DataGeneral

SOFTWARE ADDENDUM

Addendum to User Device Driver Implementation in the Real Time Operating System

Application Note

086-000054-00

017 000019-00

This addendum updates manual 017-000019-00.
See UPDATING INSTRUCTIONS on reverse.

Ordering No. 086-000054
©Data General Corporation, 1977
All Rights Reserved
Printed in the United States of America
Revision 00, April 1977
Licensed Material - Property of Data General Corporation

Licensed Material - Property of Data General Corporation

NOTICE

Data General Corporation (DGC) has prepared this manual for use by DGC personnel, licensees, and customers. The information contained herein is the property of DGC and shall not be reproduced in whole or in part without DGC prior written approval.

DGC reserves the right to make changes without notice in the specifications and materials contained herein and shall not be responsible for any damages (including consequential) caused by reliance on the materials presented, including but not limited to typographical, arithmetic, or listing errors.

UPDATING INSTRUCTIONS

This addendum (086-000054-00) to the User Device Driver Implementation in the Real Time Operating System (017-000019-00) supplies information for RTOS Rev. 5.00.

To update your copy of 017-000019-00, remove and insert the following pages:

Remove	*Insert*
Title/Notice	Title/Notice
1-1/1-2	1-1/1-2
1-5/1-6	1-5/1-6
2-9/2-10	2-9/2-10
2-13 through 2-16	2-13 through 2-16
3-1 through 3-4	3-1 through 3-4
3-13 through 3-18	3-13 through 3-18

Insert this sheet immediately behind the new Title/Notice page.

A vertical bar or an asterisk in the margin of a page indicates substantive change or deletion, respectively, from 017-000019-00.

The addendum number appears on the lower inside corner of only those pages changed by this addendum.

The following are trademarks of Data General Corporation, Westboro, Massachusetts:

U.S. Registered Trademarks			Trademarks
CONTOUR I	NOVA	NOVALITE	DASHER
DATAPREP	NOVADISC	SUPERNOVA	INFOS
ECLIPSE			

User Device Driver Implementation in the Real Time Operating System

Application Note

017-000019-00

For the latest enhancements, cautions, documentation changes, and other information on this product, please see the Release Notice (085-series) supplied with the software.

Ordering No. 017-000019
©Data General Corporation, 1975
All Rights Reserved
Printed in the United States of America
Revision 00, April 1975
Licensed Material - Property of Data General Corporation

DataGeneral
SOFTWARE DOCUMENTATION

Licensed Material - Property of Data General Corporation

NOTICE

Data General Corporation (DGC) has prepared this manual for use by DGC personnel, licensees, and customers. The information contained herein is the property of DGC and shall not be reproduced in whole or in part without DGC prior written approval.

DGC reserves the right to make changes without notice in the specifications and materials contained herein and shall not be responsible for any damages (including consequential) caused by reliance on the materials presented, including but not limited to typographical, arithmetic, or listing errors.

User Device Driver Implementation in the Real Time Operating System Application Note 017-000019

This document is intended for ECLIPSE® system users. It was derived from 017-000006, which is primarily for NOVA® system users. Any references to NOVA equipment, documents, and processes within this manual also apply to ECLIPSE systems.

Revision History:

017-000006

(RTOS 3.00) Original Release - March 1974
(RTOS 4.00) First Revision - March 1975

017-000019

Original Release - April 1975
(RTOS 5.00) 086-000054-00 - April 1977

A vertical bar or an asterisk in the margin of a page indicates substantive change or deletion, respectively, from 017-000006-00.

The following are trademarks of Data General Corporation, Westboro, Massachusetts:

U.S. Registered Trademarks			Trademarks
CONTOUR I	NOVA	NOVALITE	DASHER
DATAPREP	NOVADISC	SUPERNOVA	INFOS
ECLIPSE			

Licensed Material - Property of Data General Corporation

CHAPTER 1

RTOS DRIVER IMPLEMENTATION

INTRODUCTION

In all real time computer control systems, the CPU reacts to input data from a real world environment and provides output data to correct or control the environment. The incoming data is normally the result of a process device interrupt or an input-output operation completion interrupt. In general these interrupts differ from one another only in the way in which they are serviced.

When a significant event occurs, a signal is transmitted to the computer as an interrupt requiring a special subroutine to take appropriate action. Interrupts are usually assigned in order of urgency or priority, so that if two interrupts occur at the same moment, the more important interrupt is serviced first by the computer.

How well a computer is able to respond to interrupts generally determines the maximum capability of the real time system. A significant element in the responsive ability of any real time system is the inclusion of a powerful and flexible multi-priority interrupt control program.

This document outlines the methods of adding customer-assignable multi-priority servicing routines. These device drivers can be included as part of the resident RTOS operating system or they can be implemented as part of the user application program and attached to the interrupt dispatch program.

CHARACTERISTICS OF INTERRUPTS

Interrupts can be generated by conditions internal to the computer hardware itself (power monitor option), or by events which originate in the plant or the environment that is being controlled (an external hardware interrupt like an analog-to-digital converter completion).

Non-process interrupts may be caused by an error condition being detected, an interval timer run-down, an input-output (I/O) complete interrupt, etc. The I/O interrupt is characterized by the completion of an I/O device operation such as a paper tape or disk storage transfer function, which may involve the reading or writing of one character or word in the case of program control or multiple words in the case of data channel operation. The concept of this type of interrupt is based on the importance of keeping I/O devices active, thus improving job throughput.

Process interrupts reflect process conditions which have been detected and which require an immediate change in program execution, such as may be caused by the closing of an electrical switch or contact, a rise in temperature above a prescribed limit, etc.

Licensed Material - Property of Data General Corporation

INTERRUPT PRIORITY SCHEME

There are several ways in which priorities are determined for or assigned to devices on the I/O bus. An elementary priority is established by the hardware for devices that are requesting interrupts simultaneously: among those devices waiting with an active interrupt, the one which is physically closest to the processor on the bus is at the highest priority.

The most significant method is to specify which devices can interrupt a servicing routine currently in progress. This is done by using a 16-bit priority mask. Each device is wired to a particular bit of this mask word.

By means of the mask word, the interrupt servicing program can inhibit specified levels of interrupts and thus establish any priority structure. The biggest advantage of this means of priority level control is a near optimum priority response. To guarantee minimum response time to an interrupt, the mask bit assigned to this device should not be set for long periods of time. Through the judicious use of masking, data channels can be kept functioning for the transmission of data into and out of core storage while process interrupts are prevented from occurring. The function of masking is used to delay recognition of an interrupt (the important fact is that the interrupt is not lost). See PARR.SR for current mask definitions.

The interrupt mask assignments for standard devices supplied by DGC are as follows:

Mask	Assigned Devices
3	$TTO, $TTI
4	RTC
7	$PTP
10	$LPT
13	QTY (4060 MUX)
17	$PLT, MCA (4038 multiprocessor communications adapter transmitter/receiver).
617	moving head disk
717	fixed head disk
1737	$PTR
1777	$CDR, magnetic tape, cassette

Although slower devices are assigned to higher numbered bits in the mask, there is no established priority, since the program can use any mask configuration. All devices which have their mask bits set cannot cause an interrupt and are therefore regarded by the program as being of lower priority.

Licensed Material - Property of Data General Corporation

DEVICE CONTROL TABLE (DCT) STRUCTURE

Each device defined in the RTOS system must have the address of its DCT included in the interrupt branch table (ITBL) as explained in later sections. For interrupt servicing routines defined in the user's area, only the first three entries are needed. For a thorough understanding of DCTs, a complete DCT as used in an RTOS interrupt servicing routine is described.

Word 0, DCTSV: Address of a seven-word interrupt state save area with the following structure:

Word 0, IPCC	PC and Carry
Word 1, IAC0	AC0
Word 2, IAC1	AC1
Word 3, IAC2	AC2
Word 4, IAC3	AC3
Word 5, ICMSK	Current Hardware Mask
Word 6, IRLOC	Contents of system page zero temporary

Word 1, DCTMS: Mask Word. Set a bit for every priority considered lower than the priority of this device. The devices corresponding to the priority bits that are left cleared will be permitted to interrupt the current device. A complete list of system masks is provided in INTERRUPT PRIORITY SCHEME. *

Word 2, DCTIS: Address of the device interrupt service routine.

Word 3, DCTCH: Device characteristic word. A list of device characteristics is given in the table below.

MNEMONIC	BIT	MEANING
DCCPO	1B15	Device requiring leader/trailer
DCCGN	1B14	Device requiring tab simulation
DCIDI	1B13	Buffered input device
DCCNF	1B12	Device requiring form feed simulation
DCTO	1B11	Teletype output device
DCKEY	1B10	Keyboard input device (uncontrollable)
DCNAF	1B09	Device requiring 16 nulls after form feeds
DCRAT	1B08	Device requiring rubouts after tabs
DCPCK	1B07	Device requiring even parity check on input, even parity computation on output

*

Licensed Material - Property of Data General Corporation

DEVICE CONTROL TABLE (DCT) STRUCTURE (Continued)

MNEMONIC	BIT	MEANING
DCLAC	1B06	Device requiring line feeds after carriage returns
DCPFR	1B05	Auto restart mode bit (internal to RTOS)
DCFWD	1B04	Full word device (any size greater than a byte)
DCFFO	1B03	Form feed sent on .OPEN
DCLTU	1B02	Convert lower to upper case ASCII
DCC80	1B01	80 column device (suppress line output after 80 columns)

Word 4, DCTCD: Device Code.

Word 5, DCTEX: Address of variable I/O instruction routine.

Word 6, DCTDT: Address of the device command dispatch table. One entry for every RTOS I/O function. The table order must correspond exactly to the order of the functions given below. Each entry is an address to the routine implementing the named RTOS function. If the device does not permit a command, a -1 should be entered in place of the address.

MNEMONIC	DISPLACEMENT	MEANING
OF	0	Open a file
CF	1	Close a file
RS	2	Read Sequential
RL	3	Read Line
WS	4	Write Sequential
WL	5	Write Line
RB	6	Read Block
WB	7	Write Block
OA	10	Open a file for appending

Word 7, DCTST: Address of the device start routine. The Device Start Routine specification is as follows:

Input device:	Activate the device and return
Output device:	STOUT or address within the device handler; performs the equivalent of an interrupt service.

Writing a Custom I/O Routine (Continued)

If another task is awaiting the use of this device, provision is made to start the device again and to ready the current task; control then returns either to the scheduler or to the dismissal routine as described above.

DISMISS, location 11, contains the address of an entry, D.ISM, in the interrupt dispatch module INTD. This routine is entered with the device DCT address in AC2. The original hardware mask is restored. If dismissing to another interrupt servicing routine, interrupts are re-enabled and return is made to the interrupted routine. If dismissing to a user task, a rescheduling check is made. If no rescheduling is to occur, interrupts are re-enabled and return to the task is made. Otherwise, the state of the interrupted task is saved in its TCB and the scheduler is entered.

Opening a User File

The system call .OPEN uses the file name supplied as an argument to identify the device to which subsequent I/O operations will be directed. If channel-oriented I/O calls are to be usable with a new device, there must be a way for the .OPEN routine to trace each file name to the DCT for that device, e.g., $PTR for the paper tape reader. A more complex file name construction is necessary if the device consists of multiple communication lines or circuits, channels or units, or if the device is a storage medium with compartmentalized data.

During the processing of a .OPEN call, a series of attempts is made to match the file name with entries in tables built at RTOSGEN time. Without modifying either the RTOSGEN program or the resident RTOS .OPEN logic, the user may add a routine to recognize file names for new devices. After the RTOS routine has tried and failed to recognize a file name, it will pass control to a user routine with the entry label .CKUS, provided that such an entry was defined at relocatable load time.

The .CKUS routine may then check for other forms of device specification and, if successful, locate the corresponding control blocks for opening the channel. Having opened the channel, the user returns control to entry .FOPN which finishes the .OPEN process. This includes calling the routine found in the dispatch table pointed to by displacement DCTDT of the DCT.

When control arrives at .CKUS, AC3 contains a return address which is to be used only if the file name is not recognized. The system ensures that page zero location CTCB contains the address of the caller's TCB. Displacement TXUFT of the caller's TCB contains the address of a two-word block which must be set up to open the software channel. The .CKUS routine must place the appropriate DCT address in the first word of this pair. The second word of the pair is optionally available for a

Opening a User File (Continued)

pointer to a control block for a unit, circuit, etc., within the device. This word pair is called a user file pointer table (UFPT) entry. On completion, if the file name was recognized, the .CKUS routine must branch to entry .FOPN in RTOS with the DCT address in AC3. If the file name was not recognized, control must be returned to the address input in AC3 upon entry to .CKUS. This will cause rejection of the file name and an error return for the caller (error code ERDLE: non-existent file).

I/O SERVICE MODULE (IOSER)

The IOSER module is used by system drivers generally for byte-oriented devices. One use of IOSER is to execute certain types of I/O instructions; this permits the writing of reentrant drivers used by multiple devices (e.g., a multiple TTY system requiring only one TTY driver). Another use of IOSER is to provide common input/output character device interrupt processing. IOSER also provides routines to place a bead at the end of a bead string, and to priority enqueue a bead at the head of the queue.

The following list summarizes the entries in IOSER:

Entry	Use
.COSE	Common output interrupt service.
.STOU	Start an output device.
.ENQB	Add a bead to the end of a string.
.PENQ	Priority enqueue a bead at the head of a string.
.FINP	Common input interrupt service.
..IAC	DIAC instruction processor.
.DIAS	DIAS instruction processor.
.DOAS	DOAS instruction processor.
.NIOC	NIOC instruction processor.
.NIOS	NIOS instruction processor.
.SKPB	SKPBZ instruction processor.

Calls to any of the instruction processor entries cause that instruction to be built (for the specific device in question), stored in the driver's "execute I/O instruction" area (pointed to by DCTEX of the device's DCT), and executed. Control is then returned to the caller. Calls to the instruction processors are issued solely by the system; specific call types depend upon the entries selected in the device dispatch table or in displacement DCTST of the Device Control Table.

Licensed Material - Property of Data General Corporation

Declaring the Device Name and DCT Address

The names of user devices and the addresses of their DCTs are declared when the RTOS module, output by RTOSGEN, is created. Declaration of the DCT names of user devices is accomplished by responding in the affirmative to question 19 of RTOSGEN: "USER SUPPLIED DRIVERS?". By striking "1" on the console keyboard, you trigger the further pair of questions:

```
DEVICE CODE?
NAME?
```

Respond with the octal device code for the first user device, and follow this with a carriage return. Then input a 3 alphanumeric character name for the device, and follow this with a carriage return. The pair of questions is repeated for more devices. Respond with a simple carriage return to the device code query to terminate this interrogation. As described in Appendix B of the RTOS Reference Manual, 093-000056, RTOS then outputs a complete list of system and user device codes, DCT names (which are simply the device names with a DC suffix), and device names. RTOS will place table entries for all user devices into .ITBL; each table entry will contain the address of that device's DCT.

Corresponding to the RTOSGEN declaration of user device names, each user device driver must have an entry with the same DCT name as that specified to RTOSGEN. That is, if the device name was specified to be ABC, the driver for this device must contain an entry to the device's DCT: .ENT ABCDC.

High Priority Interrupt Devices

High priority interrupt devices are devices which will receive interrupt service control before standard devices. System high priority devices are the real time clock and the power fail/auto restart option. Question 18 of RTOSGEN permits special user devices to also receive high priority interrupt service.

In essence, the operation of the high priority interrupt dispatcher, .HINT, is as follows. Each high priority interrupt device is examined to see if it generated the interrupt. The power fail monitor is tested first, then the real time clock, and then each of the other user devices specified at RTOSGEN time in the order that they were specified during system generation. If the source of the interrupt is found, control is dispatched to its interrupt service routine; otherwise, control is given to the ordinary interrupt dispatcher.

Licensed Material - Property of Data General Corporation

High Priority Interrupt Devices (Continued)

The name given in response to question 18 for each user high priority device with the suffix IS added becomes the name of the entry point to the service routine for each such device. Thus if the name given for one high priority user device was GHK, the entry point to this device's interrupt service routine would be GHKIS, and this address would have been entered in the service routine: .ENT GHKIS. For information describing the writing of high priority user interrupt service routines, see Chapter 3, HIGH PRIORITY INTERRUPT SERVICE.

Loading a User Driver

After specifying all user drivers at RTOSGEN time and assembling these drivers, they may be loaded with user relocatable binaries and the RTOS libraries. Depending upon the loader used, one of several relocatable load procedures can be followed. These procedures are described in Appendix B of the RTOS Reference Manual. In essence, RTOS.RB, user driver relocatable binaries, and the user program binaries must be loaded before the RTOS libraries. Consult the RTOS Reference Manual for more details.

RTOS System Library List

The following lists name and describe each of the library modules contained in the RTOS libraries for revision 5.00 of RTOS, run on both the NOVA and ECLIPSE computers.

RTOS1.LB

Relocatable Binary Title	Function(s)
BFPKG	Core Buffer I/O package (see 017-000003).
TXMT	.XMT, .REC, .XMTW, .IXMT logic.
TACALL	.AKILL, .SUSP, .ASUSP, .ARDY logic.
ABORT	.ABORT logic.
TIDC	.IDST, .TIDS, .TIDR, .TIDK, .TIDP logic.
TMIN	System call set up logic for single task programs. Single task environment task scheduler.
TCBMON	System call set up logic for multitask programs. Multitask environment task scheduler.
TUMOD	.SMSK, .UCEX, .UPEX, .UIEX logic.
TQTAS	.QTSK, .DQTSK logic.
TIDSR	Additional support for .QTSK/.DQTSK.
DEBUG	Multitask debugger.

Licensed Material - Property of Data General Corporation

RTOS System Library List (Continued)

RTOS2.LB

Relocatable Binary Title	Function(s)
SYSTEM	System call processor and error handler. End of I/O processor.
SCHED	System scheduler logic.
INTD	Interrupt processor. Interrupt dispatching and dismissal.
RTIN	Initializes system device DCTs. Resets system state variables. Initializes TCB pool and chains. Branches to system scheduler. *
TTY1D	Second Teletype driver.
TTY2D	Third Teletype driver.
TTYDR	Teletype driver.
PTR1D	Second paper tape reader driver.
PTRDR	Paper tape reader driver.
PTP1D	Second paper tape punch driver.
PTPDR	Paper tape punch driver.
RTCDR	Time of day clock calls. Get RTC frequency. Set up user clock. Remove user clock. Clock delay system call.
CDR1D	Second card reader driver.
CDRDR	Card reader driver.
LPT1D	Second line printer driver.
LPTDR	Line printer driver.
PLT1D	Second plotter driver.

Licensed Material - Property of Data General Corporation

RTOS System Library List (Continued)

Relocatable Binary Title	Function(s)
PLTDR	Plotter driver.
QTYDR	4060 data communications multiplexor.
MCADR	4038 multiprocessor communications adapter.
PWRDR	Power monitor/automatic restart.
GENIO	Generalized routines corresponding to system I/O calls (except .RDB/.WRB).
IOSER	I/O service routines, mostly interrupt level.
CNVRT	Hollerith card code to 7-bit ASCII converter.
DSKDR	Fixed head disk drive for controller 0.
DKPDR	Moving head disk drive for controller 0.
MTU0 through MTU7	Unit control tables (UCTs) and buffers for line and sequential I/O on magnetic tape units 0 through 7.
MTBUF	Read/write line/sequential I/O instruction support for magnetic tape and cassette units.
MTADR	Seven or nine-track magnetic tape device driver for controller 0.
MTSER	Magnetic tape/cassette interrupt service module.
SMTU	Truncated magnetic tape unit control tables for units MT0 through MT7 employing non-buffered tape I/O.
CTU0 through CTU7	Unit control tables (UCTs) and buffers for line and sequential I/O on cassette units 0 through 7.
CASDR	Cassette device driver for controller 0.

Licensed Material - Property of Data General Corporation

CHAPTER 3

USER PROGRAM SERVICED INTERRUPTS

SERVICING USER INTERRUPTS

Special user devices may be identified either at the time an RTOS system is generated (RTOSGEN time) or at run time. This chapter describes the procedure for identifying a user device at run time and for creating a user clock driven by the system real time clock. Considerations applying to high priority interrupt routines are given at the end of this chapter. The considerations given for identifying a user device are common to both single and multitask environments; the user clock facility may also be used in both task environments.

Upon detection of an interrupt request, the system will be dispatched through the device interrupt vector table, .ITBL (see Chapter 1). In this table are pointers to Device Control Tables (DCTs) for devices established at RTOSGEN time, whether system or user devices. Chapter 1 describes the structure of system DCTs.

User Device Driver Implementation at Run Time

In order to identify a user device to the system at run time, the user must provide a three-word DCT as an interface between the system interrupt dispatch routine and the user-interrupt servicing routine. The structure and mnemonic assignments of this three-word table are as follows:

Displacement	Mnemonic	Purpose
0	DCTSV	Pointer to a 7-word state save area.
1	DCTMS	Interrupt service mask.
2	DCTIS	Interrupt service routine address.

DCTSV is a pointer to a seven-word state variable save area used by the system to save the PC, accumulators, carry, etc. of the interrupted task. DCTIS is a pointer to the routine which services this particular device interrupt. DCTMS is the interrupt mask* that the user wants to be ORed with the current interrupt mask while in the user interrupt service routine. This mask establishes which devices--if any--will be able to interrupt the currently interrupting device.

* See "How to Use the NOVA Computers", Section 2.4.

Licensed Material - Property of Data General Corporation

User Device Driver Implementation at Run Time (Continued)

Upon transferring control to the user interrupt service routine, the system will ensure that AC3 contains the return address required for exit from the routine, and that AC2 contains the address of the DCT upon exit from the routine. In revision 05 of RTOS, exit was accomplished by a jump to the return address specified by AC3 upon entry. In subsequent releases of RTOS, task call .UIEX is issued. Rescheduling will occur immediately after the completion of interrupt dismissal, if AC1 is nonzero.

All multitask environment activity ceases at the moment that a user device interrupt is detected. Nonetheless, it is possible for a user to communicate a message to a task from a service routine. If the task in question has been expecting such a message through issuance of a .REC and is now in the suspended state, issuance of the message via .IXMT will cause that task to be readied even though multitask activity is in abeyance. If no task has issued a .REC for such a message, .IXMT simply posts the message and takes no further action. For more information on communicating to tasks from a user interrupt service routine, see Chapter 3 of the RTOS Reference Manual, 093-000056.

Identifying User Interrupt Devices (.IDEF)

In order to introduce to the system those devices (not identified at RTOSGEN time) whose interrupts the system is to recognize, the system call .IDEF must be issued. This call places an entry in the device interrupt vector table, .ITBL. Required inputs to this call are the device code of the user device and the starting address of this device's DCT. The format of this call is:

```
AC0     -     Device code of the user device.
AC1     -     Starting address of the user device's DCT.

.SYSTM
.IDEF
error return
normal return
```

Possible error messages are:

AC2	Mnemonic	Meaning
36	ERDNM	Illegal device code (>77_8). Device code 77_8 is reserved for the power monitor/auto restart option.
45	ERIBS	Interrupt device code in use.

Exit from a User Interrupt Routine (.UIEX)

Upon entering a user interrupt routine, AC3 will contain the address required for exit from the routine. In all versions of RTOS, exit may be accomplished by a jump to the return address specified by AC3 upon entry to the user routine. In revision 3.00 and later revisions, task call .UIEX can be issued to return from a user interrupt routine.

The format of this call is:

```
AC1     -     Reschedule flag (0=Do not reschedule).

AC3     -     Return address.

.UIEX
```

Control returns to the location which was interrupted by the user device. No error return need be reserved. .UIEX can be issued in a single task environment.

Remove User Interrupt Servicing Program (.IRMV)

To prevent the system's recognition of user interrupts which have been previously identified by the .IDEF command, the .IRMV command is issued. Required input to this call is the user device code corresponding to the device control table which is to be removed.

The format of this call is:

```
AC0     -     Device code.

.SYSTM
.IRMV
error return
normal return
```

One possible error message may be given.

AC2	Mnemonic	Meaning
36	ERDNM	Illegal device code (>77_8) or attempt to remove a system device (i.e., one established at RTOSGEN time).

Modifying the Current Interrupt Mask (.SMSK)

RTOS contains a task call which permits the current interrupt mask to be modified. When a user interrupt occurs, the interrupt service mask contained in DCTMS of the user device DCT is ORed with the current hardware mask. Nonetheless, it is possible to modify the current mask further by issuing task call .SMSK. The format of this call is as follows:

AC1	-	New hardware mask to be ORed with the mask in effect before the current interrupt.
AC2	-	DCT address

```
.SMSK
normal return
```

There is no error return possible from this call. This call may be issued in a single task environment.

WRITING USER POWER FAIL SERVICE

RTOS provides software support for the power fail/automatic restart option. Upon detection of a power loss, the system transfers control to a power fail routine which saves the status of accumulators 0 through 3, the PC and Carry.

When power is restored, if the console key is in the LOCK position, the message

```
**POWER FAIL**
```

is output on the system console and the state variables are restored before control resumes operation at the point where it was interrupted. If the console key was in the ON position when input power failed, the user must set the console switches to all zeroes (down) and START must be pressed when power is restored. This causes the console message to be output and state variables to be restored as when the key is in the LOCK position.

The following system devices are given power-up restart service by RTOS:

paper tape readers/punches
Teletypes
quad multiplexors
card readers
line printers
disks

Licensed Material - Property of Data General Corporation

```
!0004 AIDEV
01
02
03                ; ANALOG TO DIGITAL CONVERTER INTERRUPT ROUTINE
04                ; ------------------------------------------
05
06                ; INPUT:
07                ;       AC2=DCT ADDRESS
08                ;       AC3=INTERRUPT DISMISSAL ROUTINE ADDRESS
09
10 00063'062621 AINTS:  DICC 0 ADCV      ; READ VALUE--CLEAR CONVERTER DONE
11 00064'042723          STA 0 @DVPTR     ; SAVE NEWLY READ DATA VALUE
12 00065'014723          DSZ DTCNT        ; DECREMENT DATA COUNTER
13 00066'000405          JMP AIMORE       ; MORE TO COME
14 00067'102400          SUB 0 0          ; REQUEST COMPLETE
15 00070'042715          STA 0 @ACTIV     ; SET USER DONE FLAG
16 00071'040714          STA 0 ACTIV      ; SET CONVERTER NON-BUSY
17                       ADC 1,1          ; RESCHEDULE
18 00072'177777          .UIEX            ; DISMISS THE INTERRUPT
19
20 00073'010713 AIMORE: ISZ DAPTR        ; INCREMENT ADDRESS POINTER
21 00074'010713          ISZ DVPTR        ;     "     DATA TABLE POINTER
22 00075'022711          LDA 0 @DAPTR     ; SETUP NEXT CONVERSION
23 00076'061121          DOAS 0 ADCV      ; START IT
24                       SUB 1,1          ; NO RESCHEDULE
25 00077'000072'         .UIEX            ; DISMISS THE INTERRUPT
26
27                       .END
```

```
  0005 AIDEV

ACTIV 000005'            1/26    2/12    3/15    3/18    4/15    4/16
ADDCT 000000'            1/16    2/14
AICHK 000057' EN         1/08    3/44
AICLR 000032' EN         1/08    2/29
AIDEF 000021' EN         1/08    2/10
AIMOR 000073'            4/13    4/19
AINTS 000063'            1/20    4/10
AIROR 000040' EN         1/08    3/15
AISAV 000011'            1/18    1/31
DAPTR 000006             1/27    3/22    3/27    4/19    4/21
DCODE 000004             1/25    2/13    2/30
DTCNT 000010'            1/29    3/20    4/12
DVPTR 000007'            1/28    3/24    4/11    4/20
USDCT 000001'            1/16    1/18
.UIEX 000077' XN         1/09    4/17    4/23
```

Licensed Material - Property of Data General Corporation

External Interrupt Recognition

The driver program given at the end of this section illustrates an interrupt servicing routine which readies a user task as a result of an external interrupt. This program illustrates a type 4066 digital interface device driver.

Page 1 of the listing is almost identical in form to the first page of the A/D driver listing discussed earlier. The only difference is in the variables and storage required by the digital interface. Here, the user must supply the address of a communications core location. When a call is made to attach the interrupt routine address to the RTOS dispatch table (page 2 of the listing), AC0 must contain the communications core location address. After attaching the interrupt to the RTOS dispatch vector table, the initialization routine arms the digital interface so that it can cause an interrupt.

When the external interrupt occurs, control is dispatched to the digital interface servicing routine, starting on line 38 of page 2. After the service routine gains control, it reads a sixteen-bit digital register provided by the interface. If the register's contents is non-zero, the contents are transmitted to a user task using the task call .IXMT; after this, the interrupt is dismissed. If the register's contents is zero, the interrupt is simply dismissed. Upon dismissal of the interrupt, the system re-arms the digital interface interrupts.

A practical example of the use of such a scheme would be servicing of an operator's console. Such a console would generate an external interrupt indicating that the operator desires some form of action by the computer; the sixteen-bit register contents (selected by the console operator) would indicate exactly the type of action desired.

Licensed Material - Property of Data General Corporation

```
 0001 DIDEV MACRO REV 02.6.21            16:27:35 08/03/73
01
02
03                ; DIGITAL INTERFACE (TYPE 4066) DEVICE DRIVER
04                ; ------------------------------------------
05                ; ------------------------------------------
06
                          .TITL DIDEV
08
09                        .ENT DIDEF,DICLR
10
11                        .EXTN .IXMT .UIEX
12
13                        .NREL
14
15        000042 .DUSR    DIO=42            ; DEVICE CODE DEFINITION
16
17                ; DEVICE CONTROL TABLE LAYOUT
18                ; ---------------------------
19
20 00000'000001'DIDCT:    USDCT             ; ADDRESS OF DIO DCT
21
22 00001'000004'USDCT:    DISAVE            ; INTERRUPT STATE SAVE AREA
23 00002'000400           1B7               ; INTERRUPT MASK
24 00003'000036'          DINTS             ; INTERRUPT ROUTINE ADDRESS
25
26                ; ADDITIONAL VARIABLES AND STORAGE USED BY HANDLER
27                ; ------------------------------------------------
28
29 00004'000010 DISAVE:   .BLK 10           ; STATE SAVE AREA
30 00014'000042 DCODE:    DIO               ; DEVICE CODE
31 00015'000000 DICOM:    0                 ; COMMUNICATIONS ADDRESS
```

Licensed Material - Property of Data General Corporation

```
10002 DIDEV
01
02                ; INITIALIZE THE DIGITAL INTERFACE
03                ; --------------------------------
04
05                ; INPUT: AC0=MESSAGE ADDRESS
06
07                ; CALLING SEQUENCE:
08                ;       JSP DIDEF
09                ;       <ERROR RETURN>  ; DEVICE CODE (DTO) ALREADY IN USE
10                ;       <NORMAL RETURN>
11
12 00016'054016 DIDEF:  STA 3 USP
13 00017'040776         STA 0 DICOM     ; SAVE COMMUNICATIONS ADDRESS
14 00020'020774         LDA 0 DCODE
15 00021'024757         LDA 1 DIDCT
16 00022'006017         .SYSTM          ; ATTACH TO RTOS INTERRUPT SYSTEM
17 00023'021007         .IDEF
18 00024'001400         JMP 0 3         ; ERROR RETURN
19 00025'060142         NIOS DIO        ; ARM THE EXTERNAL INTERRUPT
20 00026'001401         JMP 1 3         ; NORMAL RETURN
21
22                ; DETACH THE INTERRUPT SERVICING ROUTINE & CLEAR DEVICE
23                ; ----------------------------------------------------
24
25                ; CALLING SEQUENCE:
26                ;       JSR DICLR
27                ;       <NORMAL RETURN>
28
29 00027'054016 DICLR:  STA 3 USP
30 00030'060242         NIOC DIO        ; CLEAR DIO DEVICE
31 00031'020763         LDA 0 DCODE
32 00032'006017         .SYSTM          ; DETACH HANDLER
33 00033'021010         .IRMV
34 00034'001400         JMP 0 3         ; NORMAL RETURN
35 00035'001400         JMP 0 3         ;    "      "
36
37
38                ; DIGITAL INTERFACE EXTERNAL INTERRUPT HANDLER
39                ; --------------------------------------------
40
41                ; INPUT: AC2=DCT ADDRESS
42                ;        AC3=INTERRUPT DISMISSAL ADDRESS
43
44 00036'064542 DINTS:  DIAS 1 DIO      ; READ 16 BIT REGISTER
45 00037'125005         MOV 1 1 SNR     ; VALUE ZERO ?
46 00040'001400         .UIEX           ; YES--DISMISS INTERRUPT
47
48 00041'054410         STA 3 DISMISS   ; NO--SAVE DISMISSAL ROUTINE ADDRESS
49 00042'102400         SUR 0 0         ; CLEAR SIGNAL ADDRESS
50 00043'042752         STA 0 @DICOM
51 00044'020751         LDA 0 DICOM     ; GET SIGNAL ADDRESS
52 00045'177777         .IXMT           ; WAKE UP USER TASK
53 00046'000401         JMP .+1
54 00047'034402         LDA 3 DISMISS   ; RESTORE AC3 FOR EXIT
55                      ADC 1,1         ; FORCE RESCHEDULE
56 00050'177777         .UTEX           ; DISMISS THE INTERRUPT
57
58 00051'000000 DISMISS: 0              ; INTERRUPT DISMISSAL ADDRESS
59
60                      .END
```

Licensed Material - Property of Data General Corporation

```
  0003 DIDEV

DCODE 000014'          1/30      2/14      2/31
DICLR 000027' EN       1/09      2/29
DICOM 000015'          1/31      2/13      2/50      2/51
DIDCT 000000'          1/20      2/15
DIDEF 000016' EN       1/09      2/12
DINTS 000036'          1/24      2/44
DISAV 000004'          1/22      1/29
DISMI 000051'          2/48      2/54      2/57
USDCT 000001'          1/20      1/22
.IXMT 000045' XN       1/11      2/52
.UIEX 000050' XN       1/11      2/55
```

HIGH PRIORITY INTERRUPT SERVICE

As described in the RTOS Reference Manual, special high priority interrupt devices may be incorporated into an RTOS system at RTOSGEN time. The real time clock and power fail/auto restart device are two such high priority interrupt devices; users may also specify custom high priority interrupt service routines.

All high priority devices have entries in a high priority interrupt dispatch table, .HINT. Entries in this table are scanned whenever an interrupt occurs, and only if no high priority device has caused an interrupt will control branch through the normal interrupt table, .ITBL. All other system devices, and user devices announced at run time (via system call .IDEF), have entries in .ITBL.

Interrupts are disabled whenever high priority interrupt service is being performed. Users writing high priority interrupt service routines must also conform to several programming conventions. In general, these conventions are as follows:

1) Issue no task or system calls.
2) Save and restore accumulators and Carry.
3) Save the contents of location 0, and place a HALT instruction in location 0 (optional).

The state of Carry and the contents of accumulators AC0 through AC2 must be saved within the routine, and these state variables must be restored before leaving the routine. AC3 is saved by the system in location I.AC3 (a location within module INTD), and AC3 too must be restored before exit is accomplished even if the routine didn't use AC3. The contents of location 0 will contain the return address

Licensed Material - Property of Data General Corporation

needed for exit; this address should be stored in a user-provided location, e.g., RET, and a HALT instruction should be stored in location 0. This practice is adhered to by RTOS to capture program control in the event of system failure.

The final two instructions which must be executed when leaving a high priority interrupt service routine are to enable interrupts and then to perform an indirect JMP to the location containing the original return PC, e.g., RET. Control will then pass to the next instruction which would have been executed had no high priority interrupt occurred. The following instruction sequence accomplishes these operations.

```
                .EXTN I.AC3
                .
                .                       ;RESTORE AC0, AC1, AC2, CARRY
                .
                LDA 3 @ SAC3
                INTEN
                JMP @ RET
SAC3:           I.AC3
RET:            .BLK 1
                .END
```

*****