE(A9+1,A) : REM INSTALL LSB OF USER VECTOR
0260 FOR X=0 TO 113
0270 READ A
0280 POKE(A0+X,A)
0290 NEXT X
0300 PRINT
0330 POKE(32769,0) : POKE(32768,255) : POKE(32769,60)
0340 POKE(32771,0) : POKE(32770,0) : POKE(32771,60)
0350 REM INITIALIZE PIA A SIDE FOR ALL OUTPUTS, CA2 HIGH
0360 REM INITIALIZE PIA B SIDE FOR ALL INPUTS
0370 A=USER(0) : REM RESET S2M2
0380 A=19991 : GOSUB 2000
0390 PRINT
0400 PRINT "SPEECH DRIVER INSTALLED, PORT INITIALIZED."
0410 PRINT "READY FOR USE !"
0430 READ A : IF A=0 THEN 450
0440 GOSUB 2000 : GOTO 430
0450 END
1000 DATA 141,0,48,238,0,49,49,32
1010 DATA 2,0,0,166,7,230,8,167
1020 DATA 8,231,7,170,7,38,12,198
1030 DATA 5,79,141,47,90,38,250,134
1040 DATA 128,32,40,79,141,37,79,198
1050 DATA 5,141,55,141,24,90,38,249
1060 DATA 134,128,141,23,134,160,141,19
1070 DATA 134,224,141,15,79,141,12,182
1080 DATA 128,2,42,251,57,54,134,32
1090 DATA 141,1,50,67,183,128,0,182
1100 DATA 128,1,132,247,183,128,1,138
1110 DATA 8,54,50,54,50,1,183,128
1120 DATA 1,57,55,198,4,141,5,90
1130 DATA 38,251,51,57,103,7,102,8
1140 DATA 70,57
1800 DATA 19513,20492,9765,20492,-16631,0
1810 REM 1800 WILL SAY "READY TO TALK TO YOU"
2000 IF A<0 LET A=65536+A
2010 POKE( A0+9,INT(A/256))
2020 POKE( A0+10,A-256*INT(A/256))
2030 A=USER(0)
2040 RETURN
```

SBC/02 MONITOR WORKS WITH SWTBUG PROGRAMS

By Phil Sanders

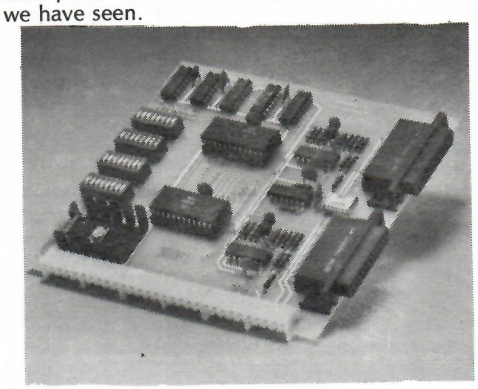
Programs written for the SWTBUG monitor can be used with the SBC/02 monitor (monitor for the 6809 version of the SBC/9) with the following transfer addresses. This is provided that the program does not make subroutine calls other than the usual (INCH, OUTCH...) as listed in Appendix D of the LFD-400 or LFD-400 EX manual. Transfer addresses can reside in either RAM or ROM from $E000 to $E3FF. These addresses being filters can add or enhance SWTBUG compatible programs by jumping to a modifying subroutine before jumping to the SBC/02 monitor's vectors.

An LFD-400/800 controller card, decoded for $E000-$EFFF instead of $C000-$CFFF, was used for a PROM board. To decode the controller, to $E000-$EFFF, (1) cut the B10 pin 3 trace (to Address 13), (2) jumper from pin 40 of the Molex bus connector (Address 13) to A4 pin 13, and (3) jumper A4 pin 12 to B10 pin 3.

```
(E1AC)          ORG     $E1AC
E1AC 7E FD33    JMP     $FD33       INCH

(E1D1)          ORG     $E1D1
E1D1 7E FD26    JMP     $FD26       OUTCH

(E0CC)          ORG     $E0CC
E0CC 7E FD9A    JMP     $FD9A       OUTS

(E06B)          ORG     $E06B
E06B 7E FD1C    JMP     $FD1C       OUTHR

(E0C8)          ORG     $E0C8
E0C8 7E FD96    JMP     $FD96       OUT4HS

(E07E)          ORG     $E07E
E07E 7E FD4B    JMP     $FD4B       PDATA1

(E0E3)          ORG     $E0E3
E0E3 7E FD65    JMP     $FC65       CONTRL

(E047)          ORG     $E047
E047 7E FDF8    JMP     $FCF8       BADDR
                END
```

NEW PRODUCTS

We introduce new products for SYSTEM-50 computers several times a year. Our latest is a dual asynchronous serial communications interface module for the 30-pin bus. This SIO card packs more features per dollar than any we have seen.



Features:

☆ Available with or without an optional on-card bit-rate generator (BRG). The BRG may be used to produce transmit/receive clock rates, thereby freeing up to five pins on the system bus for other use -- such as extended 1Mbyte addressing.

☆ Compatible with both the older and newer versions of the Systems-30 bus.

☆ Design permits control of serial-to-parallel and parallel-to-serial data conversion, parity generation and verification, and modem control -- with minimal software overhead.

☆ Output signals of each channel are current-limited for protection against a short-circuit load.

☆ Transmit and receive rate of each channel may be individually selected. Moveover, the transmit rate of a channel may be different from its receive rate. Data rate: 110, 300, 1200, 2400, 4800, 9600, and 19200 bits/second.

☆ Communications signals are compatible with RS-232-C standard levels. Clock signals are TTL compatible.

☆ Low price: Only $59.95 without the optional bit-rate generator, only $74.95 with the BRG components installed. Field upgrade BRG kit is only $19.95.

Some thumbnail descriptions of some more new products are set forth below. These briefs were prepared by Dale French of our technical staff.

M24SS Static RAM Card: A 24-Kbyte static RAM board organized into three independent 8-Kbyte blocks. Works with either the standard SS-50 bus or the 1-Mbyte extended addressing bus. Comes assembled, burned-in and tested. Users manual includes source listing of diagnostic memory test. Also available in 8- and 16-Kbyte configurations.

M48DSS Dynamic RAM Card: A low power 48-Kbyte dynamic RAM board organized into three independent 16-Kbyte blocks. Works with either the standard SS-50 bus or the 1-Mbyte extended addressing bus. Comes assembled, burned-in and tested. Users manual includes source listing of diagnostic memory test. Also available in 16- and 32-Kbyte versions.

ColoRAMa-50: A memory-mapped color VDG board. Generates alphanumerics, semi-graphic displays. Full graphic resolutions range from 64 x 64 pixels to 256 x 192 pixels. Displays in two, four or eight colors, depending on the display resolution. Two- and four-color displays may be complemented. Board is designed to accommodate a low cost Radio Shack modulator for TV set display. Comes with on Kbyte of display RAM which provides for alphanumeric, semigraphics and two low-density full graphic display formats. Also provides for 2-Kbyte EPROM. Cassette I/O provides for low-cost file storage. Users manual includes source listing of display OS. Works with 1-Mbyte extended addressing bus. The ColoRAMa-50 occupies an 8-K-byte block of memory in the upper half of a 64-Kbyte address space. Board accommodates additional RAM for higher density display modes.

The COLOR CONNECTION: A cable/circuit card assembly which is used to adapt the 6809-based TRS-80 Color Computer to the SS-50 bus. The COLOR CONNECTION allows access to LFD-400 mini-disk storage, RAM

expansion, interfacing (via the ELECTRIC WINDOW, e.g.) to a word-processing quality BW display system, etc.

SS-50 Bus Motherboard: A seven-slot system bus card that can also be used as an extender card for servicing function cards.

SS-30 Bus Motherboard Kit: Accommodates up to eight 30-pin I/0 cards. Supplied complete with PC board connectors and components required for application as an I/0 extension motherboard for the SS-50 bus.

Since the last Peripheral was issued, many new 680X programs have been released, including MPX/9, a 6809 DOS for our LFD disk systems, and a 6809 version of Percom Super BASIC.

To place an order or request product literature, call our toll-free order number, 1-800-527-1592. For additional technical information call (214) 272-3421.

Exorciser Compatible Products

Motorola's EXORciser is a versatile, industrial quality 680X microcomputer. It uses an 86-pin system bus, with standard gold-plated edge type connectors for function cards. Our EXORciser compatible boards, called ModulEX modules, are equal in quality to Motorola modules but cost far less. Included in this product line is the LFD-400/800 EX mini-disk system; a VC-EX memory-mapped video display card; and, the M64EX, a 64-Kbyte dynamic RAM card. A 6809 MPU card is in the board-layout stage of development. Software for these systems include a number of monitors -- MIKBUG, EXBUG and others -- configured for LFD-400/800 EX operation, Percom Super BASIC, an editor and business application programs.

I/0 subroutine entry addresses for EXBUG 1.1, EXBUG 1.2 and CRTBUG, plus an entry address for the MINIBUG III CONTRL subroutine have been established. These are set forth in the table below, and are to be added as an extension to Appendix D of the LFD-400/800EX Users Manual. We will soon release similar entry addresses for D3BUG1, D3BUG2, and MIKBUG 2.0. If you are interested in using the LFD-400/800EX with one of these monitors, or if you have a monitor that is not listed in Appendix D of the Users Manual, contact Percom.

ROUTINE	MIK BUG	MICRO BUG	MINI BUG II	MINI BUG III	TVBUG	EXBUG 1.1	EXBUG 1.2	EXBUG 2.1	SYS MON CMS	CRTBUG
INCH	E1AC	FD33	E11F	E133	F800	F015	F015	F015	F9D1	E508
OUTCH	E1D1	FD26	E108	E126	F803	F018	F018	F018	F9E8	E494
OUTS	E0CC	FD9A	E180	E19A	F9B4	F02A	F02A	F02A	F8EB	E427
OUTHR	E06B	FD1C	E0FE	E11C	F986	FA1C	FA0F	F0D0	F881	F3C8
OUT4HS	E0C8	FD96	E07C	E196	F80F	F01E	F01E	F01E	F8E7	F423
PDATA1	E07E	FD4B	E130	E14B	F806	F027	F027	F027	F894	E3D9
CONTRL	E0E3	FC65	E040	E065	F821	F564	F564	F564	F81C	E09B
BADDR	E047	FCF8	E0D9	E0F8	F809	F00F	F00F	F00F	F864	E3A4
I/O TYPE	PIA	ACIA	ACIA	ACIA	PIA	ACIA	ACIA	ACIA	ACIA	ACIA/PIA
ADDRESS	8004	8408	8008	8008	F404	FCF4	FCF4	FCF4	E3C0	8008/8044

A supplement to this issue of the **Peripheral** is available from Percom Data Company. The supplement includes more specific information --for example, notes on product improvement and maintenance -- and a 'short-form' product price list. This supplement, which is automatically mailed to subscribers of the **Peripheral,** may be obtained from Percom by calling our toll-free order number, 1-800-527-1592. From within Texas, call (214) 272-3421.

TRADEMARKS APPEARING IN THIS ISSUE OF THE PERIPHERAL:

Percom Data Company, Inc.: ColoRAMa-50, the COLOR CONNECTION, ELECTRIC CRAYON, ELECTRIC WINDOW, EXDOS, INDEX, LFD-400, LFD-800, ModulEX, MPX/9, PSYMON and SBC/9.

Motorola Corp.: EXBUG, EXORciser, MICROBUG, MICRO CHROMA, MIKBUG and MINIBUG.

Tandy Radio Shack Corp.: TRS-80

Microware® Systems Corporation: OS-9

Technical Systems Consultants, Inc.: FLEX, FLEX/9, UNIFLEX