ATARI UTILITIES DISKETTE

This utilities diskette is distributed to outside software developers to facilitate their efforts. This software is not in the public domain and is not to be distributed. The utilities in this package are:

DOS	SYS	039	DOS 2.0S
DUP	SYS	042	DOS 2.0S
BUILD24		010	builds self-booting BASIC programs
DIV	SRC	026	an integer divide utility
BMUL	SRC	020	a signed integer multiply routine
SMUL	SRC	027	unsigned integer multiply routine
CHRGEN		049	a simple character set editor
SOUND		095	a sound editor
XREF		052	cross-reference program for BASIC
MASHER		045	compresses BASIC programs
RENUM	BAS	030	renumbers BASIC programs
FORMTR		004	formats LIST files from ASSEMBLER
M8TXT		005	BASIC mode 8 character print routine
BCDSAV		004	makes fixed-length number records
HOBBY1		008	simple player-missile graphics demo
HOBBY2		005	simple display list interrupt demo
SCRL19	ASM	104	full fine scrolling module
SCRL19	OBJ	011	object code
FIX	OBJ	042	a disk sector utility
SCRLH	DEM	004	simple horizontal scroll demo
MDIR	OBJ	002	gives disk directories from BASIC
HORSE20	BAS	047	a character set animation demo
AUTORUN	SYS	001	the R: device handler
MDLI02	BAS	005	multiple display list interrupt demo
DEMO3	BOK	005	player priority trick
DEMO4	BOK	004	player as special character demo
DLISTA	DEM	005	alternating display list demo
SCRLF	DEM	003	simple fine scroll demo
SCRLV	DEM	004	simple vertical scroll demo

None of these programs are finished products; they all contain rough spots. If you make improvements please send a copy to our library.

DIV

This routine is composed of two assembly subroutines: Unsigned Divide of 32/16 bits and Signed Divide of 16/16 bits.

```
     Unsigned Divide:

          Enter with A,Y = Divisor (A=MSB)
                  ACC,MQ = Dividend

          Exit with  ACC = Remainder
                      MQ = Quotient

     Signed Divide:

          Enter with A,Y = Divisor (A=MSB)
                     ACC = Dividend

          Exit with  ACC = Remainder
                      MQ = Quotient
```

SMUL

This routine is composed of a 16 * 16 Unsigned Multiply and a 16 * 16 Signed Multiply.

```
     Unsigned Multiply:

          Enter with A,Y = MULTIPLIER
                     ENT = MULTIPLICAND
                     ACC = ADDEND
                (ACC,MQ) = ([A,Y]*ENT)+ACC
                (This way for chained operaton)
                 556-748 Cycles

     Signed Multiply:

          Enter with A,Y = MULTIPLIER (A=MSB)
                     ACC = MULTIPLICAND

          Exit with  ACC,MQ = PRODUCT
                        ACC = MSW

          584-821 Cycles
```

PMMOVE

This program uses a machine language routine to move Players and Missiles. The routine is called by:

```
USR(ADR(MOVE$),PMNUMB,XPOS,YPOS)
```

PMNUMB refers to the Player (0-3) or Missile (4-7) to move. The Missile number is determined by PMNUMB-4. XPOS and YPOS are the X and Y coordinates to place PM. A Relocator is used to make the PM Move routine wherever it may be placed in RAM by BASIC.

BINARY ROUTINE

This routine is included as part of the PMMOVE routine. BINARY loads or saves a binary file from BASIC.

```
On entry  CMD=7    means load a file
          CMD=11   means save a file
          STADR=   the address to load or save a file from
          BYTES=   the number of bytes to save or load
          IOCB=    the IOCB to use
          FILE$=   file name to load

On exit   ERROR=1  means successful load
          ERROR<>1 means it's an error status
```

BMUL

This routine implements BOOTHS ALGORITHM for multiplication of SIGNED binary numbers in TWO's-COMPLEMENT notation.

The MULTIPLIER is shifted to the right with the PRODUCT (as usual). Each CHANGE of the MULTIPLIER bits from zero to one causes the MULTIPLICAND to be subtracted from the PRODUCT. Each change from one to zero causes it to be added.

Like most signed arithmetic, it cannot be chained and is prone to overflow problems when given -32768, but it is smaller than the absolute-kludge wrapped around an unsigned mutiply, and not much slower (633-945 cycles).

```
Enter with     A,Y = MULTIPLIER (A=MSB)
               ACC = MULTIPLICAND

                16 * 16  SIGNED MULTIPLY

Exit with   ACC,MQ = 32 BIT PRODUCT
               ACC = MSW
                MQ = LSW

    ACC = 212
     MQ = $0CB
    ENT = $0CD
     SC = $0CF
      * = $600
```

M8TXT

M8TXT demonstrates how to mix text with graphics in BASIC mode 8. It plots the characters bit by bit onto the mode 8 screen. The technique can be adapted to any program using BASIC mode 8 displays.

BCDSAV

This program provides an alternative to the variable length records obtained when a program PRINTs values to the disk. Using the variable table values, it stores BCD values of numbers to the disk in fixed length records.

HOBBY1

This is a simple player-missile graphics demo. It sets up a player and then moves it around with the joystick. Since this is a pure BASIC program, the vertical motion of the player is too slow. An assembly language routine is necessary to get proper high-speed, motion.

HOBBY2

This is a simple display list interrupt demo program. The bottom half of the screen changes from blue to pink.

BUILD24

This program creates an AUTORUN.SYS file that will start up your BASIC programs. It asks you for a BASIC command; the command you enter will be in the AUTORUN.SYS file and will be executed on powerup. Normally your command will be of the form RUN"D:PROG.BAS". It must be less than 128 characters long. Remember to put in the terminating double quotation mark (").

MDIR

This program is an object file that can be called from BASIC to put the disk directory onto the screen. To use it, you must first load it into RAM with the BINARY LOAD command (L) in the DOS. Call it from BASIC with A=USR(1536).

AUTORUN.SYS

This program is an RS-232 handler file. It allows you to use the R: device. It is booted in automatically on powerup.

HORSE20.BAS

This is the running horse demo. It demonstrates the power of character set graphics and animation.

SCRLH.DEM

This is a simple horizontal coarse scroll demo.

SCRLV.DEM

This is a simple vertical coarse scroll demo.

SCRLF.DEM

This is a simple fine scroll demo.

MDLI02.BAS

This is a multiple display list interrupt demo. It puts gobs of color onto the screen. Do not be alarmed if the screen goes black; it takes several minutes to finish. Once it is running observe how keypresses affect the display. Also note the greatly reduced computation speed of BASIC. With so many interrupts happening, the 6502 has little time for other activities.

DEMO3.BOK

This program demonstrates a technique for increasing the resolution of a stationary player by hiding it behind a playfield cutout.

DEMO4.BAS

This program shows how a player can be used as an extended character.

DLISTA.DEM

This program demonstrates the alternating display list technique.

CHARACTER EDITOR

This program is a character editor that makes it easier to take advantage of the redefinable character set capability of the Atari. It gives you the capability to edit, load, or save character sets.

The first menu option is to create or edit a character set. If you enter this option without first loading a character set, it will default to a blank character set. You edit characters with the joystick, selecting a pixel position with the stick and the status of that pixel (on or off) with the button. When you are done editing a character, you can allocate it to a character position with a keystroke indication. Fear not, most operations are prompted, so that you can pick your way through the program rather well. The only blooper is that the prompt for an I/O operation to the disk requires that you give the D1: prefix.

This program is usable but not at all as practical as IRIDIS's FONTEDIT program. I strongly urge you to buy and study the IRIDIS program if you want to do any character set work. If only it had a display list interrupt....

SOUND EDITOR

The sound editor helps you develop new sounds. It is not appropriate for developing tunes or jingles, or any long sound. It is designed for developing short sounds (1 second long) such as clangs, croaks, rattles, and other such nonsense. It only edits two of the four sound channels.

The program needs very little external documentation, as its title page describes the commands. The joystick response is slow, but you can use the 'fast' command to speed it up and then use the normal speed to fine tune your sound. You should read the hardware manual to get an idea of what the sound registers do.

Don't overlook the possibilities this program opens up. I have heard some very convincing sound effects created with it; it just takes a little imagination.

CROSS REFERENCE UTILITY

This utility provides a cross-reference of variables and constants in a BASIC program. It requires at least 40K of RAM, a printer, and a disk. The BASIC program to be cross-referenced should be on a diskette in SAVE format. The program first gives a count of the total number of variables used. For each variable, it lists all line numbers containing references to the variable. It also gives a count of how many times each constant is used. If an error occurs during printer output, you may recover by typing GOTO 3050.

RENUMBER

This program will renumber your BASIC program on disk. The target program must be on the diskette in LIST format. The program prompts you for the values it needs. The 'input device' will normally be D:progname. The 'output device' will normally be D:newname. The 'starting number' is the new starting line number. 'From' and 'to' are the beginning and ending line numbers of the section of code you want renumbered.

FORMATTER

This program will format the output of your Assembler/Editor cartridge so that you can get a list file that looks good out of your Atari 825 printer. Make the first instruction of your assembly code a .OPT NOEJECT. Then assemble the list file to the diskette. Then drop into BASIC and run this program. Respond to the name prompt with D:progname. This program is designed for use with the Atari 825 printer, so good luck with anything else.