COP820CJ

Application Note 953 LCD Triplex Drive with COP820CJ

Literature Number: SNOA329
LCD Triplex Drive with COP820CJ

INTRODUCTION
There are many applications which use a microcontroller in combination with a Liquid Crystal Display. The normal method to control a LCD panel is to connect it to a special LCD driver device, which receives the display data from a microcontroller. A cheaper solution is to drive the LCD directly from the microcontroller. With the flexibility of a COP8 microcontroller the multiplexed LCD direct drive is possible. This application note shows a way how to drive a three way multiplexed LCD with up to 36 segments using a 28-pin COP800 device.

ABOUT MULTIPLEXED LCD'S
There is a wide variety of LCD's, ranging from static devices to multiplexed versions with multiplex rates of up to 1:256.

The multiplex rate of a LCD is determined by the number of its backplanes (segment-common planes). The number of segments controlled by one line (with one segment pin) is equal to the number of backplanes on the LCD. So, a three way multiplexed LCD has three backplanes and three segments are controlled with one segment pin. For example in a three way multiplexed LCD with three segment inputs (SA, SB, SC) one can drive a 7-segment digit plus two special segments. These are 3 x 3 = 7 + 2 = 9 segments. The special segments can have an application specific image. (“+”, “−”, “.”, “mA”, … etc).

FIGURE 1. Schematic for LCD Triplex Driver
A typical configuration of a triplex LCD is a four digit display with 8 special segments (thus having a total of 36 segments). Fifteen outputs of the COP8 are needed; 4 x 3 segment pins and 3 backplane pins.

Common to all LCD’s is that the voltage across backplane(s) and segment(s) has to be an AC-voltage. This is to avoid electrochemical degradation of the liquid crystal layer. A segment being “off” or “on” depends on the r.m.s. voltage across a segment.

The maximum attainable ratio of “on” to “off” r.m.s. voltage (discrimination) is determined by the multiplex ratio. It is given by:

\[
\frac{V_{\text{ON}}}{V_{\text{OFF}}} \max = \sqrt{\frac{\sqrt{N} + 1}{\sqrt{N} - 1}}
\]

N is the multiplex ratio.

The maximum discrimination of a 3 way multiplexed LCD is 1.93, however, it is also possible to order a customized display with a smaller ratio. With the approach used in this application note, it may not be possible to acheive the optimum contrast acheived with a standard 3 way muxed driver. As a result of decreased discrimination (1.93 to 1.73) the user may have to live with a tighter viewing angle and a tighter temperature range.

In this application you get a \(V_{\text{rmsOFF}}\) voltage of 0.408\(V_{\text{op}}\) and a \(V_{\text{rmsON}}\) voltage of 0.707\(V_{\text{op}}\). Vop is the operating voltage of the LCD. Typical Vop values range from 3V–5V.

With the optoelectrical curve of the LCD you can evaluate the maximum contrast of the LCD by calculating the difference between the relative “OFF” contrast and the relative “ON” contrast.

In this example:
\[V_{\text{rmsON}} = 0.707V_{\text{op}}\]
\[V_{\text{rmsOFF}} = 0.408V_{\text{op}}\]
The backplane signals are generated with the voltage steps 0V, Vop/2 and Vop at the backplanes; also see Figure 4. Two resistors are necessary for each backplane to establish all these levels.

The backplane connection scheme is shown in Figure 1.

The Vop/2 level is generated by switching the appropriate COP's port pin to Hi-Z.

The following timing considerations show a simple way how to establish a discrimination ratio of 1,732.

**TIMING CONSIDERATIONS**

A Refresh cycle is subdivided in 6 timephases. Figure 4 shows the timing for the backplanes during the equal distant timephases 0…5.

**TABLE 1. Possible Segment ON/OFF Variations**

<table>
<thead>
<tr>
<th>Tablent Address</th>
<th>Segment A</th>
<th>Segment B</th>
<th>Segment C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>off</td>
<td>off</td>
<td>off</td>
</tr>
<tr>
<td>1</td>
<td>on</td>
<td>off</td>
<td>off</td>
</tr>
<tr>
<td>2</td>
<td>off</td>
<td>on</td>
<td>off</td>
</tr>
<tr>
<td>3</td>
<td>on</td>
<td>on</td>
<td>off</td>
</tr>
<tr>
<td>4</td>
<td>off</td>
<td>off</td>
<td>on</td>
</tr>
<tr>
<td>5</td>
<td>on</td>
<td>off</td>
<td>on</td>
</tr>
<tr>
<td>6</td>
<td>off</td>
<td>on</td>
<td>on</td>
</tr>
<tr>
<td>7</td>
<td>on</td>
<td>on</td>
<td>on</td>
</tr>
</tbody>
</table>

Figures 5, 6, 7, 8, 9, 10, 11, 12 below show all possible combinations of controlling a "Segment Triple" with help of the 3 backplane connections and one segment pin. The segment switching has to be done according to the ON/OFF combination required (see also Table 1).

Each figure shows in the first 3 graphs the constant backplane timing. The 4th graph from the top shows the segment control timing necessary to switch the 3 segments (SA/SB/SC), activated from one pin, in the eight possible ways.

The 3 lower graphs show the resulting r.m.s. voltages across the 3 segments (SA, SB, SC).
Segment/Backplane Control-Timing

FIGURE 7.

FIGURE 8.
### Segment/Backplane Control-Timing

<table>
<thead>
<tr>
<th>TIMEPHASE</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BP1</strong></td>
<td>VOP</td>
<td>VOP/2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>BP2</strong></td>
<td>VOP</td>
<td>VOP/2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>BP3</strong></td>
<td>VOP</td>
<td>VOP/2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SA</strong></td>
<td>VOP</td>
<td>VOP/2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SB</strong></td>
<td>VOP</td>
<td>VOP/2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SC</strong></td>
<td>VOP</td>
<td>VOP/2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**FIGURE 9.**

<table>
<thead>
<tr>
<th>TIMEPHASE</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BP1</strong></td>
<td>VOP</td>
<td>VOP/2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>BP2</strong></td>
<td>VOP</td>
<td>VOP/2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>BP3</strong></td>
<td>VOP</td>
<td>VOP/2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SA</strong></td>
<td>VOP</td>
<td>VOP/2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SB</strong></td>
<td>VOP</td>
<td>VOP/2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SC</strong></td>
<td>VOP</td>
<td>VOP/2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**FIGURE 10.**

**tiphtab address = 4**

**tiphtab address = 5**
FIGURE 11.

FIGURE 12.
REFRESH FREQUENCY

One period with six timephases is called a refresh cycle (also see Figure 4).

The refresh cycle should be in a frequency range of 30…60 Hz. A frequency below 30 Hz will cause a flickering display. On the other hand, current consumption increases with the LCD's frequency. So it is also recommended to choose a frequency below 60 Hz.

In order to periodically update the µC’s port pins (involved in backlight or segment control) at the beginning of a new timephase, the COP8 needs a timebase of typ. 4 ms which is realized with an external RC-circuit at the G0/INT pin.

The G0 pin is programmable as input (Schmitt Trigger). The conditions for the external interrupt could be set for a low to high transition on the G0 pin setting the IPND-flag (external interrupt pending flag) upon an occurrence of such a transition. The external capacitor can be discharged, with the G0 pin configured as Push/Pull output and programmed to "0". When, switching G0 as input the Cap. will be charged through the resistor, until the threshold voltage of the Schmitt-Trigger input is reached. This triggers the external interrupt. The first thing the interrupt service routine has to do is to discharge the capacitor and switch G0 as input to restart the procedure.

This timing method has the advantage, that the timer of the device is free for other tasks (for example to do an A/D conversion).

The time interval between two interrupts depends on the RC circuit and the threshold of the G0 Schmitt Trigger V_th.

The refresh frequency is independent of the clock frequency provided to the COPs device.

The variations of “threshold” levels relative to V_CC (over process) are as follows:

\[
\begin{align*}
\frac{V_{\text{th}}}{V_{\text{CC}}} \min & = 0.376 \\
\frac{V_{\text{th}}}{V_{\text{CC}}} \max & = 0.572
\end{align*}
\]

at V_CC = 5V

Charge Time:

\[ T = \frac{-\ln(1-V_{\text{th}}/V_{\text{CC}})}{RC} \]

To prevent a flickering display one should aim at a minimum refresh frequency of f_min = 30 Hz. This means an interrupt frequency of f_int = 6 x 30 Hz = 180 Hz. So, the maximum charge up time T_max must not exceed 5.5 ms (T_min = 2.78 ms).

With the formula:

\[
RC_{\max} = T_{\max}/(-\ln(1-(V_{\text{th}}/V_{\text{CC}})\max)) = 5.5 \text{ ms} \times 0.849
\]

\[ RC_{\max} = 6.48 \text{ ms} \]

\[ RC_{\min} = 5.98 \text{ ms} \]

The maximum RC time-constant is calculated. The minimum RC time constant can be calculated similarly.

A capacitor in the nF-range should be used (e.g. 68 nF), because a bigger one needs too much time to discharge. To discharge a 68 nF Cap., the G0 pin of the device has to be low for about 40 µs.

On the other hand the capacitor should be large enough to reduce noise susceptibility.

When the RC combination is chosen, one can calculate the maximum refresh frequency by using the minimum value of the RC constant and the minimum threshold voltage:

\[
T_{\min} = RC_{\min}/(-\ln(1-(V_{\text{th}}/V_{\text{CC}})\min))\times RC_{\min} \times 0.472
\]

and

\[
f_{\text{ref},\max} = f_{\text{int},\max}/6 = 1/(T_{\min} \times 6)
\]

In the above example one timephase would be minimum 2.82 ms long. This means that about 250 instructions could be executed during this time.

SOFTWARE

The software for the triplex LCD drive-demo is composed of three parts:

1. The initialization routine is executed only once after resetting the device, as part of the general initialization routine of the main program. The function of this routine is to configure the ports, set the timephase counter (tiphase) to zero, discharge the external capacitor and enable the external interrupt.

The initialization routine needs 37 bytes ROM. Figure 13 shows the flowchart of this routine.

![Flowchart for Initialization Routine](FIGURE 13. Flowchart for Initialization Routine)

2. The update routine calculates the port-data for each time-phase according to the BCD codes in the RAM locations ‘digit1’… ‘digit4’ and the special segments. This routine is only called if the display image changes.

The routine converts the BCD code to a list 1st, which is used by the refresh routine. Figure 14 gives an overview and illustrates the data flow in this routine.

In Figure 15 the data flow chart is filled with example data according to the display image in Figure 16.

First the routine creates the seg1st (4 bytes long), which contains the “on/off” configuration of each segment of the display. The display has 36 segments but the 4 bytes have only 32 bits, so the four special segments S1 are stored in the specbuf location. The bcdsegtab table (in ROM) contains the LOOK-UP data for all possible Hex numbers from 0 to F.

The routine takes three bits at the beginning of each time-phase from the seg1st.

These 3 bits address the 8 bytes of the tiphtab table in ROM. Each byte of this table contains the time curve for a segment pin (only 6 bits out of 8 are used). Using this information, the program creates the lists for port D and port L.
Every byte of this list contains the timing representatives for the pins D0–D3 and L0–L7, to allow an easy handling of the refresh routine.

The external interrupt has to be disabled while the copy routine is working, because the mixed data of two different display images would result in improper data on the display. Figure 17 shows the flowchart of the update routine. The Flowchart of the convert subroutine is shown in Figure 18.

**MEMORY REQUIREMENTS**

ROM: 152 bytes incl. look up tables

RAM: 43 bytes (Figure 15 illustrates the RAM locations)

**FIGURE 14. Data Flow Chart for Update Routine**
FIGURE 15. Data Flow Chart for Update Routine
3. The refresh routine is the interrupt service routine of the external interrupt and is invoked at the beginning of a new timephase. First the routine discharges the external capacitor and switches the G0/INT pin back to the input mode, to initialize the next timephase. The backplane ports G2, G4 and G5 and the segment pin ports D and L are updated by this routine according to the actual timephase. For the backplanes the data are loaded from the \textit{bptab} table in ROM. Table 2 shows how the \textit{bptab} values are gathered. Figure 20 shows the flowchart for the refresh routine.

**TIME REQUIREMENTS**

The routine runs max. 150 cycles.

For a non flickering display, the refresh frequency must be 30 Hz minimum. One refresh cycle has six timephases and is max. 33 ms long. So each timephase is 5.5 ms long. With an oscillator (CKI) frequency of 2 MHz, one instruction cycle takes $1/(2 \text{ MHz}/10) = 5 \mu s$ to execute. During one timephase the controller can execute:

$$\frac{5.5 \text{ ms}}{5 \mu \text{s}} = 1100 \text{ cycles}.$$ 

So the refresh routine needs $\frac{134}{1100} = 0.122 = 12.2\%$ of the whole processing time (in this case).

With a refresh frequency of 50 Hz the routine needs about 20.1\% of the whole processing time.

The refresh routine needs about 103 ROM bytes.

**TABLE 2. Phase Values**

<table>
<thead>
<tr>
<th>Tiphase</th>
<th>G5</th>
<th>G4</th>
<th>G2</th>
<th>Portg Data</th>
<th>Hex</th>
<th>Portg Config.</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0/0</td>
<td>0/0</td>
<td>1/1</td>
<td>XX00X1XX</td>
<td>04</td>
<td>XX00X1XX</td>
<td>04</td>
</tr>
<tr>
<td>1</td>
<td>0/0</td>
<td>1/1</td>
<td>0/0</td>
<td>XX01X0XX</td>
<td>10</td>
<td>XX01X0XX</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>1/1</td>
<td>0/0</td>
<td>0/0</td>
<td>XX10X0XX</td>
<td>20</td>
<td>XX10X0XX</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>0/0</td>
<td>0/0</td>
<td>0/1</td>
<td>XX00X0XX</td>
<td>00</td>
<td>XX00X1XX</td>
<td>04</td>
</tr>
<tr>
<td>4</td>
<td>0/0</td>
<td>0/1</td>
<td>0/0</td>
<td>XX00X0XX</td>
<td>00</td>
<td>XX01X0XX</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>0/1</td>
<td>0/0</td>
<td>0/0</td>
<td>XX00X0XX</td>
<td>00</td>
<td>XX10X0XX</td>
<td>20</td>
</tr>
</tbody>
</table>

Data/configuration register of portg

- 0/0: Hi-Z input
- 0/1: output low
- 1/1: output high

**FIGURE 16. Display Example**

[Display Example Image]
### SUMMARY OF IMPORTANT DATA

<table>
<thead>
<tr>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCD type</td>
<td>3 way multiplexed</td>
</tr>
<tr>
<td>Amount of segments</td>
<td>36</td>
</tr>
<tr>
<td>$V_{CC}$ = (range)</td>
<td>2.5V to 6V</td>
</tr>
<tr>
<td>Oscillator frequency</td>
<td>2 MHz (typ.)</td>
</tr>
<tr>
<td>Instruction cycle time</td>
<td>5 µs</td>
</tr>
<tr>
<td>ROM requirements: init routine</td>
<td>37 bytes</td>
</tr>
<tr>
<td>update routine</td>
<td>152 bytes</td>
</tr>
<tr>
<td>refresh routine</td>
<td>103 bytes</td>
</tr>
<tr>
<td>total</td>
<td>292 bytes</td>
</tr>
<tr>
<td>RAM requirements:</td>
<td></td>
</tr>
<tr>
<td>permanent use</td>
<td>25 bytes</td>
</tr>
<tr>
<td>temporary use</td>
<td>18 bytes</td>
</tr>
<tr>
<td>stack</td>
<td>6 bytes</td>
</tr>
<tr>
<td>total</td>
<td>49 bytes</td>
</tr>
<tr>
<td>(also see Figure 19)</td>
<td></td>
</tr>
<tr>
<td>Timer</td>
<td>not used</td>
</tr>
<tr>
<td>External Interrupt</td>
<td>with RC circuit used as time-base generator</td>
</tr>
<tr>
<td>Ports D, L</td>
<td>used for LCD control</td>
</tr>
<tr>
<td>Port G</td>
<td>3 G-pins are still free for other purposes +</td>
</tr>
<tr>
<td>Port I</td>
<td>can be used as key-inp.</td>
</tr>
</tbody>
</table>
FIGURE 17. Flowchart for Update Routine
FIGURE 18. Flowchart for Convert Subroutine
FIGURE 19. RAM Assignment
FIGURE 20. Flowchart for Refresh-Routine

- DISCHARGE EXTERNAL C
- STORE ACCU, B, CARRY, MCARRY
- POINTER = TIPHASE * 2
- LOAD PORTD BYTE FROM LST
- STORE IT TO PORTD BUFFER
- LOAD PORTL BYTE FROM LST
- STORE IT TO PORTL BUFFER
- POINTER ON BACKPLANE TABLE
- LOAD PORTG BYTE FROM ROW
- STORE IT TO PORTG BUFFER
- LOAD PORTGC BYTE FROM ROW
- STORE IT TO PORTGC BUFFER
- COPY BUFFER BYTES TO PORTD, PORTL, PORTG AND PORTGC
- INC TIPHASE
- TIPHASE = 6?
  - NO
  - YES
    - SET TIPHASE = 0
    - RESTORE CARRY, HAFICARRY BIT, B, ACCU
    - RETURN FROM REFRESH ROUTINE

www.national.com 16
Listing

; DEMO FOR COP820CJ:
; 3 WAY MULTIPLEXED LCD DRIVER DEMO
; CONSTANT DISPLAY "01A3" and two special segments on
; .incl cop820cj.inc

;RAM assignments

tipase=01E
special=01F ;this byte must contain the
;on/off configuration of
;the extra segments
;('-','low bat',etc.)

digit1=020 ;in these RAM locations the
digit2=021 ;BCD code of the display
digit3=022 ;digits are stored.
digit4=023 ;

acceso=024 ;accu buffer used during
bsto=025 ;interrupt service routine
pswsto=026 ;b buffer
;psw buffer

;register definition:

podbuf=0f0 ;portd buffer
polbuf=0f1 ;portl buffer
pogbuf=0f2 ;portgd buffer
pogcbuf=0f3 ;portgc buffer
flags=0f4 ;flag byte for podfla

;flag definition in flags byte
podfla=07

;************* initialization routine **************************************************

init:

ld sp,#02f ;initialize stackpointer

ld portlc,#0ff ;port 1 output
ld portgc,#037 ;port g,G1,G2,G4,G5 are
;outputs
ld portgd,#00 ;all outputs low, all
;inputs Hi-z
;C at G0 is discharged
ld tipase,#00 ;begin with timephase 0
ld psaw,#002 ;ext. interrupt enable

AVR1076-01
begin: sbit #gie, psw
    ; interrupts are welcome now
rbit #00, portgc
    ; now the external C can be
    ; charged
ld b, #special
    ; two special segments
ld [b+], #088
    ; are 'ON'
    ; display: "01A3"
ld [b+], #00
    ; digit1
ld [b+], #001
    ; digit2
ld [b+], #00A
    ; digit3
ld [b], #003
    ; digit4

;************* main program ****************************

; ********* main program **************

loop:
    ;jsr update
    ;jp loop

; ************ update subroutine ****************************

; RAM definitions:

specbuf=01C ; buffer for 'special'
temp=01D    ; temporary used

; pointer on tables:

podlist=010 ; address of list for port d
pollist=016 ; address of list for port l
lst =000    ; main list for display
            ; routine to refresh
            ; port d, l each timephase

seglist=00C ; this list contains the
            ; on/off configuration of
            ; the segments

=0200
.local

update:
    ld a, special
    ; load 'special' register
    x a, specbuf
    ; to the buffer 'specbuf'
    ld x, #seglist
    ; x points the segmentlist
    ld b, #digit1
    ; b points digitlist

nxtdig:
    ld a, [b+] ; load BCD code of
    ; current digit
    add a, #L(bcdsegtab) ; set pointer on look up
    ; table for segment setting
    ; load segment data of
    ; current digit
    ; store it to RAM
    ; load special bit
    ; to carry
; prepare for next segment
ifnc
rbit #2,temp
; special bit not set ?
; then reset it in the temp byte
ld a,temp
; store temp
x a,[x+]
; to the seglst list
ifbne #04
; if not last digit
jp nxtdig
; load data for next digit
sbit #podfla,flags
; set flag for working at port d list
jsr convert
; convert 3 bits from the segment bytes to the timephaselst for port d

; shift with carry
shwc:
ld b,#seglist
; b points seglst
nxtshwc:
ld a,specbuf
; load special segment bit
trc a
; to carry
x a,specbuf
; prepare for next special segment
ld a,[b]
; shift the segmentbyte
trc a
; three positions right
trc a
; and append the special segment byte
rcc a
; store shifted byte
ifbne #00
; end of segment list
; not reached ?
jp nxtshwc
; then shift the next segment byte
rbit #podfla,flags
; reset flag for working at port 1 list
jsr convert
; convert 3 bits of the segment bytes to the timephaselst for port 1

; shift (without carry)
shift:
ld b,#seglist
; b points segment list
nxtshift:
ld a,[b]
; load segment byte
rcc a
; shift the segmentbyte
trc a
; three positions right
rcc a
; store shifted byte
ifbne #00
; end of segment list
; not reached ?
jp nxtshift
; then shift the next segment byte
jsr convert ;convert 3 bits of the
 ;segment bytes to the
 ;timephaselst for port 1

;copy portdata to the list on which the refresh routine will access

copy:
  rbit #eni,psw ;disable interrupt to
  ld b,#podlst ;prevent fail display
  ld x,#lst ;b points podlst
  nxt1:
    ld a,[b+] ;x points refresh list
    swap a ;load portbyte
    x a,[x+] ;swap it
    ld a,[x+] ;store it to refresh list
    inc x ;increment x
    ifbne #06 ;if the end of the podlst
    ;is not reached
    jp nxt2 ;then next timephase
    ld b,#pollst ;b points pollst
    nxtl:
      ld a,[x+] ;x points refresh list
      ld a,[b+] ;increment x
      swap a ;load portbyte
      x a,[x+] ;swap it
      ifbne #0C ;store it to refresh list
      ;if the end of the pollst
      ;is not reached
      ;then next timephase
      ;refresh routine allowed
      ;again
    jp nxt1 ;end of update routine
    ret

;subroutines for update routine:

convert:
  ld x,#seglist ;x points segment list
  nxtsg1:
    ld a,[x+] ;load segment byte
    and a,#007 ;mask out first three bits
    add a,#L(tiptab) ;pointer on timephasetable
    ld x,#laid ;load timephaset curve for
    ;one segment pin
    ;b points list for portd
    ;ifbit #podfla,flags ;working at podlst?
    ;then b points on podlst
    ;shift timephaset data according to 3 bits (8 combinations are
    ;possible with 3 segments)
    ;tipsh:
    ;store timephaset curve to
    ;temp buffer
    nxtphsh:
      ld a,temp ;load timephaset curve again
      rrc a ;shift out one bit into
x a,temp
ld a, [b]
rrc a
x a, [b+]
ld a, #pollst
ifeq a, b
jp eplst
ifbne #0C
jp nxtpsh

eplst:
ld a, #L(segment+4)
ifgt a, x
jp nxtegl
ret

bcdsegtab:
; in this bytes are the on/off configuration of the segments
; for a digit are stored. there are only 7 bits of each byte
; the configuration of the 2 special segments is stored
; in the 'special' byte.

.BYTE 0EF, 007, 0BD, 03F  ; '0'...3'
.BYTE 057, 07E, 0FE, 00F  ; '4'...7'
.BYTE 0FF, 07F, DFF, 0F6  ; '8'...B'
.BYTE 0EC, 0B7, 0FC, 0DC  ; 'C'...F'

tiphtab:
; one pin controls 3 segments. there are 8 possible
; combinations. for each combination there is one byte.
; 6 bits of one byte control the pin for each timephase.

.BYTE 007, 00E, 015, 01C, 023, 02A, 031, 038

;*************** interrupt service routine **********************

refresh:
x a, accsto
ld a, b
x a, bsto

ld b, #portgd
rbit #00, [b]
ld a, [b+]
shl #00, [b]

;carry bit
;store shifted curve
;load portbyte
;shift in one bit from
;carry bit
;store shifted portbyte
;again
;end of podlist?

; then return
; else end of podlist
; if the end of the segment
; list is not reached
; work at next segment byte

=0ff
rbit #00,[b] ; C can be charged again
ld b,#psw
rbit #ipnd,[b] ; reset ext. interrupt
; pending flag

ld a,[b] ; load psw
x a,pswsto

ld a,tiphaser
add a,tiphaser

x a,b ; store accu in b
ld a,[b+]

x a,podbuf
ld a,[b+]

x a,polbuf
ld a,b
add a,#L(bptab)-2

x a,b
ld a,b

laid
x a,pogdbuf

ld a,[b+]
ld a,b

laid
x a,pogcbuf

ld b,#podbuf
ld a,[b+]

x a,portd
ld a,[b+]

x a,portld

ld portgc,#00

ld a,[b+]

x a,portgd
ld a,[b+]

x a,portgc

ld a,tiphaser
inc a

tfeq a,#06
ld a,#00
x a,tiphaser
ld b,#pswsto
rc
ifbit #07,[b] ; restore carry bit
sbit #07, psw
ifbit #06,[b] ; restore halfcarry bit
sbit #06, psw

ld a, bsto ; restore b
x a, b
ld a, accsto ; restore accu

repi ; return from lcmd
; refresh routine

bptab:
.BYTE 004, 004, 010, 010, 020, 020
.BYTE 000, 004, 000, 010, 000, 020

.END
LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.

2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and National reserves the right at any time without notice to change said circuitry and specifications.

National Semiconductor
Americas
Tel: 1-800-272-9959
Fax: 1-800-737-7018
Email: support@nsc.com

www.national.com

National Semiconductor Europe
Fax: +49 (0) 1 80-530 85 86
Email: europe.support@nsc.com

National Semiconductor Asia Pacific Customer Response Group
Tel: 65-2544466
Fax: 65-2504466
Email: sea.support@nsc.com

National Semiconductor Japan Ltd.
Tel: 81-3-5639-7507
Fax: 81-3-5639-7507

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and National reserves the right at any time without notice to change said circuitry and specifications.
## IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any damages arising out of the use of TI products in such safety-critical applications.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

<table>
<thead>
<tr>
<th>Products</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audio</td>
<td>Communications and Telecom</td>
</tr>
<tr>
<td>Amplifiers</td>
<td>Computers and Peripherals</td>
</tr>
<tr>
<td>Data Converters</td>
<td>Consumer Electronics</td>
</tr>
<tr>
<td>DLP® Products</td>
<td>Energy and Lighting</td>
</tr>
<tr>
<td>DSP</td>
<td>Industrial</td>
</tr>
<tr>
<td>Clocks and Timers</td>
<td>Medical</td>
</tr>
<tr>
<td>Interface</td>
<td>Security</td>
</tr>
<tr>
<td>Logic</td>
<td>Space, Avionics and Defense</td>
</tr>
<tr>
<td>Power Mgmt</td>
<td>Transportation and Automotive</td>
</tr>
<tr>
<td>Microcontrollers</td>
<td>Video and Imaging</td>
</tr>
<tr>
<td>RFID</td>
<td></td>
</tr>
<tr>
<td>OMAP Mobile Processors</td>
<td></td>
</tr>
<tr>
<td>Wireless Connectivity</td>
<td></td>
</tr>
</tbody>
</table>

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2011, Texas Instruments Incorporated