WELCOME TO THE FALL ISSUE!

Where did the summer go? It seems like only yesterday that the kids got out of school and the weather got nice. Now here we are in September with the kids going back to school and the days getting shorter. I hope all of you had a pleasant summer.

Greg Morse has come through again for us with this month's lead article. Greg is the author of Basic09 Corner, our regular tutorial on Basic09. This month, in addition to his regular article, Greg has written an excellent piece on the EIA RS232 interface. Almost anyone involved with computers has had to deal with interfacing a printer or modem and has had to figure out what this mysterious RS232 standard is and what it does. Greg's knowledge and expertise really shows as he explains these mysteries. Thanks Greg!

Dave Kaleita has been working overtime adding new disks and programs to the software library. It seems like more and more of you are starting to contribute your favorite programs to the User Group Library. Great! We now have 40 finished disks ready for distribution with many more in the works. See Dave's article for the list of all the new disks and software. If you want to order any of the disks see page 14.

Of course we have all of our other regular columns such as Jim Schmidt's CoCo Advocate and George Dorner's Treasurer Report. These people have become the backbone of the MOTD and the User Group. Just a few people can make a big difference in an organization such as this. Why not consider becoming one of these people yourself. The UG needs input from more than just a handful of members. The depth of expertise out there is awsome and mostly untapped. We need your articles and input if we expect to continue as a successful organization. As of this month we will have grown to 1000 strong. Let's let that strength begin to show.

Just one final note. Those of you who subscribe to 68 Micro Journal may have read the very unkind and uncalled for remarks that Don Williams had about the OS-9 Users Group. Just what he intended to gain by these accusations is still unclear to me. Let me say, for the record, that as long as I have been involved with the User Group I have never seen a more dedicated, unselfish, and hardworking group of individuals as the officers and volunteers in this organization. It is unfortunate that those of us interested in the advancement of 68XXX products cannot at least have peace within our small world. In response to Mr. William's remarks we have given him some free advertising in this MOTD. Hopefully, this will show our intent for ALL of us to prosper.
ED

EIA RS232 Interface   by G MORSE
                                             13 July, 1985

"Standards are wonderful. There are so many to choose from".

1. INTRODUCTION

The RS 232C specification was designed to interface a modem to a CRT. The modem is called "Data Communications Equipment", or DCE, and the CRT "Data Terminal Equipment" or DTE. All RS232 signal directions are referred to the DTE. Thus pin 2, Transmit Data (TxD), means data transmitted from the DTE to the DCE.

I work for the phone company so I remember it as: "The Customer (Terminal) comes first."

The opening quotation in this column is NOT a joke. The RS232 interface suffers from two drawbacks:

1) It offers so many features that no one implements them all

2) It is not symmetrical. In particular while a DCE can prevent a DTE from sending, there is no standard way to do the reverse.

The following discussion will be limited to those parts of RS232 necessary to interface most common asynchronous terminals, computer ports and modems. Note that RS232 is a SERIAL interface, and thus has no bearing on the operation of Centronics, or other, parallel ports.

Further since this column is oriented twords micro users and their problems I will be using the descriptive abbreviations in common use rather than the more obscure standard designations.

2. OVERVIEW

A complete list of RS232 pins is given in appendix A. In this column we will mostly be concerned with those pins marked with an "*".

The first thing to notice is that the RS232 interface uses a 25 pin connector. If the connector on your device is NOT 25 pins, then you must consult the hardware documentation to determine how the pins have been assigned.

The second thing to notice is that there are many control pins defined. These control pins, if they are implented, all affect the internals of the device in some way. I will try to indicate the more common ways the manufacturers have chosen to have these pins affect their chips (UARTs) and modems.

The RS232 interface is NOT symmetrical. The DCE is expected to provide more status and control leads to the DTE than vice versa. It is in the area of control and status leads that most of the confusion occurs.

3. DTE, DCE, or BOTH?

A CRT or a serial printer is obviously a DTE, just as a modem is a DCE. But which is a computer port? The answer depends. If the computer port is driving a (serial) printer then it must behave as a DCE; if it is driving a modem then it must emulate a DTE. This dual nature of the computer port is another source of

it a DCE? To resolve this issue we adopt the following definitions:

If a device sends out (transmits) data on pin 2 then it is a DTE and should PROVIDE DTE control signals and MONITOR DCE status lines. In most cases the DTE control signals are DTR and RTS and the DCE lines are DSR, CTS, and DCD.

If a device sends data on pin 3 (RxD) then it is a DCE. Remember that all signal names are from the DTE's point of view (The customer comes first) and so if a DCE wants the DTE to receive data, the DCE must send it on pin 3, the DTE's receive data pin.

At this point we need an example.

The simplest possible arrangement, and one not often found in practice, is:

```
DTE          CABLE              DCE
TxD 2  -)------------------->-  2 TxD
RxD 3  -(-------------------<-  3 RxD
Gnd 7  ---------------------    7 Gnd
```

The cable is called a "straight through" cable for obvious reasons.

What if we wish to connect a CRT (DTE) directly to a computer port which normally talks to a modem? In that case we need:

```
CRT (DTE)        CABLE          CPU (DTE)
TxD 2  ------)------------------>---- 3 RxD
RxD 3  ------<------------------<---- 2 TxD
Gnd 7  --------------------------- 7 Gnd
```

Notice that pin 2 is crossed over to pin 3 and vice versa. This cable is the simplest possible "Null Modem" cable.

These two sketches are by way of example only. In

the real world things are more complex because of status leads.

To summarize what we have discussed thus far:

1) A DTE transmits on pin 2, receives on pin 3.
2) A DCE transmits on pin 3, receives on pin 2.
3) Two DTEs cannot communicate without a special cable. A DTE can communicate with a DCE directly.

4. STATUS AND CONTROL LEADS.

If the RS232 connection is really between a CRT and a modem then the status leads such as Clear To Send (CTS pin 5) and Data Carrier Detect (DCD pin 8) really indicate the state of the circuit, and so are important. If the circuit is simply a cable between a printer and a computer port, then the status leads are often unnecessary overhead. Thus some manufacturers omit many of the status leads.

Briefly the function of each of the "starred" leads is as follows:

Pin #	Abbreviation	Use
1	FGnd	Protective ground. Not usually required. When required it can be jumpered to pin 7 with no harm.
2	TxD	DTE transmit data.
3	RxD	DTE receive data.
4	RTS	DTE asks DCE permission to transmit. This lead is often the only one a DTE may set arbitrarily.
5	CTS	DCE gives DTE permission to send. Usually in response to RTS. Many UART chips such as the 6850 and 6551 will not transmit if CTS is not enabled (true). Since CTS follows RTS, an enabled RTS is also required in many cases.
6	DSR	The DCE has established a connection to the far end. In a full duplex environment this lead is almost redundant since DCD and CTS together provide enough status information. Most 300bps and 1200 bps modems used by micro owners are full duplex.
7	SGnd	A signal ground is always required Provides a zero volt reference for all the other leads. If it is not properly connected the voltage levels on other leads are not really meaningful and the behaviour of the circuit will be erratic.
8	DCD	The DCE is advising the DTE that The far end is sending Carrier. In a half duplex environment the DTE cannot send and should prepare to receive. In a full duplex environment, the usual case for most micro users, this signal implies that the far end and modem is on line. In the 6551 this pin enables the receiver much as the CTS pin enables the transmitter.
20	DTR	The DTE is ready to communicate. May occur at power up for CRTs and printers, or only after the software has been loaded and initialized in the case of computers. Most modems interpret a loss of DTR as a signal to clear the call, release the phone line and hang up. Many modems will not respond at all if DTR is not ON. Thus the DTE cannot toggle this lead as a means of flow control(*).

(*) Note: "Flow control" means the ability of the receiver to shut up the transmitter for a period of time, (perhaps because of lack of buffer space, or because the human asked for a pause), and then to allow the transmitter to resume. The two most common means of flow control are the use of RS232 leads and sending DC1(XON) and DC3(XOFF) characters.

5. UARTS and BIT BANGERS

In order to understand how the RS232 signals affect the operation of a CRT or computer port, we must look a bit behind the EIA interface at the internals of the equipment.

There are two approaches to handling the chores of parallel to serial conversion, monitoring RS232 status leads and so forth: 1)special purpose hardware and 2)software.

Special purpose hardware usually means a chip or UART In the 68xx community the most common UARTS are Motorola's 6850 and Rockwell's 6551.

A software implementation often consists of some general purpose hardware such as a transistor, and some special software. The cost of this approach may be low, but the overhead in terms of CPU cycles is high. For a multi-tasking operating system as OS9 this approach is not ideal. Since the software implementation is completely at the discretion of the manufacturer and is tailored to his hardware it is often non standard. It is impossible to discuss these implementations without actual experience of them, and I have none on the version used on the CoCo. The software approach is nick-named "Bit Banger" because a program is constantly turning bits on and off.

In addition to the RS232 connector, and the UART (or bit banger) there is one other link in the chain which is important.

Most computers, and UARTs, work with voltage levels between 0 and 5 volts. The RS232 spcification calls for levels between -25V and +25V. Further any level between -3.5V and +3.5V means nothing to RS232. It is neither a 1 nor a 0. What this means to us is that in addition to a UART the manufacturer must provide some chips to convert 0V-5V into -12V to +12V. Most UART based boards provide these chips, many computers with bit banger ports do not.
(RS232 does not require 25V, most computers use 12V).

6. ACIAs.

The term UART stands for Universal Asynchronous Receiver and Transmitter. It has come to mean any chip which does the conversion between the serial bit stream on the RS232 interface and the parallel byte used by the micro. The two most common UARTs used by 6809 users, the 6850 and the 6551, are called Asynchronous Communication Interface Adapters, or ACIA for short. These ACIAs connect, through voltage conversion level shifters, to both the data and status/control leads of the RS232 interface.
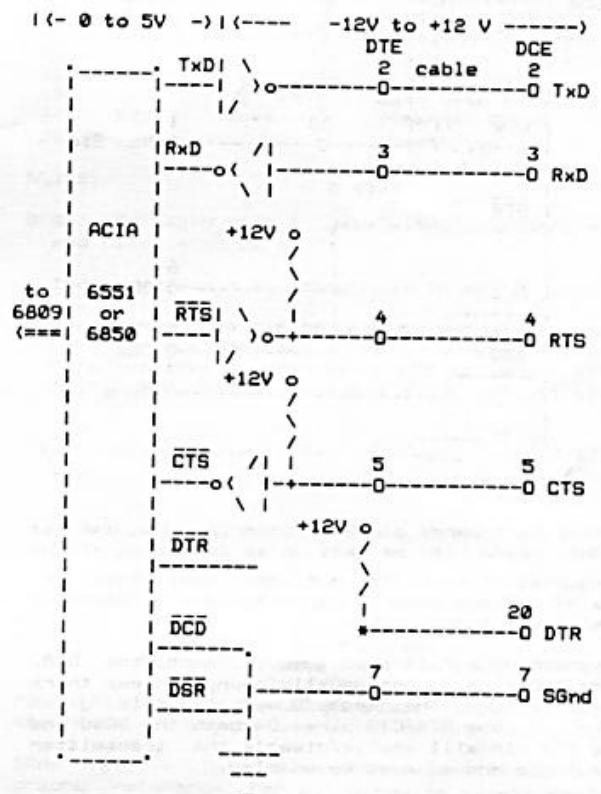
The pins on the ACIA are given the names of the RS232 lead they usually connect to. Thus on the 6850 we find leads labelled CTS, DCD, and RTS. The 6551 has in addition DTR and DSR. Since ACIAs are usually found in DTEs, the RxD, CTS, DCD, and DSR pins are inputs and the TxD, RTS, and DTR pins are outputs. Naturally the manufacturer of the serial board is not REQUIRED to connect the DTR output of the ACIA to pin 20, he might chose to connect it to pin 6 (DSR) instead, or not to use it at all.

At this point let's consider another example: (see next figure)

This figure shows a DTE computer port (or CRT) communicating with a DCE (computer port or modem). It also shows the level changing and the ACIA chip. Note that because of the action of the level shifters the status leads at the ACIA pins are the inverse of what

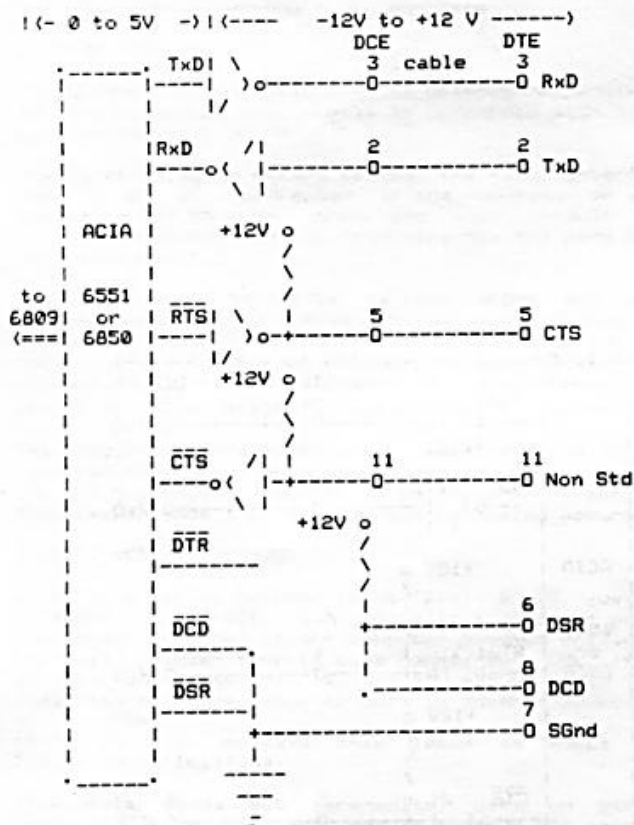they are on the RS232 Connector.

The unused inputs are grounded which has the effect of enabling them as far as the ACIA is concerned. The DTR output is not used, instead pin 20 is enabled whenever the power is turned on. The RTS and CTS pins are connected to +12V in such a way that they are enabled unless specifically disabled.

```
 |(- 0 to 5V  -)|(----    -12V to +12 V ------)
                             DTE          DCE
 .------. TxD| \             2  cable      2
 |------|    |----| )o--------0----------O TxD
 |      |    |  1/
 |      |RxD  /|            3           3
 |      |    |----o( |--------0----------O RxD
 | ACIA |    |    \ |
 |      |    +12V o
 |      |          /
to |    |          \
6809|6551|  ___    /
(===|  or |  RTS| \  |        4           4
 |  |6850 |----| )o-+------0----------O RTS
 |  |     |    |  1/
 |  |     |    +12V o
 |  |     |          /
 |  |     |          \
 |  |     |          /
 |  |     | CTS  /| |        5           5
 |  |     |----o( |-+------0----------O CTS
 |  |     |    \ |
 |  |     |   ___    +12V o
 |  |     |   DTR    /
 |  |     |--------  \
 |  |     |          /
 |  |     | ___      |        20
 |  |     | DCD      *------------O DTR
 |  |     |--------.
 |  |     | ___    |        7           7
 |  |     | DSR    |--------0----------O SGnd
 |  |     |--------|
 |  |     |        |
 .------.  ------
             ---
              -
```

The ACIA pins are connected to the RS232 pins of the same name. If the ACIA were installed in a DCE type port this would not be the case. For example see the next figure.

Notice that the ACIA "RTS" in this next case is used simply as a general purpose output to raise and lower CTS. Notice too that the CTS input is driven from pin 11, a non standard, but quite common usage.

The configuration below is a DCE because the output data is on pin 3, and the input on pin 2.

```
!(- 0 to 5V -)!(----    -12V to +12 V ------)
                     DCE          DTE
        .-------. TxD! \     3 cable   3
        !       !----! )o--------O----------O RxD
        !       !     !/
        !       !
        !       !RxD   /!       2          2
        !       !----o( !--------O----------O TxD
        !       !      \ !
        !       !
        ! ACIA  !    +12V o
        !       !         /
        !       !         \
  to  ! 6551  ! ___      /
68091  ! or    !RTS! \    !   5          5
(===!  6850  !----! )o-+------O----------O CTS
        !       !     !/
        !       !    +12V o
        !       !         \
        !       !         /
        !       ! ___   /!  !    11          11
        !       !CTS  /! !  !
        !       !----o( !-+------O----------O Non Std
        !       !      \ !  !
        !       ! ___    +12V o
        !       ! DTR       /
        !       !--------   \
        !       !           /
        !       !          !       6
        !       ! DCD      +------------O DSR
        !       !-------.   !
        !       !       !   !       8
        !       ! DSR   !   .------------O DCD
        !       !       !           7
        !       !-------!   +--------------------O SGnd
        !       !       !
        .-------.   ------
                     ---
                      =
```

Most serial port cards contain jumpering options to allow the board to be set up as in either of the above figures. If such is not the case then the transposing of the pins is usuallly done by means of a special "null modem" cable.

In an asynchornous full duplex environment the DTR, DSR, and DCD pins do not usually change unless there is a power failure. The "Ready/Busy" status is given by means of the RTS/CTS pins. On both the 6850 and 6551 the CTS pin will enable/disable the transmitter hardware logic and so must be enabled.

On the 6551 the DTR and RTS bits in the internal registers must have certain settings, but these settings are not always reflected on the RS232 connector.
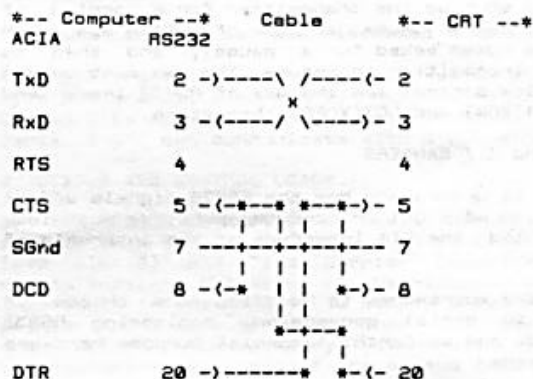
In summary to properly option a serial card, one should know a bit about how the ACIAs are being used at both ends of the cable.

7. NULL MODEM CABLES

There are many times when it is neither convenient nor possible to change the connections between the UART and and RS232 pins. In these cases the reconfiguring is done in the cable which runs between the two pieces of equipment. Because such cables often end up simulating two modems back-to-back they are called "NULL MODEM" cables.

These cables generally perform two functions, cross one pin to another (EG TxD to RxD) and jumper two pins together.
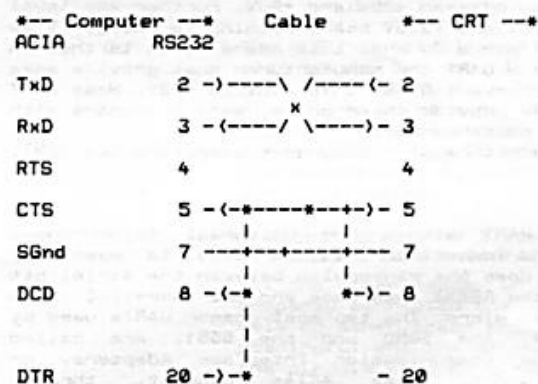
Example 1: Computer (DTE) to CRT (DTE)

```
*-- Computer --*    Cable      *-- CRT --*
ACIA    RS232
TxD       2 -)----\ /----(- 2
                   x
RxD       3 -(----/ \----)- 3
RTS       4                4
CTS       5 -(-*--* *--*-)- 5
              ! ! ! !
SGnd      7 ---+-+-+-+--- 7
              ! ! ! !
DCD       8 -(-*  ! !  *-)- 8
                  ! !
                  *-+-*
                  ! !
DTR      20 -)------*  *-(- 20
```

This simple cable performs two functions:
1) It crosses the transmit and receive data leads so that there is no conflict
2) It uses the high level output on the DTR lead to enable the ACIA transmitter on the other end of the circuit.

Thus the computer ACIA transmitter can send whenever the CRT is powered on. (The above is true if the CRT outputs DTR, which is the usual case).
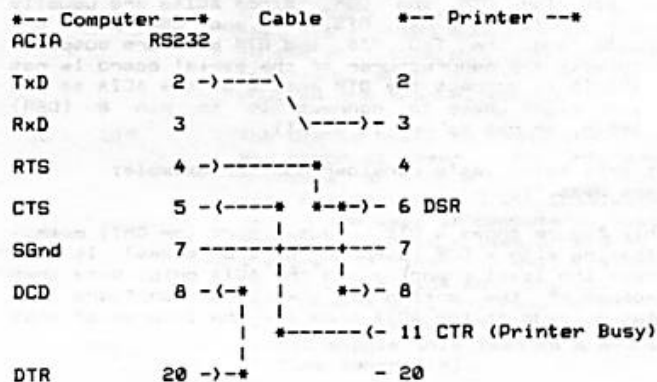
If the CRT does not output DTR (unlikely) then a cable could be made up as :

Example 2: Computer (DTE) to CRT (DTE with no DTR)

```
*-- Computer --*    Cable      *-- CRT --*
ACIA    RS232
TxD       2 -)----\ /----(- 2
                   x
RxD       3 -(----/ \----)- 3
RTS       4                4
CTS       5 -(-*----*--+-)- 5
              !    !  !
SGnd      7 ---+--+----+--- 7
              !  !
DCD       8 -(-*  *-)- 8
              !
              !
DTR      20 -)-*       - 20
```

In which the CPU uses its own DTR to enable itself and the CRT.

Example 3: Computer (DTE) to Printer

```
*-- Computer --*    Cable      *-- Printer --*
ACIA    RS232
TxD       2 -)----\        2
                    \
RxD       3          \----)- 3
RTS       4 -)-------*      4
                     !
CTS       5 -(----*  *-*-)- 6 DSR
                  !  ! !
SGnd      7 ------+----+--- 7
                  !  ! !
DCD       8 -(-*  !  *-)- 8
              !   !
              !   *------(- 11 CTR (Printer Busy)
DTR      20 -)-*        - 20
```

In this example the printer can prevent the CPU from sending by dropping pin 11. The CPU raises the printer's DSR and DCD inputs in case that device needs those pins enabled.

By now the idea behind null modem cables should be fairly clear.

Example 4: Common All-Purpose Null Modem Cable

```
*-- DTE # 1 --*     Cable       *-- DTE # 2 --*
ACIA     RS232

TxD ------- 2 -)----\ /----(- 2 TxD
                     x
RxD ------- 3 -(----/ \----)- 3 RxD

RTS ------- 4 -)----\ /----(- 4 RTS
                     x
CTS ------- 5 -(----/ \----)- 5 CTS

Vss         7 --------------- 7 SGnd

DCD ------- 8 -(----\ /----)- 8  DCD
                     x
+12 ------- 20 -)----/ \----(- 20 DTR
```

This cable assumes that the ACIA has one general purpose output, RTS, and one general purpose input CTS. Crossing them allows each device to control the other. The DCD input is used as a general indication of power on at the far end. With the ACIA driver supplied by MW it is not adviseable to use this input since a low on DCD can hang the system.

8. EIA Levels (Again)

As stated earlier the RS232 standard calls for levels to be either (-25V to -3.5V) or (+3.5V and +25V). A level between (-3.5V and +3.5V) including zero is deemed to be too weak a signal to have any meaning.

This means that a computer output of +5V can simulate an RS232 ON condition but an output of 0V will not simulate an OFF. Many "Bit Banger" ports use a bit in

a PIA which they toggle between +5 and zero volts. Such ports will not work with most equipment.

There is a further curve ball in RS232 levels. On all of the control and status leads a voltage greater than +3.5V is an ON condition and is usually represented in the UART's status register by a 1. A voltage less than -3.5V is an OFF and is represented by a zero. For the TXD and RXD leads this is reversed i.e. a voltage less than -3.5V is a one bit (MARK) and a level greater than +3.5V is a zero bit (SPACE).

Fortunately the UART manufacturers take all that into account when they design their chips, so we don't have to worry about it, (unless we are writing a bit banger driver!) but it does explain why in the UART data sheets the control leads have a bar over their names while the data leads do not.

9. SUMMARY

To apply all of this background to a given piece of equipment requires that we be marginally familiar with some internal requirements of both its hardware and software. We have to know in other words which pins the device is expecting to see, and which ones it uses to control its opposite number.

Exerpts from the RS232 Standard

RS 232-C Pin Designations

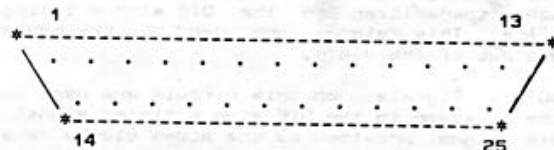| Pin #	Common Abbr.	Direction DTE	DCE	Name	Offcl Abbr. (@)
* 1	FGnd	....		Prot (frame) Ground	AA
* 2	TxD	========>		Transmit Data	BA
* 3	RxD	<========		Receive Data	BB
* 4	RTS	========>		Request To Send	CA
* 5	CTS	<========		Clear To Send	CB
* 6	DSR	<========		Data Set Ready	CC
* 7	SGnd	....		Signal Ground	AB
* 8	DCD	<========		Data Carrier Detect	CF
9				Resvd for Testing (+V)	
10				Resvd for Testing (-V)	
11				Unassigned (see NOTE %)	
12		<--------		Secondary DCD	SCF
13		<--------		Secondary CTS	SCB
14		-------->		Secondary TxD	SBA
15	TxClk	<========		Transmit Clock	DB
16		------->		Secondary RxD	SBB
17	RxClk	<========		Receive Clock	DD
18				Unassigned	
19		-------->		Secondary RTS	SCA
*20	DTR	========>		Data Terminal Ready	CD
21		<========		Signal quality detect	CG
22	RI	<========		Ring Indicate	CE
23		<========		Data Rate Selector	CH/CI
24	XClk	========>		External Clock	DA
25				Unassigned	

NOTES:

@ Grounds begin with A, Data with B, Control with C and Clock with D.

* Indicates a pin in common use in asynch circuits

% Unassigned pins can be used by mutual agreement Pin 11 is often used as "CTR" or Clear To Receive. This provides a means for a DTE to shut off data from a DCE, in the same way a DCE can use CTS.

```
   1                                    13
 *----------------------------------------*
  \  . . . . . . . . . . . . .  /
   \                            /
    *----------------------------------------*
   14                                   25
```

Male connector. Pin side view.

Signal Descriptions
Edited Extracts from the Standard

PGnd (AA) - Conductor to be electrically bonded to the frame. It may be connected to external grounds.

SGnd (AB) - This conductor establishes a common ground reference for all circuits except protective ground.

TXD (BA) - Signals on this circuit are generated by the DTE and are transferred to the DCE for remote transmission. The DTE shall hold this signal in a MARKING condition when no data are being transmitted or during the intervals between characters and words. In all systems the DTE shall not transmit unless an ON condition is present on all four of these circuits:
RTS (CA), CTS (CB), DSR (CC), DTR (CD).

RXD (BB) - Signals on this circuit are generated by the receive section of the DCE in response to data received from the remote transmitter. This circuit shall be held in the MARKING condition whenever circuit DCD (CF) is in the OFF state.

RTS (CA) - On one way channels or full duplex channels an ON condition maintains the DCE in the transmit mode. If RTS is turned OFF it shall not be turned ON again until CTS has been turned OFF by the DCE.

CTS (CB) - This circuit indicates whether the DCE is ready to transmit data. An OFF condition is an indication to the DTE that it should not transmit data. CTS is usually a delayed response to RTS provided that DSR is already on.

DSR (CC) - AN ON condition indicates that a) the DCE is connected to a communication channel (off hook) and b) the data set is not in talk, test, or dial mode and c) the local data set has completed any timing functions required by the phone system and any answer tone generation. [In other words the data set is free for data].

DTR (CD) - An ON condition prepares the DCE to be connected to the communication channel, and maintains a connection established by external means (eg manual dial). AN OFF condition causes the DCE to be removed from the communication channel. In switched applications if DTR is turned OFF it shall not be turned ON again until DSR has been turned OFF.

RI (CE) - An ON condition indicates that a ringing signal has been received on the communications channel. It shall toggle between ON and OFF approximately coincident with the ringing/silent periods of the ringing signal.

DCD (CF) - The ON condition shows that the DCE is receiving a signal which meets its suitability criteria as established by the manufacturer. [This signal is formally called Received Line Signal Detect not Data Carrier Detect]. On half duplex channels this signal shall be held OFF whenever RTS is ON.

(CH/CI) - An ON condition selects the higher of two rates in the case of dual rate DCEs.

XClk (DA) - Signals on this circuit are used to provide the transmit section of a synchronous DCE with a timing signal. The ON to OFF transition is normally in the center of any data pulses on the TXD circuit.

TxClk (DB) - Signals on this circuit are used to provide the transmitter in the DTE with a timing signal. [i.e. This signal provided by the modem clocks data out of the USRT].

RxClk (DD) - Signals on this circuit are used to provide the receiver in the DTE with a timing signal. [i.e. This signal provided by the modem clocks data into the USRT].

Treasurer's Report
George Dorner

The balance in account 381244-3 at St. Paul Federal Bank in Chicago was $6,667.68 on June 7. A spreadsheet of this year's balances will be sent to those who provide an SASE to me at P.O. Box 8251, Rolling Meadows, IL 60008.

I get to see most of the mail that is sent to the Des Moines Users Group address. Lately lots of mail has included kudos and thanks to Tim and Lori Grovac for the improvements in the MOTD in the past few issues. Because of the enormous amount of time it takes to do a good job on production of the MOTD, we are now paying the Grovacs a modest amount to edit and publish the newsletter. I will include some details next month, together with some beginning attempts at some longer term financial planning for the Group which Dale and others have been discussing with me.

As our organization matures, we will need to make some more moves away from the heretofore total dependence on volunteer efforts. If you have ideas on what additional or improved services and support activities should be included in our fiscal planning, drop me a note at the address listed above.

It is also clear from the correspondence we get, that there are lots of willing volunteers and an enormous pool of technical expertise out there. The officers have often lamented that we can't find a better way to take advantage of the offers and talents we have. That seems to remain our biggest problem.

This was reinforced by our experiences at the recent RainbowFest in Chicago. As Dale indicated in the last MOTD, we decided to take a room at the hotel to give the Group a presence there, rather than paying the stiff fee (even at a nice break it would have been $350) for a booth. This worked out well and was a lot less wearing for those of us holding forth there. We had a Dragon (made to look like a COCO) and a 5 mB hard drive in the room, courtesy of the fellows at ITMS, Inc. of West Chicago. This allowed us to copy and demo some UG software, view some neat software which is in the works by members, and answer questions from beginners and the curious who wandered in and out. I also had a terminal with which I called a Gimix by phone to demo the fact that Byron Blanchard of Ratio Metrics has solved the "hang-up problem", the problem he took on at the Des Moines meeting. Including supplies and the room and phone charges, we spent a little more than $200, took in a dozen or so new members, did a bunch of renewals, fulfilled some diskette orders, and resolved some membership and address problems. The fun part was meeting and chatting with OS-9/COCO users from all around the country and Canada and the world.

Instead of renting the booth, we put the money in an ad in Rainbow Magazine for a three month trial. It has already appeared and seems to have increased the mail at Des Moines. This cost us $855. We will watch the results and will probably follow up with trials in 68 Micro Journal, Computer Shopper, Dr. Dobbs' or BYTE, and one of the industrial magazines. Your advice on these efforts will be welcomed, too.

I had three interesting OS-9 contacts of an international nature since the last MOTD appeared. I have corresponded a couple of times with Hiroshi Nishiwaki who edits the very slick newsletter "OS-9 News" for Seikou Electronics Co., Ltd. in Japan. Seikou is the OS-9 "implementor" for Japan and this newsletter is a very nice sixteen page magazine with an intriguing mix of Kanji characters and English headings and technical words. There are many interesting articles and reports on OS-9 software in Japan in the five issues which Hiroshi sent me. (He is also responsible for the logo on the masthead of the MOTD.) It is clear that there are some nice commercial products there for which we don't yet have parallels. These are for the Fujitsu, Hitachi, and various 68000 machines. I hope we can reprint (after translation) some of the articles and notes. How about the one comparing IBM disk format with that of OS-9? Or something about Prolog under OS-9. Or LispO9!? Or LAN software for OS-9. We'll get some of this ready for future MOTDs.

Another interesting contact was the correspondence, followed by a phone call, followed by an eyeball visit with Mr. G. Fontaine of the Laboratoire de Physique Corpusculaire of the College de France. He was at O'Hare on the way to and from Des Moines (setting up a license for OS-9/68K) and Madison (attending a conference of particle physicists). Before he left I gave him some copies of software to take back for use in their lab where OS-9 and OS-9/68K are busy helping them count particles.

At the RainbowFest we visited with Jackie Cockinos and his father of Paris Radio Electronics in Kingsford, NSW, Australia. They have been operating an extensive BBS for over three years under UniFlex (wuzzat?) and have recently converted it to OS-9. Jackie will contribute a copy to the UG and plans to distribute it here are under way. He also ordered the whole UG library and should soon have it for helping out our software distribution Down Under.

That's it for this month. Hope you had a pleasant summer.

COCO ADVOCATE
Jim Schmidt

Hello again. Welcome to my column.

Back when Dale Puckett asked me to do this column it was believed that the influx of Coco users to the UG would cause a problem. It was reasonable back then to guess that most Coco users would be novices to computers and require a lot of help getting up to speed. Accordingly I was charged to be an advocate for the Coco users in the big bad world of OS9.

As an advocate I was to direct my energies and this column to helping these forlorn Coco folks keep their collective heads above water and steer them thru the white water and a whole bunch of other analogous phrases all of which say the same thing: HELP THE COCO USER!!!

Now some seven columns later, I find that the Coco user has no need of these services. At least, none of my appeals to them has produced any great surge of problem mail. On the other side of this coin, none of my appeals to the OS9 gurus has produced any offers of help either.

While I'm NOT REALLY saying, "a pox on both your houses", there does seem to be no real need for this column. What few software or hardware releases occur are quickly reviewed by the officers in their columns. MOTD has all the tutorials it needs on all the subjects of interest to you. What am I to advocate then?

I'll probably continue to write about computing as it is a mighty big part of my life. But, I will so do only when it makes sense. It no longer makes sense in these pages. So, unless and until someone can come up with a valid reason for my writing this column, I will cease and desist.

I leave you in good hands...

EOF

Editors note: This column was as much as suprise to me as it must be to all you. Jim has done a very fine service to the OS-9 community by writing the COCO Advocate over the last year and I offer a hearty thanks from all of us in the UG. Unfortunately, I don't agree with Jim's opinion that there are no special subjects that need to be covered about the COCO. How about all you COCO users? What subjects would you like to see covered? How about writing an article yourself? Maybe you could even write a regular column for the MOTD... Think about it, we have a void to fill.

OS-9 U.G. SOFTWARE COMMENTARY
by Bert Schneider

Although volumes one and two are very large, this month's commentary will be on the short side. Eric Williams did a fantastic job with Introl C. Volume #1: Spelling Checker includes four utilities which enable the user to check the spelling of text files and to look up words in the dictionary (Volume #2). The dictionary itself has 80,000 words compressed using a compression algorithm written by Eric Williams. The other two utilities, "words" and "unwords" are used to decode (uncompress) and to create a compressed dictionary file from an alphabetized list of words.

Originally, there was a problem with the format that the dictionary was copied onto for the library. The dictionary volume came only in standard OS-9 format. Thus, Color Computer OS-9 users were unable to read these disks. However, with the help of D.P. Johnson's SDISK, I was able to read the standard format. Dave Kaleita has informed me that all future disks will be in the proper form that you specify on your order request (i.e. CoCo or standard).

I find that this dictionary is much easier to use than the SpellStar spelling checker associated with CPM's Wordstar. I especially like the look-up feature ("dict"). With I/O redirection, you can have all misspelled words sent to a file onto disk for later reference. I suppose that a hard disk would make the program move along a great deal faster, but with the way I spell, this program is spellbinding!

Just as a side note, are there any Color Computer OS-9 users out there using WordPak for a device driver that have modified the CCIO device driver to allow graphics as well? I renamed the CoCo /TERM to /VERM and loaded both CCIO and /VERM modules in memory. I was able to direct text to /VERM but my keyboard hung up after that. I imagine that there is a contention for the keyboard since both drivers are expecting some input via the keyboard. I would just like to be able to use graphics and 80 columns at the same time (since I have two monitors). In this case, we should be able to have our cake (80 columns) and eat is too (graphics)! I will leave on that note. Perhaps someone will take the challenge and come up with a new CCIO driver. If Coloware can come up with "COCOMAX" for the Color Computer, someone somewhere must be coming up with some neat OS-9 graphics packages. I see this as an endless avenue of expansion. 'Till next time......

BASIC09 Corner   By: Greg Morse   1 Sep 1985

The last column dealt with some of the more straight-forward aspects of the RUN statement. In this column we will cover some of its more advanced aspects. These are:

1. Variable procedure names
2. Recursion
3. Non BASIC09 procedures

Point number 1 first. The statement "RUN SomeProc" is exactly equivalent to the pair of statements:

```
ProcName$="SomeProc"
RUN ProcName$
```

That is, the procedure to be run can be a variable of type string.

Another slightly more complicated example:

```
PROCEDURE RunString
ru$="Hello"
count=3
ru2$="Hello2"
ru3$="Hello2(count)" \REM running ru3$ will give err
RUN ru$          \REM this will work
RUN ru2$(count) \REM this is the correct way
RUN ru3$         \REM This statement will give an error
END

PROCEDURE Hello
PRINT "hello world"
END
PROCEDURE hello2
PARAM count:REAL
FOR i=1 TO count
PRINT "Hello2 world"
NEXT i
END
```

The above example shows how to RUN a string variable with parameter passing. This feature would be quite useful in menu driven systems.

This next point is slightly complex so read it carefully. Since the procedure to be RUN can be a variable name, then this variable can be passed as a parameter to a "ProcRunner" procedure. Say you want to write a procedure to integrate a function. You don't know which functions the user is going to want to integrate, so you make the function name a variable which is passed to the integrate routine. Here is how it would work:

```
PROCEDURE ProcRunner
(* Simple example to show:

(* 1. The procedure to be run can be a variable
(* 2. The variable holding the procedure name
(* can be passed to another procedure
(* this allows implementation of the DEFN
(* statement of some other basics
DIM EvalProc$:STRING
DEG
PRINT SIN(60.),COS(60.),TAN(60.)
RAD
theta=PI/3.
(* what follows are variable proc names
EvalProc$="Eval"
psin$="SinProc"
pcos$="CosProc"
(* do something complicated to sine function
(* such as integrate
RUN EvalProc$(result,theta,psin$)
PRINT "sin of 60.0= "; result
(* do same to cosine function
RUN EvalProc$(result,theta,pcos$)
PRINT "cos of 60.0= "; result
END

PROCEDURE EvalProc
(* normally this would be some complex
(* process such as integrating or root
(* finding done on some arbitrary function
PARAM value,angle:REAL; func:STRING
RUN func(value,angle)
(* more fancy processing using value returned
(* by RUN statement
END

(* now of course we must supply the procedures
(* to be evaluated
PROCEDURE sinproc
PARAM result,angle:REAL
result=SIN(angle)
END
PROCEDURE cosproc
PARAM result,angle:REAL
result=COS(angle)
END
```

Well I think that beats that subject to death. On to point 2, using the RUN statement to implement recursion.

Recursion is worth at least a column on its own. It is a very powerful programming tool. Here I'll provide one example, taken from the B09 book, and point out how essential the RUN statement is.

```
PROCEDURE HANOI
REM By T.F. Ritter
REM Tower of Hanoi Game
REM See Grogano "Programming in PASCAL"
REM See also BYTE Oct 1980 pg 279

PARAM DiscCount:INTEGER
PARAM FromPeg,ToPeg,WorkPeg:STRING[8]

IF DiscCount > 1 THEN
    (* Recursion happens when HANOI runs itself
    RUN HANOI(DiscCount-1, FromPeg, WorkPeg, ToPeg)
    (* above RUN moves a stack from FROMPEG to WORKPEG
    (* move #1 see fig 2
    PRINT "Move # ";DiscCount; " From ";FromPeg;
    PRINT " To ";ToPeg
    (* above print moves bottom disc to TOPEG (move #2)
    (* now move stack from WORKPEG to TOPEG
    RUN HANOI(DiscCount-1, WorkPeg, ToPeg, FromPeg)
ELSE
    (* this next line stops the recursion
    (* all recursive routines must do this somehow
    PRINT "Move # ";DiscCount;" From "; FromPeg;
    PRINT " To "; ToPeg
ENDIF
END
```

Now just type:
```
RUN HANOI (5,"LEFT PEG","RIGHTPEG","MID PEG")
```
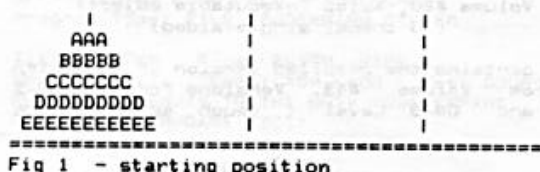and stand well back!!

```
         AAA           I            I
        BBBBB          I            I
       CCCCCCC         I            I
      DDDDDDDD         I            I
     EEEEEEEEEE        I            I
================================================
Fig 1  - starting position


         I            I            I
         I           AAA           I
         I          BBBBB          I
         I         CCCCCCC         I
         I        DDDDDDDD    EEEEEEEEEE
================================================
Fig 2 - position after move #2
```

For a detailed discussion of this problem see the book by Grogano. Let me just say that "moving a stack" is not a legal move so the program needs a routine to accomplish the stack move. The ideal routine turns out to be itself since it is designed to .move tower of HANOI stacks! You could not use a GOSUB because every time HANOI is run it is with a different arrangement of discs and pegs.
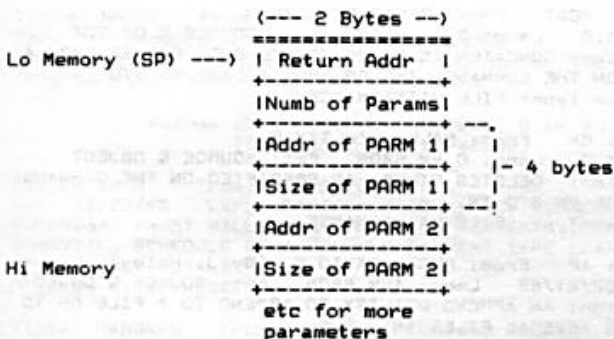
Because OS9 uses tree structured directories many OS9 utilities are recursive, for example DSAVE and the HDIR utility on the UG#0 disk.

Now for any of you still with me ONWARD TO THE HARD STUFF!!! Calling machine language (ML) subroutines from B09.

I am going to cover this subject again when we discuss the structure of OS9 modules. For now we just have to keep in mind the following points:

a) When a RUN statement is issued BASIC09 sets up the stack in a very particular way. If the procedure being RUN understands BASIC09's way of doing things then it can receive info from, and pass results to, the B09 calling program.

The stack structure is explained in detail in the B09 manual, but briefly it looks like:

```
                    (--- 2 Bytes --)
                    ================
Lo Memory (SP) ---) I Return Addr  I
                    +--------------+
                    INumb of Params|
                    +--------------+ --,
                    IAddr of PARM 1I   I
                    +--------------+   |- 4 bytes
                    ISize of PARM 1I   I
                    +--------------+ --!
                    IAddr of PARM 2I
                    +--------------+
Hi Memory           ISize of PARM 2I
                    +--------------+
                    etc for more
                    parameters
```

The B09 program does not need to know this but a ML subroutine must keep this stack structure in mind.

b) No B09 program can control the registers in the machine. These are under control of B09 itself. So all communications must be done through memory areas provided by the main program. The way to provide these is of course with a DIM statement. Since most ML subroutines use registers a lot it turns out to be convenient to set up an area in the B09 calling program to hold all the registers as follows:

```
(* Define offsets
TYPE RegAreaTyp: CC,AccA,AccB,DP:BYTE;
                 Xreg,Yreg,Ureg,PC : INTEGER
(* Now reserve storage
Dim Regs : RegAreaTyp
```

The variable "Regs" now has room to store all registers except the stack pointer. (This is just convenience. We could store the stack pointer as well.)

Now we can RUN a ML routine to get the registers and copy them into our structure so we can look at them.

For example:

```
RUN GetRegs(Regs)
Print "AccA= ";Regs.AccA; " AccB= ";Regs.AccB
etcetera.
```

The trick is that the ML subroutine must know about the stack setup, and also know what our structure looks like.

Well I almost want to appologise to all you novices who have stuck with me so far. Congratulations! As a consolation prize if you have any questions on these last couple of columns please write to MOTO or to me. I'll do my best to explain any points which are not crystal clear. The MOTO address is on the cover and my address is:
Greg Morse 10871 Roseland Gate Richmond B.C. Canada

SOFTWARE EXCHANGE NEWS
By Dave Kaleita

Upon reading the Catalog listing in the May, 1985 issue of MOTO, I noticed a typographical error: Volume #22 was listed TWICE. Please note that the first listing for this volume, which begins at the bottom of page 11 and continues at the top of page 12 is INCORRECT. The correct listing for Volume #22 is the one which is printed completely in the left column on page 12. We apologize for any inconvenience this typo may have caused.

Also, we have found it necessary to redefine the previously- unavailable Volume #20 so that we could put the object code for XLisp in the volume right after #19, which contains the XLisp source code. This shouldn't cause any problems for anyone who has been paying attention and NOT ordering any volumes that are not yet completed. Just in case, in the future we do not plan to tell about volumes we are still working on until they are ready to be released. This should stop some of the confusion and frustration for those people too anxious to wait for the announcement before they send in their orders for disks.

We have received quite a large number of new donations to the Library in the last couple of months and have been able to create a few new Library Volume disks (eleven, to be exact). One of the volumes deserves some special explanation:

We received an interesting donation from Mr. Greg Morse, called "XCom9". "XCom9" is a fairly complete and powerful modem program which should be very useful to many people. The unusual thing about this submission is that Greg decided to classify his software as "Freeware". For those of you not familiar with this term, "Freeware" is software which is not actually "free" (as is most other "public domain" software). The way it works, is that you may order and try the software for free, but if you like it, and use it, the author will expect you to mail him a donation (in this case, $20.00). Please note that, in this particular case, you will be under no LEGAL obligation to send Greg a donation if you decide to order this disk and use the program, but Greg has made his position very clear that he sure EXPECTS you to. The UG does not have any official policy regarding "Freeware" at this time, so, as an experiment, we have decided to release this program in the form that the author wished. I have heard arguments both for and against allowing this type of software submission, but haven't fully made up my own mind about it yet. Some feel that allowing Freeware in the Library will encourage more "professional" software developers to make some of their works available for free evaluation through the UG Library. The opponents to Freeware in the UG Library fear that,

9

if this were allowed, nobody would ever donate any truly "public domain" software to the Library any more; everyone would call their programs "Freeware" in the hope that they might make some money out of the deal. In any case, at least as long as I am in charge of the Library, I plan to make sure that ALL software donations are placed in the PUBLIC DOMAIN before distribution, thereby LEGALLY making payments optional.

We would like to know how YOU feel about the concept of "Freeware", as it applies to the UG Software Library. Please write the MOTD with your comments.

SOFTWARE LIBRARY CATALOG
(New Releases and Updates as of 08/08/85)

Volume #10, Math & Electronics, Disk #1
(35 track, single-sided)

Title: FAST_FOURIER   From: MORSE, GREG
Ver: 04/01/84   Lang: BASICO9.6809   Fmt: SOURCE
Function: PERFORMS THE COMPLEX FAST FOURIER TRANSFORM OF ARRAYS UP TO 2048 ENTRIES. ALSO INVERSE TRANSFORM.
Program Type: MISC.

Title: LINEFIT   From: MORSE, GREG
Ver: 01/01/84   Lang: BASICO9.6809   Fmt: SOURCE
Function: FINDS LEAST SQUARE FIT LINES THRU A SET OF DATA POINTS x y AS WELL AS THEIR MEANS STD DEV AND CORRELATION COEFF.
Program Type: MISC.

Title: NETWORK   From: KREIDER, CARL R.
Ver: 08/14/83   Lang: BASICO9.6809   Fmt: SOURCE
Function: ELECTRONIC CIRCUIT NETWORK DESIGN/ANALYSIS
Program Type: MISC.

Title: NORMAL   From: DORNER, GEORGE C.
Ver: 02/05/85   Lang: BASICO9.6809   Fmt: SOURCE
Function: NORMAL CALLS RND_SMPL REPEATEDLY TO DEMONSTRATE THIS RND FUNCTION WHICH RETURNS RANDOM VARIATES FROM A NORMAL DISTRIBUTION
Program Type: MISC.

Title: RC   From: DORNER, GEORGE C.
Ver: 02/05/85   Lang: BASICO9.6809   Fmt: SOURCE
Function: GENERATES DATA ILLUSTRATING BEHAVIOR OF SIMPLE RESISTOR-CAPACITOR CIRCUIT.
Program Type: MISC.

Title: RESRAT1HC   From: KREIDER, CARL R.
Ver: 08/14/83   Lang: BASICO9.6809   Fmt: SOURCE
Function: PRINTS ALL 1% RESISTOR COMBINATIONS THAT FIT THE SPECIFIED RESISTOR RATIO AND ERROR (TOLERANCE) LIMITS.
Program Type: MISC.

Title: UNIVARIATE   From: DORNER, GEORGE C.
Ver: 02/05/85   Lang: BASICO9.6809   Fmt: SOURCE
Function: COMPUTES ALL STANDARD STATISTICS FOR A SET OF SINGLE VARIABLE OBSERVATIONS.
Program Type: MISC.

Title: AVERAGE_STDEV   From: MORSE, GREG
Ver: 01/01/84   Lang: BASICO9.6809   Fmt: SOURCE
Function: CALCULATES MEANS, STANDARD DEVIATIONS CROSS PRODUCTS AND SUMS OF SQUARES FOR TWO ARRAYS OF NUMBERS.
Program Type: SUBROUTINE

Volume #19, Languages 1: XLisp
(40 track, double-sided)

THIS IS THE SOURCE CODE FOR THE XLISP LANGUAGE. FOR THE EXECUTABLE OBJECT, PLEASE SEE VOLUME #20.

Title: XLISP   From: JONES, JAMES E.
Ver: 1.2   Lang: C_MW.6809   Fmt: DIRECTORY
Function: LISP-LIKE LANGUAGE WITH OBJECT-ORIENTED FUNCTIONS
Program Type: MISC.

Volume #20, XLisp (executable object)
(35 track, single-sided)

This disk contains the compiled version of the XLisp program from Volume #19. Versions for both OS-9 Level 2 and OS-9 Level 1 (such as CoCo) are included.

Volume #29, File Maintenance, Disk #3
(35 track, single-sided)

Title: DEL.A   From: TYLER, W. R. & RASPUTIN, O.
Ver: 01/25/85   Lang: ASM.6809   Fmt: SOURCE
Function: SIMILAR TO MICROWARE "DEL", EXCEPT DOES NOT ABORT ON ERROR UNTIL ALL FILENAMES PASSED ARE PROCESSED.
Program Type: FILE MAINTENANCE

Title: ERASE   From: SCUDIERE, DR. MATTHEW
Ver: 02/07/85   Lang: ASM.6809   Fmt: SOURCE
Function: ERASES FILE FROM RBF MEDIA TO PROTECT SENSITIVE INFORMATION. THIS MAKES IT TOTALLY UNRECOVERABLE.
Program Type: FILE MAINTENANCE

Title: HDIR.A   From: HEALTON, WILLIAM L.
Ver: 01/05/85   Lang: ASM.6809   Fmt: SOURCE
Function: A MULTI-LEVEL, HIERARCHICAL DIRECTORY
Program Type: FILE MAINTENANCE

Title: PURGE.B   From: SCUDIERE, DR. MATTHEW
Ver: 02/07/85   Lang: BASICO9.6809   Fmt: SOURCE
Function: PERMANENTLY ERASES UNUSED PORTIONS OF RBF MEDIA TO PROTECT SENSITIVE INFO.
Program Type: FILE MAINTENANCE

Title: SCAN   From: DURHAM, PETER E.
Ver: 06/12/85   Lang: BASICO9.6809   Fmt: SOURCE
Function: SCAN attempts to read each sector of a file/device; any errors encountered on the way are displayed.
Program Type: FILE MAINTENANCE

Title: SORTDIR   From: DURHAM, PETER E.
Ver: 06/12/85   Lang: BASICO9.6809   Fmt: SOURCE
Function: Sorts directory entries in increasing ASCII order.
Program Type: FILE MAINTENANCE

Title: FCAT   From: BALL, WILLIAM H.
Ver: 1.0   Lang: C_MW.6809   Fmt: SOURCE & OBJECT
Function: CONCATENATE FILES TO STD OUT, EITHER FROM A LIST ON THE COMMAND LINE OR FROM A LIST ON STD IN
Program Type: FILE MAINTENANCE

Title: RM   From: BALL, WILLIAM H.
Ver: 1.0   Lang: C_MW.6809   Fmt: SOURCE & OBJECT
Function: DELETES FILES, AS SPECIFIED ON THE COMMAND LINE OF ON STD IN.
Program Type: FILE MAINTENANCE

Title: AP   From: BUCK, DAVID K. (By J. Haley)
Ver: 02/21/85   Lang: ASM.6809   Fmt: SOURCE & OBJECT
Function: AN APPEND UTILITY TO APPEND TO A FILE OR TO APPEND SEVERAL FILES INTO ONE.
Program Type: FILE MAINTENANCE

Title: CHOWN   From: BUCK, DAVID K.
Ver: 02/21/85   Lang: C_MW.6809   Fmt: SOURCE & OBJECT Function: CHANGES THE OWNERSHIP OF A NON-DIRECTORY FILE
Program Type: FILE MAINTENANCE

Volume #30, File Processing Utilities, Disk #4
(35 track, single-sided)

Title: COM   From: TYLER, WILLIAM R.
Ver: 05/11/85   Lang: ASM.6809   Fmt: SOURCE
Function: FILTER WHICH CLEARS THE MOST SIGNIFICANT BIT OF EVERY BYTE IN A TEXT FILE
Program Type: FILE PROCESSING FILTER

Title: PAGE   From: DREXLER, KEN
Ver: 05/16/85   Lang: ASM.6809   Fmt: SOURCE
Function: BREAKS TEXT FILES INTO 58-LINE PAGES,

SEPARATED BY FORM FEED CHARACTERS.
Program Type: FILE PROCESSING FILTER

Title: UPPER   From: SMITH, MARK W.
Ver: 04/07/85   Lang: ASM.6809   Fmt: SOURCE & OBJECT
Function: EQUIV TO THE UNIX COUNTERPART...CONVERTS A
FILE TO UPPER-CASE ASCII.
Program Type: FILE PROCESSING FILTER

Title: ASC   From: SMITH, MARK W.
Ver: 04/07/85   Lang: C_MW.6809   Fmt: SOURCE &
OBJECT
Function: CONVERTS FILE TO UPPER OR LOWER CASE ASCII.
Program Type: FILE PROCESSING FILTER

Title: WCL   From: BUCK, DAVID K.
Ver: 02/21/85   Lang: C_MW.6809   Fmt: SOURCE &
OBJECT
Function: COUNTS THE CHARACTERS, WORDS, AND LINES IN
A FILE USING 32 BIT NUMBERS
Program Type: FILE PROCESSING FILTER

Title: STRIPARITY   From: DREXLER, KEN
Ver: 1.0   Lang: BASICO9.6809   Fmt: SOURCE
Function: CLEARS THE PARITY BIT ON ALL CHARACTERS IN
A FILE.
(See "STRIP.A" if you want to do this AND remove all
control characters).
Program Type: TEXT FILE PROCESSING

Title: UPPERCASE   From: SMITH, MARK W.
Ver: 05/09/85   Lang: BASICO9.6809   Fmt: SOURCE &
OBJECT
Function: Converts a text file to upper-case.
Program Type: TEXT FILE PROCESSING

Title: ROF   From: BLOCH, RALPHF.
Ver: 05/12/85   Lang: ASM.6809   Fmt: OBJECT
Function: TEXT FORMATTER; READS STANDARD TEXT FILE
WITH COMMAND CODES, GENERATES FORMATTED, PRINTABLE
OUTPUT
Program Type: TEXT FILE OUTPUT ROUTINE

Title: PAGPRT   From: MORSE, GREG
Ver: 05/21/85   Lang: BASICO9.6809   Fmt: SOURCE
Function: PAGINATES A FILE LIST OBEYING SETTINGS FOR
TOP BOTTOM & LEFT MARGINS PAGE LENGTH SPACING AND
STARTING POINT IN FILE.
Program Type: TEXT FILE OUTPUT ROUTINE

Title: NROFF   From: NAPOLITANO, TOM
Ver: 1   Lang: C_DYNA.6809   Fmt: SOURCE
Function: TEXT PROCESSOR BASED ON "SOFTWARE TOOLS"
Program Type: TEXT FILE OUTPUT ROUTINE

                Volume #36, General Interest, Disk #3
                     (35 track, single-sided)

Title: AMORT   From: CONLEY DENNIS L
Ver: 06/23/85   Lang: BASICO9.6809   Fmt: SOURCE
Function: AMORT WILL PRINT TO YOUR PRINTER A COMPLETE
AMORTIZED SCHEDULE OF A MONTHLY PAYMENT TYPE LOAN AND
GIVE YEARLY TOTALS
Program Type: MISC.

Title: HANGMAN   From: DREXLER, KEN and GROVAC, TIM
Ver: 02/03/85   Lang: BASICO9.6809   Fmt: SOURCE
Function: REVISED VERSION OF TIM GROVAC'S HANGMAN
GAME, DOWNLOADED FROM THE OS-9 FORUM ON COMPUSERVE.
Program Type: MISC.

Title: KALAH   From: KNUDSEN, MICHAEL J.
Ver: 02/21/85   Lang: C_MW.6809   Fmt: SOURCE &
OBJECT
Function: PLAYS TWO VERSIONS OF THE INTERACTIVE
STRATEGY BOARD GAME.
Program Type: MISC.

Title: EPS   From: NAPOLITANO, TOM
Ver: 1   Lang: C_DYNA.6809   Fmt: SOURCE
Function: ALLOWS ONE TO SET UP AN EPSON PRINTER USING
ENGLISH COMMANDS
Program Type: SYSTEM UTILITY

Title: OKI   From: NAPOLITANO, TOM
Ver: 1   Lang: C_DYNA.6809   Fmt: SOURCE
Function: ALLOWS ONE TO SETUP OKIDATA 82/92 PRINTER
USING ENGLISH LANGUAGE COMMANDS
Program Type: SYSTEM UTILITY

                Volume #37, Communication, Disk #2
                     (40 track, double-sided)

Title: KERMIT   From: SEATON, WM. GLENN
Ver: VER 1.2   Lang: C_MW.6809   Fmt: SOURCE & OBJECT
Function: FILE TRANSFER PROGRAM WITH ERROR CHECKING;
SIMILAR TO VERSIONS AVAIL FOR IBM-PC/XT; HONEYWELL;
HP;+40 PROCESSORS.
Program Type: COMMUNICATION

                Volume #38, Programming Utilities, Disk #4
                     (35 track, single-sided)

Title: S1FLEX   From: PASS, EDGAR M.
Ver: 02/09/85   Lang: C_MW.6809   Fmt: SOURCE &
OBJECT
Function: CONVERTS MOTOROLA S1 FORMAT TO FLEX BINARY
FORMAT
Program Type: PROGRAMMING AID

Title: S1INTEL   From: PASS, EDGAR M.
Ver: 02/09/85   Lang: C_MW.6809   Fmt: SOURCE &
OBJECT
Function: CONVERTS MOTOROLA S1 FORMAT TO INTEL FORMAT
Program Type: PROGRAMMING AID

Title: S1LOAD   From: PASS, EDGAR M.
Ver: 02/09/85   Lang: C_MW.6809   Fmt: SOURCE &
OBJECT
Function: CONVERTS MOTOROLA S1 FORMAT TO OS9 BINARY
FORMAT
Program Type: PROGRAMMING AID

Title: S1UNFLEX   From: PASS, EDGAR M.
Ver: 02/09/85   Lang: C_MW.6809   Fmt: SOURCE &
OBJECT
Function: CONVERTS FLEX BINARY FORMAT TO MOTOROLA S1
FORMAT
Program Type: PROGRAMMING AID

Title: S1UNLOAD   From: PASS, EDGAR M.
Ver: 02/09/85   Lang: C_MW.6809   Fmt: SOURCE &
OBJECT
Function: CONVERTS OS-9 BINARY FORMAT TO MOTOROLA S1
FORMAT
Program Type: PROGRAMMING AID

Title: S1XREF   From: PASS, EDGAR M.
Ver: 02/09/85   Lang: C_MW.6809   Fmt: SOURCE &
OBJECT
Function: PRINTS SORTED X-REFERENCE FROM A MOTOROLA
S1 FORMAT FILE
Program Type: PROGRAMMING AID

Title: REPLACE   From: PASS, EDGAR M.
Ver: 02/09/85   Lang: C_MW.6809   Fmt: SOURCE &
OBJECT
Function: REPLACES STRINGS IN TEXT FILES
Program Type: TEXT FILE PROCESSING

                Volume #39, Communication, Disk #3 (FREEWARE)
                     (40 track, double-sided)

Title: BREAK   From: MORSE, GREG
Ver: 85/04/10   Lang: ASM.6809   Fmt: SOURCE & OBJECT
Function: SENDS A BREAK ON A 6850 ACIA PORT BY
DIRECTLY ACCESSING THE CHIP CSR. HAS SOME TUTORIAL
VALUE FOR LEVEL 2 USERS.
Program Type: COMMUNICATION

Title: XCOM9   From: MORSE, GREG
Ver: 85/04/10   Lang: ASM.6809   Fmt: DIRECTORY
Function: MODEM PGM WITH XMODEM SUPPORT, NON XMODEM
CAPTURE AND UPLOAD MODES, EXPANDABLE CAPTURE BUFFER
AND MUCH MORE.
Program Type: COMMUNICATION

                Volume #40, System Utilities, Disk #4
                     (35 track, single-sided)

Title: MAKE   From: CASLER, DAVE R.
Ver: 1.1   Lang: ASM.6809   Fmt: SOURCE & OBJECT
Function: MAKE A FILE BASED ON DEPENDENCY FILES
Program Type: SYSTEM UTILITY

Title: SETIME.NEW   From: GROVES, ORAL E.
Ver: 1   Lang: ASM.6809   Fmt: SOURCE & OBJECT
Function: USER FRIENDLY VERSION OF OS9 SETIME PROGRAM
Program Type: SYSTEM UTILITY

Title: HELP.C   From: SEATON, WM. GLENN
Ver: 05/29/84   Lang: C_MW.6809   Fmt: SOURCE
Function: UPGRADE OF ASM VERSION; LIST HELP FILES ON
THE HELP DIRECTORY.
Program Type: SYSTEM UTILITY

Title: SST   From: SEATON, WM. GLENN
Ver: 2.00   Lang: C_MW.6809   Fmt: SOURCE
Function: SHOWS SYSTEM BLOCK SIZE; TOTAL SYSTEM RAM;
BLOCK ALLOCATION MAP FOR SYSTEM STATE & SYSTEM STATE
FREE MEMORY.
Program Type: SYSTEM UTILITY

Volume #42, Color Computer Graphics, Disk #1
(35 track, single-sided)

Title: GFXDUMP2   From: DURHAM, PETER E.
Ver: 06/12/85   Lang: BASIC09.6809   Fmt: SOURCE
Function: Dumps the CoCo graphics display to a Line
Printer VII or compatible.
Program Type: MISC.

Title: THREE_D_GRAPHICS   From: McLEAN, BRUCE
Ver: 05/25/85   Lang: BASIC09.6809   Fmt: SOURCE
Function: This program generates 3-dimensional
graphics using the CoCo hires screen. It also has
routines for disk and printer.
Program Type: MISC.

Title: RATMAZE   From: NAPOLITANO, TOM
Ver: 1   Lang: C_DYNA.6809   Fmt: SOURCE
Function: USES THE COCO GRAPHICS TO SIMULATE A RAT'S-
EYE VIEW OF A MAZE
Program Type: MISC.

LETTERS TO THE EDITOR

OS-9 USERS GROUP
P. O. Box 7586
Des Moines, IA 50322

Dear USERS GROUP,

I am a beginning OS-9 user. I bought my Color
Computer some 3 years ago and was a true believer at
Purchase Time. At work I probably convinced some
parents on the Color Computer as a wonderful
Christmas gift.

However, with the advent of OS-9, there cannot be
enough praises heaped. I worked in the accounting
department for 5 years and grew bored sitting at a
desk shuffling papers. I went to the MIS department
two years ago to try to understand what all the
screaming was about concerning the new "COMPUTER
AGE".

I work now as a computer operator in RCA. We use
11/34 and 11/44 PDP computers using RSX11M as the
operating system. The 11/44 has 4 megabytes of
internal memory with a minimum of 2 gigabytes of disc
storage available.

Comparing the aforementioned computer with the 64K
Color Computer would be absurd. However, considering
the thousands of dollars spent for software to
acquire the RSX11M operating system and the #300.00
(TANDY) + BASIC09 + BASIC09 & OS-9 TOUR GUIDES I
received quite a deal for my money.

Since purchasing OS-9 I became involved in System
Generations (Sys Gens) on the DEC PDP computers.
Prior to Sys Gen(ing) the PDPs, I reconfigured my OS-
9 Color Computer to 6 milliseconds and 40 tracks.
This process helped me to really understand what an
operating system is and does. The process helped me
to understand what is meant by "building" a device
driver, utility, installation of application
languages, and how certain hardware characteristics
must be accounted for.

Enough already: I have enclosed my check. My name
and address are:

JOHN R. SPENCER
552 REYNOLDS AVENUE
LANCASTER, PENNA.   17602-2147

EDITOR'S NOTE: John also indicated that Garry Kling
of the Computer Center of York, Pennsylvania has been
holding informal OS-9 meetings on Saturday mornings
on a Level II system and COCO meetings on the third
Thursday evening of the month.

SOUTHEAST MEDIA
5900 Cassandra Smith Road
P. O. Box 794
HIXSON, TENNESSEE 37343

Memo

LETTER

(615) 842-4601

Date   July 24, 1985

Editors, MOTD
25825 104th Ave. S.E.
Suite 344
Kent, WA 98031

OS-9 PRODUCTS ANNOUNCEMENT

K-BASIC now makes the multitude of TSC XBASIC Software available
for use under OS-9. Transfer your favorite BASIC Programs to OS-
9, compile them, Assemble them, and BINGO -- usable, multi-
precision, familiar software is running under your favorite
Operating System!
K-BASIC including the Assembler $199.00

O-F -- Written in BASIC09 (with Source), includes: REFORMAT, a
BASIC09 Program that reformats a chosen amount of an OS-9 disk to
FLEX Format so it can be used normally by FLEX; and FLEX, a
BASIC09 Program that does the actual read or write function to the
special O-F Transfer Disk; user-friendly menu driven. Read the
FLEX directory, Delete FLEX Files, Copy both directions, etc.
Flex users use the special disk just like any other FLEX disk.
OS-9 $79.95

OSM Extended 6809 Macro Assembler -- Provides local labels,
Motorola S-records, and Intel Hex records; XREF. Generate OS-9
Memory modules under FLEX. $99.00.

TRUE CROSS ASSEMBLERS Supports 1802/5, Z-80, 6800/1/2/3/8/11/HC11,
6804, 6805/HC05/146805, 6809/00/01, 6502 family, 8080/5,
8020/1/2/35/C35/39/40/48/C48/49/C49/50/8748/49, 8031/51/8751, and
68000 Systems. Assembler and Listing formats same as target CPU's
format. Produces machine independent Motorola S-Text.
each - $50.00; any 3 - $100.00; complete set w/ C Source (except
the 68000 source) - $200

CRASMB -- 8-Bit Macro Cross Assembler with same features as OSM;
cross-assemble to 6800/1/2/3/4/5/8/9/11, 6502, 1802, 8048 Sers,
80/85, Z-8, Z-80, TMS-7000 sers. Supports the target chip's
standard mnemonics and addressing modes. Full package - $399.00.

CRASMB 16.32 -- Cross Assembler for the 68000. $249.00

SUPER SLEUTH -- Interactive Disassembler; extremely POWERFUL!
Disk File Binary/ASCII Examine/Change, Absolute or FULL
Disassembly. XREF Generator, Label "Name Changer", and Files of
"Standard Label Names" for different Operating Systems.
CCO, Obj. Only $50.00; OS-9 w/source $101.00

BasicO9 XRef -- This BasicO9 Cross Reference Utility is a BasicO9
Program which will produce a "pretty printed" listing with each
line numbered, followed by a complete cross referenced listing of
all variables, external procedures, and line numbers called. Also
includes a Program List Utility which outputs a fast "pretty
printed" listing with line numbers. Requires BasicO9 or RunB. O
& CCO Obj. Only $39.95; w/source $79.95

OS-9 VDisk -- For Level I only. Use the Extended Memory
capability of your SWTPC or Gimix CPU card (or similar format DAT)
for FAST Program Compiles, CMD execution, high speed inter-process
communications (without pipe buffers), etc. - SAVE that System
Memory. Virtual Disk size is variable in 4K increments up to
960K. Some Assembly Required. OS-9 Obj. only $79.95; w/Source
$149.95

CMODEM Telecommunications Program -- Menu-Driven; supports Dumb-
Terminal Mode, Upload and Download in non-protocol mode, and the
CP/M "Modem7" Christensen protocol mode to enable communication
capabilities for almost any requirement. Written in "C".
OS-9 with complete source - $100.00; without source - $50.00

No.	Done?	Title:		Format:	Who?
0.03	Y	New Member Intro		+++(40 track, ss)	DK
1.00	Y	Spelling Checker		(35 track, ss)	DK
2.00	Y	Spelling Dictionary		(40 track, ds)	DK
3.01	Y	Word Processing Utils		+++(35 track, ss)	DK
4.01	Y	Programming Utilities		+++(35 track, ss)	DK
5.00	Y	File Processing Utils		+++(35 track, ss)	DK
6.01	Y	Adventure Game (source)		(40 track, ds)	CK
7.01	Y	Adventure Game (object)		(40 track, ss)	CK
8.00	Y	General Interest (demo, games, finance)		(35 track, ss)	CK
9.00	Y	C Programmer's Tool Kit		(35 track, ss)	CK
10.00	Y	Math & Electronics 1		(35 track, ss)	GD
11.00	Y	Word Processing Utils	(disk #2)	(35 track, ss)	CK
12.00	Y	Programming Utilities	(disk #2)	(35 track, ss)	CK
13.00	Y	File Processing Utils	(disk #2)	(35 track, ss)	CK
14.02	Y	File Maintenance		(35 track, ss)	CK
15.01	Y	Communication		(35 track, ss)	CK
16.00	Y	Hardware Customizations		(35 track, ss)	CK
17.00	Y	Basic09 Programmer's Tool Kit		(35 track, ss)	CK
18.00	Y	System Utilities		(35 track, ss)	CK
19.01	Y	Languages 1: XLisp (source)		(40 track, ds)	CK
20.00	Y	XLisp (object)		(35 track, ss)	DK
21.00	Y	File maintenance	(disk #2)	+++(35 track, ss)	CK
22.00	Y	Programming Utilities	(disk #3)	(35 track, ss)	CK
23.00	Y	File Processing Utils	(disk #3)	(35 track, ss)	CK
24.00	Y	General Interest	(disk #2)	(35 track, ss)	CK
25.00	Y	Word Processing Utils	(disk #3)	(35 track, ss)	CK
26.00	Y	C Language Math Library		(35 track, ss)	CK
29.00	Y	File Maintenance	(disk #3)	(35 track, ss)	CK
30.00	Y	File Processing Utils	(disk #4)	(35 track, ss)	CK
31.00	Y	Hardware Customizations	(disk #2)	(35 track, ss)	CK
32.00	Y	Hardware Customizations	(disk #3)	(35 track, ss)	CK
33.00	Y	System Utilities	(disk #2)	(35 track, ss)	CK
34.00	Y	Hardware Customizations	(disk #4)	(35 track, ss)	CK
35.00	Y	System Utilities	(disk #3)	(35 track, ss)	CK
36.00	Y	General Interest	(disk #3)	(35 track, ss)	CK
37.00	Y	Communication	(disk #2)	(40 track, ds)	CK
38.00	Y	Programming Utilities	(disk #4)	(35 track, ss)	CK
39.00	Y	Communication (Morse)	(disk #3)	(40 track, ds)	CK
40.00	Y	System Utilities	(disk #4)	(35 track, ss)	CK
42.00	Y	Coco Graphics		(35 track, ss)	CK

NOTES: 1) Above formats describe STANDARD (non-CoCo) versions.
All RS CoCo versions are 40 track, single-sided,
double-density.

2) All STANDARD (non-CoCo) format disks are single-density.

+++ For these volumes to fit in the specified format, the
disk is created with a default sector allocation of 1
sector per directory (made by doctoring the "segment
allocation size" byte (offset $20) in the device
descriptor of the drive on which the master disk is
made).

OS-9 SOFTWARE LIBRARY ORDER FORM

DISK #	TITLE	FORMAT (RS,STANDARD)	SIZE (8 or 5)	PRICE

TOTAL:------------

NAME:---

ADDRESS:---

CITY,STATE:--------------------------------------

ZIP CODE:--

PRICES: $3.00 PER DISK IN 5 INCH FORMAT   ALL ORDERS MUST BE PREPAID
$5.00 PER DISK IN 8 INCH FORMAT

SEND ORDER FORM TO: OS-9 USERS GROUP
P.O. BOX 7586
DES MOINES, IA 50322

Classified

One DD/SS 8 inch disk drive, just rebuilt, runs perfectly...$50.00.   206/630-3091

2 LMB1A Smoke Signal Broadcasting motherboards...One never used, retail 400.00 on sale for $100.00 each. This is a good way to start your SS50 system cheap...all gold Pins and extended addressing for OS-9 level Two... call 206-630-3091

CORRECTIONS/CHANGES OF ADDRESS

Corrections/changes of address MUST be sent to the OS-9 Users Group address. Mailings are by bulk presorted third class mail. Any incorrect addresses will usually result in missed issues.

OS-9 USERS GROUP ADDRESS:
P.O. Box 7586
Des Moines, IA 50322

ARTICLES AND LETTERS TO EDITOR:

The Users Group invites all members to submit articles or letters for publication in the MOTO Newsletter. The preferred method of submission is on a 5 or 8 inch disk in OS-9 format. Stylograph or Screditor3 commands can be used if indicated in read me file. Short letters or articles can also be submitted in printed form or dumped on-line through the MOTO Editors Compuserve number below. Disks will be returned to sender ONLY if a self addressed and stamped disk mailer is included with submission. If possible all printed or on-disk articles should be only 53 columns wide. All submissions should be mailed to MOTO Publishing through the Users Group address above.

The Classified Section of this newsletter is free and open to all User Group members. The only requirement we have is that any equipment for sale must run OS-9 or be compatible with an OS-9 machine. Include price and phone number with all submissions.

Editors: Tim & Lori Grovac   206-630-3091
         Compuserve ID: 70370,720

HOW TO JOIN THE OS-9 NATIONAL USERS GROUP

The National OS-9 Users Group has been formed to foster communication between the users of OS-9. Our plans and expectations include the establishment of several means for members to communicate their interests, concerns, questions, and software to others. Presently the specific activities of the Group include the Group's Newsletter, MOTO (message of the day), a bulletin board system on Compuserve, a users software library with 36 volumes, and an annual User Group meeting usually held in Des Moines, Iowa.

When you join the Users Group, you will receive a free disk of OS-9 software and a copy of the latest MOTO. You will also be eligible to buy for cost only any of the 38 software library disks. This software is setup on disks for either the standard SS-50 or Radio Shack format. Be SURE to advise them on the enclosed application form which version you want. The current officers of the OS-9 Users Group are:
                    President: Dale Puckett
                    Vice Pres: Peter Dibble
                    Treasurer: George Dorner

The OS-9 Users Group is independent of Microware Systems Corp. and may not be reached through their offices or phones.

To join the OS-9 Users Group, send check or money order for $25.00 to the OS-9 User Group address along with the Membership Application below.

OS-9 USERS GROUP MEMBERSHIP APPLICATION

NAME_____

ADDRESS_____

CITY, STATE, ZIP_____

COUNTRY (IF NOT USA)_____

PHONE NUMBER_____

COMPUSERVE NUMBER_____

TYPE & LEVEL OF OS-9:

STANDARD SS-50_____

COLOR COMPUTER_____

LEVEL ONE_____

LEVEL TWO_____

HARDWARE MFG_____

FLOPPY DISK SIZE   5"_____   8"_____

IS IT OK TO DISTRIBUTE YOUR NAME WITH A LIST OF THE USER GROUP MEMBERS?

    YES_____   NO_____

WOULD YOU LIKE TO WORK ON ANY COMMITTEE OR GROUP?

    YES_____   NO_____

OS-9 USERS GROUP ELECTIONS

As most of you know the present officers of the OS-9 Users Group have served well over their one year term of office. Many attempts have been made to replace our officers but for one reason or another have been delayed. We have finally come up with a group of candidates that we feel have the background and expertise to run the UG in the professional manner to which we have all become accustomed. The candidates are as follows:

FOR PRESIDENT: BRIAN A. LANTZ
   Brian is a self-employed freelance programmer whose programs are familiar to COCO and other OS-9 and 68XXX users. He currently has three 68000 systems and two 6809 systems. Brian will have a column appearing in the Rainbow soon. He lives in Tampa, Florida.

FOR VICE-PRESIDENT: WILLIAM W. TURNER III
   Bill is Senior Systems Programmer for Telecommunications Support for GTE Data Services in Tampa, Florida. He has worked with OS-9 both on SS-50 and COCO systems. He was a columnist for Interface Age Magazine for some time.

FOR TREASURER: STEVE ODNEAL
   Steve has worked with computers since 1966 and with Motorola products since 1975. He is now Director of Microprocessor Management at a bank in Kansas City, Missouri. His hobby interests in computers are focused on the COCO and system programming. A number of his products are familiar to COCO users. Steve has worked with OS-9 on the COCO since before it's public release.

FOR SECRETARY: DAVE GIBSON
   Dave has served as the acting secretary with the current officers. He works for the US Government in Traffic Engineering applications. Dave lives in Alexandria, Va. and has been an OS-9 user "since Level I was still buggy". His current computer system is a Smoke Signal Cheiftain.

These individuals have all indicated a willingness to serve and have displayed the enthusiasm to carry forward and to expand the activities and services of the OS-9 Users Group. Other candidates may be added on a write in basis.

OS-9 USERS GROUP BALLOT

INSTRUCTIONS: Cast your vote by placing an X by the candidate of your choice. Blank line is provided for write in candadate. Cut out ballot along dotted line above and mail to: OS-9 Membership Committee   13229 Blue Quail Rd.   Yukon OK 73099. Final date for ballots to be received is September 30, 1985.

PRESIDENT:

    BRIAN A. LANTZ   _____

    _____

VICE-PRESIDENT:

    WILLIAM W. TURNER   _____

    _____

TREASURER: STEVE ODNEAL   _____

    _____

SECRETARY: DAVE GIBSON   _____

    _____