INTERNATIONAL ATOMIC ENERGY AGENCY
UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION
INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
I.C.T.P., P.O. BOX 586, 34100 TRIESTE, ITALY, CABLE: CENTRATOM TRIESTE

UNITED NATIONS INDUSTRIAL DEVELOPMENT ORGANIZATION

INTERNATIONAL CENTRE FOR SCIENCE AND HIGH TECHNOLOGY
c/o INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS 34100 TRIESTE (ITALY) VIA GRIGNANO, 9 (ADRIATICO PALACE) P.O. BOX 586 TELEPHONE 040-224572 TELEFAX 040-224559 TELEX 460449 APH I

SMR/474 - 13

COLLEGE ON
"THE DESIGN OF REAL-TIME CONTROL SYSTEMS"
1 - 26 October

PROJECT SOFTWARE

L. CARCIONE
Depto de Computacion
Facultad Ciencias Exactas y Naturales
Universidad de Buenos Aires
Buenos Aires
Argentina

HOW TO SEND A COMMAND

* Check if Multi is busy by testing if bit 1 of Status Register is set, if you don't want to abort any command being executed.

* **Write command code into Data-Command Register.**

* **Set bit 1 of Control Register.**

* Follow the data transfer procedure to send parameters.

* Check exit error.

Example:

Execute a RESET:

Assembly Language:

```
BASE equ $EF30
        ldd #2
        std BASE
```

C Language

```
int *dcr=0xEF30;
*dcr=0x0002;
```


DATA TRANSFER FROM ROSY JR. TO MULTI

* **Rosy Jr. waits for bit 0 of Status Register to be set (acknowledgment).**

* **Rosy Jr. writes Data Register.**

* **Rosy Jr. sets bit 0 of Control Register .**

Examples:

Assembly Language

```
Data    equ $EF30       Data Command Register
Cont    equ $EF31       Control Register
Stat    equ $EF31       Status Register
.....
Wait    lda Stat
        anda #01
        beq Wait        Wait for Multi's acknowledgment
        lda ,x+
        ldb #1
        std Data
.....
```

C Language

```
char *data=0xEF30,      /* Data Command Register */
     *cont=0xEF31,      /* Control Register */
     *st=0xEF31;        /* Status Register */
char byte;
.....
while (*st&0x01);       /* Wait for Multi's acknowledgment */
*data=byte ;
*cont=0x01;
.....
```

DATA TRANSFER FROM MULTI TO ROSY JR.

* **Rosy Jr. waits for bit 0 of Status Register to be set (acknowledgment).**
* **Rosy Jr. reads Data Exchange Register.**
* **Rosy Jr. sets bit 0 of Control Register.**

Examples :

Assembly language

```
Data    equ $EF30           Data Exchange Register
Cont    equ $EF31           Control Register
Stat    equ $EF31           Status Register
.....
Wait    lda Stat
        anda #1
        beq Wait
        lda Data
        sta ,x+
        ldb #1              Send acknowledgment
        stb Cont
.....
```

C Language

```
char *data=0xEF30,          /* Data Exchange Register */
     *cont=0xEF31,          /* Control Register */
     * st=0xEF31;           /* Status Register */
char byte  ;
.....
while (*st&0x01);
byte=*data;
*cont=0x01;
.....
```


HOW TO INSTALL A NEW COMMAND

* Write the command in assembly language

```
fcb $87
fcc /MODULE -/
fcb $xx         Command Code
fcc /-START/
.....

Command instructions
.....
swi
```

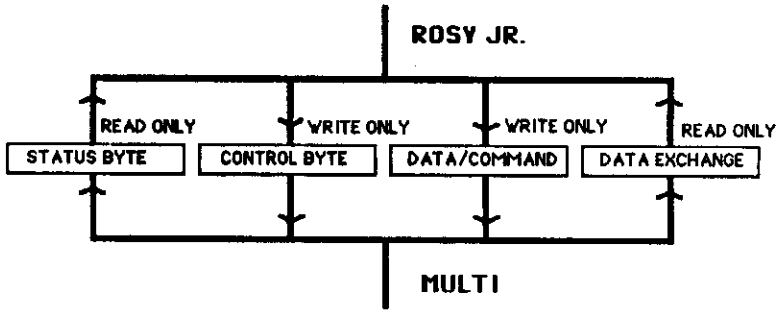
* Load the object file into Multi's memory and perform a RESET

SETTING INTERRUPTS

Rosy Jr. will be IRQ interruped if bit 2 of Control Register is set. OS-9 handles IRQ interrupts with a prioritized polling system that identifies the source of the interrupt. As the system call F$IRQ that adds or removes a device from the IRQ table is a privileged request , interrupts should be handled within a driver.

We can write a simple monitor program to write and read Multi's registers and to count interrupts.
The menu of this monitor may be as follows:

1- Display Status Register.
2- Write Control Register.
3- Write Data/Command Register.
4- Display Data Exchange Register.
5- Display number of Interrupts.



We operate Multi from our program by just reading and writing into the addresses in memory where the four registers are mapped. This implementation is strongly hardware dependent.

To make hardware features transparent and Multi easy to use we will build a special structure:

- We'll let only a driver to access Multi's registers. Multi will be then considered one more device in the operating system.
- We program C library functions to perform Multi's basic commands using the I/O system calls.
- All our application programs will call the library functions.

Example of library functions: