MICRO TECHNOLOGY UNLIMITED

APPLICATION NOTE #2

USE OF THE K-1008 VISIBLE MEMORY FOR GRAY SCALE DISPLAY

DATE: MAY 1979

PRICE:	SINGLE BOARD PRICE:	SLAVE BOARD	$270.00
		MASTER BOARD	$270.00
	3 BOARD SET:		$750.00
	4 BOARD SET:		$1,000.00

PRICE GOOD THROUGH DECEMBER 1979.

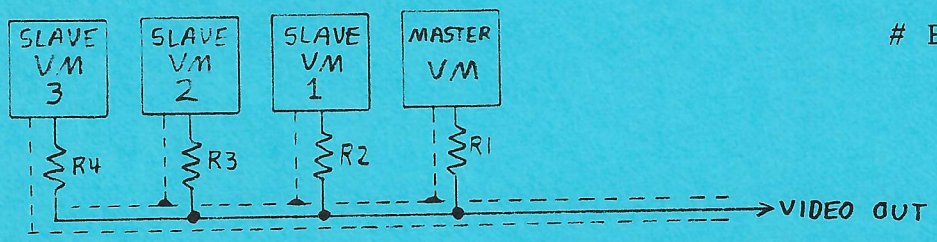
Gray Scale and Color With the Visible Memory

MTU Application Note 2. 8/78

Although a single model K-1008 Visible Memory provides a display capability unmatched anywhere in the KIM/6502 accessory market, there are many applications where various shades of gray or even colors need to be displayed. The majority of these applications can be satisfied by synchronizing 2 or more Visible Memories together so that more than one bit per pixel is available. Most applications will call for two to four boards to be combined but an even greater number could be accomodated if a buffered backplane bus is available. Up to 12 for a full 4096 color display is feasible. Please contact MTU for possible discounts on the purchase of 5 or more boards.

The method employed for synchronizing the boards is termed master-slave. One board is designated as the master. It generates a special synchronization signal (via added logic) that is sent to the slaves. There can be only one master in a set of synchronized boards. The remainder of the boards are slaves. They are specially modified to accept the synchronization signal from the master in lieu of their internal timing. The modifications made to a board to make it a master in no way affects its normal operation. The slave modifications however make it dependent on the master for its timing and therefore it will no longer run unless connected to a master.

Once the boards have been synchronized, it is necessary to combine their video outputs into the form required by the display monitor. For gray scale the combination is easy and is accomplished by making a special video cable according to the schematic below:



# Boards	R1	R2	R3	R4
2	39	150		
3	56	180	470	
4	62	200	470	1K

Essentially an ultra high speed digital-to-analog converter is formed which produces the gray scale voltages. Since the video sync pulses are in phase on all of the boards, they emerge unmodified by the combination network.

When set up for gray scale, each pixel position on the screen is addressed in each visible memory as before. Thus assuming that the set of VM boards is addressed at 6000, 8000, A000, etc., the upper left pixel on the screen is influenced by what is stored in location 6000 bit 7, 8000 bit 7, A000 bit 7, etc. for as many boards as are tied together. A good convention to follow is to assign the master VM the lowest address and the slaves successively higher addresses. Also when the special mixing cable is plugged into the boards, the smallest resistor should be associated with the master, the next smallest with the slave addressed just beyond the master, etc. When this is done, the gray scale is as below:

GRAY LEVEL	6000	8000	A000	GRAY LEVEL	6000	8000	A000
Black	0	0	0	Somewhat Dim	1	0	0
Faint	0	0	1	Bright	1	0	1
Very Dim	0	1	0	Quite Bright	1	1	0
Dim	0	1	1	Very Bright	1	1	1

If 4 boards are tied together, the additional brightness levels are inserted between each of those above plus a Blinding level is added.

Combining the VM outputs together for color suitable for a standard NTSC color monitor is, unfortunately, not so simple. A special, rather complex, circuit called an NTSC Encoder is necessary to insert the color subcarrier and properly modulate it with the color information. Also the encoding process destroys a considerable amount of resolution, particularly in the reds.

The recommended method of implementing a color display requires modification of the color monitor or TV set for direct Red Blue Green input. If a TV is being modified, the three color amplifier inputs will have to be made available and each connected to a corresponding VM board. Also the sync separator will have to be connected to one of the VM boards (doesn't matter which one) for proper synchronization. If a fourth VM board is being used, it should be connected to the luminance amplifier so that 7 colors with two levels of brightness can be displayed.

MASTER Theory of Operation

The master modifications essentially generate a pulse which is coincident with all of the flip-flops and counters in the counter chain being zero. This pulse when distributed to slave boards will force their counters to reset at the same time thus synchronize them with the master.

The added 74LS30 (a plain 7430 can also be used) detects when all of the counters have reached their "terminal count", that is, the count just prior to them all resetting. Although the counter chain is of the ripple carry type, there is absolutely no chance of a decoding spike when decoding this particular state. This is demonstrated on the Master timing diagram.

When the next Dot Clock comes, the counters start ripple resetting with DOT 1 resetting first. This terminates the terminal count detect from the 74LS30 which then causes the 74121 to fire producing a pulse of approximately 30NS. This pulse is guarenteed to start after the resetting commences and terminate before the next Dot Clock pulse.

SLAVE Theory of Operation

Essentially all that the slave modification must accomplish is separation of the counter reset lines from ground and internal slave reset logic. The reset lines are then tied together and to the syncronization pulse from the master. After this separation is done, the slave is no longer able to operate without the "pacemaker" signal from the master.

Adjustments

For best display quality, particularly with 3 or 4 boards tied for gray scale, the Dot Sync adjustment of each board should be checked with a scope. They should be adjusted so that the dot timing from each board is exactly in phase with all of the others. If they are not in phase, there will be an excessive black or white dot marking some transitions between adjacent levels of gray. Any remaining dot can be reduced by replacing R17 and R18 with 560 ohm resistors and R15 with a 39 ohm resistor.

Modification Service

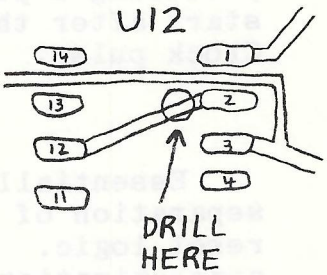
For those customers who are not dyed in the wool hackers, MTU has a board modification service. Please write or call for details.

MODIFICATIONS TO MAKE A MASTER VM BOARD

__1. Obtain a 74LS30 and a 74121 type integrated circuit.
__2. Mount these IC's near the video output jack. They may be glued topside down on top of the board or mounted in sockets held in a small piece of vectorboard glued to the back edge of the board (same edge as the video output jack).
__3. Make the following connections. U99 is the added 74LS30 and U98 is the added 74121.
- __ A. Run a wire from U45-16 to U99-14
- __ B. Run a wire from U44-14 to U98-14
- __ C. Run a wire from U45-8 to U99-7
- __ D. Run a wire from U44-7 to U98-7
- __ E. Connect a .1uF bypass capacitor between U98-14 and U98-7
- __ F. Run a wire from U99-1 to U45-6
- __ G. Run a wire from U99-2 to U47-2
- __ H. Run a wire from U99-3 to U32-1
- __ I. Run a wire from U99-4 to U29-1
- __ J. Run a wire from U99-5 to U12-11
- __ K. Run a wire from U99-6 to U12-13
- __ L. Run a wire from U99-11 to U99-12
- __ M. Run a wire from U99-11 to U12-10
- __ N. Run a wire from U99-8 to U98-5
- __ O. Run a wire from U98-3 to U98-4
- __ P. Run a wire from U98-4 to U98-7
- __ Q. Run a wire from U98-9 to U98-14

__4. U98-6 is the synchronization signal that must connect to the slave Visible Memories. It may just hang off the board or be connected to a plug and socket arrangement that mates with the slaves.

MODIFICATIONS TO MAKE A SLAVE VM BOARD



__1. U12-2 must be disconnected from ground. This is most easily accomplished by carefully drilling a 1/16" hole beside U12-2 as shown to the right. Alternatively, the pin may be cut away from the board and bent out horizontally.
__2. Cut line going from U30-15 to R10
__3. Cut line going from U30-15 to U30-2
__4. Cut line going from U31-3 to U46-1
__5. Cut line going from U31-3 to U32-2
__6. Connect a wire from U46-2 to U30-15
__7. Connect a wire from U30-2 to R10 (end cut away in step 2)
__8. Connect a wire from U12-2 to U12-12 (drilled away in step 1)
__9. Connect a wire from U46-1 to U32-2
__10. Connect a wire from U32-12 to U12-12
__11. U46-1 is to be connected to the synchronization signal from the master. A wire may just hang off the board or it may be connected to a plug ans socket arrangement that mates with the master and other slaves, if any.

NOTE With these modifications, a slave board will not function at all if not connected to a master. Memory operation may be restored by grounding the slave's sync input but normal standalone graphic operation can only be restored by removing the modifications.

```
                        .PAGE  'EQUATES AND DATA STORAGE'
                ;       PATTERN GENERATOR FOR GRAY SCALE SETUP OF TWO OR MORE K-1008
                ;       VISIBLE MEMORY BOARDS.
                ;           THE FIRST PATTERN GENERATOR PRODUCES A GRAY SCALE CHECKER-
                ;       BOARD PATTERN.  ALL SQUARES ARE THE SAME SIZE BUT THE SIZE IN
                ;       EACH PASS IS DETERMINED BY A RANDOM NUMBER GENERATOR.  THE
                ;       SQUARES GET BRIGHTER FROM LEFT TO RIGHT AND TOP TO BOTTOM.
                ;       THIS IS A PATTERN GENERATOR ROUTINE ONLY, NO
                ;       MEMORY FUNCTION TEST IF PERFORMED.


                ;       KIM SYSTEM EQUATES

1C22            KIMMON  =       X'1C22          ; ADDRESS OF SAVE MACHINE STATE ENTRY POINT
2000            VMSIZ   =       8192            ; SIZE OF EACH VISABLE MEMORY BOARD

                ;       BASE PAGE DATA STORAGE
0000                    .=      0

                ;       MAIN PROGRAM DATA STORAGE

0000 00         T1ITCT: .BYTE   0               ; ITERATION COUNT FOR PATTERN 1
0001 0060       VMORG:  .WORD   X'6000          ; ADDRESS OF FIRST VM BOARD (MOST
                                                ; SIGNIFICANT)
0003 04         VMNO:   .BYTE   4               ; COUNT OF VISIBLE MEMORIES CONNECTED
                                                ; TOGETHER


                ;       DATA STORAGE FOR CHECKERBOARD PATTERN

0004 0000       VMADR:  .WORD   0               ; ADDRESS POINTER FOR VM DATA MANIPULATION
0006 00         VMDATA: .BYTE   0               ; DATA DESTINED FOR VM
0007 00         CKXSZ:  .BYTE   0               ; X SIZE (WIDTH) OF CHECKER RECTANGLE
0008 00         CKYSZ:  .BYTE   0               ; Y SIZE (HEIGHT) OF CHECKER RECTANGLE
0009 00         CKDTA:  .BYTE   0               ; COLOR OF UPPER LEFT CHECKER RECTANGLE
000A 00         CKDTAX: .BYTE   0               ; WORK COLOR DURING HORIZONTAL SCAN
000B 00         CKDTAY: .BYTE   0               ; WORK COLOR DURING VERTICAL SCAN
000C 00         CKYCT:  .BYTE   0               ; COUNT OF CHECKER HEIGHT DURING VERTICAL
                                                ; SCAN
000D 00         HBYTCT: .BYTE   0               ; BYTE COUNT DURING HORIZONTAL SCAN
000E 00         VMCNT:  .BYTE   0               ; COUNT OF VM BOARDS PROCESSED
000F 00         VMPGCT: .BYTE   0               ; COUNT OF VM PAGES PROCESSED
0010 D204       RANDNO: .WORD   1234            ; RANDOM NUMBER REGISTER
```

```
                                .PAGE  'MAIN PATTERN GENERATION PROGRAM'
 47 0012                        .=     X'200          ; START PROGRAM CODE AT 200
 48
 49 0200 A9E0       MTEST:      LDA    #X'E0          ; INITIALIZE STACK POINTER
 50 0202 9A                     TXS
 51 0203 D8                     CLD                   ; INSURE BINARY ARITHMETIC
 52
 53                 ;           PATTERN 1, 16 GRAYSCALE CHECKERBOARD PATTERS.
 54
 55 0204 A910       MAIN:       LDA    #16            ; SET 16 ITERATION COUNT
 56 0206 8500                   STA    T1ITCT
 57 0208 20B702     MAIN1:      JSR    RNDEXP         ; GET AN EXPONENTIALLY DISTRIBUTED RANDOM
 58                                                   ; NUMBER IN A
 59 020B 20D102                 JSR    CKSZAD         ; ADJUST ACCORDING TO NUMBER OF VM'S
 60 020E F0F8                   BEQ    MAIN1          ; MUST NOT BE ZERO
 61 0210 8507                   STA    CKXSZ          ; MAKE IT THE X CHECKER SIZE
 62 0212 20B702                 JSR    RNDEXP         ; GET ANOTHER
 63 0215 20D102                 JSR    CKSZAD         ; ADJUST IT
 64 0218 F0EE                   BEQ    MAIN1          ; MUST NOT BE ZERO
 65 021A 8508                   STA    CKYSZ          ; MAKE IT THE Y CHECKER SIZE
 66 021C A503                   LDA    VMNO           ; INITIALIZE VM BOARD COUNT
 67 021E 850E                   STA    VMCNT
 68 0220 A501                   LDA    VMORG          ; INITIALIZE VMADDR TO ADDRESS OF MOST
 69 0222 8504                   STA    VMADR          ; SIGNIFICANT VM
 70 0224 A502                   LDA    VMORG+1
 71 0226 8505                   STA    VMADR+1
 72 0228 A900       MAIN2:      LDA    #0             ; SET INITIAL COLOR TO BLACK
 73 022A 8509                   STA    CKDTA
 74 022C 204F02                 JSR    CKGEN          ; GENERATE A CHECKERBOARD
 75 022F 4607                   LSR    CKXSZ          ; PREPARE FOR NEXT MOST SIGNIFICANT
 76 0231 4608                   LSR    CKYSZ          ; CHECKERBOARD, DIVIDE CKXSZ AND CKYSZ BY 2
 77 0233 C60E                   DEC    VMCNT          ; TEST IF ALL VM BOARDS DONE
 78 0235 D0F1                   BNE    MAIN2          ; GO GENERATE SQUARES ON NEXT MOST SIG VM
 79 0237 A200                   LDX    #0             ; WAIT AWHILE
 80 0239 A000       MAIN3:      LDY    #0
 81 023B A904       MAIN4:      LDA    #4
 82 023D 18         MAIN5:      CLC
 83 023E 69FF                   ADC    #-1
 84 0240 D0FB                   BNE    MAIN5
 85 0242 88                     DEY
 86 0243 D0F6                   BNE    MAIN4
 87 0245 CA                     DEX
 88 0246 D0F1                   BNE    MAIN3
 89 0248 C600                   DEC    T1ITCT         ; DECREMENT AND CHECK 16 ITERATION COUNTER
 90 024A D0BC                   BNE    MAIN1          ; LOOP UNTIL 16 ITERATIONS DONE
 91 024C 4C0402                 JMP    MAIN           ; CONTINUE
 92
```

```
                                .PAGE  'CHECKERBOARD PATTERN GENERATOR ROUTINES'
  93                    ;       CHECKERBOARD PATTERN GENERATOR
  94                    ;       STARTS AT UPPER LEFT CORNER OF SCREEN AND GENERATES A CHECKER-
  95                    ;       BOARD PATTERN.
  96                    ;       ENTER WITH CKXSZ SET TO CHECKER SQUARE WIDTH AND CKYSZ SET TO
  97                    ;       CHECKER SQUARE HEIGHT AND CKDTA SET TO 0 FOR A BLACK UPPER LEF
  98                    ;       SQUARE OR SET TO X'FF FOR A WHITE UPPER LEFT SQUARE.
  99                    ;       USES ALL REGISTERS, PRESERVES CKXSZ, CKYSZ, CKDTA
 100
 101 024F A920      CKGEN:  LDA     #VMSIZ/256      ; SET NUMBER OF VM PAGES TO FILL
 102 0251 850F              STA     VMPGCT
 103 0253 A008              LDY     #8              ; INITIALIZE BIT COUNT
 104 0255 A509              LDA     CKDTA           ; COPY CKDTA TO VERTICAL WORK LOCATION
 105 0257 850B              STA     CKDTAY
 106
 107                    ;       START A ROW OF CHECKER BLOCKS
 108
 109 0259 A508      CKGNV:  LDA     CKYSZ           ; SET Y SIZE IN CKYCT
 110 025B 850C              STA     CKYCT
 111
 112                    ;       START A HORIZONTAL SCAN
 113
 114 025D A50B      CKGNH:  LDA     CKDTAY          ; COPY VERTICAL CKDTA TO HORIZONTAL WORK
 115 025F 850A              STA     CKDTAX          ; LOCATION
 116 0261 A928              LDA     #40             ; INITIALIZE COUNT OF BYTES GENERATED IN
 117 0263 850D              STA     HBYTCT          ; A HORIZONTAL SCAN
 118
 119 0265 A607      CKGNH1: LDX     CKXSZ           ; SET X SIZE IN INDEX X
 120 0267 A50A      CKGNH2: LDA     CKDTAX          ; GENERATE A DOT = TO CURRENT VALUE OF
 121 0269 2A                ROLA                    ; CKDTAX
 122 026A 2606              ROL     VMDATA
 123 026C 88                DEY                     ; COUNT DOTS GENERATED
 124 026D D014              BNE     CKGNH4          ; SKIP AHEAD IF NOT 8 YET
 125 026F A506              LDA     VMDATA          ; STORE A COMPLETED BYTE IN VM
 126 0271 9104              STA     (VMADR),Y
 127 0273 E604              INC     VMADR           ; INCREMENT VM ADDRESS
 128 0275 D006              BNE     CKGNH3
 129 0277 E605              INC     VMADR+1
 130 0279 C60F              DEC     VMPGCT          ; DECREMENT VM PAGE COUNT TO SEE IF DONE
 131 027B F01F              BEQ     CKGENF          ; JUMP OUT IF SO
 132 027D A008      CKGNH3: LDY     #8              ; RESTORE 8 BIT COUNT
 133 027F C60D              DEC     HBYTCT          ; TEST IF FINISHED WITH A HORIZONTAL SCAN
 134 0281 F00C              BEQ     CKGNV1          ; JUMP IF SO
 135 0283 CA        CKGNH4: DEX                     ; DECREMENT SQUARE WIDTH COUNT
 136 0284 D0E1              BNE     CKGNH2          ; GO GENERATE NEXT DOT
 137 0286 A50A              LDA     CKDTAX          ; AT SQUARE BOUNDARY, FLIP COLOR
 138 0288 49FF              EOR     #X'FF
 139 028A 850A              STA     CKDTAX
 140 028C 4C6502            JMP     CKGNH1          ; GO GENERATE NEXT DOT
 141
 142                    ;       FINISH VERTICAL SCAN
 143
 144 028F C60C      CKGNV1: DEC     CKYCT           ; DECREMENT SQUARE HEIGHT
 145 0291 D0CA              BNE     CKGNH           : GO GENERATE NEXT LINE
 146 0293 A50B              LDA     CKDTAY          ; AT SQUARE BOUNDARY, FLIP COLOR
```

```
147 0295 49FF                EOR     #X'FF
148 0297 850B                STA     CKDTAY
149 0299 4C5902              JMP     CKGNV          ; GO GENERATE NEXT LINE
150
151 029C 60          CKGENF: RTS                    ; RETURN
152
```

```
                                .PAGE  'MISCELLANEOUS SUBROUTINES'
  153                   ;       RANDOM NUMBER GENERATOR SUBROUTINE
  154                   ;       ENTER WITH SEED IN RANDNO
  155                   ;       EXIT WITH NEW RANDOM NUMBER IN RANDNO
  156                   ;       USES 16 BIT FEEDBACK SHIFT REGISTER METHOD
  157                   ;       DESTROYS REGISTER A AND Y
  158
  159 029D A008         RAND:   LDY     #8              ; SET COUNTER FOR 8 RANDOM BITS
  160 029F A510         RAND1:  LDA     RANDNO          ; EXCLUSIVE-OR BITS 3, 12, 14, AND 15
  161 02A1 4A                   LSRA                    ; OF SEED
  162 02A2 4510                 EOR     RANDNO
  163 02A4 4A                   LSRA
  164 02A5 4A                   LSRA
  165 02A6 4510                 EOR     RANDNO
  166 02A8 4A                   LSRA
  167 02A9 4511                 EOR     RANDNO+1        ; RESULT IS IN BIT 3 OF A
  168 02AB 4A                   LSRA                    ; SHIFT INTO CARRY
  169 02AC 4A                   LSRA
  170 02AD 4A                   LSRA
  171 02AE 4A                   LSRA
  172 02AF 2611                 ROL     RANDNO+1        ; SHIFT RANDNO LEFT ONE BRINGING IN CARRY
  173 02B1 2610                 ROL     RANDNO
  174 02B3 88                   DEY                     ; TEST IF 8 NEW RANDOM BITS COMPUTED
  175 02B4 D0E9                 BNE     RAND1           ; LOOP FOR MORE IF NOT
  176 02B6 60                   RTS                     ; RETURN
  177
  178                   ;       EXPONENTIALLY DISTRIBUTED RANDOM NUMBER SUBROUTINE
  179                   ;       RULES OF USE SAME AS RAND, 8 BIT RESULT RETURNED IN A
  180                   ;       AN EXPONENTIAL DISTRIBUTION MEANS THAT THE PROBABILITY OF A
  181                   ;       RESULT BETWEEN 10 AND 20 IS THE SAME AS THE PROBABILITY OF A
  182                   ;       RESULT BETWEEN 100 AND 200.
  183                   ;       NOTE THAT THE PROBABILITY OF A ZERO RESULT IS ZERO.
  184
  185 02B7 209D02       RNDEXP: JSR     RAND            ; GET TWO NEW RANDOM BYTES
  186 02BA 209D02               JSR     RAND
  187 02BD A510                 LDA     RANDNO          ; CONVERT ONE OF THE BYTES TO A RANDOM
  188 02BF 2907                 AND     #7              ; VALUE BETWEEN 0 AND 7 AND PUT IN Y AS A
  189 02C1 A8                   TAY                     ; SHIFT COUNT
  190 02C2 C8                   INY
  191 02C3 A511                 LDA     RANDNO+1        ; GET THE OTHER RANDOM NUMBER AND SHIFT IT
  192 02C5 88           RNDXP1: DEY                     ; RIGHT ACCORDING TO Y
  193 02C6 F004                 BEQ     RNDXP2
  194 02C8 4A                   LSRA
  195 02C9 4CC502               JMP     RNDXP1
  196 02CC 0900         RNDXP2: ORA     #0              ; TEST FOR A ZERO RESULT
  197 02CE F0E7                 BEQ     RNDEXP          ; PROHIBIT ZERO RESULTS
  198 02D0 60                   RTS                     ; RETURN
  199
  200                   ;       RECTANGLE SIZE TEST SUBROUTINE
  201                   ;       FORCES LOW VMNO-1 BITS OF A TO BE ZERO AND TESTS FOR ZERO
  202                   ;       USES A AND X
  203
  204 02D1 A603         CKSZAD: LDX     VMNO            ; GET NUMBER OF VM'S IN USE
  205 02D3 3DD602               AND     SZADTB-1,X      ; MASK OUT APPROPRIATE NUMBER OF LOW BITS
  206 02D6 60                   RTS                     ; RETURN
  207
  208 02D7 FFFEFCF8     SZADTB: .BYTE   X'FF,X'FE,X'FC,X'F8
  209
  210 0000                      .END
NO ERROR LINES
```

MASTER TIMING

41.6 NS