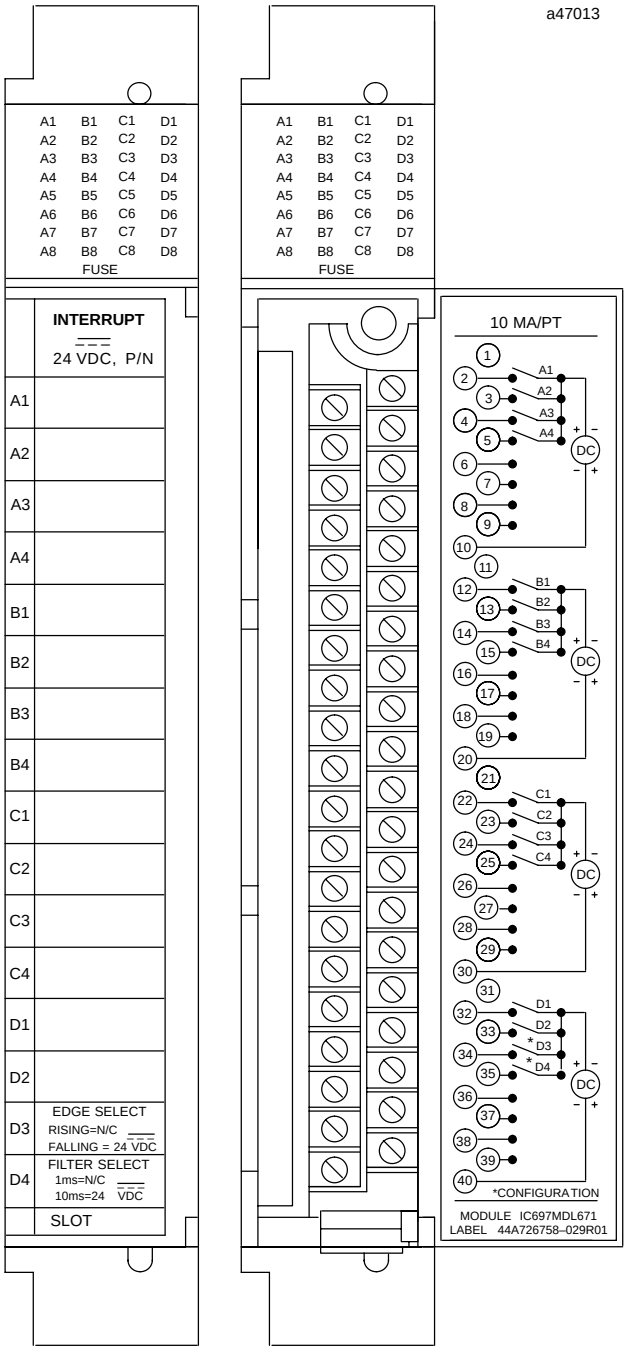
GFK-0880C
August 1997

Interrupt Module, 14 Point, 24 Volt DC Positive/Negative Logic

Features

- 16 points total - 4 isolated groups of 4 points each
- 14 interrupt points; 2 configuration points
- Positive / Negativelogiccompatibility
- Proximity switch compatible
- Input filter selectable - 1 ms or 10 ms
- All 14 interrupts can run at maximum I / O interrupt rate (See Table 1 for module specifications).
- First input configured as interrupt



Functions

The **24 Volt DC Positive/Negative Logic Interrupt Input Module** provides 16 points total in four isolated groups of four points each. The first 14 points (A1 through D2) are interrupt inputs and the last 2, D3 and D4 are configuration inputs. The input current-voltage characteristics meet IEC standard (type 1) specifications.

LED indicators which give the ON-OFF status of each point are located at the top of the module.

The module is mechanically keyed to ensure correct replacement with a similar type in the field. I / O references are user configurable without the use of jumpers or DIP switches on the module.

Configuration is done using the configuration function of the MS-DOS® or Windows® programming software running on Windows® 95 or Windows NT® over Ethernet TCP / IP or through the SNP port. The Programming Software configuration function is installed on the programming device. The programming device can be an IBM® XT, AT, PS / 2® or compatible Personal Computer.

® IBM and PS / 2 are registered trademarks of International Business Machines Corporation.
® MS-DOS, Windows, Windows 95, and Windows NT are registered trademarks of Microsoft Corporation.

GFK-0880C
August 1997

Interrupt Module, 14 Point, 24 Volt DC Positive/Negative Logic

Operation - 24 Volt DC Positive/Negative Logic Interrupt Module

Operation of the Interrupt module is described following the illustration of a block diagram of the module.

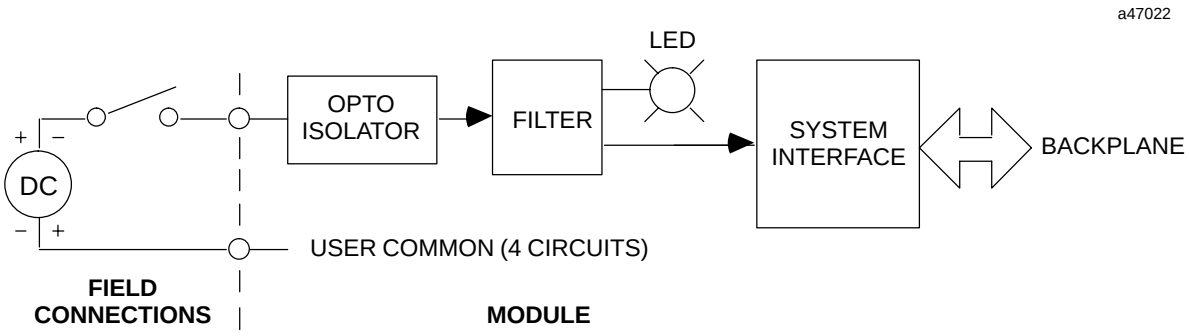


Figure 1. Block Diagram for IC697MDL671

This interrupt module will generate an interrupt to the PLC, allowing the execution of an interrupt block, when any or all of its 14 interrupt inputs satisfy its configured edge and filter selection. All interrupts are reported as one word at the module's selected reference address. The PLC reads the interrupt word and acknowledges the interrupt. The module is then ready to generate the next PLC interrupt.

Note

If you are using CPU software earlier than Release 6.00 this module requires a ladder diagram to perform the above functions The ladder diagram is provided as **Attachment A** at the end of this data sheet.

Interrupts are edge triggered and require a minimum pulse width (as configured) for the trigger to occur. Interrupts on a single input may not occur at a frequency greater than 500 Hz with a model CPM915 CPU (frequency depends on model of CPU, see Table 1 for more information). Within an interrupt module, interrupts will be reported as they are received which means that all interrupts have equal priority. Multiple interrupts may be reported simultaneously.

Software Configuration Requirements

Release 4.02 or later of IC641 programming software is required for configuration of the module. When the module is selected via the Configuration function, the proper configuration is automatically set. The only available configuration option is to select the %I reference address.

Module Configuration

The configuration inputs D3 and D4 configure all 14 interrupt inputs at power-up. The module will default to *rising edge trigger* and *1 ms interrupt filter* with no configuration input connection. Configuration inputs D3 and D4 are read only at system power-up. Activating input D3 will reconfigure the module (after a power cycle) for falling edge detection. Activating input D4 will reconfigure the module (after a power cycle) for a 10 ms interrupt filter. The module configuration cannot change after power-up.

I/O Triggered Interrupt Block

Each Interrupt module can call one LD (Ladder Diagram) program block triggered by the first input address configured for the module (similar to other IC697 Input modules). This means that an Interrupt module configured at %I0001 must be programmed to execute an LD program block triggered by %I0001. To do this, the program block must first be included in the program block declarations and then entered in the interrupt declaration editor.

Event Triggered Programs

Each interrupt module can cause an event-triggered program to be scheduled for execution. The trigger must be the first %I reference configured for this module.

Interrupt Module, 14 Point, 24 Volt DC Positive/Negative Logic

GFK-0880C
August 1997

Interrupt Word Default

At power-up, configuration dependent default data is present at the %I reference for the Interrupt module regardless of the level of each interrupt input. The default data for rising edge selection is all zeros; the default data for falling edge selection is all ones. The level of the configuration bits detected at power-up will also be present in the default word.

Interrupt Reporting

An interrupt received on any of the 14 interrupt inputs will cause an interrupt to be sent to the CPU. The CPU will read the interrupt module and capture the interrupt conditions at the time the CPU interrupt was generated. This data will be written to the %I reference that has been configured for the module. Current interrupt data is valid within the interrupt block only. Each interrupt input corresponds to a bit in the interrupt word. The interrupt word also contains the module configuration read at power-up.

Modules configured for positive edge trigger will report a 1 (one) for active interrupts while negative edge interrupts will report a 0 (zero) for active interrupts. The interrupt handling logic must reference the %I data to determine which subset of the interrupt points are active (see *Interrupt Handling Logic* below).

Interrupt Word

Two examples of interrupt reports are shown below.

RISING EDGE

D4	D3	D2	D1	C4	C3	C2	C1	B4	B3	B2	B1	A4	A3	A2	A1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

D4 = 10 ms Filter Select
D3 = Rising Edge Select
A2 = Positive Interrupt

FALLING EDGE

D4	D3	D2	D1	C4	C3	C2	C1	B4	B3	B2	B1	A4	A3	A2	A1
0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1

D4 = 1 ms Filter Select
D3 = Falling Edge Select
B4 = Negative Interrupt

Interrupt Handling Logic

Any user application logic you have developed for an interrupt module should be contained in the interrupt handling logic associated with that module. Logic outside of the interrupt handling logic will have no meaningful data to act upon. *The interrupt bits and associated transitions are undefined outside of the interrupt handling logic.*

If the interrupt handling logic is an interrupt block within the RLD program, then the global %I contacts and / or %I transition contacts can be referenced to determine which interrupt points are active.

If the interrupt handling logic is an event-triggered standalone C program, then the %I word should be included in the program's input specification list. This data can then be referenced using the input specification macros to determine which interrupt points are active. Refer to the *C Programmer's Toolkit Reference Manual* for more information.

Note

Referencing the global %I data directly from within a standalone C program is not recommended since the global data may not reflect the state of the %I bits at the time at which the interrupt occurred.

If the interrupt handling logic is an event-triggered RLD program, then the global %I data can be referenced to determine which interrupt points are active. However, this is not recommended since the global %I data may not reflect the state of the %I bits at the time that the interrupt occurred.

Refer to the *IC697 Programmable Controller Reference Manual* for more information on interrupt handling logic.

Interrupt Buffer

Interrupts received while the CPU is processing a previous interrupt will be stored in a buffer. When the CPU acknowledges the previous interrupt, the module will immediately generate a new CPU interrupt to report all of the buffered interrupts. For buffered interrupts, data is not available about the number of interrupts received on a single input. Information about the order in which interrupts from different inputs occurred is also not available for buffered interrupts.

Important Module Information

- Interrupt modules cannot be used in a Remote I/O Scanner rack since a remote drop *cannot* have any I/O module interrupts.
- If a DOIO instruction is used with an I/O interrupt, transition contacts associated with scanned inputs may not operate as expected.
- When using the Mask/Unmask I/O Interrupt Service Request #17 (SVCREQ #17), the mask applies to all interrupts on the module. When entering the address of the interrupt to mask, use the first %I reference of the module.

Caution

Caution should be exercised since an I/O interrupt can interrupt the execution of a function block in the main block or any program block. This means that unexpected results may occur if the interrupt block and a program block access the same data.

Input Characteristics

This input module is designed to have both positive and negative logic characteristics in that it sinks or sources current from the input device to the user common. The input device is connected between the power bus and the module input as shown in Figure 1.

This module is compatible with a wide variety of input devices, such as:

- Pushbuttons, limit switches, selector switches;
- Electronic proximity switches, both 2-wire and 3-wire.

In addition, inputs on this module may be directly driven by any IC697 PLC voltage compatible output module.

The input circuitry provides sufficient current to ensure reliable operation of the switching device. Input current is typically 10mA in the ON state, and can accept up to 2 mA leakage current in the OFF state without turning on.

3-wire proximity switches are easily applied, since they provide low voltage drop in the ON state and low leakage current in the OFF State.

2-wire proximity switches derive their power from the signal connections; thus both the ON state voltage and the OFF state leakage current are higher than for 3-wire devices. This module is designed to be compatible with many such 2-wire devices; however each device type must be carefully evaluated for compatibility in both the ON and OFF states.

To determine compatibility with a specific proximity switch, find the ON state characteristics of the switch in the diagram shown in Figure 2. If that point falls to the left of the input load line, the ON state characteristics are compatible. As an example, the ON state requirements of a compatible proximity switch of 3mA at 5 volts drop are shown in Figure 2.

OFF state compatibility is assured if the proximity switch leakage is less than 2mA with a module input voltage of 5 volts or less.

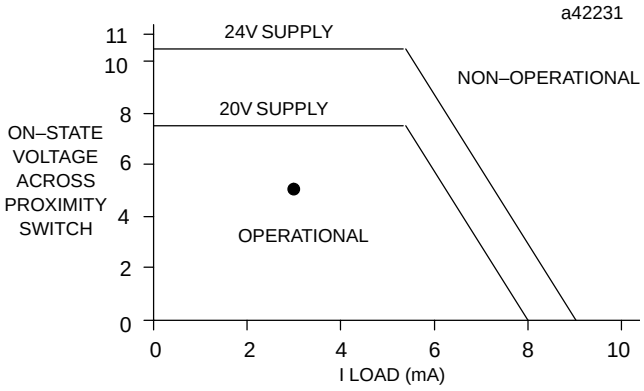


Figure 2. Proximity Switch Compatibility

Interrupt Module, 14 Point, 24 Volt DC Positive/Negative Logic

GFK-0880C
August 1997

Module Mechanical Keying

This module includes a mechanical key that prevents inadvertent substitution of one module type for another in a given slot. The key fits a uniquely shaped area on the board below the connector. Each module has a key packaged with it.

When the module is first installed, the key latches onto the backplane center rail. When the module is extracted, the key remains in the center rail, configuring the slot to accept only identical module types.

If it is necessary to change the module location in the rack after the key has been latched onto the center rail of the rack, the key can be removed by pushing it upward to unhook the latch while pulling it off the rail. It may then be reinserted onto the module and the module inserted into the rack in the desired location.

Note that in an IC697 PLC rack only the power supply can be placed in the leftmost rack position, and slot 1 (adjacent to the power supply) must always contain a CPU (in rack 0 - the CPU rack), or a Bus Receiver Module (in an expansion rack).

Field Wiring

The module is wired as shown in Figure 3. Since each group of four inputs is isolated from the others, a wire from the power source to the power input terminal (10, 20, 30, or 40) for each group is required (power input terminals for each group are not connected inside the module). Each group can be powered to operate as either positive logic or negative logic.

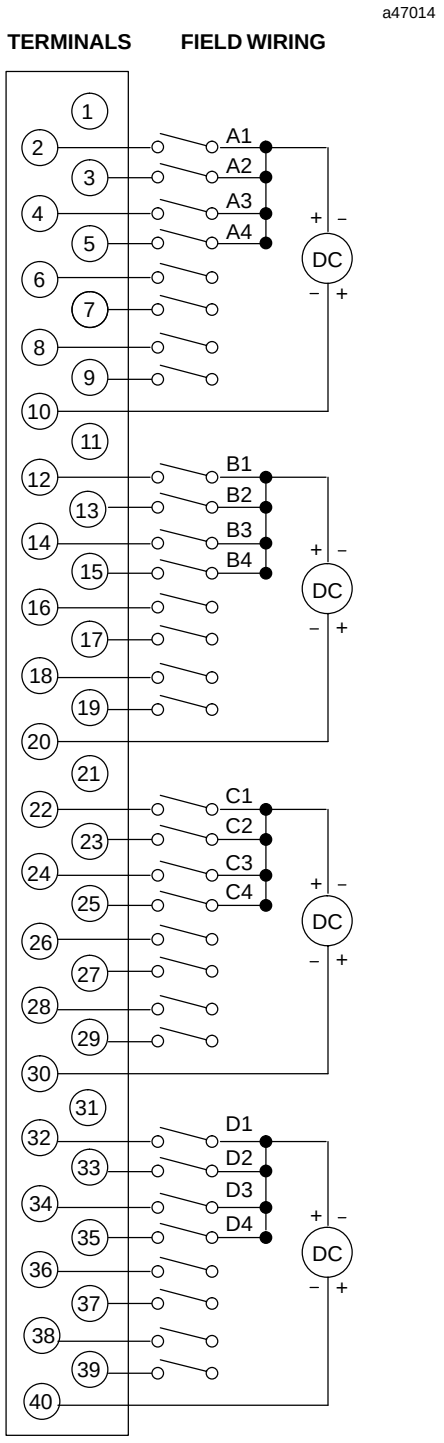


Figure 3. Field Wiring Connections for IC697MDL671

Recommended Field Wiring Procedures

The following procedures are recommended when connecting field wiring to the detachable terminal board on this input module. Module features referenced in the following procedures which are common to all IC697 I/O modules are illustrated in the following figure.

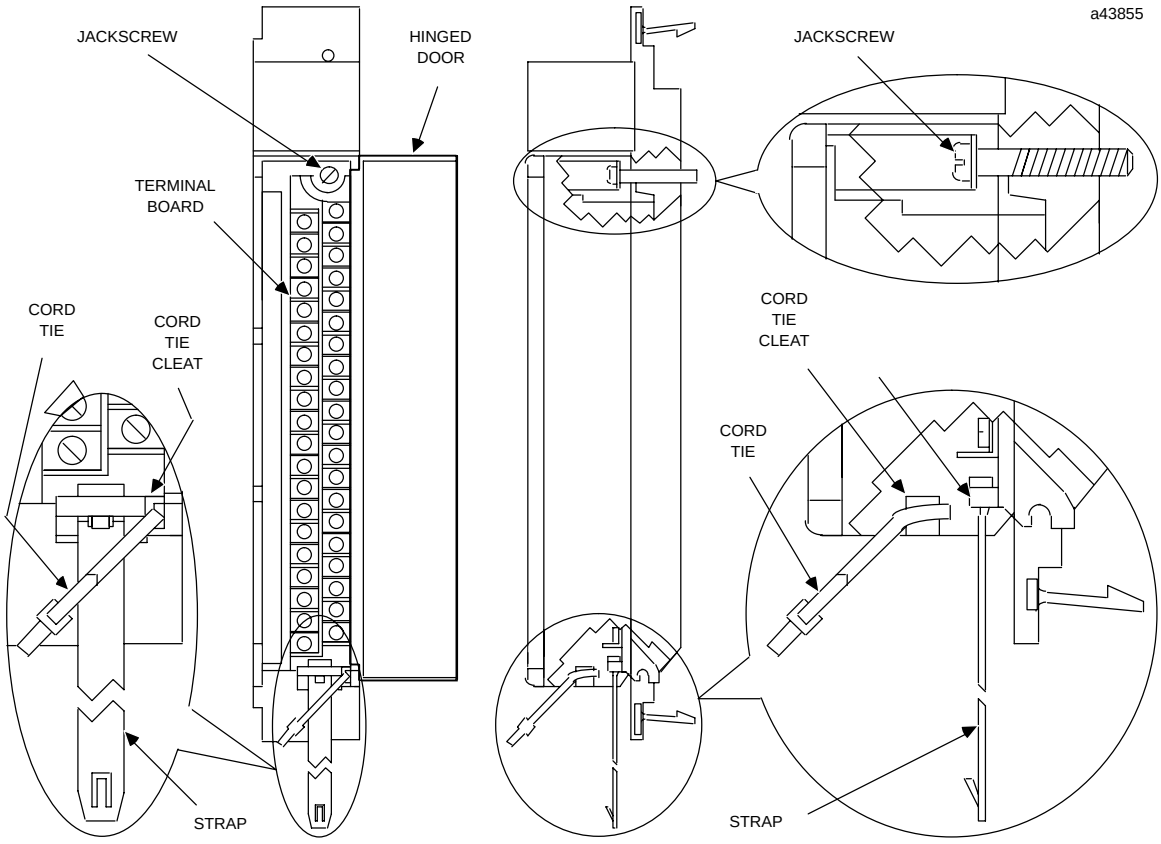


Figure 4. I/O Module Features

1. Turn off power before removing or installing terminal boards. Open the hinged door on the module to access a jackscrew which holds the terminal board securely in place. The detachable field wiring terminal board can now be removed from the module by turning the jackscrew counter-clockwise until it is fully disengaged.
2. To remove the terminal board, grasp the top of the terminal board and swing it outward.

Caution

Do not use the hinged door to remove the terminal board. The hinged door could be damaged if this is done.

3. The terminal board is designed to accept wire sizes from AWG #22 (0.36 mm^2) through AWG #14 (2.10 mm^2). It is important that when using AWG #14 (2.10 mm^2) wire for wiring all points, that a maximum insulation diameter of .135 inch (3.43 mm) not be exceeded. To ensure proper connection, two wires may be terminated on any one terminal only if both wires are the same size.
4. The terminal board is designed to accept a maximum of (40) AWG #14 (2.10 mm^2) wires. If AWG #14 (2.10 mm^2) wires are to be used, then wire markers should be placed at least 8 inches (203 mm) from termination end to provide sufficient space for the hinged door to close.

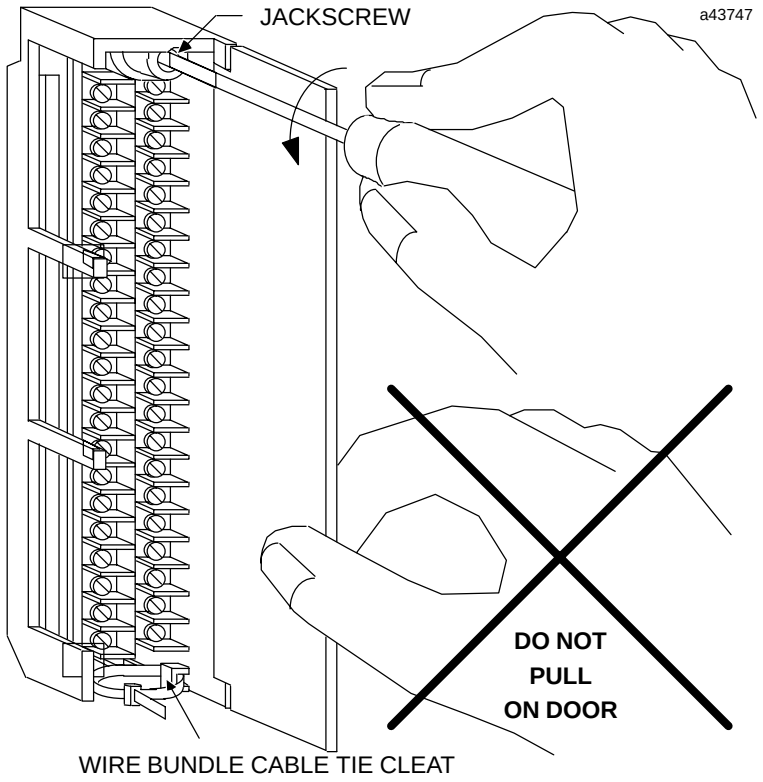


Figure 5. Removal of I/O Terminal Board

5. After completing connections to all modules in a rack, the wire bundle must be secured. To ensure that the wire bundle is secured properly, it is recommended that a cable tie be wrapped around the wire bundle and tightly secured through the cable tie cleat located at the lower right corner of the terminal board. For extremely large wire bundles, additional cable ties should be used.
6. A door label insert is included with each module to indicate circuit wiring information and provide space to record user circuit wiring identification. A slot is provided on the hinged door to allow for insertion of this label. If the label is difficult to insert, crease the scored edge before insertion. The outside label has a color coded stripe to allow quick identification of the module voltage type (blue: low voltage; red: high voltage).
7. After field wiring is completed, the terminal board should be securely fastened to the rack by inserting the terminal board strap (attached to each module) into the small rectangular slots in the bottom card guide grill on the rack. This strap not only secures the terminal board to the rack, it also provides a way of identifying the wired terminal board with its correct mating rack slot location.
8. For adequate module ventilation, it is recommended that at least a 6 inch (152 mm) clearance be allowed above and below the rack grill. Wire bundles should not obstruct the rack grill work.

Removing an I/O Module

The instructions below should be followed when removing an I/O module from its slot in a rack.

- Grasp the board firmly at the top and bottom of the board cover with your thumbs on the front of the cover and your fingers on the plastic clips on the back of the cover.
- Squeeze the rack clips on the back of the cover with your fingers to disengage the clip from the rack rail and pull the board firmly to remove it from the backplane connector.
- Slide the board along the card guide and remove it from the rack.

Attachment A - Using CPU Software Earlier Than Release 6.00

Unsupported CPU Interface to Module

Full CPU support for this module was not available with CPU software prior to Release 6.00. The Interrupt module requires IC641 programming software ladder logic when using CPU software earlier than Release 6.00. This logic is available on the PLC Bulletin Board with a file name of gfk671.zip. For information on how to use the Bulletin board, contact your local authorized PLC distributor or sales office. Also, the following rules must be followed when using the module:

- The Interrupt module will only function when installed in Rack 0.
- Calls to other program blocks and external blocks are not allowed inside an interrupt block.
- The Mask / Unmask I / O Interrupt Service Request #17 (SVCREQ #17) cannot be used.
- Event triggered programs are not supported.

IC641 Programming Software Support Ladder Logic

The following logic is required for each Interrupt module used without the availability of full PLC CPU support. This logic contains the following required operations:

- First scan interrupt acknowledge to clear interrupts occurring during the PLC CPU STOP mode.
- Reads the interrupt word at slot address +92H using a VME READ instruction.
- Interrupt acknowledge: two VME WRITE instructions using inputs 1DH (Hexadecimal) and 15H (Hexadecimal) at slot address +81H (Hexadecimal).
- Toggle module's %I reference for transition contacts.

Note

Do not use this ladder logic with CPU software Release 6.00, or later. It is only intended to be used with CPU software prior to Release 6.00

```
|[   PROGRAM BLOCK DECLARATIONS   ]
               +-------+
               | INT3  |  LANG: LD        (* Interrupt bd slot 3      *)
               +-------+
|[           INTERRUPTS           ]
  (*************************************************************************)
  (* Interrupt Block I/O triggered by first %I reference address for the   *)
  (* module.                                                               *)
  (*************************************************************************)
                        +-------------+
+-------->[%I00001]---->|CALL   INT3  |
                        +-------------+
 [     START OF PROGRAM LOGIC - MAIN BLOCK   ]
| (*************************************************************************)
| (* First scan call resets board and clears interrupts which may have  *)
| (* occurred during PLC stop mode. Should be the first rung in main.   *)
| (*************************************************************************)
|
| << RUNG 5 >>
|
|FST_SCN +-------------+
+--] [---+CALL   INT3  +
|        +-------------+
|
  [      END OF PROGRAM LOGIC      ]
```

```
+[  START  OF  LD  BLOCK  INT3    ]
|
|[     VARIABLE DECLARATIONS      ]

             V A R I A B L E   D E C L A R A T I O N   T A B L E

          REFERENCE     NICKNAME          REFERENCE DESCRIPTION
          ---------     --------     -------------------------------
          %I00001       ITABLE3      Interrupt Table
          %L00001       SLOTTAB      Slot Address Table
          %L00010       SLOT#        Slot number
          %L00011       INTADD       Interrupt word address.
          %L00012       HINT         High Interrupt Address. (0)
          %L00013       MODEADD      Mode register Address.
          %L00014       HMODE        High Mode Address (0).
          %L00015       INTWD3       Interrupt Word
          %L00016       TIME1        Time from start of sweep.
          %L00017       DELAYT       Time1 + delay time.
          %L00018       TIME2        Time after delay elapse.

| (*********************************************************************)
| (* Data Init below is look-up table for Rack 0 slot addressing.       *)
| (* %L12 and %L14 clear upper address for INTADD and MODEADD.          *)
| (*********************************************************************)

 ____________________________________________________________________________
|                                                                            |
|                    D A T A _ I N I T   F U N C T I O N                     |
|                                                                            |
|   New Value >                     Element: 00001        Length: 00009      |
|                                                                            |
|          (Ordered left to right)                                           |
|     1    +00000        +04096        +06144        +08192        +10240    |
|     6    +12288        +14336        +16384        +18432                  |
|____________________________________________________________________________|

| << RUNG 4 >>
|
|FST_EXE +-----+                +-----+                +-----+
+--] [---+DATA_+----------------+MOVE_+----------------+MOVE_+-
|        |INIT_|                | INT |                | INT |
|        | INT |                |     |                |     |
|        |    Q+-SLOTTAB  CONST -+IN  Q+- HINT    CONST -+IN  Q+- HMODE
|        | LEN |          +00000 | LEN |          +00000 | LEN |
|        |00009|                |00001|                |00001|
|        +-----+                +-----+                +-----+
|
| (*********************************************************************)
| (* User to provide slot# for Interrupt board at constant 3 example below.*)
| (* INTADD = Interrupt word address; slot + 92h                        *)
| (* MODEADD = Reset address; slot + 81h.                               *)
| (*********************************************************************)
|
```

```
| << RUNG 6 >>
|
|FST_EXE +-----+                +-----+                +-----+
+--] [---+MOVE_+----------------+ ADD_+----------------+ ADD_+-
|        | INT |                | INT |                | INT |
|        |     |                |     |                |     |
| CONST -+IN  Q+- SLOT#   CONST -+I1  Q+-INTADD   CONST -+I1  Q+-MODEADD
| +00003 | LEN |          +00146 |     |          +00129 |     |
|        |00001|           (92h) |     |           (81h) |     |
|        +-----+         @SLOT# -+I2   |         @SLOT# -+I2   |
|                                +-----+                +-----+
|
£| << RUNG 7 >>
|
|FST_EXE
+--]/[-------------------------------------------------------------------->> SKPFST
|
| (*********************************************************************)
| (* First scan logic clears interrupts which may have occurred while  *)
| (* PLC-CPU was in stop mode.                                          *)
| (*********************************************************************)
|
| << RUNG 9 >>
|                +-----+                +-----+
+----------------+ VME_+----------------+ VME_+-
|                | WRT_|                | WRT_|
|                | BYTE|                | BYTE|
|         CONST -+IN   |          CONST -+IN   |
|           001C | LEN |            0014 | LEN |
|                |00001|                 |00001|
|         CONST -+AM   |          CONST -+AM   |
|           0029 |     |            0029 |     |
|                |     |                 |     |
|        MODEADD-+ADR  |         MODEADD-+ADR  |
|                +-----+                 +-----+
|
| << RUNG 10 >>
|
|        +-----+         +-----+
+--------+ SVC_+---------+ ADD_+-
|        | REQ |         | UINT|
|        |     |         |     |
| CONST -+FNC  |  TIME1 -+I1  Q+-DELAYT
|  00009 |     |         |     |
|        |     |         |     |
| TIME1 -+PARM |  CONST -+I2   |
|        +-----+   00002 +-----+
|
| << RUNG 11 >>
|
| DELAY :
|
```

```
| << RUNG 12 >>
|
|        +-----+         +-----+
+--------+ SVC_+---------+ LT_ +-
|        | REQ |         | UINT|
|        |     |         |     |
| CONST -+FNC  |  TIME2 -+I1  Q+---------------------------------------->>  DELAY
|  00009 |     |         |     |
|        |     |         |     |
| TIME2 -+PARM | DELAYT -+I2   |
|        +-----+         +-----+
|
| << RUNG 13 >>
|                +-----+                 +-----+
+----------------+ VME_+-----------------+ VME_+-
|                | WRT_|                 | WRT_|
|                | BYTE|                 | BYTE|
|         CONST -+IN   |          CONST -+IN   |
|           001C | LEN |            0015 | LEN |
|                |00001|                 |00001|
|         CONST -+AM   |          CONST -+AM   |
|           0029 |     |            0029 |     |
|                |     |                 |     |
|        MODEADD-+ADR  |         MODEADD-+ADR  |
|                +-----+                 +-----+
| << RUNG 14 >>
|
+-------------------------------------------------------------------->>  EXIT
|
| << RUNG 15 >>
|
|SKPFST :
|
| (*********************************************************************)
| (* Read interrupt word at slot address + 92h.                        *)
| (* Length must be two bytes.                                         *)
| (*********************************************************************)
|
| << RUNG 17 >>
|
|                +-----+
+----------------+ VME_+-
|                |  RD_|
|                | BYTE|
|         CONST -+AM   |
|           0029 | LEN |
|                |00002|
|        INTADD -+ADR Q+-INTWD3
|                +-----+
|
| (*********************************************************************)
| (* Interrupt board reset (HSI toggle for interrupt acknowledge = slot *)
| (* address + 81h).                                                   *)
| (*********************************************************************)
|
```

```
| << RUNG 19 >>
|
|                +-----+                 +-----+
+----------------+ VME_+-----------------+ VME_+-
|                | WRT_|                 | WRT_|
|                | BYTE|                 | BYTE|
|         CONST -+IN   |          CONST -+IN   |
|           001D | LEN |            0015 | LEN |
|                |00001|                 |00001|
|         CONST -+AM   |          CONST -+AM   |
|           0029 |     |            0029 |     |
|                |     |                 |     |
|        MODEADD-+ADR  |         MODEADD-+ADR  |
|                +-----+                 +-----+
|
| (*********************************************************************)
| (* Tests for falling edge configuration from config input D3.         *)
| (* Bit test Interrupt word (INTWD3) bit 15.                           *)
| (*********************************************************************)
|
| << RUNG 21 >>
|
|        +-----+
+--------+ BIT_+-
|        |TEST_|
|        | WORD|
|INTWD3 -+IN  Q+-------------------------------------------------------->> FALLEDG
|        | LEN |
|        |00001|
| CONST -+BIT  |
|  00015 +-----+
|
| (*********************************************************************)
| (* Toggles transition table for positive transition contacts/coils    *)
| (* which function within the interrupt block only.                    *)
| (*********************************************************************)
|
| << RUNG 23 >>
|
|        +-----+                 +-----+
+--------+ AND_+-----------------+MOVE_+-
|        | WORD|                 | WORD|
|        |     |                 |     |
|INTWD3 -+I1  Q+-ITABLE3 INTWD3 -+IN  Q+-ITABLE3
|        | LEN |                 | LEN |
|        |00001|                 |00001|
| CONST -+I2   |                 +-----+
|   C000 +-----+
|
| << RUNG 24 >>
|
+-------------------------------------------------------------------->>  USER
|
```

```
|
| << RUNG 25 >>
|
|FALLEDG :
|
|
| (*************************************************************************)
| (* Toggles transition table for negative transition contacts/coils    *)
| (* which function within the interrupt block only.                    *)
| (*************************************************************************)
|
| << RUNG 27 >>
|
|        +-----+                 +-----+
+--------+ OR_ +-----------------+MOVE_+-
|        | WORD|                 | WORD|
|        |     |                 |     |
|INTWD3 -+I1  Q+-ITABLE3 INTWD3 -+IN  Q+-ITABLE3
|        | LEN |                 | LEN |
|        |00001|                 |00001|
| CONST -+I2   |                 +-----+
|   3FFF +-----+
|
|
| << RUNG 28 >>
|
| USER  :
|
| (*************************************************************************)
| (* User logic to be inserted here.  A DOIO at the end of user logic on   *)
| (* outputs associated with the interrupts is suggested for best thru-put.*)
| (*************************************************************************)
|
| << RUNG 30 >>
|
+  EXIT :
|
+[       END OF BLOCK LOGIC       ]
```

GFK-0880C
August 1997

Interrupt Module, 14 Point, 24 Volt DC Positive/Negative Logic

Table 1. Specifications for IC697MDL671 †

Rated Voltage:	24 volts DC
Inputs per Module:	14 interrupts (total of 16 inputs with four groups of four inputs each)
Isolation:	1500 volts - any input to backplane 500 volts between input groups
Input Voltage Range (V_s):	–3 to +30 volts DC
Input Current:	10mA (typical) at rated voltage
Input Impedance:	2.6K ohm, typical
Input Characteristics	
On-state Voltage:	Positive: 13.5 volts to 30 volts Negative: –3 volts to V_s –13.5 volts
On-state Current:	6mA to 15mA
Off-state Voltage:	Positive: –3 volts to 5 volts Negative: V_s –5 volts to 30 volts
Off-state Current:	0 to 2mA (2mA minimum at 5V input)
Filter Delay Time:	1 ms or 10 ms configurable
Minimum Pulse Width	1 ms Filter Select: 1 ms on and off 10 ms Filter Select: 11 ms on and off
Minimum Interrupt Burst (1 ms Filter Selection) † ***Single Point:*** **with IC697CPM915 CPU** **at 1 second PLC Watchdog Setting**	500 Hz
with IC697CPU731 CPU **at 2.5 seconds PLC Watchdog Setting**	290 Hz
Minimum Through-Put * **Interrupt Input to Discrete Output Response** **with IC697CPM915 CPU and IC697MDL740 Output Module with DOIO Function** ** Amount of user logic in the interrupt block may affect performance.*	1.5 ms
Current Required from 5V Bus:	0.30 amp
VME	System designed to support the VME standard C.1

† Refer to GFK-0867B, or later for product standards and general specifications. For installations requiring compliance to more stringent requirements (for example, FCC or European Union Directives), refer to *Installation Requirements for Conformance to Standards.*

Table 2. Ordering Information

Description	Catalog Number
Interrupt Input Module, 24 Volt DC Positive / Negative Logic, 14 interrupt points (16 points total)	IC697MDL671

Note: For Conformal Coat option, or Low Temperature Testing option please consult the factory for price and availability.