MTU

UPDATE

2nd Quarter '83 Vol. 1 #4

Stylish New MTU-140™ Maintains Compatibility

Pictured above is the newest member of MTU's line of advanced microcomputers. The MTU-140 is functionally equivalent to the '130. The most obvious difference is the integrated monitor-CPU-disk with detached keyboard which is becoming so popular. The CPU and Disk Controller boards inside are the same as the MTU-130, however, which gives total software and hardware compatibility.

Externally, the display monitor is still 12 inches measured diagonally, but with separate sync inputs for absolutely no bending at the top of the image. The disk drives are the latest thinline design which allows two 8" double-sided drives to be fitted in the space normally taken by one. The keyboard is the latest ultra low profile design with identical layout to that of the '130 except for reduced

A forum for the exchange of ideas and information exclusively for MTU-130 users by MICRO TECHNOLOGY UNLIMITED, a Division of Consolidated Sciences, Inc.

clearance on either side of the cursor keypad. The cable is sufficiently long to allow the keyboard to be placed in the user's lap if desired. Improved keyboard circuitry allows the cable to be extended up to 20 feet if necessary. Total unit weight is about 48 pounds, which is easily movable.

Internally, the MTU-140 contains a 4-slot card file and a power supply with sufficient capacity to operate all 4 slots. The monitor brightness and contrast controls are easily reached from the front. The speaker gives good sound quality while the heat activated variable speed fan keeps things cool in severe environments without excessive noise under normal conditions. The rear panel has all of the jacks and connectors the MTU-130 has except for the cassette and audio jacks. These can still be accessed internally however. Everything, including the fan and disk drive motors, is DC powered which means that 50Hz European operation is accomplished simply by moving a jumper on the power supply board. The top cover is easily lifted for servicing by removing six hidden screws.

Pricing for the MTU-140-2D is the same as a comparably equipped MTU-130 which is $3900. Shipments will begin in August.

FROM THE PRESIDENT

What happened to MTU this year? As some of you already know, we have reorganized our administrative and marketing staff. Bill Smith, our V.P. of Marketing is no longer with us. He now has a highly responsible position with an international organization to help manage an overseas naval facility. We were sorry to loose Bill and appreciate the time he spent with us. I have taken over the marketing role at MTU since Bill's departure.

To fill the general management position left behind, we brought Chuck McGill on-board. Chuck has a history of 15 years in the electronics industry and a depth and understanding of management operations that was needed at MTU for continued growth. You may also have noticed a new voice on the phone. This is Linda Shuster, our new administrative assistant. We are very pleased to have Linda with us. Additionally, we have grown to the point where we needed a full-time accountant. After an extended search, we have employed Shirley Uhrinec, a true professional.

These three persons have given us the administrative strength that we needed. We started as a small company, and now with the grace of God and your support, we continue to grow. With this growth comes the added pressures of administrating a larger organization. We are now ready to face new challenges.

Many of you have been asking for a nameplate for your "ZENITH" systems. The existing MTU logo could not easily be implemented as a nameplate because of its shape. After evaluating a number of new logo designs from various commercial artists, we have made our selection. The new one is shown at the end of this column. You will be receiving nameplates for your MTU-130 systems shortly. You may see our new logo soon in print, because we are kicking off a new round of advertising. We do not plan to simply say "look what we have", but will address vertical markets where the MTU-140 provides a viable solution.

We have listened to your requests and feedback and are changing the MTU-130 as a result. A frequent request has been to change the "package" appearance. The cover of this issue shows the MTU-140 -- the MTU-130 in its new package. The external attractiveness of the MTU-140 now reflects its inner beauty. We hope you like it. Color prints of the cover photo will be mailed to you in several weeks. All products undergo evolution during their lifetime. The important thing is that this evolution retains compatability with existing systems.

Many of you have told us what you were doing with your MTU-130's. In several cases, requests were made to change or add

functions to the system. We respond to these ideas, as many of you can attest to. Some of the most notable developments are: the extensive WORDPIC upgrades, the CHESTR graphic design program, a 16 bit A/D and D/A system capable of signal analysis, and larger mass storage. We are looking for your ideas for applications for the MTU-130 in solving problems. If you don't talk to us, we do not hear your needs or desires. I am the person to ask for when you want to discuss or suggest an application.

A current project just begining was triggered by an MTU customer, the president of a steel fabrication house. It is a two dimensional graphic design system. Many of you have relayed your uses of GREDIT, but mentioned that other features should be added to it. GREDIT was not designed to be expanded (it was a trial program for testing the BASIC libraries). We now have enough user input to begin the final specification and design of that package. More on this in a later issue.

Before closing, I would like to mention that the major purpose for this newsletter is to provide a forum for your ideas. MTU receives questions which, unlike many other computer companies, we answer. We do want you to continue submitting general interest articles and ideas to this newsletter for all to read. In the absence of your input, it simply becomes an MTU public relations tool, which is not our intent. So think of what you have done with your '130 or why you are glad you purchased it and write about it. With the new WORDPIC upgrade you just received, that should be a snap!

Sincerely,

David B. Cox

NEW PRODUCTS

Upgraded WORDPIC 1.0

After more than a year of intense development work, Version 1.0 of the WORDPIC word processing system is now available for the MTU-130. All users who purchased the preliminary version of WORDPIC (described in UPDATE #2) were shipped the upgraded version without charge in May.

Perhaps foremost of the many improvements to WORDPIC is the addition of a PAGINATE module to the system. The PAGINATE function key is used to divide a document up into pages of a specified size automatically, with the operator having the ability to make "judgement calls" interactively to avoid "widow" lines, paragraphs split across page boundaries, etc. PAGINATE supports automatic page numbering, and automatic headers and footers. Headers and footers may contain any sort of information (including graphics such as a company logo), and can be changed any number of times in a document. Page numbers can be placed anywhere on the page or not used at all, and compound page numbers (for example page 3-17 or INDEX-6) are permitted.

Graphics printing has been vastly improved. The size of figures is now determined automatically from the image file. You may specify a different size to reduce the amount of the figure that is printed, or increase "whitespace" after the figure. A DENSITY command has been added which allows specification of the resolution of the figure. For the NEC 8023 printer, densities up to 144 by 160 dots per inch are permitted, with figures up to 1280 by 2047 dots in size.

The "shell" program which runs the WORDPIC system now provides fully "user-friendly" functions for all file maintenance operations such as copying, renaming, and

deleting files, backing up disks, etc. To further increase ease of use, the 24 page Preliminary User Manual has been replaced with a 76 page Tutorial Manual and a 102 page Reference Manual. The Tutorial Manual provides a "hands-on" guide for the complete novice, with many figures and a glossary. It is coordinated with sample document files provided on the distribution disk, and steps a new user through operations from how to turn on the machine to use of advanced features such as multicolumn printing with figures. The fully indexed Reference Manual provides detailed information on every aspect of WORDPIC operation.

If you are one of the few MTU-130 users who has not yet purchased WORDPIC, you really owe it to yourself to try it. We think that WORDPIC, with its ability to process very large documents and combine graphics with text, represents the finest word processor available on any personal computer.

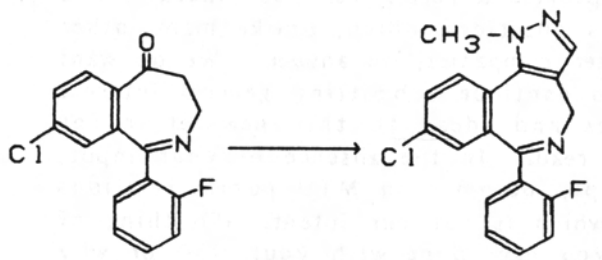
If you have been using a Preliminary version of WORDPIC, we strongly recommend you delete the Preliminary version of the WORDPIC programs (not your document files of course!) from all your working disks and replace them with the new version. WORDPIC 1.0 will abort with an error message if you attempt to run it on a disk with modules from the Preliminary version. The minimum files required to run WORDPIC 1.0 are WORDPIC, WPEDIT, PRINT, PAGINATE, NEWDISK, WSBACKUP, and WORDPICSAVES.Z, plus the operating system files. You may notice that the new PRINT module prints somewhat differently than the Preliminary version, especially when using proportional justification. If you are trying to duplicate a proportionally printed document printed with the Preliminary version, you may want to specify PROPPITCH=13.3.

The development of WORDPIC was no small task. The project took more than 18 months to complete with a four person development team. The completed software package, written completely in assembly language for maximum performance, consists of about two megabytes of source code. The program listing is more than 1200 pages long (that's about seven inches thick)! Programmers contemplating undertaking a large project may be interested to know that this package was developed 100 percent on the MTU-130, not on a minicomputer or mainframe.

Of course the job is not over yet. The MTU team is already assimilating feedback from users for improvements for the next version, and planning other related programs to enhance the usefulness of the WORDPIC system.

CHESTR™ Draws Chemical Structures

If you have a need to do any type of documentation involving chemical structures, you'll love the new CHESTR program for the MTU-130. CHESTR (CHEmical STRucture image generator) lets you quickly "sketch" your structure on the screen using the light pen, and prepare a high quality image which can be printed by WORDPIC. A sample image is shown below, which was input with the light pen in five minutes.



The image above is reproduced actual size to show the high quality which is attainable using CHESTR, WORDPIC, and the SANTEC printer.

To create an image, the CHESTR user draws on the screen by "picking" predefined shapes such as lines and hexagons from a "menu" with the light pen. The shapes may be drawn in any size by selecting the endpoints from a gridded work area on the screen. Chemical symbols may be positioned anywhere desired, and digits are automatically subscripted as shown above. An UNDO "button" on the screen can be used to make corrections. A simple database is

automatcially created on a disk file which describes the image as it is entered. This database can subsequently be appended or edited.

Once the desired image has been defined, a graphic image file is created which can be printed by WORDPIC. You specify how large you want the picture (in inches), and CHESTR automatically scales the data to create a dot-matrix image file of the appropriate size. The image can then be included with your WORDPIC text and printed on the NEC or SANTEC printer.

The CHESTR software package includes a tutorial User Manual, sample images, and SOURCE CODE to the program, which is written in BASIC 1.5 and well documented for ease of modification. A special 68000 machine language library is included for generating image files much larger than the 480 by 256 pixel MTU-130 graphic display. CHESTR requires a DATAMOVER, MTU BASIC 1.5, WORDPIC, and an NEC 8023 or SANTEC printer. CHESTR is available now for $200.

NOTES FROM THE FACTORY

Note: The bug-correction patches described in this section should be made to an existing work copy of the program in question. The correction should not be applied to the original which comes on the Distribution disk for that product.

BASIC 1.5 Bugs

There are presently two known bugs in BASIC 1.5 (not to be confused with standard BASIC 1.0), which aren't too hard to fix. The first bug allows array subscripts to go 1 too high. The second bug occurs when BASIC 1.5 loads a library which contains a block that loads into a bank other than 0. BASIC 1.5 will execute the wrong initialization entry point for the library, probably causing BASIC to crash.

The following patches should be applied to versions of BASIC 1.5 which do not show a creation date, i.e. only shows:

```
MTU-130 BASIC  V1.5     (C) 1981 MTU
```

(Future versions of BASIC will contain a creation date in the startup message.) To correct the bugs enter the following commands, or execute them as a job file. BASIC 1.5 is expected to be on the disk in drive 0, and to be named "BASIC".

```
GET BASIC
SET 9EC AD B8 09 48 AD B9 09 48 20 F3 1A
SET 9F7 68 8D B9 09 68 8D B8 09 60
SET 1B9E 20 EC 09
SET 1530:2 64 ;FIX SUBSCRIPT CHECKING.
SET AFBE 0 0
SAVE NEWBASIC=A00 9EC 402F 3A1 3F5 700 77D
    AFBE =BFBE AFBF 1000:2 1BDD 200:3 2F8
```

Note: the last two lines above should be entered as one line. Once you have made the patch, verify that subscript range checking has been corrected and that libraries still load properly. If it checks out okay, you can delete the old version and rename NEWBASIC to just BASIC.

CPM Cursor Bug

A bug in the BOOTCPM program causes visible cursors to remain on the screen when the cursor is positioned elsewhere. This occurs only in programs which use X,Y cursor positioning, such as WORDSTAR.

The following patches should be applied to BOOTCPM programs which show the CBIOS being created on 2/1/83 or earlier. To correct the problem enter the following commands, or execute them as a job file.

```
GET BOOTCPM
SET DC9 38
SET DD4 18
SAVE NEWBOOTCPM =800 800 122E 4A00 4C70
```

Once you verify that the patch works correctly, you can delete the old version and rename NEWBOOTCPM to BOOTCPM. To correct the assembly language source code, make the following changes:

```
CSRON   BIT     CSRVIS
        BMI     CSRON1
        SEC                 (*** was CLC)
```

and

```
CSROFF   BIT     CSRVIS
         BPL     CSROF1
         CLC              (*** was SEC)
```

DMXMON Bugs

During extensive use by customers and internally, several bugs in DMXMON have surfaced. The bugs listed below can be fixed by entering the patches to be given:

1. If a 68000 program file has more than one block, the entry point is lost when it is run by entering the file name as a "user command".

2. The RFSHSP SVC (#201) does not stop automatic screen refresh.

3. The CPYDSP SVC (#202, reverse copy screen memory to 68000 memory) does not work properly.

4. DMXMON does not properly save and restore SR (status register) when leaving or entering a program.

5. SAVE with an alternate entry point specified gives an error message.

To fix these bugs, first verify that you have the affected version of DMXMON by entering: GETLOC DMXMON and then checking that the printout looks like this:

```
DMXMON.C:0 =0700   0700     0945
                   0B00     2007
                   FC00:1   FC38
                   2008     3949
                   BFBE     BFBF
                   0000:2   002F
                   0060:2   00BF
                   0100:2   015F
                   0700:2   0DC9
```

Then enter the following CODOS commands (be sure that a Datamover is installed):

```
GET DMXMON
SET 3859 4C EE 0A
SET 0AEE F0 06 20 AF 1F 4C 56 38 8D 52 9 60
SET 1194 20 FA 0A
SET 0AFA 20 29 20 4C 52 11
SET 1FA5 26
SET 73C:2 21 F8 02 44 07 54 20 78 02 40 4E
SET 747:2 60 4C F8 FF FF 02 00 46 F8 02 48
SET 1BC1 C9
SET AFBE 0 0
RESAVE DMXMON 700 945 AEE 3949 FC00:1 FC38
          AFBE=BFBE AFBF 0:2 15F 700:2 DC9
```

Note that the last two lines should be typed in as one line. You may wish to use the editor to prepare a job file with these commands. User disk #4 has this job file already prepared under the name DMXMON_PATCH.J

In addition, mark the following corrections in the DMXMON manual:

1. Page 15 section 2.3.15: change the first A2 to A0 in Note 2.

2. Page 91 section 5.1.2: address 000028 (emulator vectors) should include a reference to page 66 section 5.5.4 of the Motorola 16 Bit Microprocessor User's Manual, third edition.

3. Page 73 change SETVP68 to SETVP65 and SETWD65 to SETWD68.

Nondestructive Screen Saves

Have you ever wanted to save an image on the MTU-130 screen for printing or re-display, but couldn't figure out how to save it without having the command you type in clutter the image on the screen? Or, have you ever wanted to be able to get an image to the screen without having the "CODOS>" prompt overwrite part of it? This is actually quite simply done using commands described in the CODOS manual, but may not be obvious.

First let's review the commands used for saving and recalling images. The easiest way to save a screen image is to use the new GSAVE Utility, supplied on the WORDPIC

1.0 disk and on User Group Disk #4. Simply type GSAVE IMAGEFILE.G and the contents of the entire screen will be saved on a file named IMAGEFILE.G. This image can then be recalled to the screen later by typing GGET IMAGEFILE.G, or can be printed from WORDPIC using a figure code line specifying FILE=IMAGEFILE.G.

The trouble is, after the IMAGEFILE.G is redisplayed, the CODOS> prompt will appear on the screen, possibly messing up the image you wish to display. You can disable all prompts and messages generated by CODOS by typing ASSIGN 2 N before doing the GGET. No more CODOS prompts will appear until you make an error or do an ASSIGN 2 C to restore normal prompts.

There is still the problem of how to keep the actual command you type in from overwriting the image you wish to save. This is easily accomplished by using the CTRL/E key. Pressing CTRL/E disables the echo of keyboard characters to the screen until CTRL/E is pressed again. The cursor will still move on the screen as you type, but no characters will appear, and the screen will not be affected.

If you don't have User disk #4 or WORDPIC, you can still save and restore images to the screen using the same techniques for disabling echo and suppressing prompts, but using the command:

```
SAVE IMAGEFILE.G C000:1 FBFF
```

to save the screen image, and:

```
GET IMAGEFILE.G
```

to redisplay it.

CLASSIFIED ADS

FOR SALE - Rockwell R65C02 CPU chips. Replace the 6502 in your MTU-130 and get extra instructions like PHX, PLX, BRA, BIT#, abs-indirect addressing and more! $10 each. Keith Sproul, 1368 Noah Rd., North Brunswick, NJ 08902. (201)-246-3749.

INPUTS & OUTPUTS

Typesetting With the MTU-130

Bruce D. Carbrey
MTU

If you love the appearance of a professionally typeset document, but thought that it was too expensive and too time consuming to consider, read on!

If you have an MTU-130, WORDPIC, and a modem, you have just about everything you need to typeset a document for about one fourth to one fifth the cost of conventional typesetting. Several typesetting services now have fully automated digital phototypesetters which you can dial up and send your manuscripts to using AUTOTERM. Special codes embedded in your document select the typestyle, character size ("points"), width, etc. You have complete control over the final document. The typeset text is returned to you by mail, with many typesetters offering same-day or next-day shipment.

We have just started using one such typesetting service:

INTEGRAPHICS, Inc.
106-A South Columbus St.
Alexandria, VA 22314
(703) 683-9414

They have an excellent tutorial manual, "Personal Publishing", which tells you everything you need to know to do your own typesetting, available for $10.00. The cost to typeset a document is $2.00 per 1000 characters, with a five dollar minimum.

The INTEGRAPHICS system uses special embedded codes to control the typesetting process. All embedded codes are enclosed in curly brackets. For example {f385} tells the typesetter to use font number 385 (which happens to be Helvetica Bold,

which the headline above is set in). There are 400 different fonts (typefaces) on line in the INTEGRAPHICS system, which are shown in their manual. The commands {p10} {l12} {m42} select 10 point characters with 12 point line spacing and a column width ("measure") of 42 picas (one pica is 1/6th of an inch, and 12 points is 1 pica).

The macros available in WORDPIC are well suited for generating the necessary embedded codes. You can simply define one macro for each of the commonly needed code sequences, which can then be entered with only two keystrokes.

To send text to INTEGRAPHICS, you can simply prepare an ordinary WORDPIC text file, avoiding any special embedded characters (such as figure code lines) peculiar to WORDPIC. Then add the special control codes needed by the INTEGRAPHICS system using Macros. Before actually sending the file using AUTOTERM, there is one more required step. The WORDPIC text file needs to be converted to a form which supports the idiosyncrasies of the INTEGRAPHICS system (such as the fact that it totally ignores all ASCII control codes including carriage returns). I prepared a special utility program called PRETYPESET which does this conversion for you. In addition, it checks for common errors such as unbalanced { } bracket pairs and unprintable codes in the document. This program, including source code and instructions, is included on User Disk #4.

AUTOTERM can be used to transmit the file to INTEGRAPHICS. They have a toll-free number. You will receive the typeset text in a tube in about 3 working days (faster if you make special shipping arrangements). If corrections are required, you can make the changes and send the whole thing again, or only the part that needs changing.

The INTEGRAPHICS manual is extremely readable and is a good introduction to typesetting concepts even if you don't plan to use their service. You don't need any prior knowledge of typesetting at all. Everything is explained thoroughly. The manual even tells you how to do your own "pasteup", and makes suggestions for style and format selection. Further information about INTEGRAPHICS and the PRETYPESET program is included on User Disk #4.

New Graphic Utilities

Hal Chamberlin
MTU

In the process of improving figure support for WORDPIC 1.0, a new graphic image file format was defined and four new utility programs were written to help support it. In addition, the GREDIT and ANNOTATE graphic editor programs were modified to utilize the new graphic file format. Besides being included on the WORDPIC 1.0 distribution disk, these programs (including source) are also being put in the public domain on User Group disk #4 in order to encourage use of the new file format.

As you know, graphic files (.G files) are currently just dumps of the MTU-130 screen memory onto a standard "loadable" file using the CODOS SAVE command. For example, SAVE FILE.G C000:1 FBFF may be used to save the current screen image. Conversely, a GET command is used to read the file back into screen memory so it may be seen. Owners of Datamover 68000 boards may have tried saving the 1024x1024 hat image in a similar way using the DMXMON SAVE command. A disadvantage of this method (besides being relatively obscure) is that the image width must always be the screen width.

In designing WORDPIC's figure handling, it became clear that images of widely varying widths and heights were needed. Further, it is desirable for the graphic file itself to contain size and other information rather than requiring the user to remember and specify it whenever a figure was called for in the text. To meet these goals, a graphic file header was defined.

The graphic file header is much like the loadable file header that has been used all along and in fact is the same length (10 bytes). Instead of load address/entry

point information, it describes the image data in the file using this format:

BYTE	USE
0	Graphic file flag, $C7 = "G"+$80
1	Data type, $00 = straight bit image
2&3	Width in dots, low byte first
4&5	Height in dots, low byte first
6&7	Horizontal density in dots/inch
8&9	Vertical density in dots/inch
10-	Data

The graphic file flag simply identifies this as a graphic header as opposed to a memory image or other kind of header. The data type byte identifies how the graphic image is coded; 0 is for a straight uncompressed bit image with others (such as a compressed bit image and vector lists) to be defined later. The width and height entries give the image size in dots and can be any positive value (WORDPIC currently imposes a limit of 2047). The two density values are used by WORDPIC to indicate what density the printer should be set to when the figure is printed. Their ratio also establishes the aspect ratio for which the image was prepared. Following the header is the graphic data. For TYPE=0, it is bit images left-to-right and top-to-bottom. If the width is not divisible by 8, the last byte of each "scan line" is padded on the right with zeroes.

GCONVERT is used to convert your existing memory dump graphic files to the new header. You simply enter: GCONVERT <name> and it converts the file in-place. If it was a 6502 memory image (saved by CODOS) the width will be set to 480 and the height will be set to the file length divided by 60. If it was a 68000 memory image (saved by DMXMON), the program will ask you what the width is (1024 in the case of the hat image dump). It will then compute the height from the file length. The density is set to 72x72 in either case.

GSAVE is used to save some or all of the '130 screen as a new format graphic file. The simplest form is GSAVE <name> which will save the entire screen as a 480 wide by 256 high image file. GSAVE <name> NL will save the screen less function key legends (No Legends) as a 480 wide by 240 high image file. GSAVE <name> GC will allow you to interactively select the upper left and lower right corner of exactly what you wish to be saved with the crosshair (GRIN) cursor. The resulting width and height could be anything. Finally, GSAVE <name> Xl Xr Yt Yb can be used to explicitly define the borders of the saved image in X and Y coordinates. The screen area covered by GSAVE's operator prompts is actually saved and restored so that the image remains intact both in the file and on the screen. GSAVE runs entirely in the utility area of memory and uses no page zero memory (except for the pseudo registers) so it can be called from BASIC (with the SYSTEM command), C (using the system() function) and assembly (using SVC #13).

GGET is used to display the contents of a graphic file on the '130 screen. A simple GGET <name> will display the specified file starting at the upper left corner of the screen. If the image is smaller than the screen, that part of the screen not covered by the image will be undisturbed. If the image is larger than the screen, only the leftmost and/or topmost portion that will fit on the screen will be displayed. You can also specify horizontal and vertical offset arguments to control the image display. Positive arguments will put the upper left corner of the image further to the right or further down on the screen. Negative arguments allow you to see various parts of an image that is larger than the screen. Like GSAVE, GGET runs in the utility area of memory and can be called from a variety of languages.

GETDIM is used to display header information for a graphic file like GETLOC does for a memory image file. Entering GETDIM <name> will display the type, size, and density of the specified file or else inform you that the file does not have the new graphic header.

GREDIT has also been modified to produce the new graphic file when the SAVE IMG function is used. ANNOTATE has been similarly modified and also substantially

improved. The mystery bug has been fixed and you may now change character sizes and fonts without having to restart the program. The included font file now has lower case letters and three different matrix sizes: 5x7, 7x9, and 9x11, which when combined with the magnification feature gives an even wider variety of character sizes.

68000 Numeric I/O Routine

Bruce D. Carbrey
MTU

If you're one of the many proud new owners of MTU-130 DATAMOVERs who have been exploring the world of 68000 assembly language, you have probably discovered how easy the 68000 is to program, especially with DMXMON to give you an assist. DMXMON provides more than 70 built-in functions your program can call, for nearly every I/O and utility function you are likely to need.

However, there are four functions which you may have gotten used to having with 6502 CODOS which are absent from DMXMON. These are SVC numbers 8 through 11, which provide for conversion of numeric values to and from ASCII character strings, in decimal and in hex. These functions (as well as the 16-bit pseudo-processor) were omitted because, quoting from the DMXMON manual, "the 68000's more powerful instruction set makes these functions trivial to program directly." Well, OK, maybe "trivial" is a little strongly worded! Perhaps we should have said "easier". Anyway, it took me four hours to program and debug the the numeric I-O routines, so I am passing the results along to you on User Group Disk #4.

There are two routines on the 68000 source file NUM_IO_68K.A which perform the equivalent functions of CODOS SVC numbers 8 through 11, and then some. The "then some" part is that these routines can do conversions not just in hex and decimal, but in any base between 2 and 36. The two assembly language routines are ASC_VAL, which converts a numeric ASCII character string to the value it represents, and VAL_ASC, which generates an ASCII string given a value. The comments in the program explain the usage details. Briefly, you call the routines with the desired radix in D2, and the buffer pointer in A0. The value is passed in D0. For example, the following program segment inputs a decimal number from the keyboard and displays the equivalent value in hex:

```
MOVE     #1,D1      CHANNEL 1
SVC$     5          INPUT LINE
MOVE     #10,D2     SELECT BASE 10
MOVE.L   $0260,A0   BUFFER POINTER
JSR      ASC_VAL    GET VALUE IN D0
MOVE.L   $0264,A0   OUT BUF PTR.
MOVE     #16,D2     SELECT HEX
JSR      VAL_ASC    CONVERT TO STRING
MOVE.L   A0,D0
SUB.L    $0264,D0   COMPUTE CHAR COUNT
MOVE     #2,D1      CHANNEL 2
SVC$     6          OUTPUT STRING
```

Block Move for 68000

Bruce D. Carbrey
MTU

In harnessing the super speed of the Motorola 68000 processor on the MTU DATAMOVER board, one of the most common needs is a memory block move subroutine. Included on User Disk #4 is a block move routine I wrote after a fair amount of experimentation. The routine is written with the objective of moving large blocks of memory as quickly as possible without having an excessively large routine.

The BLKMOV routine can move a block of words of any size between any two addresses in memory (including a destination which overlaps the block being copied). The source and destination must be on word boundaries (even addresses). For large blocks, the routine averages 5.3 cycles per byte for moving from a higher to a lower address, or 5.8 cycles per byte for moving from a lower to a higher address. It takes longer in one direction than the other because the 68000 takes longer to do an

autodecrement operation than an autoincrement operation. Anyway, this is about 1.5 megabytes per second, more than 20 times faster than can be done on the 6502 processor.

Here's a challenge for you hackers out there: can you significantly improve this routine? If so, I'd like to hear about it! I've heard that MOVEM (move multiple registers) can be used for super-fast block moves by loading up all the registers from the source and then storing them into the destination, but I could not find a suitable combination of legal addressing modes to make it work. Can you?

MTU USER GROUP

User Group Disk #4 Contents

NOTE: If a particular submission is accompanied by an assembly source (.A extension), instruction (.T extension), or job (.J extension) file, they will be listed after "Accompanied by" at the end of the submission description.

Tools and Utilities

AKKAD.Z – submitted by Gordon Newby
A font with a few Akkadian characters for use with the LANGUAGE.B program below.

BLKMOV.A – submitted by Bruce Carbrey
The 68000 assembly language source for the BLKMOV routine described elsewhere in this newsletter.

DMXMONPATCH.J – submitted by Hal Chamberlin
A job file to patch existing bugs in the DMXMON.C program (standard with the DATAMOVER Board).

GCONVERT.C – submitted by Hal Chamberlin
A Utility program for converting an image file written to disk with a CODOS or DMXMON SAVE command to a bit-mapped graphics file with a graphics header. See description elsewhere in this newsletter. Accompanied by GCONVERT.A and GCONVERT.T.

GETDIM.C – submitted by Hal Chamberlin
A Utility program which displays the size and density information found in the graphics header of a bit-mapped graphics file. See description elsewhere in this newsletter. Accompanied by GETDIM.A and GETDIM.T.

GGET.C – submitted by Hal Chamberlin
A Utility program for loading a bit-mapped graphics file which contains a graphics header, such as file generated with the GSAVE Utility. See description elsewhere in this newsletter. Accompanied by GGET.A and GGET.T.

GSAVE.C – submitted by Hal Chamberlin
A Utility program to save the screen image as a bit-mapped graphics file containing a graphics header. See description elsewhere in this newsletter. Accompanied by GSAVE.A and GSAVE.T.

HEB.Z – submitted by Gordon Newby
A Hebrew font for use with the LANGUAGE.B program below.

NUM_IO_68K.A – submitted by Bruce Carbrey
The 68000 assembly language source code for the 68000 numeric I/O routines described elsewhere in this newsletter.

PRETYPESET.C – submitted by Bruce Carbrey
Prepares a WORDPIC file for typesetting by the INTEGRAPHICS phototypesetting service, as described elsewhere in this newsletter. Accompanied by PRETYPESET.A and PRETYPESET.T

PR_USING.C – submitted by Ralph O. Erickson
Provides MTU BASIC with the PRINT USING facility. The routine merges a mask string and a data string to form an output string. Adapted from C.R. Monclova's article in MICRO no. 58, p. 39, March 1983. Accompanied by PR_USING.A.

PRUSING1.B – submitted by Ralph O. Erickson
Demonstration of the use of PR_USING with a BASIC program.

PRUSING2.B – submitted by Ralph O. Erickson
Another demonstration of PR_USING.

SHARPCHECK.C – submitted by Bruce Carbrey
A utility program which checks for missing '#' characters in 6502 assembly language source files. Any numeric constant less than $100 that is not preceeded by a '#' character is flagged with a warning. Accompanied by SHARPCHECK.A.

STDFONT.Z – from MTU Demo Disk
The standard MTU System font, used by the LANGUAGE.B program.

TKL.Z – submitted by Ralph O. Erickson
Library enhancement of MTU BASIC to drive Tektronix 4662 vector-type plotter, for professional quality drawings and graphs. Accompanied by TKL.A.

TKLDEMO.B – submitted by Ralph O. Erickson
Draws a graph of a computed function with graduated and labelled axes, using the Tektronix 4662 plotter.

Applications

ANNOTATE.B – submitted by Hal Chamberlin
An improved version of the ANNOTATE.B BASIC program on User Group Disk #1. The program is used for adding headlines, captions, etc. to a screen image file. Accompanied by ANNOTATE.T.

STDVBFONTS.S – submitted by Hal Chamberlin and Bruce Carbrey
A set of three complete upper and lower case fonts for use with ANNOTATE.B program. Each font provides a different dot matrix size, i.e 5X7, 7X9, and 9X11.

Entertainment and Demo

AESCULUS.B – submitted by Ralph O. Erickson
Draws attractive patterns, some resembling horse-chestnut, or palm leaves.

ARITH3.B – submitted by Gordon Newby
A BASIC program which does arithmetic drills. Includes drills for add, subtract, multiply, divide, approximation, and square-root. Accompanied by MATH.J.

CANARY.B – submitted by Ralph O. Erickson
Let your computer sing like a bird.

FIFTEEN.B – submitted by Ralph O. Erickson
Adaptation for MTU-130 of Asakawa's Number Square puzzle. Easy. A demonstration of the facilities of the Keyword Graphics Library, KGL.Z.

LANGUAGE.B – submitted by Gordon Newby
A BASIC program which demos the use of special fonts. This program will allow you to display and enter some text using an Akkadian or Hebrew font. Accompanied by LANGUAGE.J.

LEGEND.J – submitted by Gordon Newby
A CODOS job file which sets up legends to startup the LANGUAGE.B and ARITH3.B programs.

Ordering User Group Disks

All of the programs listed in the User Group Disk #4 Contents can be running on your system for only $15! In order to keep costs to a minimum we will only be able to fill prepaid orders or orders which charge the purchase to Visa or MasterCard. Why not submit an article or program on disk so you can get the next User Group disk free?

Programs submitted by MTU-130 users on this disk are provided as is, unchecked by MTU. The MTU-130 User Group and MTU disclaims responsibility for any direct or consequential damages incurred in using any of these programs. User Group programs are believed but not verified to be free of patent and copyright restrictions. Any operational questions about User Group programs should be directed to the submitter. Any questions about copyright infringement should be directed to MTU.

Submitting Articles and Programs

We welcome input from all of our users whether that input be programs, hints, music, pictures or any other item which you think may be useful and/or interesting to other MTU-130 users.

Programs should be accompanied by a separate text file containing a one paragraph abstract of the program suitable for publishing in the next newsletter as well as specific operating instructions for your program. Articles or other text should be written using a maximum width of 43 columns, one space before the first word in the paragraph, and a blank line between paragraphs.

All programs submitted to MTU will be considered in the public domain and the user submitting the program is requested to include a statement regarding the original author if other than the submitter as well as any known previous publication history.

When you submit material on a diskette, we will copy that material from your disk and then copy the current user disk contents to your disk before returning it to you, so you will get the User Disk for free!

IN THE QUEUE

10 Meg Disk for the '130!

Under development in the dark engineering caverns at MTU is a high capacity, removable cartridge disk system for the '130. The disk drive uses 10MB (formatted capacity) removable cartridges. The drives are identical in size to the 8" floppy disk drives used on the '130 and the cartridges measure about 8.5" by 11" by .5" thick. In fact, the drives will fit into the standard '130 disk drive cabinet although the power supply is somewhat different.

Actually, the disk media is "rigid" only when it is spinning at 1500RPM; it is really flexible mylar media. This means that a "head crash" is physically impossible and thus the system is as rugged as a floppy disk system. Since dust particles on the disk surface are not a problem, the drive needs no air filters thus cartridges can be changed (stop, swap, and start) in less than 5 seconds. This "rigid floppy" technology is so good that it achieves bit densities (24K/inch!) on oxide-coated flexible media nearly twice what has been achieved on plated rigid media!

Even though flexible media is used, the performance is impressive. A voice-coil positioner gives a full-stroke seek time of just 75MS (35MS average) which is several times faster than the stepper-motor driven seek of most other rigid disk drives. The burst data transfer rate is 1.13MB/second which is nearly twice as fast. Up to 4 drives may be connected to a single system for 40 megabytes of formatted storage.

The disk system has its own integrated intelligent controller which presents an SCSI interface to the host computer. SCSI is an acronym for Small Computer Standard Interface and is also known as the SASI interface. The SCSI interface is an official standard which is offered by several disk drive manufacturers. Since the disk controller is intelligent, most of the low level disk functions such as error correction and retry, mapping block numbers to track/sector numbers, read-after-write verification, seeking, etc. are handled automatically.

It is actually possible to communicate with an SCSI interface through the '130's parallel I/O port. In fact the drives are being acceptance tested in this manner. However, in order to maximize performance and provide room for an expanded CODOS to support high capacity disks, a simple interface board is being developed. This board will have 32K of on-board memory addressed from $0000 to $7FFF in Bank 1 and a DMA interface from the high capacity disk into that memory. It will also have the necessary registers and latches to simplify programming of the SCSI interface. CODOS will keep disk buffers and much of its program code in this memory which may actually free up space in Bank 0 for a better SVC processor and other enhancements.

Backup is always an issue with high capacity disk systems. Since the cartridges are removable, backup is easily accomplished in dual-drive systems by simply copying the cartridge, which takes about 2 minutes. In single drive systems, backup to floppies can be used. All 10MB can be copied to 8 double-sided floppies in less than 5 minutes. For selective file backup, the CODOS COPYF utility can be used.

Cost for the high capacity disk system will be approximately $3000 for the cabinet, first drive, controller, and interface board. The second and fourth drives will be about $2000 while the third drive (which requires a second cabinet) will be about $2300. Cartridges will be $89.

As with the color graphics board, we would appreciate feedback from Update readers prior to finalizing the design. What is your application? Is 10MB enough capacity? How do you feel about the backup issue? How should CODOS be modified to best support high capacity disks?

Color Graphics for the '130

You've asked for a high resolution color graphics board and now we're proposing one! After much research and headscratching, MTU Engineering has determined that the following features are possible in an MTU-130 compatible color graphics board:

1. The board can plug directly into the MTU-130 bus.

2. Resolution can be 768 pixels wide by 512 pixels high with 4 bit planes. This would allow 51 lines of 96 characters when using an 8x10 cell.

3. The 4 bit planes may be used for 8 colors of two intensities each or 16 shades of gray.

4. Refresh memory (non-expandable) can be 256K bytes of which 192K is actually visible when all 4 bit planes are enabled.

5. An on-board 68000 can be used as a programmable image processor for much greater flexibility than existing graphics chips provide.

6. With all bit planes enabled for display, the 68000 can run at 60-70% of its 8MHz speed.

7. The on-board memory may also be accessed in banks 2 and 3 by the '130's 6502 processor. An enable/disable bit allows its use along with Datamover and Programmover boards.

8. To accomodate the board's high resolution, a monitor with 31KHz horizontal sweep will be required. For color, a red-blue-green input monitor with fine pitch shadow mask is required.

9. Estimated cost is $1500-$2000 for the board, $300-$500 for a monochrome monitor, or $700-$1000 for a color monitor.

Now we need your help. What do you think of the proposed design? What is your application for the board? Is the resolution high enough for your anticipated application? Is the specialized monitor required for that level of resolution a severe disadvantage? Is 16 colors or shades of gray sufficient? How do you like the 68000 as a graphics processor? Remember that more features=higher cost.

Over the next month we will carefully consider all responses received before the detailed design work is started.

Digisound 16

DigiSound-16 will be a high-fidelity audio A-to-D and D-to-A converter for the MTU-130 and other computers. Typical applications include professional quality delayed playback music synthesis, audio signal acquisition and analysis in research, audio signal enhancement (such as noise removal), and as a component in a low-cost digital audio production studio.

The primary advantages of DigiSound-16 over standard A-to-D and D-to-A converters is its specialized audio processing circuitry and very large first-in-first-out (FIFO) buffer. This buffer allows uniform sampling rates with non-uniform attention from the host computer system. For some applications involving "single-shot" signal acquisition and analysis, the buffer alone may be sufficient to hold the signal. For others, the buffer makes continuous high speed data transfer to/from disk possible.

The prototypes currently under construction incorporates the following features:

1. Stand-alone packaging with an internal power supply.
2. Universal digital I/O interface connectable to any computer via parallel port or simple bus interface.
3. Large internal FIFO buffer, 32K 16 bit samples.
4. 16 bit linear, 12 bit companded, and 8 bit companded data formats.
5. Monophonic or stereo operation.
6. Maximum sample rate of 50KHZ in stereo mode and 100KHz in mono mode.
7. Internal programmable sample clock with sample rates from 3.5KHz to 100KHz including the 44.056KHz commercial standard and 48.000KHz professional standard plus an external clock.
8. Plug-in filter boards, separate for A-to-D and D-to-A and each channel.
9. Several digital inputs and outputs are included for connection to the user's equipment.
10. Provision for depopulation to D-to-A only, A-to-D only, and mono only for reduced costs.

Cost is expected to be under $1000 for stereo D-to-A only with two filter boards and under $2000 for the complete unit including A-to-D and 4 filter boards. Additional filter boards should be approximately $100 each. A complete 16 page specification for DigiSound-16 is available on request.

★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★

YOUR AD HERE!

Yes! We are now accepting advertising space in **UPDATE**. *If you've got a quality program or accessory for the MTU-130, here's your chance to reach 100 percent of the aftermarket at bargain rates!*

Full Page *(7"h x 9"v)*	***$50***
½ Page *(3.25"h x 9"v or 7"h x 4.5"v)*	***$30***
¼ Page *(3.25"h x 4.5"v)*	***$18***

Classified Ad*: 2¢ per character (including blanks between words), $3.00 minimum.*

All display ads must be camera-ready artwork. Classified ads may be typed or handwritten. All ads must be ***prepaid****. Make checks payable to Micro Technology Unlimited. Direct all advertisements to:*

Micro Technology Unlimited
Newsletter Advertising Dept.
2806 Hillsborough St.
PO Box 12106
Raleigh, NC 27605

DEADLINE FOR NEXT ISSUE: AUG. 19, 1983

☎ *(919) 833-1458*

Note: We reserve the right to reject any ad which we deem to be inappropriate to the newsletter.

★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★

****

MTU-130 USERS

* Full Support for your IDS Prism Printer

* Conversions: MTU <---> IBM 9 Track Tape

* Call for other special services available

Corporate Micro Solutions, Inc.
545 Fifth Ave.
Suite 109
New York, NY 10017

(212)-222-6689

****

****

FACTORY CLEARANCE!

The introduction of the new MTU-140 computer shown on the cover of this newsletter has enabled us to offer to existing MTU-130 owners and their friends, for a limited time only, a dramatic price reduction on our existing stock of present-style MTU-130s:

MTU-130-1D (regularly $3300.00): $2600.00
MTU-130-2D (regularly $3900.00): $3100.00

These special prices are in effect until August 1, 1983, or until our present, limited stock is exhausted. Orders will be accepted on a first-come, first-served basis. These models are fully warranted, and the MTU User Referral Plan is in effect (see UPDATE #2 or call for details). Order soon to insure your requirements are met.

Micro Technology Unlimited (919)-833-1458

****

Micro Technology Unlimited
2806 Hillsborough Street
P.O. Box 12106
Raleigh, N.C. 27605 U.S.A.
(919) 833-1458