# DEVICE SERVICE ROUTINE SPEC

# FOR TI 99/4 PERSONAL COMPUTER

## Contents




Personal Computer Division


Revised, Jan. 9, 1981

## 1.0 Introduction

This document describes the hardware and software interfaces between the console and peripherals of the 99/4 personal computer family. The purpose of this document is for the reference of a third party, who wants to design new peripherals around the 99/4 family.

### 1.1 Interface in General

The 99/4 console has a 44-pin male card edge-connector, called the I/O Bus, on its right edge. The I/O Bus contains all the informations needed for the Console to communicate with the peripherals. It is suggested that each peripheral has its own power supply, instead of sharing the voltages with the console through the I/O Bus.

Each peripheral should include a non-volatile Device Service Routine(DSR) software package to drive the peripheral. The DSR communicates with the console software through certain memory locations. The console software sets up information in these memory locations and passes them to the user selected peripheral. From there on, the selected peripheral's DSR should have the capability to interpret the information setup by the console software, physically drive the peripheral and pass the peripheral's data or status to the console software if necessary.

## 2.0 I/O Bus

The purpose of this section is to describe the pin assignments of the I/O Bus, functions of I/O signals, sources of Output Signals and destinations of Input Signals.

### 2.1 I/O Bus Pin Assignments and Descriptions

| Pin # | Notation | I/O | Description |
|---|---|---|---|
| 1 | +5V | O | 5V Power Supply |
| 2 | SBE | O | low when MPU read from >90XX / write to 94XX memor |
| 3 | RESET | O | Master Reset, low active |
| 4 | -EXT INT | I | External Interrupt, low active |
| 5 | A5 | O | Address Bit 5 |
| 6 | A10 | O | Address Bit 10 |
| 7 | A4 | O | Address Bit 4 |
| 8 | A11 | O | Address Bit 11 |
| 9 | DBIN | O | derived from MPU's DBIN pin, same parity |
| 10 | A3 | O | Address Bit 3 |
| 11 | A12 | O | Address Bit 12 |
| 12 | HOLD/READY | I | if device/memory is ready after being addressed by by MPU in memory R/W cycle, device/memory should cause this input to go high |
| 13 | -LOAD | I | to 9900's LOAD pin |
| 14 | A8 | O | Address Bit 8 |
| 15 | A13 | O | Address Bit 13 |
| 16 | A14 | O | Address Bit 14 |
| 17 | A7 | O | Address Bit 7 |
| 18 | A9 | O | Address Bit 9 |
| 19 | CRUOUT/A15 | O | CRU Output/Address Bit 15, LSB |
| 20 | A2 | O | Address Bit 2 |
| 21 | GND | | Signal Ground |
| 22 | -CRUCLK | O | inversion of MPU's CRUCLK pin |
| 23 | GND | | Signal Ground |
| 24 | -PHI3 | O | inversion of Phase 3 Clock |
| 25 | GND | | Signal Ground |
| 26 | -WE | O | derived from MPU's $\overline{WE}$ pin, same parity |
| 27 | GND | | Signal Ground |
| 28 | -MBE | O | low when MPU addressing >4000 - >5FFF Memory |
| 29 | A6 | O | Address Bit 6 |
| 30 | A1 | O | Address Bit 1 |
| 31 | A0 | O | Address Bit 0, MSB |
| 32 | -MEMEN | O | derived from MPU's $\overline{MEMEN}$ pin, same parity |
| 33 | CRUIN | I | CRU Input to MPU |
| 34 | D7 | I/O | Data Bus Bit 7, LSB |
| 35 | D4 | I/O | Data Bus Bit 4 |
| 36 | D6 | I/O | Data Bus Bit 6 |
| 37 | D0 | I/O | Data Bus Bit 0, MSB |
| 38 | D5 | I/O | Data Bus Bit 5 |
| 39 | D2 | I/O | Data Bus Bit 2 |
| 40 | D1 | I/O | Data Bus Bit 1 |
| 41 | HOLDA/IAQ | O | logic OR of MPU's HOLDA and IAQ pins |
| 42 | D3 | I/O | Data Bus Bit 3 |
| 43 | -5V | O | -5V Power Supply |
| 44 | AUDIO IN | I | to Sound Generator Controller's AUDIO IN pin |

## 3.0 Hardware Structure of DSR

### 3.1 DSR ROM

Normally, a DSR is written in 9900 Assembly Language and is housed in a ROM, which is itself part of the 99/4 family's peripheral. All DSR must begin at address >4000, and should not exceed >5FFF. All the 8 Data Pins of the DSR ROM must be buffered before connecting to the data bus of the I/O Bus. This buffer is enabled and disabled by a pre-assigned CRU output bit, which is controlled by the console software, so that not more than one DSR is accessed at any time.

### 3.2 CRU Mapping

The CRU I/O is used by the system to access the peripherals, if the speed of data transfer is not too crucial. The decoding format for the CRU addressing is indicated as :

| A0 | A1 | A2 | A3 A4 A5 | A6 A7 | A8 A9 A10 A11 A12 A13 A14 | A15/CRUOUT |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | User ID | Device | CRU I/O Bit Decode | 0 |

The CRU address space, ranging from >0 to >1FFE with A0, A1, A3 and A15 unused, is broken into 8 blocks of 512 Bits each. The User ID, indicated through the Address Decoding of A3, A4 and A5, represnets each of the 512 Bit Block. They are assigned as follows :

| MPU Address | A3 | A4 | A5 | Assignment |
|---|---|---|---|---|
| 0000-03FE | 0 | 0 | 0 | Console Use |
| 0400-07FE | 0 | 0 | 1 | Third Party Peripheral without DSR |
| 0800-0BFE | 0 | 1 | 0 | Future Peripheral without DSR |
| 0C00-0FFE | 0 | 1 | 1 | Future Peripheral without DSR |
| 1000-13FE | 1 | 0 | 0 | TI Peripheral Space #1 |
| 1400-17FE | 1 | 0 | 1 | TI Peripheral Space #2 |
| 1800-1BFE | 1 | 1 | 0 | TI Peripheral Space #3 |
| 1C00-1FFE | 1 | 1 | 1 | TI Peripheral Space #4 |

The standard device decoding through A6 and A7 gives a total of 4 Device Block within each User ID Block. The CRU I/O Bit decoding through A8-A14 allows 128 addressable bits each of input and output. In the 128 CRU Output bits, the first bit, with A8-A14 all 0's, is reserved for enabling the Data Buffer of the DSR ROM in each peripheral as mentioned in the last subsection. Setting this bit to logic one enables the DSR ROM, while setting it to logic zero disables the DSR ROM.

## 4.0 Software Structure of DSR

A DSR must follow a certain specific format in order to communicate with the console software properly. The purpose of this predefined format is to let the console software have the least overhead and the DSR have the maximum flexibility in device servicing.

A DSR, in general, contains the following elements :

1. Symbol Definition Block,
2. Header and Linkage Block,
3. Power-up Routine(optional),
4. Main Device Service Routine,
5. Interrupt Routine(optional).

Each of the above elements will be discussed in detail in the following sections.

### 4.1 Symbol Definition Block

The Symbol Definition Block serves two purposes. First, it equates frequently used Data or Address to a Symbol for ease of recognition. Secondly, it specifies the CPU RAM location in certain way, so that each DSR can be used around different 99/X consoles with the least modifications. The first purpose is common in every program, but the second one needs some more explanation.

In the 99/X family, each model has a different memory structure within the console. Therefore, care must be taken in handling memory addressing in a DSR, so that the DSR can support different 99/X models. The console software always enters the DSR through the instruction :

```
        BL    *R9,
```

where R9 contains the DSR entry address. Upon entry of the DSR, the Work Space Pointer contains the beginning address of the Register File, which the console 2software uses. If all the CPU RAM which the DSR may access are predefined with respect to this Work Space Pointer, the DSR does not have to know the memory map of each console. The DSR can address each CPU RAM through Index Addressing via the Work Space Pointer. For example, if one wants to move the content of a CPU RAM, having a displacement 'DISP' with respect to the Work Space Pointer, to R0, he does the following :

```
ENTRY   STWP  R4              entry of DSR
        .
        .
        .
        MOV   @DISP(R4),R0
        .
        .
```

Some of the frequently used CPU RAM's symbols and locations are listed below.

```
PAD     EQU   >8E0           start of CPU RAM in 99/4 console
FAC     EQU   PAD+4A          start of 36 bytes available to DSR
R0LB    EQU   PAD+E1          lower byte of R0
R1LB    EQU   PAD+E3          lower byte of R1
 .
 .
OPCODE  EQU   FAC+0           beginning of PAB, I/O operation code
FLGSTS  EQU   FAC+1           PAB - Flag/Status
```

```
BUFADR  EQU   FAC+2          PAB - Data Buffer Address
LRECLN  EQU   FAC+4          PAB - Logical Record Length
CHRCNT  EQU   FAC+5          PAB - Character Count
RECNUM  EQU   FAC+6          PAB - Record Number
SCNOFF  EQU   FAC+8          PAB - Screen offset
OPTLEN  EQU   FAC+9          PAB - Option Length
DEVLEN  EQU   FAC+10         PAB - Device Length
PABVDP  EQU   FAC+12         PAB - Pointer to PAB in VDP RAM
VWA     EQU   >8C02          address for VDP Write-Adddress operation
GRD     EQU   >9800          address for GROM Read-Data operation
 .
 .
```

4.2.0 Header and Linkage Block

The DSR must contain a certain header starting at >4000 so that the linkage to the console software can be established properly. This header contains the following informations :

1. Validation flag( >AA ),
2. Name(s) of device(s) being serviced by this DSR, the device name should be 7 characters or less,
3. Entry point(s) of the device(s) being serviced by this DSR,
4. Entry point of Power-up Routine if necessary,
5. Entry point of Interrupt Routine if necessary.

4.2.1 A Sample Program for Header and Linkage Block

This section gives an example of the Header and Linkage Block in a typical DSR. The syntax of DX10 Assembler is obeyed in this program.

```
*****************************************************
*                                                   *
*        Sample Program Showing DSR Header Format   *
*                                                   *
*****************************************************
        RORG >4000              start of DSR
        BYTE >AA                telling console this is a valid DSR
        BYTE 1                  version #, always 1
        DATA 0                  not used in DSR calls, so leave it 0
        DATA PWRLNK             Power-up Routines' Link.  'PWRLNK' is replaced
                                * by '0' if power-up set up is not necessary
        DATA 0                  not used in DSR calls, leave it 0
        DATA DSRLNK             DSR Link, can't be 0 here
        DATA 0                  not used in DSR calls, leave it 0
        DATA INTLNK             Interrupt Routine's Link.    'INTLNK' is
                                *replaced by '0' if interrupt is not used
        DATA 0                  not used in DSR calls, leave it 0
*
*       *** This Power-up Routine can be omitted if power-up
*       *** initialization is not necessary for this peripheral
*
PWRLNK  DATA 0                  linkage, set to 0
        DATA PWRUP              entry point of power-up routine
        BYTE 0                  name length, set to 0
        EVEN
        .
        .
PWRUP   .                       entry of power_up routine
        .
        .
        B    *R11               return to console software through R11,
        .                       if it is not destroyed
        .
        .
*
*       *** This Interrupt Routine can be omitted if Interrupt
*       *** Request is never issued by this peripheral
*
INTLNK  DATA 0                  linkage, set to 0
        DATA INTDSR             entry point of the Interrupt Routine
        BYTE 0                  name length, set to 0
        EVEN
        .
        .
INTDSR  .                       entry of interrupt routine:
        .                           if interrupt_request_from_this_peripheral
        .                                   = true
        .                           then
        .                              begin interrupt_service ;
        .                                    reset Interrupt_request;
        .                                    goto INTEND
        .                              end
        .                           else goto INTEND;
        .
INTEND  B    *R11               return to console software through R11,
        .                       if it is not destroyed
        .
        .
*
*       *** Main Device Service Routine. Assuming three devices,
*       *** with names 'DEVICE', 'DEVICE/1', 'DEVICE/2', are
*       *** supported in this peripheral.
*
```

```
DSRLNK  DATA DSRLK2             linkage to next device field
        DATA ENTRY1             entry point of 1st device
        BYTE 6                  name length of 1st device
        TEXT 'DEVICE'           name of 1st device
        EVEN
DSRLK2  DATA DSRLK3             linkage to next device field
        DATA ENTRY2             entry point of 2nd device
        BYTE 8                  name length of 2nd device
        TEXT 'DEVIC/1'          name of 2nd device
        EVEN
DSRLK3  DATA 0                  linkage to next device field(none)
        DATA ENTRY3             entry point of 3rd device
        BYTE 8                  name length of 3rd device
        TEXT 'DEVIC/2'          name of 3rd device
        EVEN
        .
        .
ENTRY1  .                       entry of 1st device servicing
        .
        .
ENTRY2  .                       entry of 2nd device servicing
        .
        .
ENTRY3  .                       entry of 3rd device servicing
        .
        .
EXIT    INCT R11                return to console software through
        B    *R11               * R11 (It must not be destroyed).
        .
        .
```

## 4.3 Power-up Routine

For some peripherals, it is necessary to initialize the hardware at power-up time. It is suggested that a power-up routine be included to do the initialization through software for these pheripherals. Power-up Link should be set in the Header Field as described above.

Power-up routines are executed whenever the system is reset by either hadrware or software. The console software searches all peripheral DSR for power-up routine addresses and executes them, if they are found. Each power-up routine can use R0-R10. Upon entry, R12 is set to the beginning address of the CRU space for that peripheral DSR(note that this address is used to enable the DSR ROM).R11 contains the return address. R13 and R15 contain the memory-mapped addresses of GROM Read Data and VDP Write Address, respectively. All VDP and GROM operatons can be indexed from these 2 registers. The power-up routine may use VDP RAM from 0 to the location pointed by CPU RAM >70, offsetted from ( Workspace Pointer - >E0 ). It can use all CPU RAM but location >55, >6D and >C0 through >DF offsetted from ( Workspace Pointer - >E0 ).

All power-up routine must return with B *R11.

## 4.4 Interrupt Routine

If the peripheral issues Interrupt Request to the 9900 CPU, the DSR should also include an interrupt routine and Interrupt Link in the Header as described above. Every interrupt that isn't recognized as a console interrupt(VDP or 9901 Timer) causes the console software to execute every interrupt routine it can find. The interrupt routine must check whether the Interrupt Request is raised by this peripheral. If it is not, exit by: B *R11.

The interrupt routine may use R1-R10. The contents of R11-R15 are the same as those of the Power-up Routine Section. The interrupt routine and Main DSR can split the allocation of CPU RAM from >4A to >6D offsetted from (Workspace Pointer - >E0 ).

All interrupt routines end with B *R11. Interrupt Request raised by the peripheral must be cleared before exit.

## 4.5 Main Device Service Routine

Please refer to the following documents :

1. GPL User's Guide, Appendix H, I
2. File Management Spec for the TI 99/4 Home Computer

To exit the DSR, do:

```
INCT   R11
B      *R11
```