# The Final OSCII

(Oliver's Standard Code for Information Interchange)
Jan '90

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | NULL | 29 | CRSR DOWN | 58 | : | 87 | W |
| 1 | NULL | 30 | CRSR RIGHT | 59 | ; | 88 | X |
| 2 | NULL | 31 | CRSR LEFT | 60 | < | 89 | Y |
| 3 | NULL | 32 | SPACE | 61 | = | 90 | Z |
| 4 | NULL | 33 | ! | 62 | > | 91 | [ |
| 5 | NULL | 34 | " | 63 | ? | 92 | ¥ |
| 6 | NULL | 35 | # | 64 | @ | 93 | ] |
| 7 | BELL | 36 | $ | 65 | A | 94 | ^ |
| 8 | BACKSPACE | 37 | % | 66 | B | 95 | _ |
| 9 | TAB | 38 | & | 67 | C | 96 | ` |
| 10 | LINEFEED | 39 | ' | 68 | D | 97 | a |
| 11 | NULL | 40 | ( | 69 | E | 98 | b |
| 12 | CLRHOME | 41 | ) | 70 | F | 99 | c |
| 13 | CR | 42 | * | 71 | G | 100 | d |
| 14 | NULL | 43 | + | 72 | H | 101 | e |
| 15 | NULL | 44 | , | 73 | I | 102 | f |
| 16 | NULL | 45 | – | 74 | J | 103 | g |
| 17 | NULL | 46 | . | 75 | K | 104 | h |
| 18 | NULL | 47 | / | 76 | L | 105 | i |
| 19 | HOME, f. | 48 | 0 | 77 | M | 106 | j |
| 20 | NULL | 49 | 1 | 78 | N | 107 | k |
| 21 | NULL | 50 | 2 | 79 | O | 108 | l |
| 22 | NULL | 51 | 3 | 80 | P | 109 | m |
| 23 | NULL | 52 | 4 | 81 | Q | 110 | n |
| 24 | NULL | 53 | 5 | 82 | R | 111 | o |
| 25 | NULL | 54 | 6 | 83 | S | 112 | p |
| 26 | NULL | 55 | 7 | 84 | T | 113 | q |
| 27 | ESC | 56 | 8 | 85 | U | 114 | r |
| 28 | CRSR UP | 57 | 9 | 86 | V | 115 | s |

|  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
| 116 | t | 145 |  | 174 | ョ | 203 | ヒ | 232 | √ |
| 117 | u | 146 |  | 175 | ッ | 204 | フ | 233 | $^{-1}$ |
| 118 | v | 147 |  | 176 | ー | 205 | ヘ | 234 | i |
| 119 | w | 148 |  | 177 | ア | 206 | ホ | 235 | ˣ |
| 120 | x | 149 |  | 178 | イ | 207 | マ | 236 | ¢ |
| 121 | y | 150 |  | 179 | ウ | 208 | ミ | 237 | ŧ |
| 122 | z | 151 |  | 180 | エ | 209 | ム | 238 | n̄ |
| 123 | { | 152 | reserved | 181 | オ | 210 | メ | 239 | ö |
| 124 | \| | 153 |  | 182 | カ | 211 | モ | 240 | p |
| 125 | } | 154 |  | 183 | キ | 212 | ヤ | 241 | q |
| 126 | → | 155 |  | 184 | ク | 213 | ユ | 242 | θ |
| 127 | ← | 156 |  | 185 | ケ | 214 | ヨ | 243 | ∞ |
| 128 |  | 157 |  | 186 | コ | 215 | ラ | 244 | Ω |
| 129 |  | 158 |  | 187 | サ | 216 | リ | 245 | ü |
| 130 |  | 159 |  | 188 | シ | 217 | ル | 246 | Σ |
| 131 |  | 160 | space | 189 | ス | 218 | レ | 247 | π |
| 132 |  | 161 | 。 | 190 | セ | 219 | ロ | 248 | x̄ |
| 133 |  | 162 | 「 | 191 | ソ | 220 | ワ | 249 | y |
| 134 |  | 163 | 」 | 192 | タ | 221 | ン | 250 | 千 |
| 135 |  | 164 | 、 | 193 | チ | 222 | ゛ | 251 | 万 |
| 136 | reserved | 165 | ・ | 194 | ツ | 223 | ゜ | 252 | 円 |
| 137 |  | 166 | ヲ | 195 | テ | 224 | α | 253 | ÷ |
| 138 |  | 167 | ァ | 196 | ト | 225 | ä | 254 | space |
| 139 |  | 168 | ィ | 197 | ナ | 226 | β | 255 | █ |
| 140 |  | 169 | ゥ | 198 | ニ | 227 | ε |  |  |
| 141 |  | 170 | ェ | 199 | ヌ | 228 | μ |  |  |
| 142 |  | 171 | ォ | 200 | ネ | 229 | σ |  |  |
| 143 |  | 172 | ャ | 201 | ノ | 230 | ρ |  |  |
| 144 |  | 173 | ュ | 202 | ハ | 231 | g |  |  |

```
8000 DATAWR  8001 CTRLWR  8002 DATARD  8003 CTRLRD  8004 LATCHWR  00A6 TEXTLO
00A7 TEXTHI  00A8 TEXTLEN  00A9 LOCRSRPOS  00AA HICRSRPOS  00AB LOSCRNPOS  00AC HISCRNPOS
00AD COL     00AE LIN     00AF LATCH   00B0 TAB     C002 GETASCII  C03B NEXT
C043 CTRL7   C049 CTRL8   C04F CTRL9   C055 CTRL10  C05B CTRL12  C061 CTRL13
C067 CTRL19  C06D CTRL28  C073 CTRL29  C079 CTRL30  C07F CTRL31  C085 PRINT
  )C SKIP1   C0A5 SKIP3   C0AB NEWLIN  C0C5 SKIP2   C0CE ADDLIN  C0D2 LOOP3
C108 NOHI    C129 CARETURN  C137 NOHI2  C15B BELL    C15C LINEFEED  C16E SETCRSR
C18F HOME    C1A1 CLRHOME  C1A3 LOOP4   C1B0 TABULATOR  C1B4 TABTEST  C1D0 NOHI4
C1D7 NEXTTAB  C1DE QUIT    C1DF CRSRUP  C1F0 SKIP6  C1F3 QUIT5  C1F4 CRSRLEFT
C204 SKIP4   C207 QUIT2   C208 BACKSPACE  C218 SKIP5   C22E QUIT4   C22F CRSRRIGHT
C242 NOHI3   C245 QUIT3   C246 CLRSCR  C252 LOOP2   C253 LOOP1   C2B1 WAIT
C2BB LCDRESET  01

C246
                ; CONOUT              AN 1990 BY  LIVER  ALTSTEIN


                ; CONTROLS THE  ITACHI  61830        RIVER

                ;ATTENTION:         SEQUENCES AND      STILL HAVE TO BE WRITTEN


8000        DATAWR =     32768
8001        CTRLWR =     DATAWR+1
  )2        DATARD =     CTRLWR+1
8003        CTRLRD =     DATARD+1
8004        LATCHWR =    CTRLRD+1            WRITEONLY

00A6        TEXTLO =     $A6                 READ & WRITE
00A7        TEXTHI =     TEXTLO+1            READ & WRITE
00A8        TEXTLEN =    TEXTHI+1            READ & WRITE

00A9        LOCRSRPOS =  TEXTLEN+1           READONLY
00AA        HICRSRPOS =  LOCRSRPOS+1         READONLY

00AB        LOSCRNPOS =  HICRSRPOS+1         READONLY
00AC        HISCRNPOS =  LOSCRNPOS+1         READONLY

00AD        COL    =     HISCRNPOS+1           READONLY
00AE        LIN    =     COL+1                 READONLY

00AF        LATCH  =     LIN+1               READ & WRITE

  30        TAB    =     LATCH+1               READ & WRITE


C000                ORG $C000

C000 A000           LDY #00
C002 98     GETASCII TYA
C003 48             PHA
C004 B1A6           LDA (TEXTLO),Y
C006 48             PHA
C007 29E0           AND #$E0             IF >= 32 THEN SEND ASCII TO LCD
C009 AA             TAX
C00A E000           CPX #$00
C00C D077           BNE PRINT
C00E 68             PLA
C00F C90D           CMP #13              CHECK FOR RETURN
C011 F04E           BEQ CTRL13
C013 C90A           CMP #10
C015 F03E           BEQ CTRL10
C017 C907           CMP #07
C019 F028           BEQ CTRL7
C01B C908           CMP #08
C01D F02A           BEQ CTRL8
```

0898 991180

changes

- init Brighness with 10 dez

- check if # of char to be printed is zero and skip.

- CLD

```
CØ1F C9Ø9          CMP #Ø9
CØ21 FØ2C          BEQ CTRL9
CØ23 C913          CMP #19
CØ25 FØ4Ø          BEQ CTRL19
CØ27 C91C          CMP #28
CØ29 FØ42          BEQ CTRL28
CØ2B C91D          CMP #29
CØ2D FØ44          BEQ CTRL29
CØ2F C91E          CMP #3Ø
CØ31 FØ46          BEQ CTRL3Ø
CØ33 C91F          CMP #31
CØ35 FØ48          BEQ CTRL31
```

Ø2

```
CØ37 C9ØC          CMP #12
CØ39 FØ2Ø          BEQ CTRL12

CØ3B 68     NEXT   PLA
CØ3C A8            TAY
CØ3D C8            INY
CØ3E C4A8          CPY TEXTLEN
CØ4Ø DØCØ          BNE GETASCII
CØ42 6Ø            RTS

CØ43 205BC1 CTRL7  JSR BELL
CØ46 4C3BCØ        JMP NEXT
CØ49 2008C2 CTRL8  JSR BACKSPACE
CØ4C 4C3BCØ        JMP NEXT
CØ4F 2ØBØC1 CTRL9  JSR TABULATOR
CØ52 4C3BCØ        JMP NEXT
CØ55 205CC1 CTRL1Ø JSR LINEFEED
CØ58 4C3BCØ        JMP NEXT
CØ5B 20A1C1 CTRL12 JSR CLRHOME
CØ5E 4C3BCØ        JMP NEXT
CØ61 2029C1 CTRL13 JSR CARETURN
CØ64 4C3BCØ        JMP NEXT
CØ67 208FC1 CTRL19 JSR HOME
CØ6A 4C3BCØ        JMP NEXT
CØ6D 20DFC1 CTRL28 JSR CRSRUP
CØ7Ø 4C3BCØ        JMP NEXT
CØ73 205CC1 CTRL29 JSR LINEFEED               USED AS CRSRDOWN
CØ76 4C3BCØ        JMP NEXT
CØ79 202FC2 CTRL3Ø JSR CRSRRIGHT
CØ7C 4C3BCØ        JMP NEXT
CØ7F 2ØF4C1 CTRL31 JSR CRSRLEFT
CØ82 4C3BC0        JMP NEXT

CØ85 20B1C2 PRINT  JSR WAIT
CØ88 A9ØC          LDA #$ØC
CØ8A 8D0180        STA CTRLWR                 SWITCH TO CHAR OUT MODE
CØ8D 20B1C2        JSR WAIT
CØ9Ø 68            PLA                        JIS FOR LCD
CØ91 8D0080        STA DATAWR                 SEND CHAR TO LCD
CØ94 E6AD          INC COL
CØ96 E6A9          INC LOCRSRPOS
CØ98 DØ02          BNE SKIP1
CØ9A E6AA          INC HICRSRPOS              UPDATES INFORMATION IN ZERO PAGE
CØ9C A5AD   SKIP1  LDA COL                    NOW IT HAS TO BE CHECKED IT CRSRPOS IS A
CØ9E C941          CMP #65                    (MULTIPLE OF 66)-2
C0A0 FØ03          BEQ SKIP3                  MARGIN REACHED
C0A2 4C3BCØ        JMP NEXT
```

```
CØA5 2ØABCØ SKIP3  JSR NEWLIN
CØA8 4C3BCØ        JMP NEXT


CØAB A9ØØ   NEWLIN LDA #ØØ                 THIS PROCEDURE SHOULD ONLY BE ACTIVATED
CØAD 85AD          STA COL                 WHEN CRSR HAS REACHED RIGHT MARGIN
CØAF A9ØC          LDA #$ØC                IT MOVES CRSR TO NEXT LINE AND SCROOLS
CØB1 2ØB1C2        JSR WAIT                UP IF NECESSARY
CØB4 8DØ18Ø        STA CTRLWR
CØB7 A92Ø          LDA #32
CØB9 2ØB1C2        JSR WAIT
CØBC 8DØØ8Ø        STA DATAWR
CØBF E6A9          INC LOCRSRPOS
CØC1 DØØ2          BNE SKIP2
CØC3 E6AA          INC HICRSRPOS           UPDATES INFORMATION IN ZERO PAGE
CØC5 A5AE   SKIP2  LDA LIN
CØC7 C9Ø7          CMP #Ø7
CØC9 FØØ3          BEQ ADDLIN              BRANCH TO CREATE NEW LINE ON LCD




Ø3

CØCB E6AE          INC LIN                 MOVE DOWN ON THE LCD WITHOUT SCOOLING
CØCD 6Ø            RTS
CØCE A242   ADDLIN LDX #66                 WRITE 66 SPACES TO ERASE FOLLOWING LINE
CØDØ A92Ø          LDA #32
CØD2 CA     LOOP3  DEX
CØD3 2ØB1C2        JSR WAIT
CØD6 8DØØ8Ø        STA DATAWR
CØD9 EØØØ          CPX #ØØ
CØDB DØF5          BNE LOOP3
CØDD A9ØA          LDA #1Ø                 CRSRPOS ADDRESS
CØDF 2ØB1C2        JSR WAIT
CØE2 8DØ18Ø        STA CTRLWR
CØE5 A5A9          LDA LOCRSRPOS
CØE7 2ØB1C2        JSR WAIT
CØEA 8DØØ8Ø        STA DATAWR
CØED A9ØB          LDA #11
CØEF 2ØB1C2        JSR WAIT
CØF2 8DØ18Ø        STA CTRLWR
CØF5 A5AA          LDA HICRSRPOS
CØF7 2ØB1C2        JSR WAIT
CØFA 8DØØ8Ø        STA DATAWR
CØFD A942          LDA #66                 MOVE SCRNSTART 66 FURTHER AWAY
CØFF 18            CLC
C1ØØ 65AB          ADC LOSCRNPOS
C1Ø2 85AB          STA LOSCRNPOS
C1Ø4 9ØØ2          BCC NOHI
C1Ø6 E6AC          INC HISCRNPOS
C1Ø8 A9Ø8   NOHI   LDA #Ø8                 UPDATE LCD SCREENPOSITION
C1ØA 2ØB1C2        JSR WAIT
C1ØD 8DØ18Ø        STA CTRLWR
C11Ø A5AB          LDA LOSCRNPOS
C112 2ØB1C2        JSR WAIT
C115 8DØØ8Ø        STA DATAWR
C118 A9Ø9          LDA #Ø9
C11A 2ØB1C2        JSR WAIT
C11D 8DØ18Ø        STA CTRLWR
C12Ø A5AC          LDA HISCRNPOS
C122 2ØB1C2        JSR WAIT
C125 8DØØ8Ø        STA DATAWR
C128 6Ø            RTS
```

```
C129 A941   CARETURN LDA #65          PROCEDURE PLACES CURSOR ONTO RIGHT MARGIN
C12B 38              SEC
C12C E5AD            SBC COL          FIND NUMBER OF SPACES TO THE RIGHT MARGIN
C12E 18              CLC
C12F 65A9            ADC LOCRSRPOS    ADD DIFFERENCE TO CRSRPOSITION
C131 85A9            STA LOCRSRPOS
C133 9002            BCC NOHI2
C135 E6AA            INC HICRSRPOS
C137 A90A   NOHI2    LDA #10          MOVE CRSR ON SCREEN INTO THE RIGHT MARGIN
C139 20B1C2          JSR WAIT
C13C 8D0180          STA CTRLWR
C13F A5A9            LDA LOCRSRPOS
C141 20B1C2          JSR WAIT
C144 8D0080          STA DATAWR       SEND LOWBYTE
C147 A90B            LDA #11
C149 20B1C2          JSR WAIT
C14C 8D0180          STA CTRLWR
C14F A5AA            LDA HICRSRPOS
C151 20B1C2          JSR WAIT
C154 8D0080          STA DATAWR       SEND HIBYTE
C157 20ABC0          JSR NEWLIN
C15A 60              RTS




04

C15B 60     BELL     RTS

C15C A5AD   LINEFEED LDA COL
C15E 48              PHA              SAVE OLD COL
C15F 2029C1          JSR CARETURN
C162 68              PLA
C163 85AD            STA COL
C165 18              CLC
C166 65A9            ADC LOCRSRPOS    MOVE CRSR FROM COL 0 TO OLD COL
C168 85A9            STA LOCRSRPOS
C16A 9002            BCC SETCRSR
C16C E6AA            INC HICRSRPOS
C16E A90A   SETCRSR  LDA #10
C170 20B1C2          JSR WAIT
C173 8D0180          STA CTRLWR
C176 A5A9            LDA LOCRSRPOS
C178 20B1C2          JSR WAIT
C17B 8D0080          STA DATAWR
C17E A90B            LDA #11
C180 20B1C2          JSR WAIT
C183 8D0180          STA CTRLWR
C186 A5AA            LDA HICRSRPOS
C188 20B1C2          JSR WAIT
C18B 8D0080          STA DATAWR
C18E 60              RTS


C18F A5AB   HOME     LDA LOSCRNPOS
C191 85A9            STA LOCRSRPOS    MAKE CRSRPOS EQUAL TO SCRNPOS
C193 A5AC            LDA HISCRNPOS
C195 85AA            STA HICRSRPOS
C197 A900            LDA #00
C199 85AE            STA LIN          UPDATE POS INFO
C19B 85AD            STA COL
C19D 206EC1          JSR SETCRSR
C1A0 60              RTS
```

```
C1A1 A21Ø   CLRHOME LDX #$1Ø
C1A3 8A     LOOP4  TXA
C1A4 48            PHA
C1A5 2Ø29C1        JSR CARETURN
C1A8 68            PLA
C1A9 AA            TAX
C1AA DØF7          BNE LOOP4
C1AC 2Ø8FC1        JSR HOME
C1AF 6Ø            RTS


C1BØ A5BØ   TABULATOR LDA TAB
C1B2 FØ2A          BEQ QUIT                IF TABSIZE IS Ø THEN FORGET IT
C1B4 AA     TABTEST TAX
C1B5 38            SEC
C1B6 E941          SBC #65                 IF TABPOS >= 64 THEN QUIT ALSO
C1B8 1Ø24          BPL QUIT
C1BA 8A            TXA                     GET THE TABPOS BACK
C1BB 38            SEC                     IF COL < TABPOS THEN JMP TO NEXT TAB
C1BC E9Ø1          SBC #Ø1                 (SUBSTITUTE FOR DEA)
C1BE 38            SEC
C1BF E5AD          SBC COL                 TABPOS - 1 - COL
C1C1 3Ø14          BMI NEXTTAB             IF COL >= TAB, INC TAB AND TRY AGAIN
C1C3 8A            TXA                     NOW PLACE CURSOR AT TABPOS
C1C4 38            SEC
C1C5 E5AD          SBC COL                 DISTANCE BETWEEN TABPOS AND COL
C1C7 18            CLC
```

```
C1C8 65A9          ADC LOCRSRPOS
C1CA 85A9          STA LOCRSRPOS
C1CC 9ØØ2          BCC NOHI4
C1CE E6AA          INC HICRSRPOS
C1DØ 8A     NOHI4  TXA
C1D1 85AD          STA COL
C1D3 2Ø6EC1        JSR SETCRSR             INFORM SCREEN OF CHANGED CRSRPOS
C1D6 6Ø            RTS                     STOP AFTER TAB HAS BEEN DONE
C1D7 8A     NEXTTAB TXA                    TABPOS=TABPOS + TAB
C1D8 18            CLC
C1D9 65BØ          ADC TAB                 PREPARE TO TEST NEXT TAB LOCATION
C1DB 4CB4C1        JMP TABTEST
C1DE 6Ø     QUIT   RTS


C1DF A5AE   CRSRUP LDA LIN
C1E1 FØ1Ø          BEQ QUIT5
C1E3 C6AE          DEC LIN
C1E5 A5A9          LDA LOCRSRPOS
C1E7 38            SEC
C1E8 E942          SBC #66                 ABOVE CRSR
C1EA 85A9          STA LOCRSRPOS
C1EC BØØ2          BCS SKIP6
C1EE C6AA          DEC HICRSRPOS
C1FØ 2Ø6EC1 SKIP6  JSR SETCRSR
C1F3 6Ø     QUIT5  RTS


C1F4 A5AD   CRSRLEFT LDA COL
C1F6 FØØF          BEQ QUIT2
C1F8 C6AD          DEC COL
```

```
C1FA C6A9           DEC LOCRSRPOS
C1FC A5A9           LDA LOCRSRPOS
C1FE C9FF           CMP #$FF
C200 D002           BNE SKIP4
C202 C6AA           DEC HICRSRPOS
  04 206EC1 SKIP4   JSR SETCRSR
C207 60     QUIT2   RTS


C208 A5AD   BACKSPACE LDA COL
C20A F022           BEQ QUIT4
C20C C6AD           DEC COL
C20E C6A9           DEC LOCRSRPOS
C210 A5A9           LDA LOCRSRPOS
C212 C9FF           CMP #$FF
C214 D002           BNE SKIP5
C216 C6AA           DEC HICRSRPOS
C218 206EC1 SKIP5   JSR SETCRSR
C21B A90C           LDA #12
C21D 20B1C2         JSR WAIT
C220 8D0180         STA CTRLWR
C223 A920           LDA #32                     SPACE
C225 20B1C2         JSR WAIT
C228 8D0080         STA DATAWR
C22B 206EC1         JSR SETCRSR
C22E 60     QUIT4   RTS


C22F A5AD   CRSRRIGHT LDA COL
C231 C940           CMP #64                     IF CRSR ON RIGHT THEN DON'T MOVE
C233 F010           BEQ QUIT3
C235 E6AD           INC COL
C237 A901           LDA #01
C239 18             CLC
```

```
C23A 65A9           ADC LOCRSRPOS
C23C 85A9           STA LOCRSRPOS
C23E 9002           BCC NOHI3
C240 E6AA           INC HICRSRPOS
C242 206EC1 NOHI3   JSR SETCRSR
C245 60     QUIT3   RTS


C246 20B1C2 CLRSCR JSR WAIT                     FILL SCREEN MEMORY WITH SPACES
C249 A90C           LDA #12
C24B 8D0180         STA CTRLWR
C24E A010           LDY #$10
C250 A200           LDX #$00                    LOOP 4096 TIMES
C252 88     LOOP2   DEY
C253 CA     LOOP1   DEX
C254 20B1C2         JSR WAIT
C257 A920           LDA #32                     SPACE
C259 8D0080         STA DATAWR
C25C E000           CPX #00
C25E D0F3           BNE LOOP1
C260 C000           CPY #00
C262 D0EE           BNE LOOP2

C264 20B1C2         JSR WAIT
C267 A908           LDA #08
```

```
C269 8DØ18Ø        STA CTRLWR
C26C 2ØB1C2        JSR WAIT
C26F A9ØØ          LDA #ØØ                 LOBYTE DISPLAY STARTING ADDRESS
C271 8DØØ8Ø        STA DATAWR
C274 2ØB1C2        JSR WAIT
  77 A9Ø9          LDA #Ø9
C279 8DØ18Ø        STA CTRLWR
C27C 2ØB1C2        JSR WAIT
C27F A9ØØ          LDA #ØØ                 HIBYTE DISPLAY STARTING ADDRESS
C281 8DØØ8Ø        STA DATAWR
C284 85AB          STA LOSCRNPOS           UPDATE ZEROPAGE
C286 85AC          STA HISCRNPOS

C288 2ØB1C2        JSR WAIT
C28B A9ØA          LDA #1Ø
C28D 8DØ18Ø        STA CTRLWR
C29Ø 2ØB1C2        JSR WAIT
C293 A9ØØ          LDA #ØØ
C295 8DØØ8Ø        STA DATAWR              LOBYTE CURSOR ADDRESS
C298 2ØB1C2        JSR WAIT
C29B A9ØB          LDA #11
C29D 8DØ18Ø        STA CTRLWR
C2AØ 2ØB1C2        JSR WAIT
C2A3 A9ØØ          LDA #ØØ                 HIBYTE CURSOR ADDRESS
  A5 8DØØ8Ø        STA DATAWR
C2A8 85A9          STA LOCRSRPOS
C2AA 85AA          STA HICRSRPOS
C2AC 85AD          STA COL
C2AE 85AE          STA LIN
C2BØ 6Ø            RTS

C2B1 48     WAIT   PHA
C2B2 ADØ38Ø        LDA CTRLRD
C2B5 298Ø          AND #$8Ø
C2B7 DØF8          BNE WAIT
C2B9 68            PLA
C2BA 6Ø            RTS

C2BB 2ØB1C2 LCDRESET JSR WAIT
C2BE A9ØØ          LDA #ØØ                  RITE  EG Ø




Ø7


C2CØ 8DØ18Ø        STA CTRLWR
C2C3 2ØB1C2        JSR WAIT
C2C6 A938          LDA #$38                 ISPLAY ON, CHARACTER BLINK,
C2C8 8DØØ8Ø        STA DATAWR              NO GRAPHICS, INTERNALS CHARSET

C2CB 2ØB1C2        JSR WAIT
C2CE A9Ø1          LDA #Ø1
C2DØ 8DØ18Ø        STA CTRLWR
C2D3 2ØB1C2        JSR WAIT
C2D6 A975          LDA #117                8 LINES, 66.6 COL
C2D8 8DØØ8Ø        STA DATAWR

C2DB 2ØB1C2        JSR WAIT
C2DE A9Ø2          LDA #Ø2
C?EØ 8DØ18Ø        STA CTRLWR
 .E3 2ØB1C2        JSR WAIT
C2E6 A941          LDA #65                 66 CHAR PER LINE
C2E8 8DØØ8Ø        STA DATAWR
```

```
C2EB 20B1C2         JSR WAIT
C2EE A903           LDA #03
C2F0 8D0180         STA CTRLWR
C2F3 20B1C2         JSR WAIT
C2F6 A91F           LDA #31                   32 VERTICAL DOTS
C2F8 8D0080         STA DATAWR

C2FB 20B1C2         JSR WAIT
C2FE A904           LDA #04
C300 8D0180         STA CTRLWR
C303 20B1C2         JSR WAIT
C306 A907           LDA #07                   CURSOR POS AT 8     ???
C308 8D0080         STA DATAWR

C30B A908           LDA #08                   SET BRIGHTNESS
C30D 8D0480         STA LATCHWR               LATCH OF LCD
C310 85AF           STA LATCH                 LATCH OF PAGE ZERO

C312 A908           LDA #08                   INIT TAB WITH 8
C314 85B0           STA TAB

C316 A5AF           LDA LATCH
C318 0940           ORA #%01000000            SET BIT 7 TO ACTIVATE PAGE 2
C31A 85AF           STA LATCH
C31C 8D0480         STA LATCHWR
C31F 2046C2         JSR CLRSCR
C322 A5AF           LDA LATCH
C324 29BF           AND #%10111111            CHANGE BACK TO FIRST PAGE
C326 85AF           STA LATCH
C328 8D0480         STA LATCHWR
C32B 2046C2         JSR CLRSCR

C32E 60             RTS
```

5.1.90

# Display Driver Port

Anschluß an C=64 Expansion Port

|  |  | Description |
|---|---|---|
| 1 | 2 | 5V |
| 2 | 3 | 5V |
| 3 | N.C. | – |
| 4 | N.C. | – |
| 5 | N.C. | – |
| 6 | N.C. | – |
| ? 7 | E (N.C.?) | $\Phi_2$ |
| 8 | N.C. | – |
| 9 | pullup | ~$\overline{RD}$~ |
| 10 | N.C. | – |
| 11 | N.C. | – |
| 12 | C | Reset |
| 13 | 5 | R/$\overline{W}$ |
| 14 | N.C. | – |
| 15 | R | $A_7$ |
| 16 | S | $A_6$ |
| 17 | T | $A_5$ |
| 18 | U | $A_4$ |
| 19 | V | $A_3$ |
| 20 | W | $A_2$ |
| 21 | X | $A_1$ |
| 22 | Y | $A_0$ |
| 23 | H | $A_{14}$ |
| 24 | F | $A_{15}$ |
| 25 | P | $A_8$ |
| 26 | N | $A_9$ |
| 27 | L | $A_{11}$ |
| 28 | K | $A_{12}$ |
| 29 | 21 | $D_0$ |
| 30 | 20 | $D_1$ |
| 31 | 19 | $D_2$ |
| 32 | 18 | $D_3$ |
| 33 | 17 | $D_4$ |
| 34 | 16 | $D_5$ |
| 35 | 15 | $D_6$ |
| 36 | 14 | $D_7$ |
| 37 | J | $A_{13}$ |
| 38 | M | $A_{10}$ |
| 39 | 1, 22, A or Z | GND |
| 40 | 1, 22, A or Z | GND |

Achtung, $\Phi$ angeschloßen, durch jumper L regelbar

Achtung:
N.C. leitungen irgendwo anschließen!

Dip Switches

|  | S1 … S16 |
|---|---|
| 32768<br>8000 | : 0001 1111 1111 1110 |
| A000h | : 00011111 1111 1010 |

SHARP

Vorsicht

## HANDLING PRECAUTIONS

**************************************

I. Installation

1) LCD module shall be mounted on the application unit with the mounting holes on the PCB. While mounting, care shall be taken not to twist or deform the PCB, which can lead mulfunction of the module due to disconnection caused by deforming pressure on the LCD panel.

2) Pressure on the LCD panel will change the thickness of LC layer, which can cause mulfunction and can shorten the service life of module.

The module shall be covered with protective board, such as glass or acrylic board because the front polarizer of the LCD panel is soft and fragile.

II. Operation

1) The module shall be operated within the operating conditions stipulated in the specifications.

2) The drive circuit of the module employs CMOS LSI's. Some messure for electro-static discharge shall be taken to prevent damage on the LSI's due to ESD.

3) The module shall be operated at the specified voltage. Operating the module at voltage higher than specified will shorten the service life.

DC voltage, if it is applied to the LCD panel, will shorten the service life, causing electro-chemical reaction and shall be strictly avoided.

Care shall be taken to insure that clock pulse and drive wave form alternating signal are supplied to keep applying AC voltage to the LCD panel while operating.

4) Condensation in the LCD module will cause electro-chemical reaction, which can cause open or short circuit of the transparent electrodes of the LCD panel.

The module shall be operated below relative humidity equivalent to 40 °C and 50 % RH to prevent condensation.

5) When the module is operated at temperatures below the specified operating temperature, the response time becomes slow. When the operating temperatyre is above the specified one, the display becomes dark.

Operating the module at temperatures below or above the specified storage temperature can cause crystalization or liquid phase of the LC material, which, sometimes, may not return to the original alignment.

- continued-

Uitvoerige beschrijving (10pag.) met uitstuurprint in Elektuur April '88

1. **Scope**

This specification applies to the 400 x 64 Dot Matrix LCD Unit.

2. **Construction and Outline**

Construction: 400 x 64 full dot graphic display unit

Outline: See Fig. 8

Connection: See Fig. 8 and Table 4

8x5 char cell for 8 line, 80 col. disp.

3. **Mechanical Specifications**

Table 1

| Items | Specifications | Unit |
|---|---|---|
| Unit outline dimensions | 260(W) x 80(H) x 25(D) | mm |
| Effective viewing area | 226(W) x 43(H) | mm |
| Display format | 400 (W) x 64(H) full dots | – |
| Dot size | 0.475(W) x 0.475(H) | mm |
| Dot spacing | 0.075 | mm |
| Dot color | Dark blue | – |
| Background color | White | – |
| Weight | Approx 250 | gr |

4. **Electrical Specifications**

4.1 **Absolute Maximum Ratings**

Table 2

| Parameter | Symbol | Min. | Max. | Unit | Remark |
|---|---|---|---|---|---|
| Supply voltage (Logic) | $V_{DD}-V_{SS}$ | 0 | 7.0 | V | |
| Input voltage | $V_{IN}$ | 0 | $V_{DD}$ | V | |
| Storage temperature | $T_{stg}$ | -25 | +55 | °C | |
| Operating temperature | $T_{opr}$ | 0 | +50 | °C | |

## 4.2 Electrical Characteristics

Table 3

(Ta = 25°C)

| Parameter | Symbol | Conditions | Min. | Max. | Unit | Nom. |
|---|---|---|---|---|---|---|
| Supply voltage (Logic) | $V_{DD}$-$V_{SS}$ | | 4.75 | 5.25 | V | 5.0 |
| Input signal voltage | $V_{IN}$ | "High" level | $0.7V_{DD}$ | $V_{DD}$ | V | 3.5. |
| | | "Low" level | 0 | $0.3V_{DD}$ | V | 1.5 |
| Input leakage current | $I_{IL}$ | $V_{IN}$ = 5.0V | - | 10 | μA | |
| Power dissipation | Pd | $V_{DD}$=5V | - | 105 | mW | |

(Temperature compensation is built-in.)

## 4.3 Interface Signals

Connector used: MOLEX 5046 10A
Compatible connector: MOLEX 5051-10 Di

Table 4

| Pin No. *1 | Symbol | Description | Effective Level | |
|---|---|---|---|---|
| 1 | S | Scan start-up signal | "H" V | Sync |
| 2 | CP₁ | Input data latch signal | H → L H | Sync |
| 3 | CP₂ | Data input clock signal | H → L | Pxclk |
| 4 | DI₁ | Display data signal (Upper half of screen) | H(ON), L(OFF) | DØ |
| 5 | M | Drive waveform alternating signal | H, L | MS |
| 6 | $V_O$ *2 | Bias voltage for LCD drive | —— | |
| 7 | $V_{DD}$ | Power supply for logic circuit | —— | +V |
| 8 | $V_{SS}$ | Ground potential | —— | GND |
| 9 | DI₂ | Display data signal (Lower half of screen) | H(ON), L(OFF) | D1 |
| 10 | NC | | —— | |

(Note) *1: For the location of Pin Nos., refer to Fig. 8.

*2: External supply voltage between $V_{DD}$ and GND. Contrast of LCD shall be controlled from external portion by changing this voltage value.

## 4.4 Interface Timing

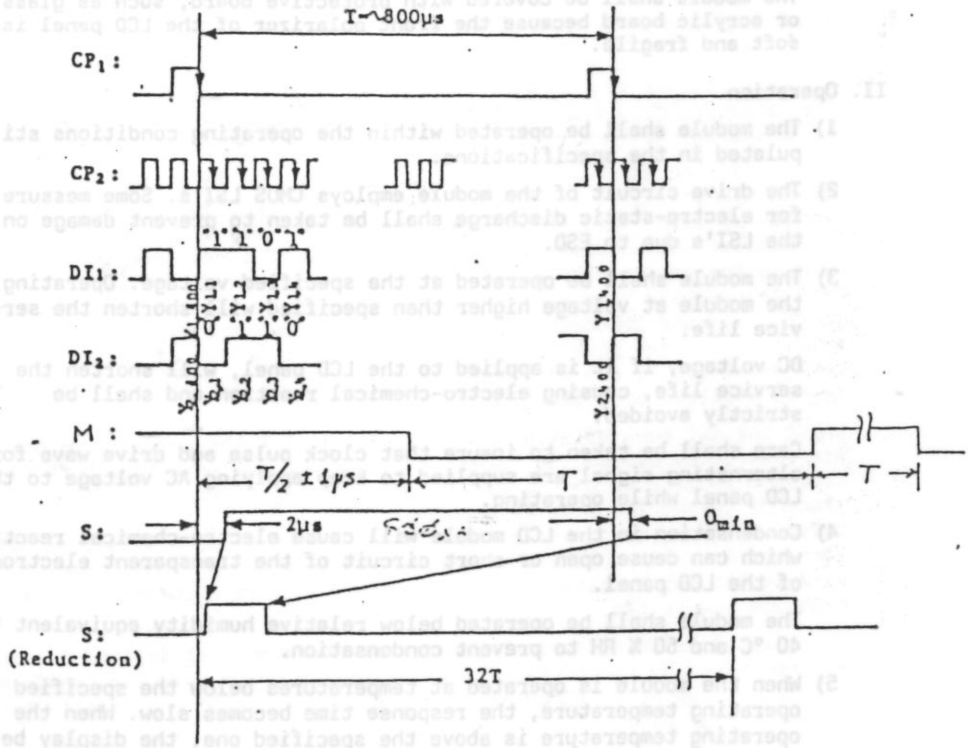
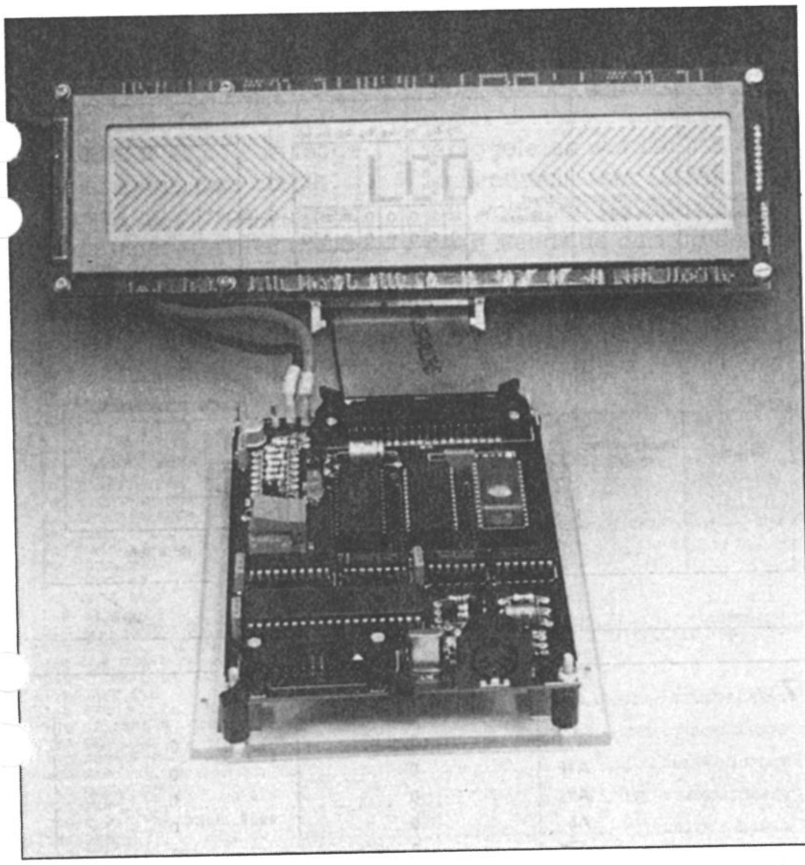


Fig. 1 Interface Timing Chart

Uitvoerige beschrijving (oppag.) met
uitstwurprint in Elektuur april '88



Tabel 3b

| Signal name |  |  | Signal Name |
|---|---|---|---|
| GND | B1 | A1 | −I/O CH CK |
| +RESET DRV | B2 | A2 | +D7 |
| +5V | B3 | A3 | +D6 |
| +IRQ2 | B4 | A4 | +D5 |
| −5VDC | B5 | A5 | +D4 |
| +DRQ2 | B6 | A6 | +D3 |
| −12V | B7 | A7 | +D2 |
| Reserved | B8 | A8 | +D1 |
| +12V | B9 | A9 | +D0 |
| GND | B10 | A10 | +I/O CH RDY |
| −MEMW | B11 | A11 | +AEN |
| MEMR | B12 | A12 | +A19 |
| −IOW | B13 | A13 | +A18 |
| −IOR | B14 | A14 | +A17 |
| −DACK3 | B15 | A15 | +A16 |
| +DRQ3 | B16 | A16 | +A15 |
| −DACK1 | B17 | A17 | +A14 |
| +DRQ1 | B18 | A18 | +A13 |
| −DACK0 | B19 | A19 | +A12 |
| CLOCK | B20 | A20 | +A11 |
| +IRQ7 | B21 | A21 | +A10 |
| +IRQ6 | B22 | A22 | +A9 |
| +IRQ5 | B23 | A23 | +A8 |
| +IRQ4 | B24 | A24 | +A7 |
| +IRQ3 | B25 | A25 | +A6 |
| −DACK2 | B26 | A26 | +A5 |
| +T/C | B27 | A27 | +A4 |
| +ALE | B28 | A28 | +A3 |
| +5V | B29 | A29 | +A2 |
| +OSC | B30 | A30 | +A1 |
| +GND | B31 | A31 | +A0 |

omdat de snelheid van BASIC nooit zo hoog is dat een nieuwe opdracht verzonden kan worden binnen de tijd die de controller nodig heeft. Machinetaalprogrammeurs moeten wel met deze vlag rekening houden.

Alvorens er ook maar een karakter op het beeldscherm kan verschijnen, moet de controller ingesteld worden. Wij gaan nu even uit van het LM40001-display, bij andere displays zullen voor een aantal registers (zoals het aantal regels en het aantal karakters per regel) andere waarden gekozen moeten worden. In ieder geval moeten de registers 0 t/m 4 gevuld worden. In tabel 4 zijn alle registers afgebeeld en is te zien dat met register 0 de mode ingesteld kan worden. In figuur 5 is te zien wat er zoal ingesteld kan worden. Bit 0 bepaalt of de interne of een externe karaktergenerator gebruikt wordt, bit 1 bepaalt de mode van de controller (grafisch of tekst). De twee volgende bits bepalen de vorm van de cursor en schakelen hem aan of uit. Bit 4 is in deze applikatie altijd "1", de

*Tabel 4. De mogelijkheden van de controller in kombinatie met de daarvoor aanwezige registers.*

Tabel 4

must be initialized

| Register |  |  |  |  |  |  |  |  | Data |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CTL | R/W | RS | DB7≈DB4 | DB3 | DB2 | DB1 | DB0 | Function | R/W | RS | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | Mode control | 0 | 0 | 0 | 0 | Mode data |  |  |  |  |  |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | Vertical/horizontal character pitch | 0 | 0 | (Vp − 1) B |  |  |  | 0 | (HP-1) B |  |  |
| 2 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | Number of characters per line/number of bytes | 0 | 0 | 0 | (HN − 1)B = * |  |  |  |  |  |  |
| 3 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | Number of vertical dots | 0 | 0 | 0 | (NX − 1) B |  |  |  |  |  |  |
| 4 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | Cursor position | 0 | 0 | 0 | 0 | 0 | 0 | (CP − 1) B |  |  |  |
| 8 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | Display starting address (least significant) (Lower) | 0 | 0 | Address data |  |  |  |  |  |  |  |
| 9 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | Display starting address (most significant) (Upper) | 0 | 0 | 0 | 0 | 0 | 0 | Address data |  |  |  |
| 10 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | Cursor address (least significant) | 0 | 0 | Address data |  |  |  |  |  |  |  |
| 11 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | Cursor address (most significant) (Upper) | 0 | 0 | 0 | 0 | 0 | 0 | Address data |  |  |  |
| 12 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | Refresh memory write | 0 | 0 | Character code/bit data |  |  |  |  |  |  |  |
| 13 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | Refresh memory read | 1 | 0 | Refresh memory data |  |  |  |  |  |  |  |
| 14 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | Bit clear | 0 | 0 | 0 | 0 | 0 | 0 | 0 | (BN) B |  |  |
| 15 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | Bit set | 0 | 0 | 0 | 0 | 0 | 0 | 0 | (BN) B |  |  |
| 14 | – | – | – | – | – | – | – | BUSY signal read | 1 | 1 | BF | • | • | • | • | • | • | • |

117
65

* HN = 66, use only 0 – 64 to avoid "dirt" on the left margin