IC693CPU321

New In Stock!

GE Fanuc

http://www.pdfsupply.com/automation/ge-fanuc/series-90-30/IC693CPU321

Series 90-30

1-919-535-3180

In Stock! 10 SLOT CPU RACK IC693C IC693CP IC693CPU

www.pdfsupply.com

Email: sales@pdfsupply.com

Important Product Information

February 9, 1994
GFK-0702G

READ THIS INFORMATION FIRST

Product:

IC693 CPU Firmware for Full Production:
IC693CPU311L
IC693CPU321L, M*
IC693CPU331M, N*
IC693CPU313D
IC693CPU323D, E*

* *Suffix changes due to the following: to add UL Class I Div 2 approval and C–UL approval to the products with added suffix letters. No hardware or software changes were required.*

This is a feature release of the IC693 PLC CPU firmware, version 4.40. This release makes available six function block enhancements, programmer and system window modes which more closely match the IC697 Programmable Logic Controller, active constant sweep mode and time settable with the Hand Held Programmer, removal of the restriction on the number of Enhanced GCMs which can be configured, support for the I/O Link Master module, and the SNPX protocol.

New Catalog Number	Replaces
IC693CPU311L	IC693CPU311K
IC693CPU321L,M	IC693CPU321K
IC693CPU313D	IC693CPU313C
IC693CPU323D,E	IC693CPU323C
IC693CPU331M,N	IC693CPU331L

Identification:

The hardware and software identification for this release is summarized in the following table.

Catalog Number	Board Identification	Board Revision	EPROM Label	EPROM Location
IC693CPU311L	BC3C1	44A729648-G01R07 or later	395-027B4.40	U7
IC693CPU321L,M	BC3D1	44A729690-G01R07 or later	395-028B4.40	U7
IC693CPU313D	BC3C2	44A731725-G01R02	395-018B4.40	U8
IC693CPU323D,E	BC3D2	44A731730-G01R04 or later	395-019B4.40	U8
IC693CPU331M,N	CP3A1	44A721765-G01R09 or later	395-029B4.40	U19

Update Information:

Upgrade kits are available to update the CPU models listed below to release 4.40. This update is optional. Existing units may be updated for a charge by ordering the appropriate field update kit.

Upgrade Kit	For Upgrading
44A731233-G05	IC693CPU311K or earlier versions
44A731234-G05	IC693CPU321K or earlier versions
44A731249-G01	IC693CPU313C or earlier versions
44A735538-G01	IC693CPU323C or earlier versions
44A731232-G05	IC693CPU331L or earlier versions

Documentation:

The following table lists the applicable documentation for each of the CPU models listed above by catalog number.

Catalog Number	IPI	User Manual
IC693CPU311L	GFK-0702G	Programmable Controller Installation Manual (for IC693 Products)
IC693CPU321L,M	GFK-0702G	Programmable Controller Installation Manual (for IC693 Products)
IC693CPU313D	GFK-0702G	Programmable Controller Installation Manual (for IC693 Products)
IC693CPU323D,E	GFK-0702G	Programmable Controller Installation Manual (for IC693 Products)
IC693CPU331M,N	GFK-0702G	Programmable Controller Installation Manual (for IC693 Products)

Operational Notes:

Time Required for Reset of PCM

1. Time required for a reset of a PCM module (model 331 CPU only) to be completed can be up to a minute for long scan times. The approximate time required in milliseconds can be computed by multiplying 300 times the scan time in milliseconds.

UVEPROM Change

2. The User Program, configuration, and status tables will automatically be cleared when the CPU 331 PROM (U19) is changed.

Rack Size Mismatch

3. Configuring a model 331 rack size which is different from the actual rack size will produce a "Non-fatal hardware failure" fault in the PLC fault table. This is only a diagnostic fault, and will not inhibit the PLC from going to RUN mode. Although RUN mode is allowed, problems may occur during RUN mode due to the mismatch.

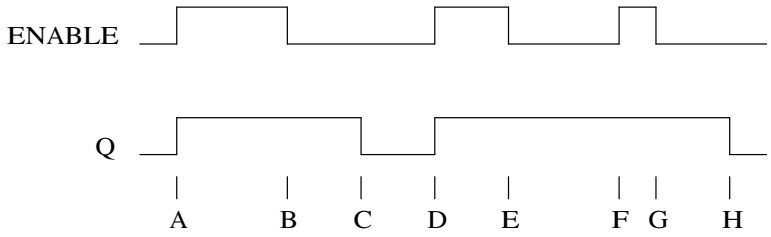
New Features:

1. Off Delay (OFDTR) Function 14

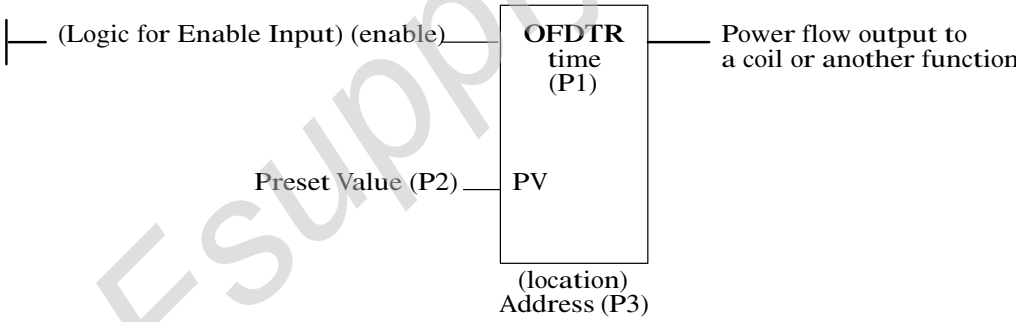
The off-delay timer (OFDTR) increments while power flow is off, and resets to zero, when power flow is on. Time may be counted in tenths of seconds (the default selection), or hundredths of seconds. The range is 0 to +32767 time units. The state of this timer is retentive on power failure; no automatic initialization occurs at power-up.

When the OFDTR first receives power flow, it passes power to the right and clears the current value (CV) located in the operating registers of the timer. The output remains on as long as the function receives power flow. If the function stops receiving power flow from the left, it continues to pass power to the right and the timer starts accumulating time in CV. Each time the function is invoked with the enabling logic set OFF, the current value is updated to reflect the time since the timer was turned off. When the current value (CV) is equal to or greater than the preset value (PV), the function stops passing power flow to the right.

When the function receives power flow again, the current value resets to zero and the output is enabled again.



A = ENABLE and Q both go high; timer is reset (CV = 0).
B = ENABLE goes low; timer start accumulating time.
C = CV reaches PV; Q goes low, and timer sops accumulating time.
D = ENABLE goes high; timer is reset (CV = 0).
E = ENABLE goes low; timer starts accumulating time.
F = ENABLE goes high; timeris reset (CV = 0).
G = ENABLE goes low; timer begins accumulating time.
H = CV reaches PV; Q goes low, and timer stops accumulating time.



Programming Elements and Sequential Order of Programming

1. Logic controlling the enable input from the left bus. Must start with an LD element.
2. Type of function (Function 14)
3. Parameter (P1) Timer Accuracy or base value for timing increments;
   1 = one hundredth of a second (.01 second),
   10 = one tenth of a second (0.1 second).
4. Parameter (P2) Preset Time, a constant number or the register that will contain the preset value.
5. Parameter (P3) Timer Location, the first register of the three sequential registers containing the operating values.

Parameters for OFDTR (Function 14)

The following table specifies which memory types are valid for each of the OFDTR function block's parameters:

Allowable Memory Types for OFDTR (Function 14)

Parameter	%I	%Q	%M	%T	%G	%S	%R	%AI	%AQ	Constant
Timer Accuracy (P1)										●
Preset Time (P2)	●	●	●	●	●		●	●	●	●
Timer Location (P3)							●			

Timer Accuracy (P1): The timer accuracy parameter indicates the time base of the timer. A constant of 1 indicates a time base of 0.01 second; 10 indicates a time base of 0.1 seconds. Other values are not accepted as a valid parameter value.

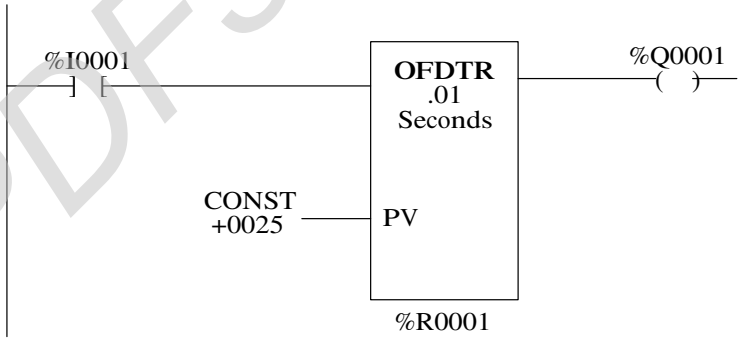
Preset Time (P2): The preset time parameter indicates the time period for the off-delay timer. It is indicated by a positive (only) 16-bit twos complement signed integer (0...32,767). A constant of -1 indicates that no preset time parameter is specified. For this case, the preset time will be accessed from the Operating Registers of the timer.

Timer Location (P3): The timer location gives the address of a three-word data structure which is used by the timer function block.

Programming Example for OFDTR Function

In the following example, power flow will be passed through the OFDTR to turn on %Q0001 at a time of 2.5 seconds after input 1 goes from being closed to opened. The Time Base or Timer Accuracy is a tenth of a second (.01); the Preset is a constant of 25, and the Location of this OFDTR is Register 1.

Ladder Diagram Representation



Statement List Representation

```
#0001:    LD          %I0001
#0002:    FUNC   14   ONDTR
          P1:         10
          P2:         25
          P3:         %R0001
#0003:    OUT         %Q0001
```

After pressing [INS] Key: Programing sequence

Key Strokes	HHP Display
Initial Display:	#0001 INS <S _
Press the key sequence [LD] [A I AI] [1] :	#0001 INS <S LD I 1 _
Press the [ENT] key:	#0002 INS <S _
Press the key sequence [FUNC] [1] [4] or [TMR ONDTR] 3 times	#0002 INS <S FUNC 14 OFDTR
Press the [ENT] key:	#0002 OFDTR <S P01

Press the key sequence [1] [0] :

```
#0002    OFDTR <S
P01    10_
```

Press the [ENT] key:

```
#0002    OFDTR <S
P02
```

Press the key sequence [2] [5] :

```
#0002    OFDTR <S
P02    25_
```

Press the [ENT] key:

```
#0002    OFDTR <S
P03
```

Press the key sequence [R] [1] :

```
#0002    OFDTR <S
P03    R 1_
```

Press the [ENT] key:

```
#0003    INS     <S
_
```

Press the key sequence

[OUT/OUTM] [B Q/AQ] [1] :

```
#0003    INS     <S
OUT       Q 1_
```

Press the [ENT] key:

```
#0004    INS     <S
_
```

2. ACTIVE AND CONSTANT SWEEP MODES

Two new CPU parameters have been added after "Power-Up Mode":

Parameter	Selections	Default Value
Active Constant Sweep Mode	DISABLE ENABLE	DISABLE
Active Constant Sweep Setting	5 - 200 msec	100 msec

In addition, the "Constant Sweep Mode" and "Constant Sweep Setting" parameters have been renamed to "Configured Constant Sweep Mode" and "Configured Constant Sweep Setting".

Active Constant Sweep Mode Parameter

The PLC may be configured during RUN mode to use a constant amount of time per sweep. The active constant sweep mode parameter allows the user the ability to enable or disable the constant sweep mode while the program is running, and have the affects noticed immediately. This parameter can be used to toggle the sweep mode of the PLC, without changing the configured constant sweep mode parameter. The active constant sweep mode parameter, once changed, is only valid during the current RUN mode. Upon going from STOP to RUN mode, the configured sweep mode parameter value is copied to the active sweep mode parameter.

Use the Right cursor key to scroll through the PLC parameters until the active constant sweep mode (ACT CNSW) parameter is displayed. Then, use the -/+ key to toggle the selection between DISABLE and ENABLE. By default, the PLC will execute every sweep as fast as possible.

Active Constant Sweep Setting Parameter

If the constant sweep mode is enabled in the PLC during RUN mode, then the Active Constant Sweep Setting Parameter can be used to adjust the sweep time. This allows the user to fine tune the sweep time while the PLC is running a program. Changing this parameter does not affect the configured constant sweep setting parameter. The active constant sweep setting is only valid during the current RUN mode, as long as Active Constant Sweep Mode is enabled. Upon going from STOP to RUN mode, the configured sweep setting parameter value is copied to the active sweep setting parameter. If the active constant sweep mode is disabled, this parameter is ignored. The active constant sweep value may range between 5 and 200 milliseconds, inclusive.

Use the Right cursor key to scroll through the PLC parameters until the active constant sweep setting (ACT CONS TM) parameter is displayed. To set the active sweep time, enter a value between 5 and 200 milliseconds, inclusive, and press the ENT key. The default setting is 100 milliseconds.

Configured Sweep Mode Parameter

The PLC may be configured to use a constant amount of time per sweep. The constant sweep mode parameter should be enabled when I/O points or register values must be polled at a constant frequency, as in control algorithms. The configured sweep mode parameter can be overridden by the active constant sweep mode parameter during RUN mode, but upon going from STOP to RUN mode, the configured sweep mode parameter

value is copied to the active constant sweep mode parameter (see Active Constant Sweep Mode Parameter). The configured sweep mode parameter can only be edited during STOP mode.

Use the Right cursor key to scroll through the PLC parameters until the configured constant sweep mode (CFG CNSW) parameter is displayed. Then, use the -/+ key to toggle the selection between DISABLE and ENABLE. By default, the PLC will execute every sweep as fast as possible.

Configured Constant Sweep Setting Parameter

If the configured constant sweep mode is enabled in the PLC, the sweep time value must also be selected. The configured constant sweep setting parameter can be overridden by the active constant sweep setting parameter during RUN mode, but upon going from STOP to RUN mode, the configured constant sweep setting parameter value is copied to the active constant sweep setting parameter. This allows the user to maintain a configured setting, while fine tuning the setting during RUN mode with the active constant sweep setting parameter. If the configured constant sweep mode is disabled, this parameter is ignored. The configured constant sweep value may range between 5 and 200 milliseconds, inclusive.

Use the Right cursor key to scroll through the PLC parameters until the configured constant sweep setting (CFG CONS TM) parameter is displayed. To set the sweep time, enter a value between 5 and 200 milliseconds, inclusive, and press the ENT key. The default setting is 100 milliseconds.

3. SVCREQ #26: Interrogate I/O

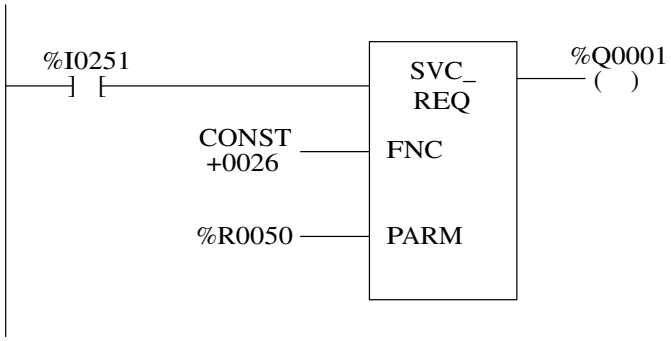
Use SVCREQ function #26 to interrogate the actual modules present and compare them with the rack/slot configuration, generating addition, loss, and mismatch alarms, as if a store configuration had been performed. This SVCREQ will generate faults on both the PLC and I/O fault table, depending on the fault. This function has no parameter block and always outputs power flow.

NOTE:

The time for this SVCREQ to execute depends on how many faults exist. Therefore, execution time of this SVCREQ will be greater for situations where more modules are faulted.

Example:

In the following example, when input %I0251 is ON the actual modules are interrogated and compared to the rack/slot configuration. Output %Q0001 is turned on after the SVCREQ is complete.



4. SVCREQ #23: Read Master Checksum

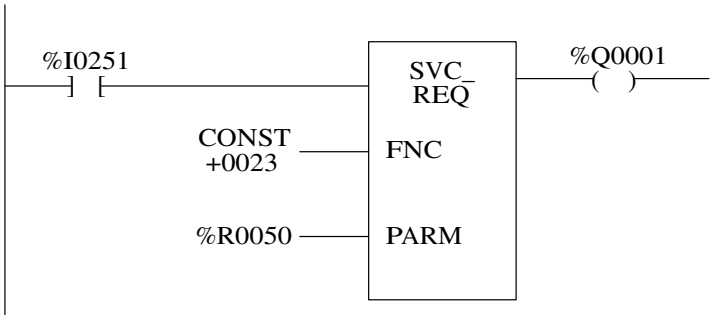
Use SVCREQ function #23 in order to read the Master Checksums for the User Program and the Configuration. The SVCREQ output is always set to ON, and the output block of information (see below) starts at the address given in parameter 3 (PARM) of the SVCREQ function.

In the instance when a Run Mode Store is active, the Program checksums may not be valid until the store is complete. Therefore, two flags for when the Program and Configuration checksums are valid are given at the beginning of the output parameter block. For this function, the parameter block output has a length of 12 words. Format of the information is shown below:

Output Parameter Block	Word Address
Master Program Checksum Valid (0 = not valid, 1 = valid)	address
Master Configuration Checksum Valid (0 = not valid, 1 = valid)	address + 1
Number of Program Blocks (including _Main)	address + 2
Size of User Program in bytes (DWORD Data Type)	address + 3
Program Additive Checksum	address + 5
Program CRC Checksum (DWORD Data Type)	address + 6
Size of Configuration Data in bytes	address + 8
Configuration Additive Checksum	address + 9
Configuration CRC Checksum (DWORD Data Type)	address + 10

Example 1:

In the following example, when input %I0251 is ON the Master Checksum information is placed into the parameter block and the output coil (%Q0001) is turned on. The parameter block is located at %R0050.



5. Masked Compare Word (MSKCMPW) Function 143 Masked Compare Dword (MSCCMPD) Function 144

The Masked Compare function is used to compare the contents of two bit strings with the ability to mask selected bits. The length of the bit strings to be compared is specified by the LEN parameter where the value of LEN specifies the number of 16 bit words for MSKCMPW and 32 bit words for MSKCMPD.

When the logic controlling the enable input to the function passes power flow to the enable (EN) input, the function begins comparing the bits in the first string with the corresponding bits in the second. Comparison continues until a miscompare is found, or until the end of the string is reached.

The BIT input is used to store the bit number where the next comparison should start where a 0 indicates the first bit in the string. The BN output is used to store the bit number where the last comparison occurred where a 1 indicates the first bit in the string. Using the same reference for BIT and BN causes the compare to start at the next bit position after a miscompare or at the beginning if all bits compared successfully upon the next invocation of the function block.

If you want to start the next comparison at some other location in the string, you can enter different references for BIT and BN. If the value of BIT is a location that is beyond the end of the string, BIT is reset to a 0 before starting the next comparison.

IF all Bits in I1 and I2 are the Same

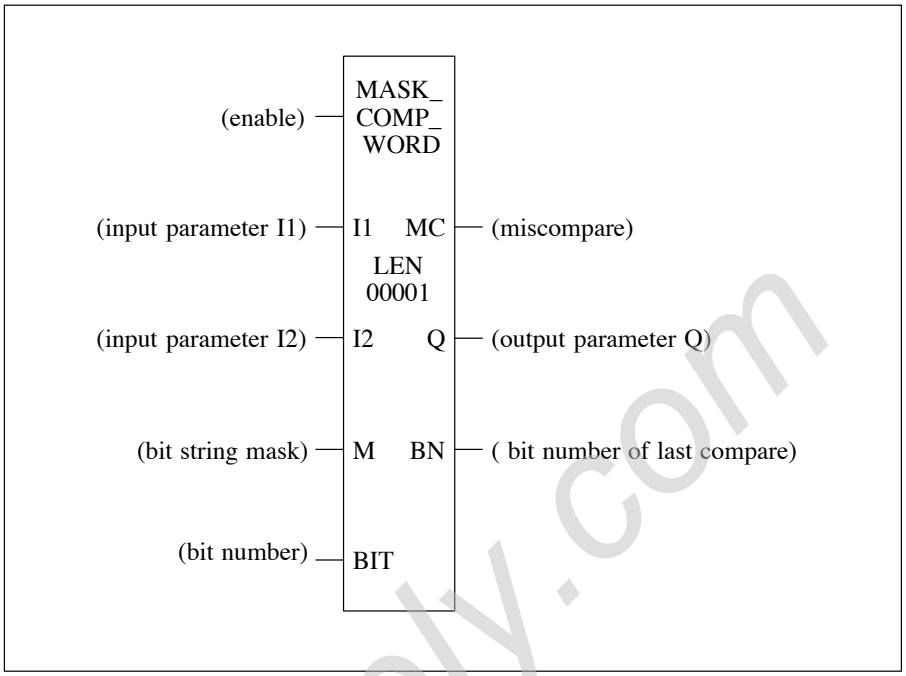
If all corresponding bits in strings I1 and I2 match, the function sets the "miscompare" output MC to 0 and BN to the highest bit number in the input strings. The comparison then stops. On the next invocation of MSKCMPW, it will be reset to 0.

If a Miscompare is Found

When the two bits currently being compared are not the same, the function then checks the correspondingly-numbered bit in string M (the mask). If the mask bit is a 1, the comparison continues until another miscompare or the end of the input strings is reached.

If a miscompare is detected and the corresponding mask bit is a 0, the function:

1. Sets the corresponding mask bit in M to a 1.
2. Sets the miscompare (MC) output to 1.
3. Updates the output bit string Q to match the new content of mask string M.
4. Sets the bit number output (BN) to the number of the miscompared bit.
5. Stops the comparison.



Parameters:

Parameter	Description
enable	Permissive logic to enable the function
I1	Reference for the first bit string to be compared.
I2	Reference for the second bit string to be compared.
M	Reference for the bit string mask.
BIT	Reference for the bit number where the next comparison should start.
LEN	The number of words in the bit string.
MC	User logic to determine if a miscompare has occurred.
Q	Output copy of the mask (M) bit string.
BN	Number of the bit where the last compare occurred.

Valid Memory Types:

Parameter	flow	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	none
enable	●											
I1		○	○	○	○	○	○	●	●	●		
I2		○	○	○	○	○	○	●	●	●		
M		○	○	○	○	○†	○	●	●	●		
BIT		●	●	●	●	●	●	●	●	●	●	
LEN											●‡	
MC	●											●
Q		○	○	○	○	○†	○	●	●	●		
BN		●	●	●	●	●	●	●	●	●		

● = Valid reference or place where power may flow through the function.
○ = Valid reference for WORD data only; not valid for DWORD.
† = %SA, %SB, %SC only; %S cannot be used.
‡ = Max const value of 4095 for WORD and 2047 for DWORD.

Programming example for MSKCMPW Function

In the following example, when %I0001 is TRUE, the MSKCMPW function block is executed. %M0001 through %M0016 is compared with %M0017 through %M0032. %M0033 through %M0048 contains the mask value. The value in %R0001 determines at which bit position the comparison starts within the two input strings. The contents of the above references before the function block is executed are as follows:

(I1) - %M0001 =

0	1	1	0	1	1	0	0	0	1	1	0	1	1	0	0

(I2) - %M0017 =

0	1	1	0	1	1	0	1	0	1	1	0	1	1	1	1

(M/Q) - %M0033 =

0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

(BIT/BN) - %R0001 = 0

(MC) - %Q0001 = FALSE

The contents of these references after the function block is executed are as follows:

(I1) - %M0001 =

0	1	1	0	1	1	0	0	0	1	1	0	1	1	0	0

(I2) - %M0017 =

0	1	1	0	1	1	0	1	0	1	1	0	1	1	1	1

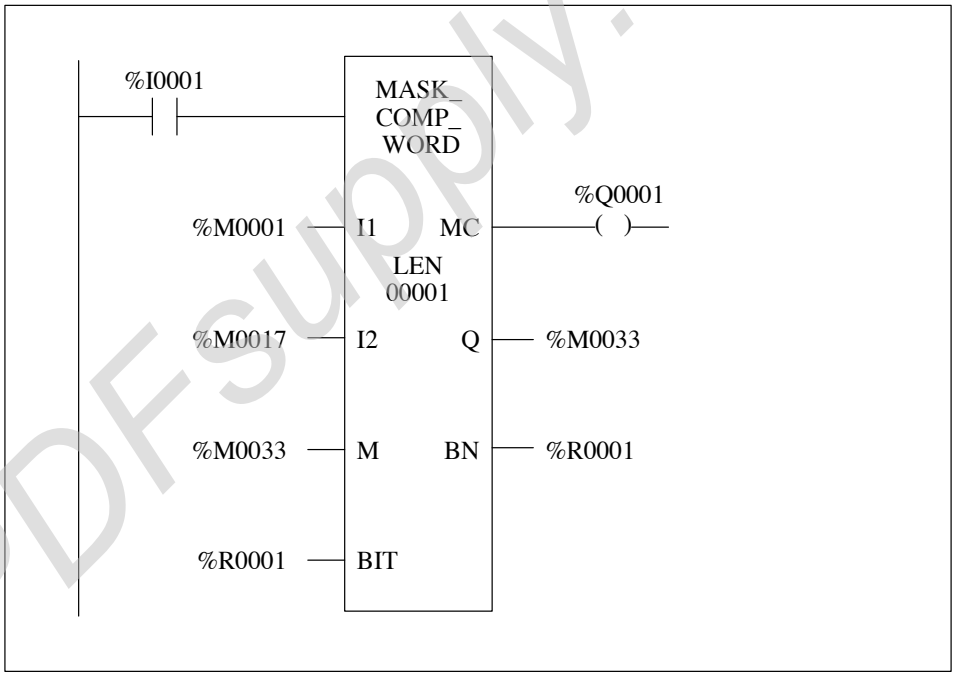
(M/Q) - %M0033

0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1

(BIT/BN) - %R0001 = 9

(MC) - %Q0001 = TRUE

Ladder Diagram Representation



Statement List Representation

#0001:	LD		%I0001
#0002:	FUNC	143	MSKCMPW
		P1:	%M0001
		P2:	%M0017
		P3:	%M0033
		P4:	%R0001
		P5:	1
		P6:	%M0033
		P7:	%R0001
#0003:	OUT		%Q0001

After pressing the INS key: Programming sequence

Key Strokes	HHP Display
Initial Display:	#0001 INS <S _
Press the key sequence: LD, A/I, 1	#0001 INS <S LD I 1_
Press the ENT key:	#0002 INS <S _
Press the key sequence: FUNC, 1, 4, 3	#0002 INS <S FUNC 143 MSKCMPW
Press the ENT key:	#0002 MSKCMPW <S P01 _

Press the key sequence:

C M T 1

```
#0002    MSKCMPW    <S
P01          M1_
```

Press the ENT key:

```
#0002    MSKCMPW    <S
P02 _
```

Press the key sequence:

C M T 1 7

```
#0002    MSKCMPW    <S
P02          M17_
```

Press the ENT key:

```
#0002    MSKCMPW    <S
P03 _
```

Press the key sequence:

C M T 3 3

```
#0002    MSKCMPW    <S
P03          M33_
```

Press the ENT key:

```
#0002    MSKCMPW    <S
P04 _
```

Press the key sequence:

R 1

```
#0002    MSKCMPW    <S
P04          R1_
```

Press the ENT key:

```
#0002    MSKCMPW    <S
P05 _
```

Press the key sequence:

1

```
#0002    MSKCMPW    <S
P05           1_
```

Press the ENT key:

```
#0002    MSKCMPW    <S
P06 _
```

Press the key sequence:

C M T 3 3

```
#0002    MSKCMPW    <S
P06          M33_
```

Press the ENT key:

```
#0002    MSKCMPW    <S
P07 _
```

Press the key sequence:

R 1

```
#0002    MSKCMPW    <S
P07          R1_
```

Press the ENT key:

```
#0003    INS        <S
_
```

Press the key sequence:

OUT OUTM B Q AQ 1

```
#0003    INS        <S
OUT          Q1_
```

Press the ENT key:

```
#0004    INS        <S
_
```

6. NUMBER OF ENHANCED GCMs ALLOWED

The I/O configuration limit of two GCM+'s has been eliminated. The number of GCM+'s that can be configured is limited only by the presence of a GCM (in which case, no GCM+'s may be configured - this is no change) or by the maximum configuration size that the PLC can hold. So if a new GCM+ added into the I/O configuration would cause the configuration size to exceed the maximum allowed, then the new GCM+ is not accepted.

7. FASTER MAIN RACK DO_IO (Model 331 only)

The ALT parameter of the DO_IO function block can now be used to enter the slot of a single module in the main rack. In this case, the DO_IO function block will execute in 80 microseconds, instead of the 236 microseconds required when the block is programmed without the ALT parameter. No error checking is performed to prevent overlapping reference addresses or module type mismatches.

8. SNPX PROTOCOL

The SNPX protocol is supported by all models in this release. Refer to the *Communications Protocols - SNP-X Protocol Specification* for description of its use.

9. I/O LINK MASTER MODULE

The I/O Link Master Module, IC693BEM321, is supported by all models except the model 211 in this release. Refer to the *I/O Link Master Module User's Manual* for a description of its use.

10. WINDOW MODES

The default window modes now work in a manner similar to the IC697 PLC window modes, except that the window times are not changeable but are fixed at a maximum of 6 ms. The default programmer window mode is LIMITED mode. In this mode, no more than 6 ms. per sweep is devoted to executing functions for the current programmer. If a function requires more than 6 ms. to execute, then only part of the function is performed each sweep. The default system communication window mode is RUN-TO-COMPLETION. In this mode, all pending requests are executed completely in the window, regardless of how many simultaneous requests are submitted to the CPU and regardless of how long the requests take to execute. Because the IC693 PLC watchdog timer is fixed, the total window time is limited to a maximum of 50 ms.

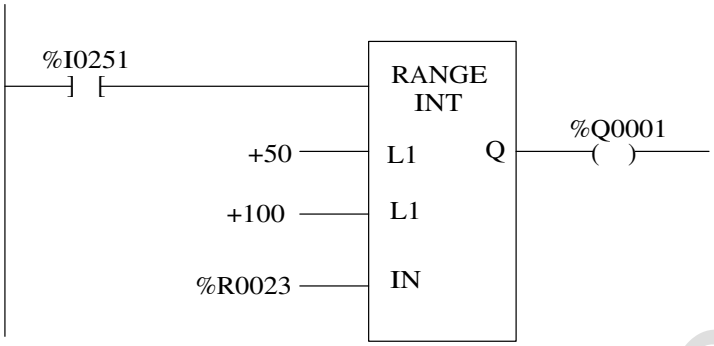
Currently, the only way to change the window modes is by submitting a service request to the CPU, either using a custom SNP application program, or using a PCM application program.

11. RUN-MODE STORE

Programs can now be stored from IC641 programming software to the PLC while the PLC is in RUN mode without clearing the non-retentive references. During the store process, the execution of the program is paused while scanning of I/O continues.

12. RANGE FUNCTION BLOCK

The RANGE function block is added to the Relation set. It has the following form:



The RANGE function has four inputs: a boolean enable (EN), limit 1 (L1), limit 2 (L2), and an input (IN). It has one output, a boolean (Q). The available types on which the RANGE function block can be performed are INT, DINT, WORD, and DWORD.

When enable (EN) is true, the RANGE function block will compare the (IN) value against the range specified by the (L1) and (L2) values. The (Q) output is set true when the (IN) value is within the range specified by the (L1) and (L2) inclusive. Otherwise (Q1) is set to false.

NOTE:

(L1) and (L2) merely represent the end points of a range. There isn't any fixed min/max or hi/lo connotation assigned to either of them. That is to say, a desired range or (1 - 100) could be specified by assigning 0 to (L1) and 100 to (L2) or by assigning 100 to (L1) and 0 to (L2).

The following table specifies which memory types are available for the RANGE function block:

ALLOWABLE MEMORY TYPES FOR RANGE INT, WORD

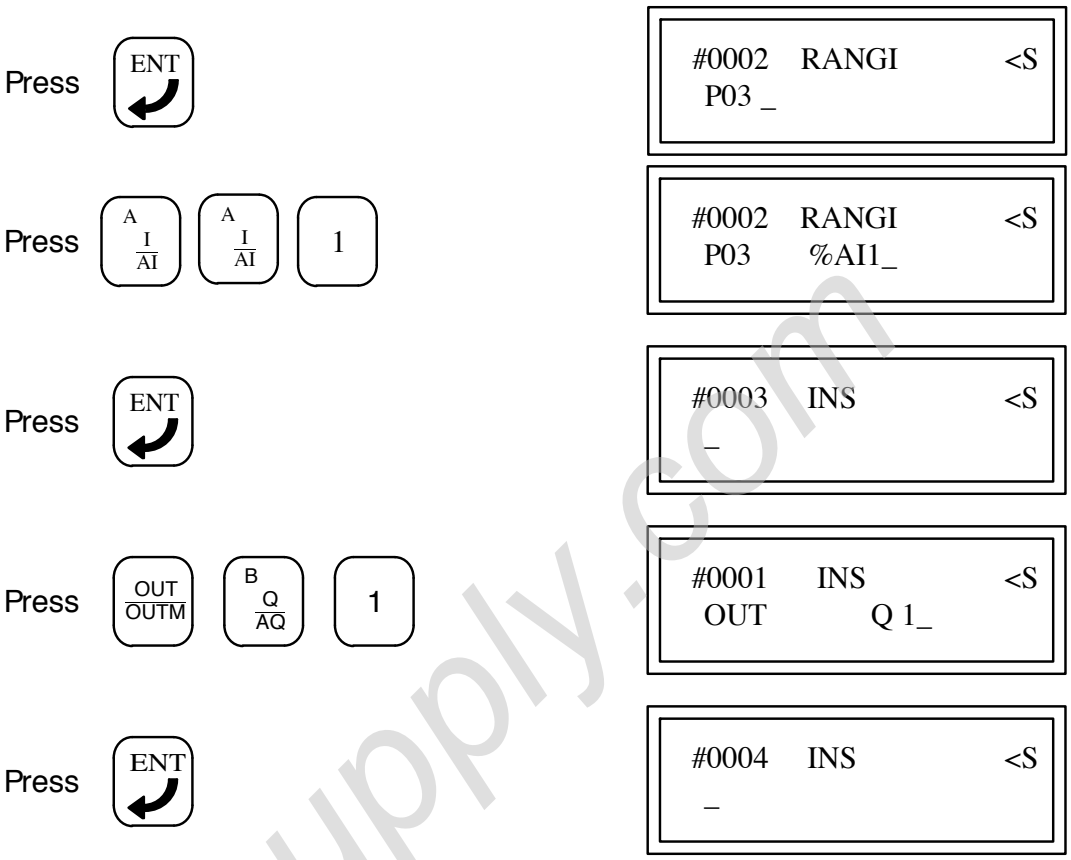
PARAM	%I	%Q	5M	%T	%G	%R	%AI	%AQ	%S	CONST	FLOW	NONE
EN											•	
L1	•	•	•	•	•	•	•	•		•		
L2	•	•	•	•	•	•	•	•		•		
IN	•	•	•	•	•	•	•	•				
Q											•	•

ALLOWABLE MEMORY TYPES FOR RANGE DINT, DWORD

PARAM	%I	%Q	5M	%T	%G	%R	%AI	%AQ	%S	CONST	FLOW	NONE
EN											•	
L1						•	•	•		•		
L2						•	•	•		•		
IN						•	•	•				
Q											•	•

After pressing the INS key: Programming sequence

Key Strokes	HHP Display
Initial Display:	#0001 INS <S _
Press LD, A I/AI, 1	#0001 INS <S LD I 1_
Press ENT	#0002 INS <S _
Press FUNC, 1, 4, 0	#0002 INS <S FUNC 140 RANGI
Press ENT	#0002 RANGI <S P01 _
Press 1, 0, 0, 0	#0002 RANGI <S P01 1000_
Press ENT	#0002 RANGI <S P02 _
Press 0	#0002 RANGI <S P02 0_



Problems Resolved by This Upgrade:

Retentiveness of %G References

1. %G references were not saved in the EEPROM or on the Memory Card. So, if RAM memory had been lost, the %G references were not be restored when the reference memories were restored from the device.

Restrictions and Open Problems:

Pid Block In Manual Mode

1. A PID function block executes at most once every 10 ms., even if the sweep time is less than 10 ms. So if the user program is very short and the sweep time is less than 10 ms., the the PID algorithm will not process every sweep. This can be avoided by setting a constant sweep time to 10 ms. or greater.