

\*TITLE XMONXM VER 1.5C UTILITIES USED BY MONITOR

\*\*\*\*\*

BASED ON

XMON/D - DISKETTE BASED MONITOR

DATE: 21/2 1979 REVISION LEVEL 5 G J S

\*\*\* MJH 15/05/79 \*\*\*

\*\*\*\*\*

public symbols

public yrcvi,yxmio,enter  
public chari,charj,ydobl,ycr1f  
public comc,ybytut,yhxoud,yhxinh  
public yhxine,ygc,yttyhx,yconvr

external symbols

extrn rcvbuf,xmibuf,tto,tti,TTI2,IOWAIT  
EXTRN BYTUT,TTYHX,GC,CONVR

equivalences

kbmax equ 30  
uarts equ 0f7h  
intco equ 0e7h  
cr equ 13  
lf equ 10  
sp EQU 20h

UTILITY TO READ FROM RECEIVER BUFFER

YRCVI:

PUSH H  
PUSH D  
RCVI1: LXI H,RCVBUF  
MOV A,M  
ORA A  
JNZ TTI2  
LHLD IOWAIT  
ORA H  
CNZ ENTER  
JMP RCVI1

ENTER:

PCHL

UTILITY TO STORE IN XMIT BUFR

YXMIO:

PUSH H  
PUSH D  
ANI 7FH  
PUSH PSW

XMIO1:

LXI H,XMIBUF 25C0  
MOV A,M  
CPI KBMAX ;Buffer full ?  
JNZ XMIO2 ;No  
LHLD IOWAIT ;Is alternative wait loop installed ?  
ORA H  
CNZ ENTER ;Yes  
JMP XMIO1

XMIO2:

INX H  
MOV E,M  
MOV D,H  
ADD E  
MOV E,A  
SUB L ;Top of buffer ?  
SBI KBMAX+1  
JM XMIO3 ;No  
ADD L

KBMAX=30

25C0  
25A

```

ADD      L
INR      A
MOV      E,A
XMI03:   POP      PSW
STAX     D      ;Load character to buffer D->H      E = Adr +
DCX      H
DI
INR      M      ;Update character count
IN       UARTS
CMA
ANI      4
RLC
OUT      INTC0   ; ENABLE XMIT INTERRUPT
EI
POP      D
POP      H
RET

;
;CHARI:   CALL     GC      ; CHARACTERS IN
;CHARJ:
CPI      CR      ;      HL: ADDRESS
RZ
cpi      ','
RZ
MOV      M,A
INX      H
DCR      B
JNZ      CHARI
STC
RET

;
; SUBROUTINES
;
YDOBLU:  PUSH     PSW      ;***
CALL     COMC      ; DOUBLE BYTE HEX OUTPUT, ADDRESS IN HL.
db
MOV      A,M
CALL     BYTUT
DCX      H
MOV      A,M
CALL     BYTUT
POP      PSW      ;***
RET      ;***

;
; ygc:
call     tti      ; get char with echo
CPI      60H      ; get character
JC       TTO      ; ADJUST TO CAPITALS
ANI      5FH
jmp      tto

;
; YCRLF:
PUSH     PSW      ; NEW LINE
CALL     COMC
DB       CR
CALL     COMC
DB       LF
POP      PSW
RET

;
; COMC:
XTHL
MOV      A,M      ; SINGLE CHAR OUTPUT
INX      H
XTHL
JMP      TTO

;
; BYTUT:
PUSH     PSW      ; BYTE OUTPUT AT TWO HEX
RAR
RAR

```

```

RAR
RAR
CALL    HXUT
POP      PSW
HXUT:   ANI    0FH      ; FOUR RIGHT BITS OUT AS HEX
        ADI    90H
        DAA
        ACI    40H
        DAA
        JMP    TTO

```

```

;
YHXOUD:
        PUSH   PSW      ;***
        MOV    A,D      ; DE OUT AS 4 HEXA CHARS
        CALL   BYTUT    ; FOLLOWED BY SPACE
        MOV    A,E
        CALL   BYTUT
        CALL   COMC
        DB     SP
        POP    PSW      ;***
        RET

```

```

;
; HEXA VALUE INPUT TO HL
;

```

```

YHXINH:
        LXI    H,0
HXH1:   CALL   TTYHX
        RC
        DAD    H
        DAD    H
        DAD    H
        DAD    H
        ORA    L
        MOV    L,A
        JMP    HXH1

```

```

;
; GET CHARACTER FROM TTY
; IF VALID HEXA: RETRN HEXA VALUE IN A,
; IF NOT: RETURN CHARACTER IN A WITH CARRY SET
;

```

```

YTTYHX:
        PUSH   B
        CALL   GC
        CALL   CONVR
        JC     TTYHY
        POP    B
        RET
TTYHY:  MOV     A,B
        POP    B
        RET

```

```

;
YCONVR:
        MOV     B,A
        SUI     30H
        RC      ; 0 - 2F OUT
        ADI     0E9H ; 47-FF OUT
        RC
        ADI     6
        JP      CON1 ; A - F
        ADI     7
        RC      ; 3A - 40 OUT
CON1:   ADI     10
        ORA     A      ; CLEAR CARRY
        RET

```

```

;
; GET HEXA VALUE TO E, A(0)=ZERO: NO HEXA DIGITS ENTERED
;

```

```

YHXINE:
        LXI    D,0
        PUSH   B
HXE2:   CALL   TTYHX
        JC     HXE1

```

|       |     |       |
|-------|-----|-------|
|       | MOV | C,A   |
|       | MOV | A,E   |
|       | ADD | A     |
|       | ADD | A     |
|       | ADD | A     |
|       | ADD | A     |
|       | ADD | C     |
|       | MOV | E,A   |
|       | MVI | D,80H |
|       | JMP | HXE2  |
| ;     |     |       |
| HXE1: | POP | B     |
|       | CP  | SP    |
|       | JZ  | HXE3  |
|       | CPI | CR    |
|       | JZ  | HXE3  |
|       | STC |       |
|       | RET |       |
| HXE3: | ORA | D     |
|       | RET |       |
| ;     |     |       |
|       | end |       |