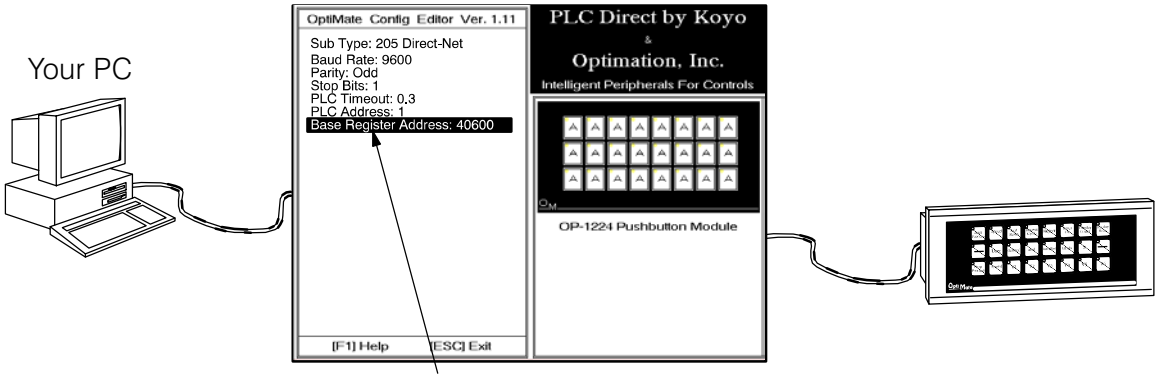
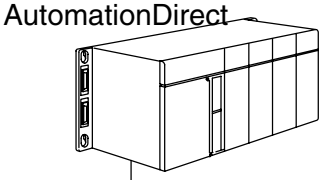
Applying Ladder Logic

General Concepts

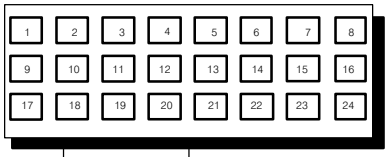
Memory Mapping

On Pages 2 and 3 of this manual, we introduced you to the basic concept of memory mapping. The OP-1224 uses memory mapping in order to link itself to a PLC. Memory mapping is a technique that maps the memory of the OP-1224 into the memory of the PLC. During initial configuration, you indicate where in the PLC memory you want to start the mapping process (See Step 7B on Page 18). By knowing where the data of the specific panel is mapped, this data can be moved, changed or monitored using ladder logic.





During configuration, you determine the starting address for the memory mapping process.

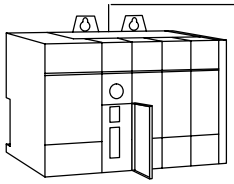


Mapping Assignments

Mapped Memory Location	Function
m (such as V40600) C0-C17	Pushbuttons 1-16 ON/OFF
m+1 (such as V40601) C20-C37	Pushbuttons 17-24 ON/OFF
m+2 (such as V40602) C40-C57	LEDs 1-16 flash
m+3 (such as V40603) C60-C77	LEDs 17-24 flash
m+4 (such as V40604) C100-C117	LEDs 1-16 ON/OFF
m+5 (such as V40605) C120-C137	LEDs 17-24 ON/OFF
m+6 (such as V40606) C140-C157	Force Function Data (1-16)
m+7 (such as V40607) C160-C177	Force Function Mode/Data (17-24)

The pushbuttons are numbered left to right starting in the upper left corner.

Allen-Bradley

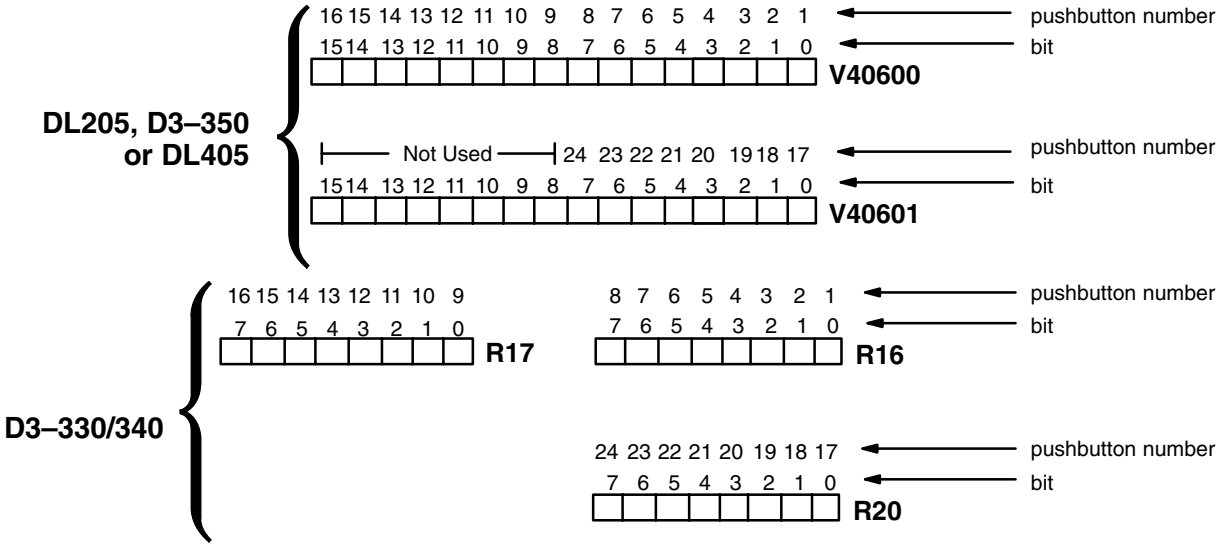


Mapped Memory Location	Function
m (such as N7: 0/0– 0/15)	Pushbuttons 1-16 ON/OFF
m+1 (such as N7: 1/0 1/15)	Pushbuttons 17-24 ON/OFF
m+2 (such as N7: 2/0 2/15)	LEDs 1-16 flash
m+3 (such as N7: 3/0 3/15)	LEDs 17-24 flash
m+4 (such as N7: 4/0 4/15)	LEDs 1-16 ON/OFF
m+5 (such as N7: 5/0 5/15)	LEDs 17-24 ON/OFF
m+6 (such as N7: 6/0 6/15)	Force Function Data (1-16)
m+7 (such as N7: 7/0 7/15)	Force Function Mode/Data (17-24)

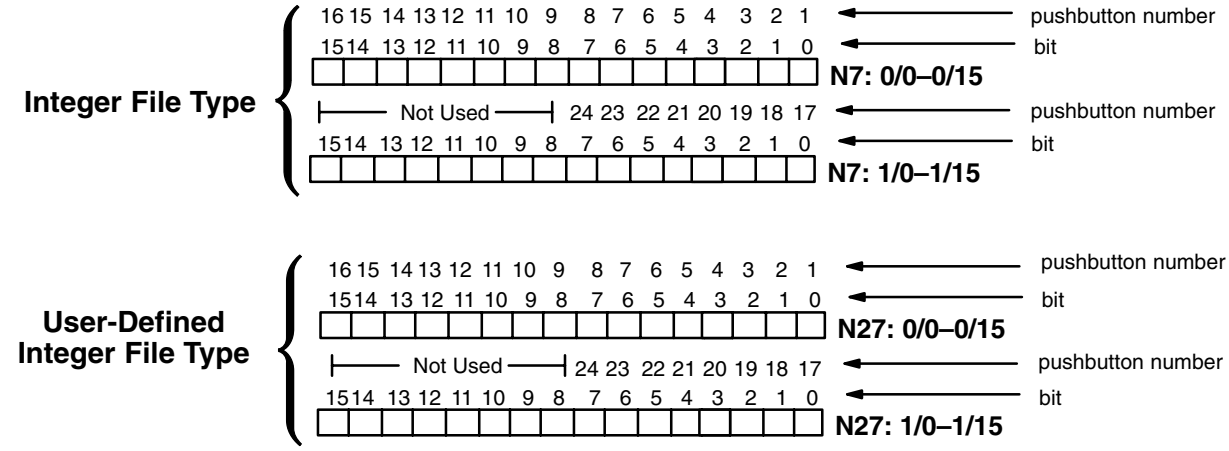
Addressing Conventions

Before we jump into ladder logic programming, let's take a moment to review and compare the addressing conventions used by AutomationDirect and Allen-Bradley.

AutomationDirect Memory–A typical address within a AutomationDirect programmable controller is Vxxxx (such as V40600 for the DL205 or DL405 families) or Rxxx (such as R16 for the DL330/340 family). The V-memory in the DL205 and DL405 is divided into 16-bit boundaries, and the R-memory in the DL330/340 is divided into 8-bit boundaries. Refer to your individual User Manuals for complete memory information. The two diagrams below show you how the lamps of the OP-1224 could be mapped during configuration. In this example, we have arbitrarily chosen V40600 and R16 as starting boundaries to map the pushbuttons to the PLC, but it could actually be any available user or internal relay memory areas as long as they are consecutive:



Allen-Bradley Memory–A typical address for Allen-Bradley might be N7:0/0 or N27:0/0. The OP-1224 will allow you to define your starting address for mapping purposes using either Allen-Bradley's integer (N7) file type or user-defined integer file types (N9–N255). *If you plan to use an integer file between N9 and N255, you must define these in the Allen-Bradley memory map before configuring the panel.* Below we have shown you how 16-bit integer files could be used to map the pushbuttons to the Allen-Bradley PLC.



Three Different Ways to Use the Panel

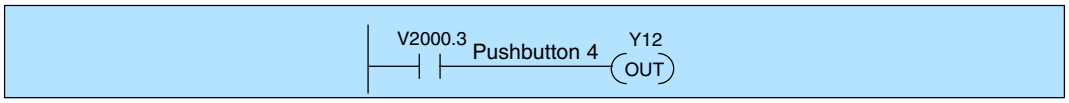
Depending on the type of CPU and the number of OP-1224 functions selected, there are three different ways to interface your ladder logic with the panel.

1. **Bit-of-Word**
2. **Internal Relays**
3. **User Memory Combined with Internal Relays**

Which of these methods is best for you depends on the make and model of the PLC you are using. Let's look at each of these three methods and discuss their relative merits.

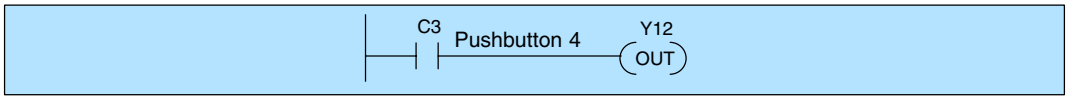
Method 1: Bit-of-Word DL250/350/450 and Allen-Bradley

The most direct way to address the individual bits with your ladder logic is to use "bit-of-word". This method is available to the DL250/350/450 (**AutomationDirect**) and SLC 5/03 and 5/04 (Allen-Bradley). Below is a rung of logic that shows how the DL250/350/450 might use the status of bit 3 to control a process connected to Y12. Don't worry about understanding exactly how it works at this point. We will cover that just a few pages later. **Refer to Pages 24–28 for DL250/350/450 examples, Pages 40–47 for Allen-Bradley.**



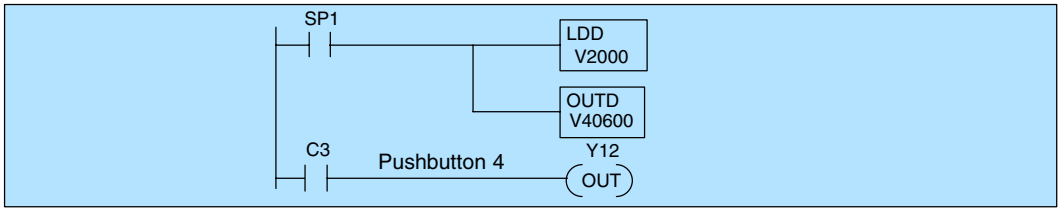
Method 2: Internal Relays (All Options Used)

This method is only available to **AutomationDirect** programmable controllers. If you are already familiar with the DL205, DL305 and DL405 PLCs, then you know about internal relays. These relays, by PLC design, are mapped to certain bits in reserved memory areas. You can make use of these relays during configuration with the OPEditor by mapping directly to the control relay reserved memory area. **This method should only be used if you plan to use all of the functions of the panel; otherwise it will consume internal relays unnecessarily.** Using this method automatically consumes 128 internal relays. In the example below, we have used one of the mapped pushbuttons to control the output Y12. **Refer to Pages 29–33.**



Method 3: Remapping (Selected Options)

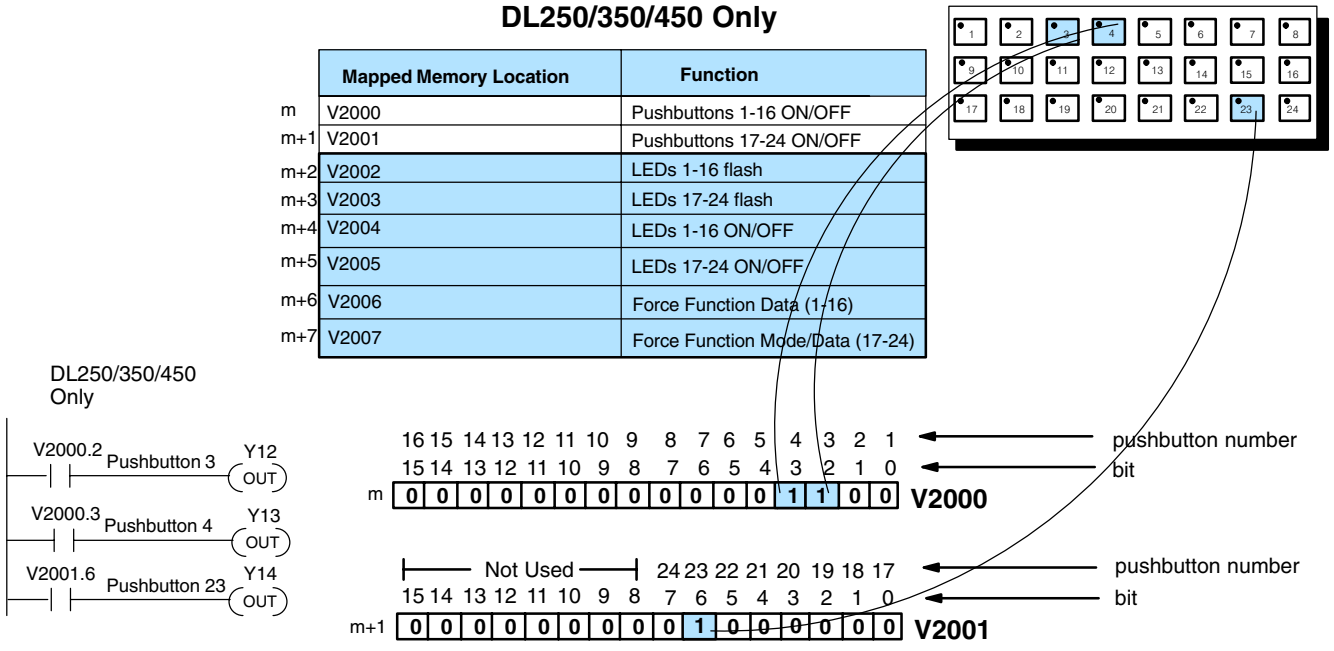
A better way to make use of internal relays when you are not using all of the OP-1224 options is to use a process of "remapping". With this technique you map your panel to user memory (such as V2000), and then map parts of your user memory only to those relays you actually need to use. The example below shows ladder logic necessary to detect when a pushbutton has been pressed. It maps V2000/V2001 to V40600/V40601 and consumes only 32 relays. The point is–it uses only the relays necessary for the option you have selected. We'll make this clearer in a moment when we give you specific ladder logic examples that use this technique. **By convention, in this manual we will use syntax of the form V2000:V40600 to refer to memory locations that have been mapped together**. **Refer to Pages 34–39 for ladder logic examples.**



Using the OP-1224 with the DL250/350/450

Using Pushbutton Status Via Ladder Logic

By convention we are using the letter m to refer to consecutive memory locations in the PLC. Memory locations **m** and **m+1** reflect the state of the pushbuttons. If you have the DL250/350/450, the status of the individual bits of these two words is easily determined by using the bit-of-word instruction. The example shown below uses a base register address of V2000 to map the status of the pushbuttons. When Pushbutton 3 is pressed it affects bit 2 of V2000. Likewise, Pushbutton 4 affects bit 3. Pushbutton 23 affects bit 6 of V2001.



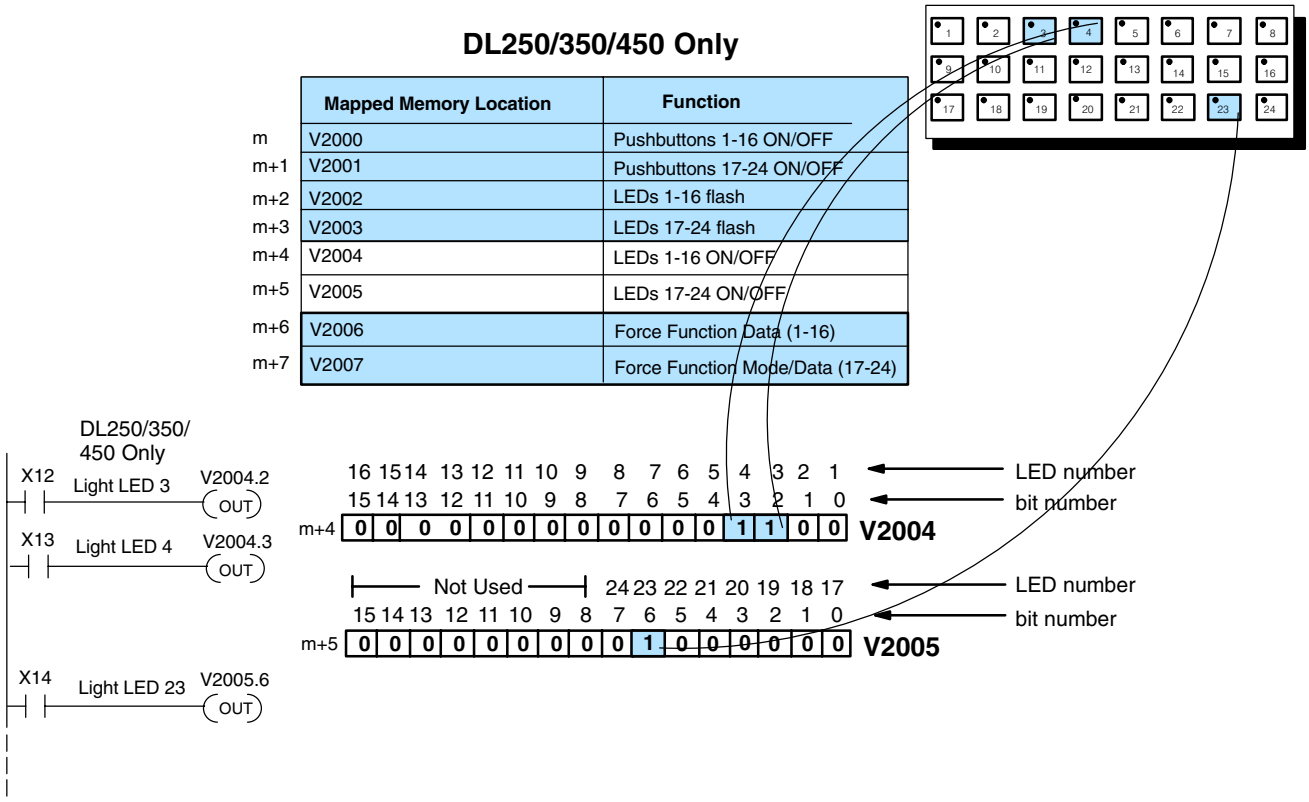
Controlling LEDs Separately with the DL250/350/450

By default, the LED simply shows the state of the pushbutton–ON or OFF. If a pushbutton is configured for momentary operation, there are two options available for the LED. It can show the state of the pushbutton or it can be controlled independently by enabling the **LED Separation** feature. When you have enabled the LED Separation feature, the ON/OFF state of the LED is controlled only by the status of the bits in **m+4** and **m+5**. These bits can be manipulated via your ladder logic. **Remember: Any pushbutton configured for maintained (alternate action) will ignore the bits in these two words.**

In the example below, we show how the bit-of-word instruction can control LEDs 3, 4 and 23 when you have designated V2000 as the base address during configuration with your OPEditor software. X12 turns ON LED3, X13 turns ON LED4, and X14 turns ON LED23. **Remember: Independent control of the LEDs can only be accomplished if you have Enabled LED Separation during your initial configuration. (See Page 19, Step 9.)**

DL250/350/450 Only

	Mapped Memory Location	Function
m	V2000	Pushbuttons 1-16 ON/OFF
m+1	V2001	Pushbuttons 17-24 ON/OFF
m+2	V2002	LEDs 1-16 flash
m+3	V2003	LEDs 17-24 flash
m+4	V2004	LEDs 1-16 ON/OFF
m+5	V2005	LEDs 17-24 ON/OFF
m+6	V2006	Force Function Data (1-16)
m+7	V2007	Force Function Mode/Data (17-24)



Adding Flashing with the DL250/350/450

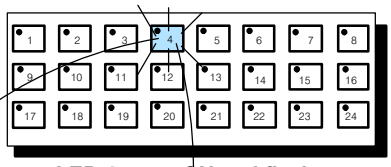
If you plan to use this feature with one or more pushbuttons, there are three things you must always remember during configuration:

1. Flashing is only available for those buttons that have been configured as **Momentary**.
2. **LED Separation** must be Enabled.
3. The **Flash Option** must be Enabled.

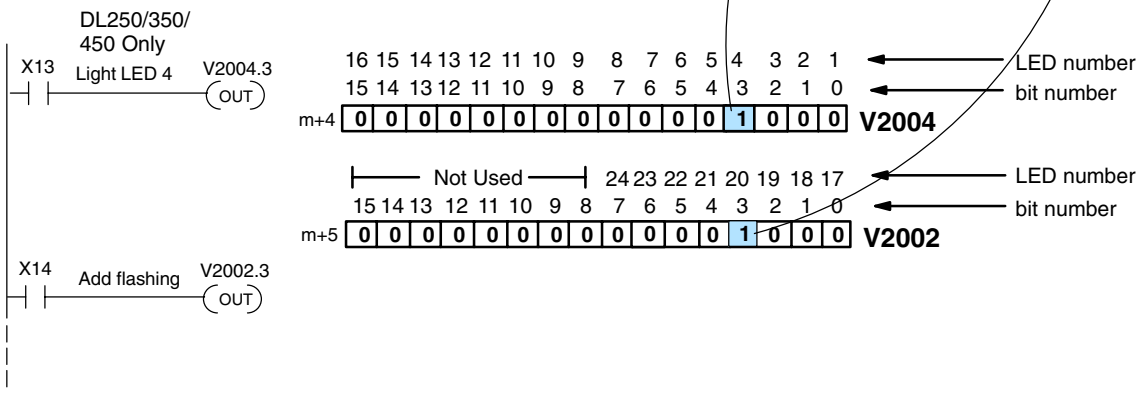
The Flashing Option is triggered through your ladder logic. On the previous page, we showed you how to turn ON an LED, this example shows you how to add flashing to an LED that has been turned ON. The flashing feature is controlled by the status of the bits in **m+2** and **m+3** memory areas. In the example below, we have begun our mapping at V2000 during the initial configuration. We are turning ON LED4 and then making it flash. Bit 3 of V2004 turns the LED ON, and bit 3 or V2002 makes it flash.

DL250/350/450 Only

	Mapped Memory Location	Function
m	V2000	Pushbuttons 1-16 ON/OFF
m+1	V2001	Pushbuttons 17-24 ON/OFF
m+2	V2002	LEDs 1-16 flash
m+3	V2003	LEDs 17-24 flash
m+4	V2004	LEDs 1-16 ON/OFF
m+5	V2005	LEDs 17-24 ON/OFF
m+6	V2006	Force Function Data (1-16)
m+7	V2007	Force Function Mode/Data (17-24)



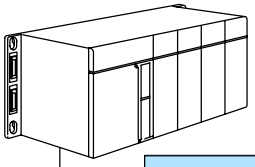
LED 4 turns ON and flashes



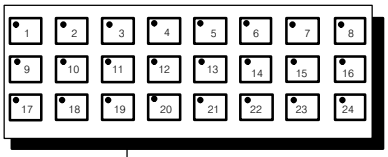
Force Function Registers

The OP-1224 has the capability to "force" a pushbutton ON or OFF through your ladder logic. If you plan to use this function, you must enable the force option during configuration. (See Page 20, Step 10.)

NOTE: The Force Function will only work for those pushbuttons that you have configured as "maintained" (alternate action). It will not work for momentary pushbuttons.



Mapping Assignments



Automation

Mapped Memory Location	Function
m (such as V2000)	Pushbuttons 1-16 ON/OFF
m+1 (such as V2001)	Pushbuttons 17-24 ON/OFF
m+2 (such as V2002)	LEDs 1-16 flash
m+3 (such as V2003)	LEDs 17-24 flash
m+4 (such as V2004)	LEDs 1-16 ON/OFF
m+5 (such as V2005)	LEDs 17-24 ON/OFF
m+6 (such as V2006)	Force Function Data (1-16)
m+7 (such as V2007)	Force Function Mode/Data (17-24)

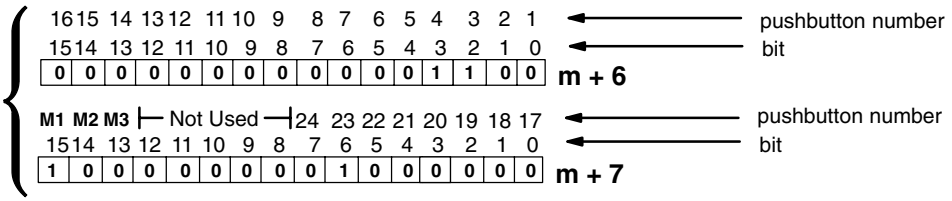
How the Memory is Used–Looking at the above memory map**, m+6** stores the forcing data for Pushbuttons 1-16 and **m+7** stores forcing data for Pushbuttons 17-24. There are three modes of the force function. These modes are controlled by the 3 most significant bits of m+7.

Mode 1 (M1)– This <u>forces all of the Pushbuttons to reflect the status</u> stored in m+6 and m+7. For example, the data shown below would force Pushbuttons 3, 4 and 23 to ON and all the others would be forced OFF. Notice that bit 15 of m +7 is set to 1 for this mode. M2 and M3 are set to 0's.

Mode 2 (M2)– This <u>forces ON only those Pushbuttons matching the bits set</u> in registers m+6 and m+7. The bits not set do not affect the status of the Pushbuttons. You would set M2 to 1 while M1 and M3 are set to 0.

Mode 3 (M3)– This <u>forces OFF only those Pushbuttons matching the bits set</u> in registers m+6 and m+7. The bits not set do not affect the status of the Pushbuttons. You would set M3 to 1, while M1 and M2 are set to 0.

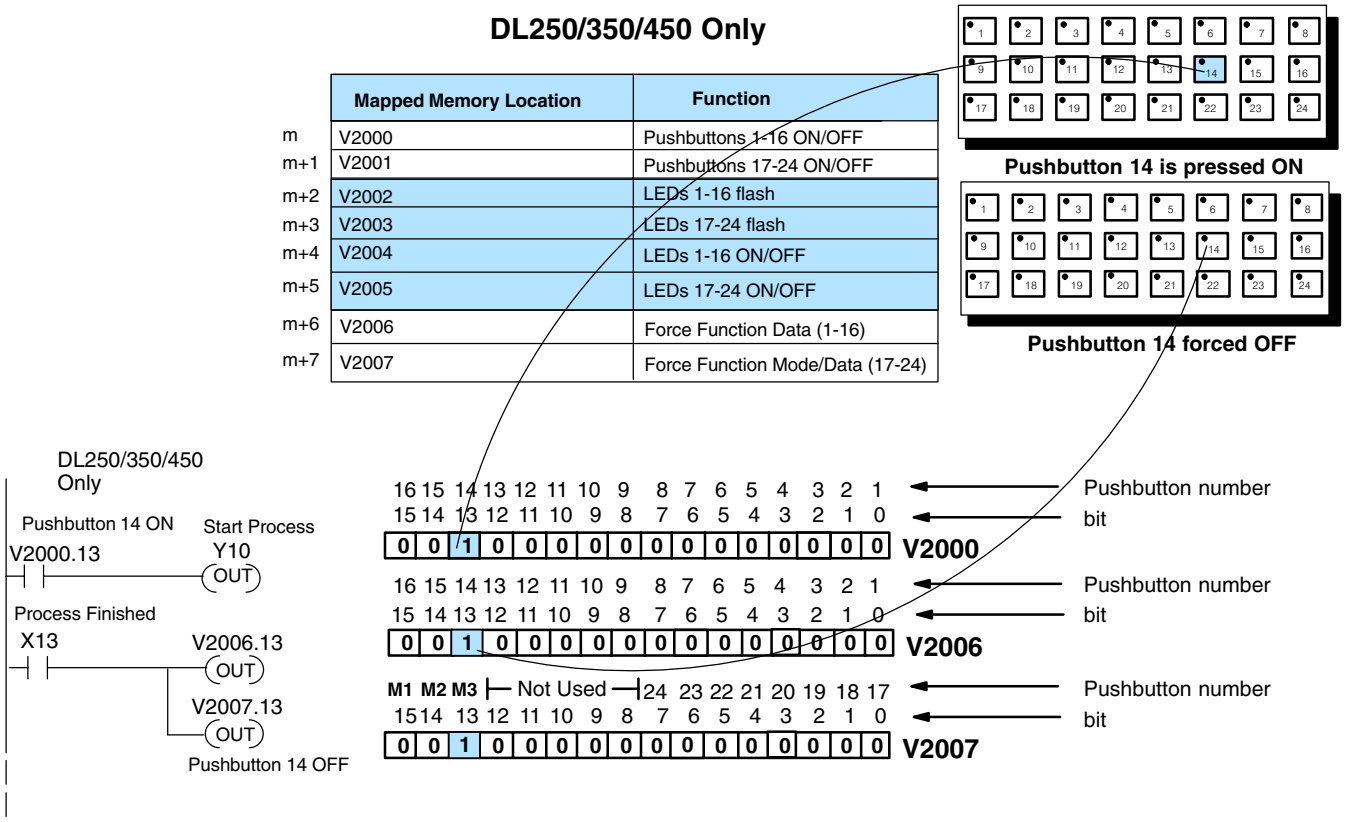
Force Function Registers



Think of forcing as a one-shot process. That is, once you have set the mode in m+7, the bit patterns in m and m+1 are changed (according to the mode selected), and then, all of the bits in m+6 and m+7 <u>are set to zero</u>. What this means is that all pushbuttons return to normal manual operation after the forcing is completed.

Forcing Pushbuttons ON/OFF with DL250/350/450

You can also use bit-of-word with the DL250/350/450 to force pushbuttons ON or OFF. Here, during configuration with the OPEditor, we have chosen V2000 as our base address for the mapping in the PLC. In this example, we are using Pushbutton 14 to start a process, and then forcing the pushbutton OFF when the process is completed. Memory location **m** (V2000 in this case) holds the bit that reflects the status of Pushbutton 14. Memory locations **m+6** and **m+7** hold the data for the forcing. Here we have chosen to use Mode 3. With this mode, whichever bits are set to 1 in m+6 and m+7, the corresponding pushbuttons will be forced to OFF. In the following example, we only set bit 13 in m+6; so only Pushbutton 14 is turned OFF.

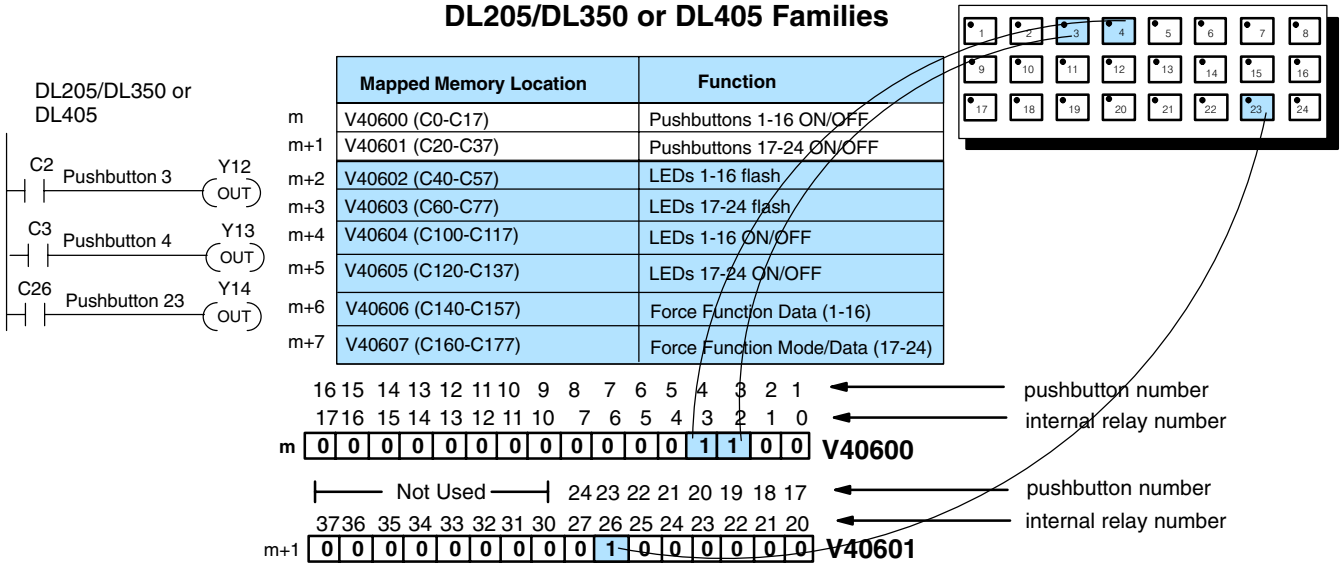


DL205/DL305/DL405 (Using All Functions)

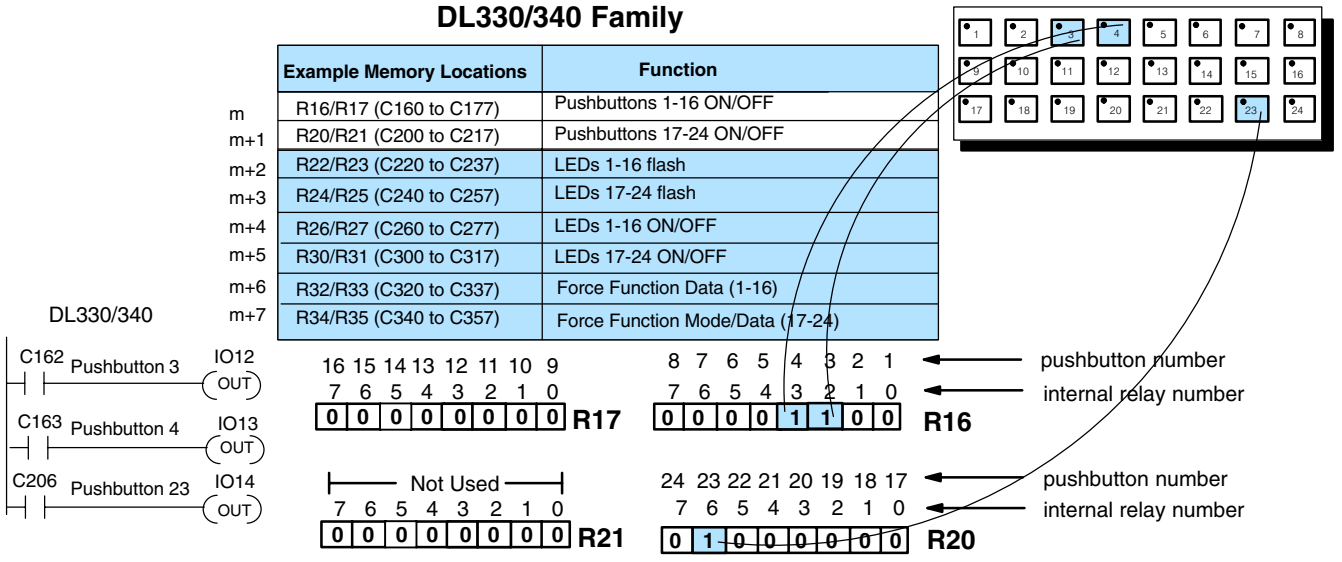
Using Pushbutton Status Via Ladder Logic

When you configure the OP1224, you must choose a base address in the CPU. This address can be a direct mapping to the reserved memory locations that are tied to internal relays. The internal relays of the DL205 and DL405 families start at V40600 and the internal relays of the DL305 family start at R16. Using this method, the total mapping consumes 128 internal relays. You should only use this method when using all of the OP-1224 functions. In the examples below, we have chosen V40600 as the starting address for either a DL205 or DL405. We have chosen R16 as our starting address for the DL305. *Notice that the internal control relays are numbered in octal and not decimal.* In the examples below, our ladder logic is interacting with Pushbuttons 3, 4 and 23.

DL205/DL350 or DL405 Families



DL330/340 Family



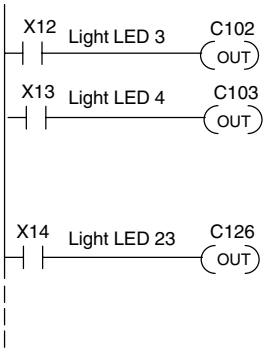
Note: To determine the control relay number, use the register number as the first two digits and the bit number as the last digit. For example, Bit 3 of R16 is referenced as C163.

Controlling LEDs Separately

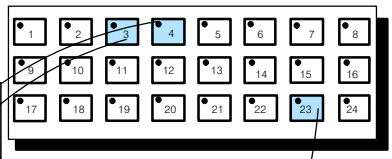
By default, the LED simply shows the state of the pushbutton–ON or OFF. If a pushbutton is configured for momentary operation, there are two options available for the LED. It can show the state of the pushbutton or it can be controlled independently by enabling the **LED Separation** feature. When you have enabled the LED Separation feature, the ON/OFF state of the LED is controlled only by the status of the bits in **m+4** and **m+5**. These bits can be manipulated via your ladder logic. In the examples below, our ladder logic is controlling LEDs 3, 4 and 23. **Remember: Any pushbutton configured for maintained (alternate action) will ignore the bits in these two words. Independent control of the LEDs can only be accomplished if you have Enabled LED Separation during your initial configuration. (See Page 19, Step 9.)**

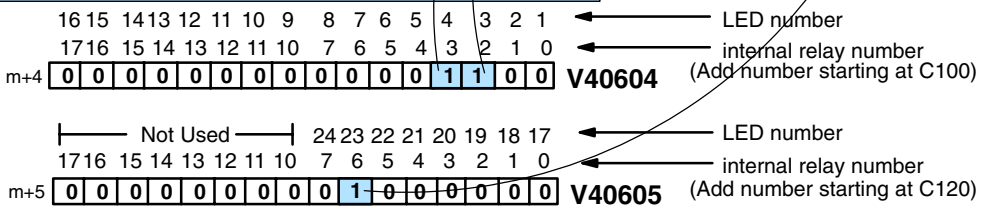
DL205/DL350 and DL405 Family

DL205/DL350 or DL405



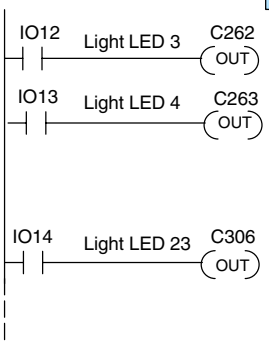
	Mapped Memory Location	Function
m	V40600 (C0-C17)	Pushbuttons 1-16 ON/OFF
m+1	V40601 (C20-C37)	Pushbuttons 17-24 ON/OFF
m+2	V40602 (C40-C57)	LEDs 1-16 flash
m+3	V40603 (C60-C77)	LEDs 17-24 flash
m+4	V40604 (C100-C117)	LEDs 1-16 ON/OFF
m+5	V40605 (C120-C137)	LEDs 17-24 ON/OFF
m+6	V40606 (C140-C157)	Force Function Data (1-16)
m+7	V40607 (C160-C177)	Force Function Mode/Data (17-24)



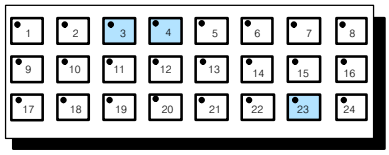


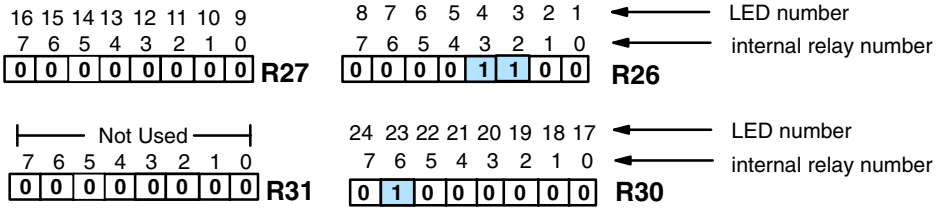
DL330/340 Family

DL330/340



	Example Memory Locations	Function
m	R16/R17 (C160 to C177)	Pushbuttons 1-16 ON/OFF
m+1	R20/R21 (C200 to C217)	Pushbuttons 17-24 ON/OFF
m+2	R22/R23 (C220 to C237)	LEDs 1-16 flash
m+3	R24/R25 (C240 to C257)	LEDs 17-24 flash
m+4	R26/R27 (C260 to C277)	LEDs 1-16 ON/OFF
m+5	R30/R31 (C300 to C317)	LEDs 17-24 ON/OFF
m+6	R32/R33 (C320 to C337)	Force Function Data (1-16)
m+7	R34/R35 (C340 to C357)	Force Function Mode/Data (17-24)





Note: To determine the control relay number, use the register number as the first two digits and the bit number as the last digit. For example, Bit 3 of R26 is referenced as C263.

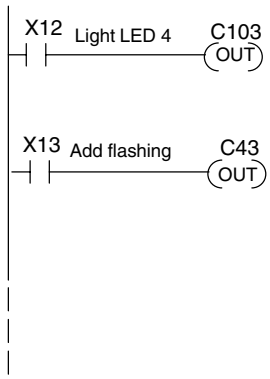
Adding Flashing

If you plan to use this feature with one or more pushbuttons, there are three things you must always remember during configuration:

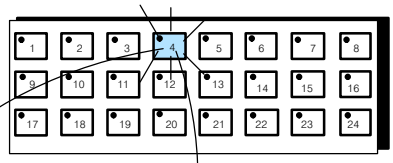
1. Flashing is only available for those buttons that have been configured as **Momentary**.
2. **LED Separation** must be Enabled.
3. The **Flash Option** must be Enabled.

The Flashing Option option is triggered through your ladder logic. On the previous page, we showed you how to turn ON an LED, this example shows you how to add flashing to an LED that has been turned ON. The flashing feature is controlled by the status of the bits in **m+2** and **m+3** memory areas. In the example below, we have begun our mapping at V40600 during the initial configuration. We are turning ON LED4 and then making it flash. Bit 3 of **m+4** turns the LED ON, and bit 3 of **m+2** makes it flash.

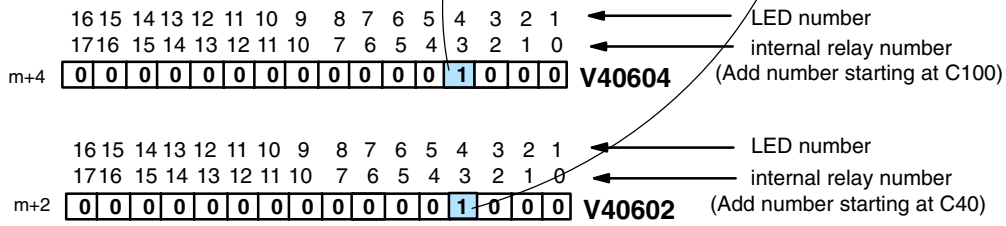
DL205/DL350 or DL405 Families



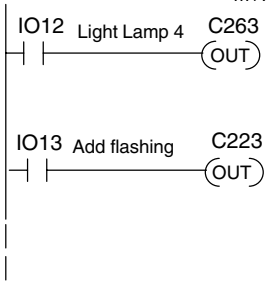
	Mapped Memory Location	Function
m	V40600 (C0-C17)	Pushbuttons 1-16 ON/OFF
m+1	V40601 (C20-C37)	Pushbuttons 17-24 ON/OFF
m+2	V40602 (C40-C57)	LEDs 1-16 flash
m+3	V40603 (C60-C77)	LEDs 17-24 flash
m+4	V40604 (C100-C117)	LEDs 1-16 ON/OFF
m+5	V40605 (C120-C137)	LEDs 17-24 ON/OFF
m+6	V40606 (C140-C157)	Force Function Data (1-16)
m+7	V40607 (C160-C177)	Force Function Mode/Data (17-24)



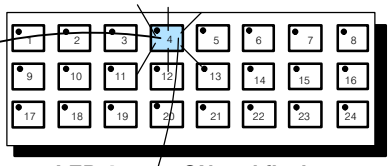
LED 4 turns ON and flashes



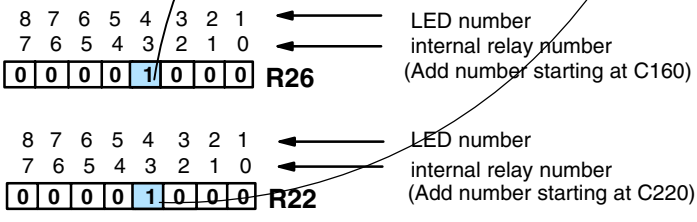
DL330/340 Family



	Example Memory Locations	Function
m	R16/R17 (C160 to C177)	Pushbuttons 1-16 ON/OFF
m+1	R20/R21 (C200 to C217)	Pushbuttons 17-24 ON/OFF
m+2	R22/R23 (C220 to C237)	LEDs 1-16 flash
m+3	R24/R25 (C240 to C257)	LEDs 17-24 flash
m+4	R26/R27 (C260 to C277)	LEDs 1-16 ON/OFF
m+5	R30/R31 (C300 to C317)	LEDs 17-24 ON/OFF
m+6	R32/R33 (C320 to C337)	Force Function Data (1-16)
m+7	R34/R35 (C340 to C357)	Force Function Mode/Data (17-24)



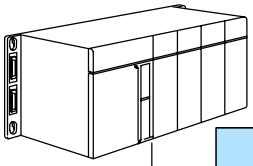
LED 4 turns ON and flashes



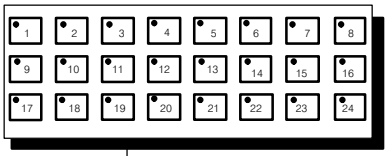
Force Function Registers

The OP-1224 has the capability to "force" a pushbutton ON or OFF through your ladder logic. If you plan to use this function, you must enable the force option during configuration. (See Page 20.)

NOTE: The Force Function will only work for those pushbuttons that you have configured as "maintained" (alternate action). It will not work for momentary pushbuttons.



Mapping Assignments



AutomationD

Mapped Memory Location	Function
m (such as V40600, C0-C17)	Pushbuttons 1-16 ON/OFF
m+1 (such as V40601, C20-C37)	Pushbuttons 17-24 ON/OFF
m+2 (such as V40602, C40-C57)	LEDs 1-16 flash
m+3 (such as V40600, C60-C77)	LEDs 17-24 flash
m+4 (such as V40604, C100-C117)	LEDs 1-16 ON/OFF
m+5 (such as V40605, C120-C137)	LEDs 17-24 ON/OFF
m+6 (such as V40606, C140-C157)	Force Function Data (1-16)
m+7 (such as V40607, C160-C177)	Force Function Mode/Data (17-24)

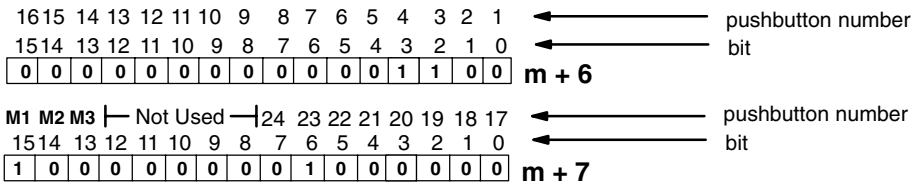
How the Memory is Used–Looking at the above memory map**, m+6** stores the forcing data for Pushbuttons 1-16 and **m+7** stores forcing data for Pushbuttons 17-24. There are three modes of the force function. These modes are controlled by the 3 most significant bits of m+7.

Mode 1 (M1)– This <u>forces all of the Pushbuttons to reflect the status</u> stored in m+6 and m+7. For example, the data shown below would force Pushbuttons 3, 4 and 23 to ON and all the others would be forced OFF. Notice that bit 15 of m +7 is set to 1 for this mode. M2 and M3 are set to 0's.

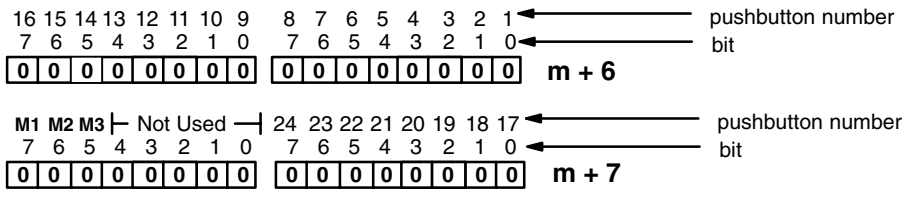
Mode 2 (M2)– This <u>forces ON only those Pushbuttons matching the bits set</u> in registers m+6 and m+7. The bits not set do not affect the status of the Pushbuttons. You would set M2 to 1 while M1 and M3 are set to 0.

Mode 3 (M3)– This <u>forces OFF only those Pushbuttons matching the bits set</u> in registers m+6 and m+7. The bits not set do not affect the status of the Pushbuttons. You would set M3 to 1 while M1 and M2 are set 0.

Force Function Registers for DL205/DL350/ DL405



Force Function Registers for DL3330/340



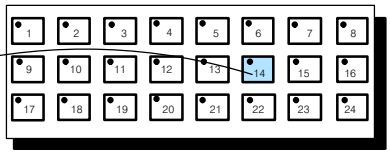
Think of forcing as a one-shot process. That is, once you have set the mode in m+7, the bit patterns in m and m+1 are changed (according to the mode selected), and then, all of the bits in m+6 and m+7 <u>are set to zero</u>. What this means is that all pushbuttons return to normal manual operation after the forcing is completed.

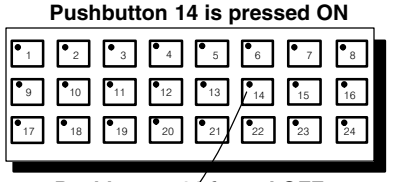
Forcing Pushbuttons ON or OFF

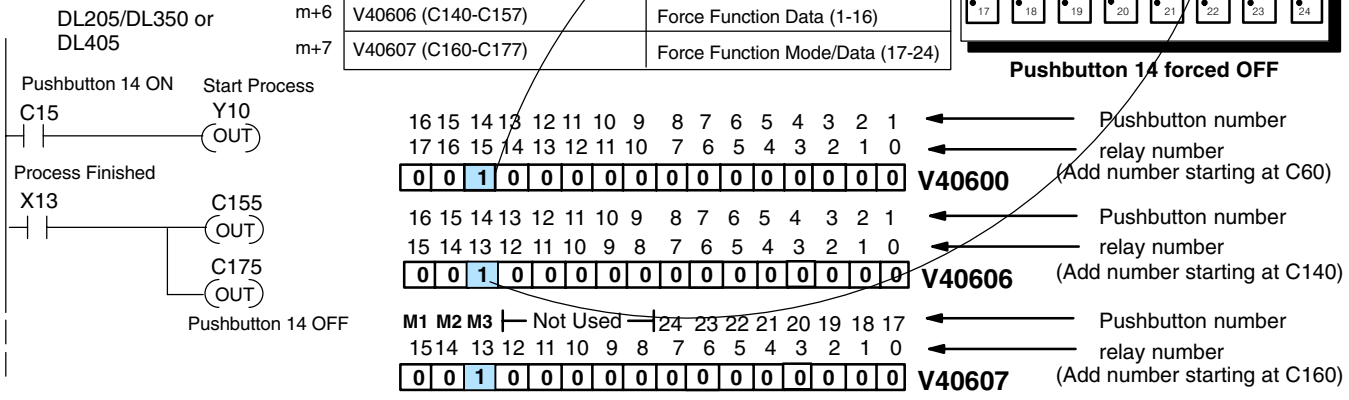
In this example, we have used Mode 3 of the Force Function to force Pushbutton 14 OFF when a process has been completed. Be sure and read Page 32 (if you haven't already done so) to learn the function of all three modes. For the DL205/DL405 example, we have used a base address of V40600. And for the DL305, we have used R16.

DL205/DL350 or DL405 Families

	Mapped Memory Location	Function
m	V40600 (C0-C17)	Pushbuttons 1-16 ON/OFF
m+1	V40601 (C20-C37)	Pushbuttons 17-24 ON/OFF
m+2	V40602 (C40-C57)	LEDs 1-16 flash
m+3	V40603 (C60-C77)	LEDs 17-24 flash
m+4	V40604 (C100-C117)	LEDs 1-16 ON/OFF
m+5	V40605 (C120-C137)	LEDs 17-24 ON/OFF
m+6	V40606 (C140-C157)	Force Function Data (1-16)
m+7	V40607 (C160-C177)	Force Function Mode/Data (17-24)

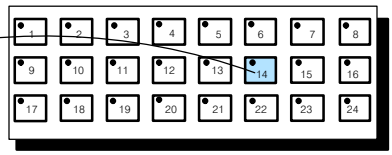


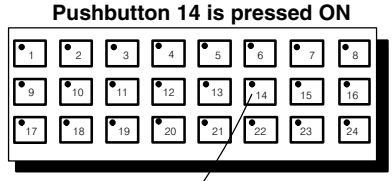


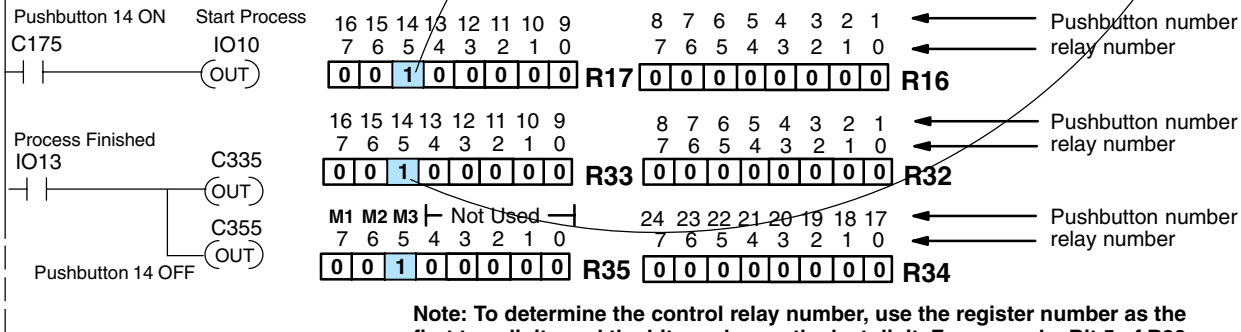


DL330/340 Family

	Example Memory Locations	Function
m	R16/R17 (C160 to C177)	Pushbuttons 1-16 ON/OFF
m+1	R20/R21 (C200 to C217)	Pushbuttons 17-24 ON/OFF
m+2	R22/R23 (C220 to C237)	LEDs 1-16 flash
m+3	R24/R25 (C240 to C257)	LEDs 17-24 flash
m+4	R26/R27 (C260 to C277)	LEDs 1-16 ON/OFF
m+5	R30/R31 (C300 to C317)	LEDs 17-24 ON/OFF
m+6	R32/R33 (C320 to C337)	Force Function Data (1-16)
m+7	R34/R35 (C340 to C357)	Force Function Mode/Data (17-24)





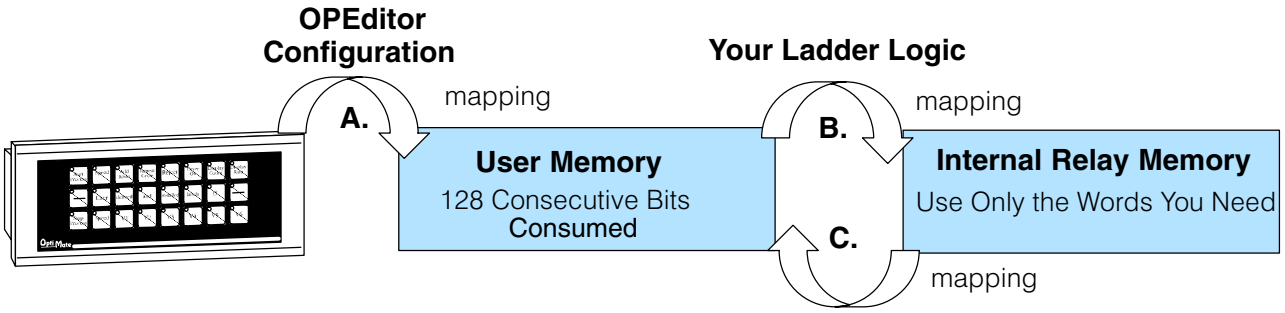


Note: To determine the control relay number, use the register number as the first two digits and the bit number as the last digit. For example, Bit 5 of R33 is referenced as C335.

DL205/DL305/DL405 (Using Selected Functions)

Using the Remapping Process

We briefly discussed the "remapping" process back on Page 23 as a method that allows you to easily manipulate individual bits to take advantage of the panels several functions. All the functions are bit-controlled. By using this method, we only consume the number of relays we actually need for the functions we select.

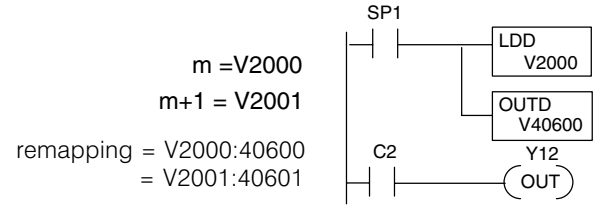


Using the remapping method, when you first indicate a base register address with your OPEditor software and download it to the panel, the panel configuration will automatically consume 128 consecutive memory bits in PLC User Memory. This is indicated by the arrow A. But since User Memory doesn't give you bit control, you need to remap the User Memory with Internal Relay Memory. By remapping between User Memory and Internal Relay Memory, you only consume the Relay Memory you need. You have two directions in which you can have your ladder logic do the remapping between User Memory and Internal Relay Memory:

1. For using the Pushbutton Status to control outputs, you will want to write ladder logic that maps User Memory to Internal Relay Memory (arrow B). This affects the User Memory in the **m** and **m+1** locations.
2. For controlling all other functions of the panel, you will want to write your ladder logic to map Internal Relay Memory to User Memory (arrow C). This affects the User Memory in locations **m+2** through **m+7**.

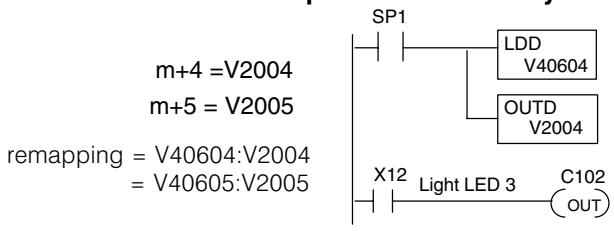
Let's look at two examples of remapping accomplished with ladder logic that demonstrate the two types of remapping that can be used with this technique. We will assume here that V2000 was used as the base register address:

Example of User Memory being mapped to Internal Relay Memory



Here we are using SP1 to map V2000/V2001 to V40600/V40601. This consumes 32 relay bits, 24 of which are tied to the 24 pushbuttons of the panel. By pressing Pushbutton 3, you affect the status of the third relay in V40600 which is C2. In turn, C2 will control output Y12.

Example of Internal Relay Memory being mapped to User Memory

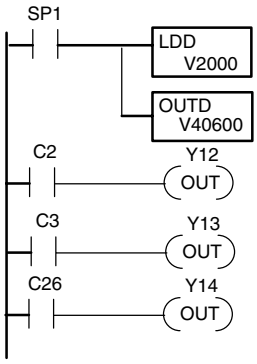


Here we are using SP1 to map V40604/V40605 to V2004/V2005. This consumes 32 relay bits, 24 of which are tied to the 24 LEDs of the panel. When a relay is ON, its corresponding LED is ON. By turning ON X12 with our ladder logic, we can thus turn on the LED corresponding to C102. C102 is bit 2 of the V40604 word and is tied to LED3 through the mapping process. See your PLC User Manual for relay number assignments

Using Pushbutton Status Via Ladder Logic

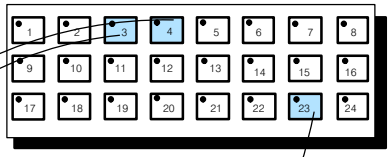
In this example, we are going to remap user memory to internal relay memory in order to use the the pushbutton status to control outputs. The internal relays of the DL205 and DL405 families start at V40600 and the internal relays of the DL305 family start at R16. In the examples below, we have chosen V2000 as the base address for either a DL205 or DL405 and then used SP1 (always ON relay) in our ladder logic to map it to V40600. We have chosen R400 as our base address for the DL305 and then used normally closed C374 in our ladder logic to map it to R16. Using SP1 and normally closed C374, the remapping is done every scan, otherwise m and m+1 would not be updated. In the examples below, our ladder logic is interacting with Pushbuttons 3, 4 and 23.

DL205/DL350 or DL405

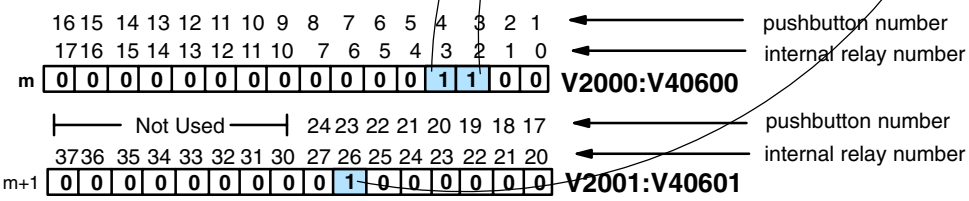


DL205/DL350 or DL405 Families

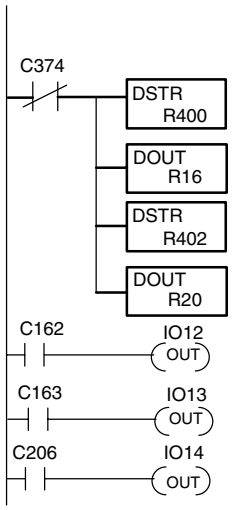
	Mapped Memory Location	Function
m	V2000:V40600	Pushbuttons 1-16 ON/OFF
m+1	V2001:V40601	Pushbuttons 17-24 ON/OFF
m+2	V2002	LEDs 1-16 flash
m+3	V2003	LEDs 17-24 flash
m+4	V2004	LEDs 1-16 ON/OFF
m+5	V2005	LEDs 17-24 ON/OFF
m+6	V2006	Force Function Data (1-16)
m+7	V2007	Force Function Mode/Data (17-24)



Syntax shown in the form of of V2000:V40600 refers to two memory locations that have been mapped together.

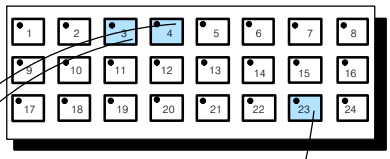


DL330/340

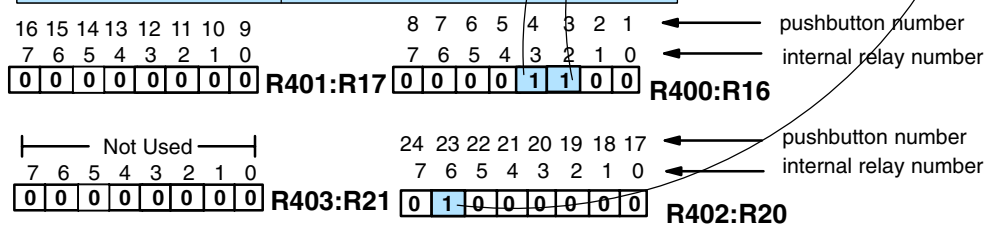


DL330/340 Family

	Example Memory Locations	Function
m	R400/R401:R16/R17	Pushbuttons 1-16 ON/OFF
m+1	R402/R403:R20/R21	Pushbuttons 17-24 ON/OFF
m+2	R404/R405	LEDs 1-16 flash
m+3	R406/R407	LEDs 17-24 flash
m+4	R410/R411	LEDs 1-16 ON/OFF
m+5	R412/R413	LEDs 17-24 ON/OFF
m+6	R414/R415	Force Function Data (1-16)
m+7	R416/R417	Force Function Mode/Data (17-24)



Syntax shown in the form of of R400/R401:R16/R17 refers to two consecutive memory registers mapped to two other consecutive memory registers.

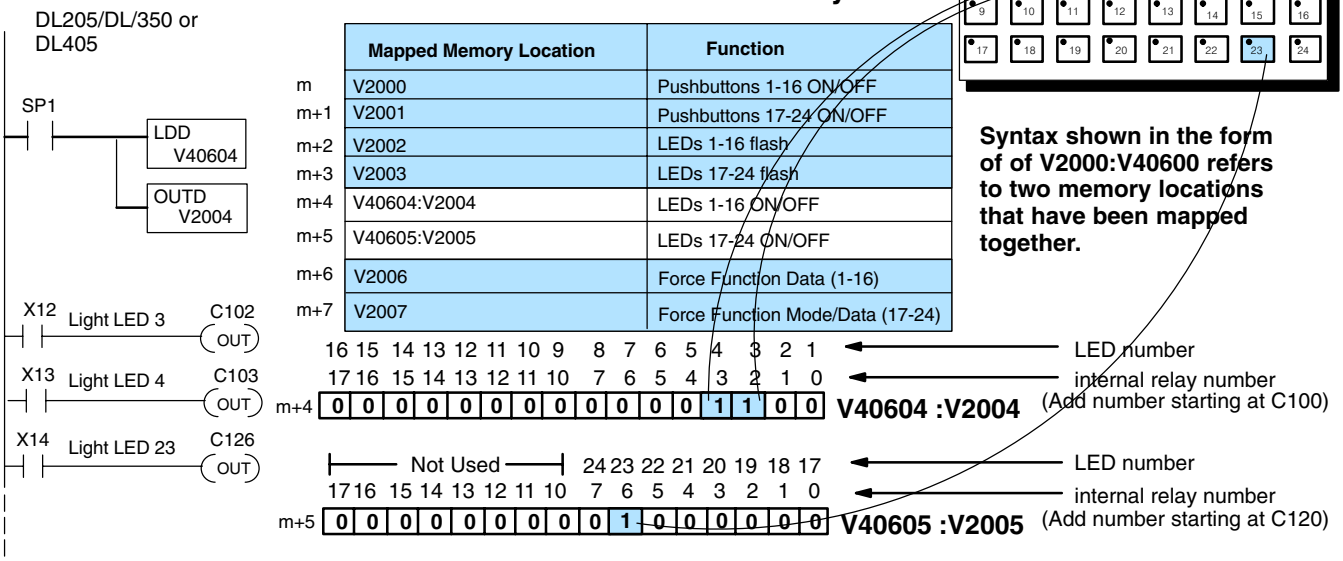


Note: To determine the control relay number, use the register number as the first two digits and the bit number as the last digit. For example, Bit 3 of R16 is referenced as C163.

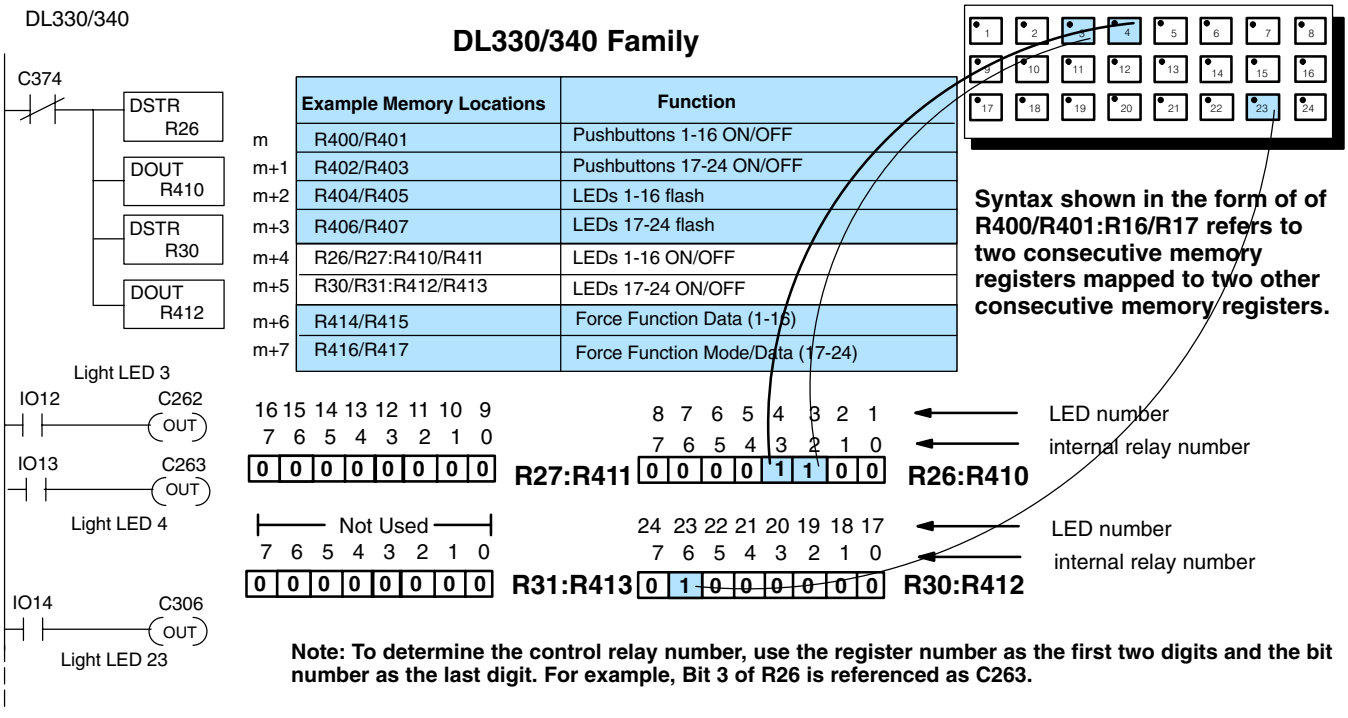
Controlling LEDs Separately

By default, the LED simply shows the state of the pushbutton–ON or OFF. If a pushbutton is configured for momentary operation, there are two options available for the LED. It can show the state of the pushbutton or it can be controlled independently by enabling the **LED Separation** feature. When you have enabled the LED Separation feature, the ON/OFF state of the LED is controlled only by the status of the bits in **m+4** and **m+5**. These bits can be manipulated via your ladder logic. In the examples below we have remapped the user memory to control relay memory to control LEDs 3, 4 and 23. **Remember: Any pushbutton configured for maintained (alternate action) will ignore the bits in these two words. Independent control of the LEDs can only be accomplished if you have Enabled LED Separation during your initial configuration. (See Page 19, Step 9.)**

DL205/DL350 or DL405 Family



DL330/340 Family



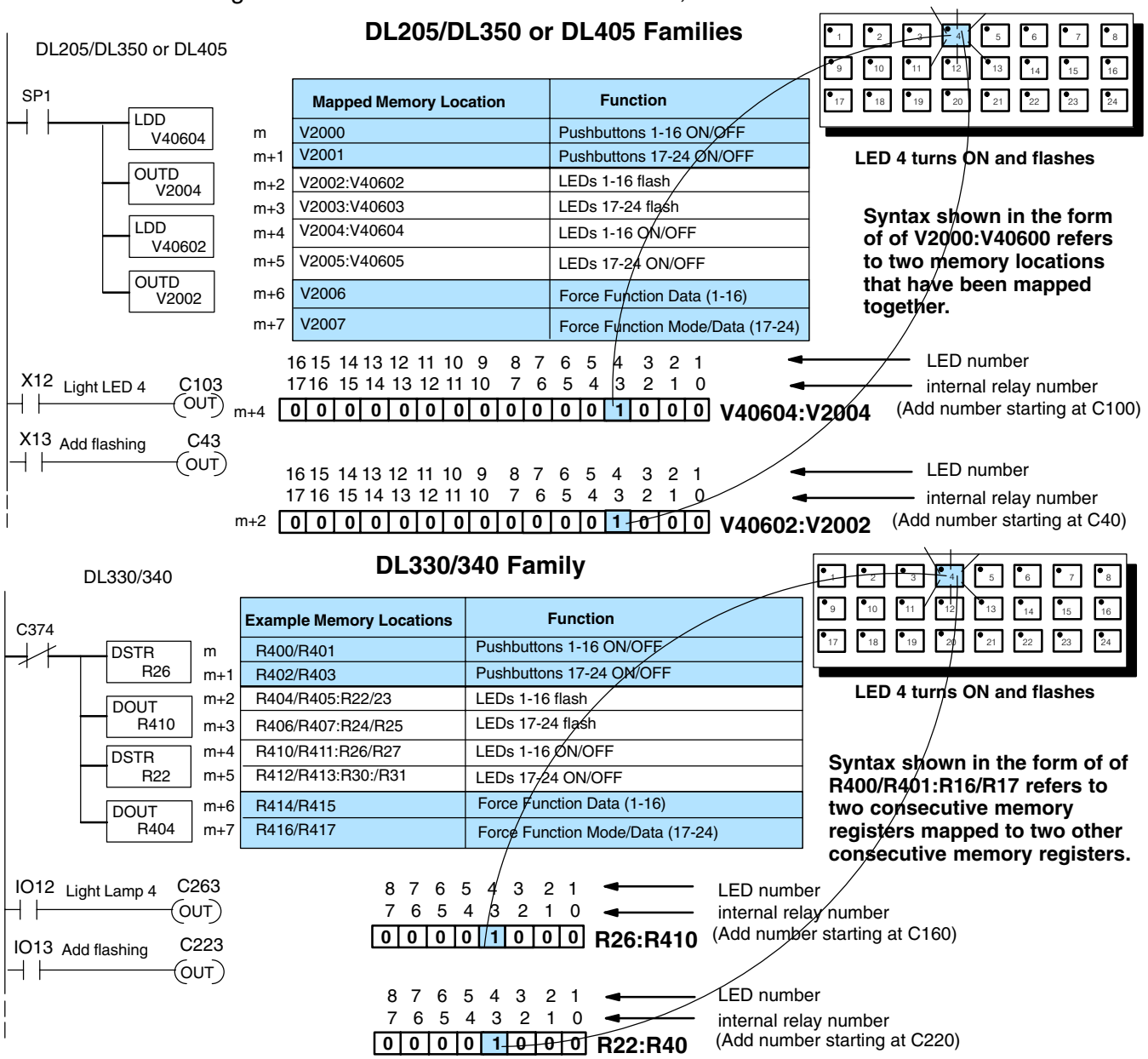
Note: To determine the control relay number, use the register number as the first two digits and the bit number as the last digit. For example, Bit 3 of R26 is referenced as C263.

Adding Flashing

If you plan to use this feature with one or more pushbuttons, there are three things you must always remember during configuration:

1. Flashing is only available for those buttons that have been configured as **Momentary**.
2. **LED Separation** must be Enabled.
3. The **Flash Option** must be Enabled.

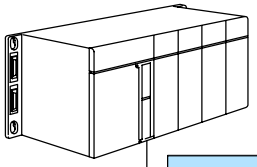
The Flashing Option is triggered through your ladder logic. On the previous page, we showed you how to turn ON an LED, this example shows you how to add flashing to an LED that has been turned ON. The flashing feature is controlled by the status of the bits in the **m+2** and **m+3** memory areas. We have mapped the user memory in these locations to internal relay memory. In the example below, we have begun our mapping at V2000 during the initial configuration. We are turning ON LED4 and then making it flash. Bit 3 of **m+4** turns the LED ON, and bit 3 of **m+2** makes it flash.



Force Function Registers

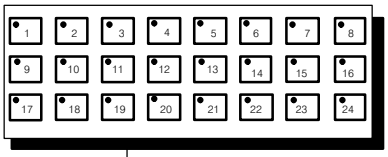
The OP-1224 has the capability to "force" a pushbutton ON or OFF through your ladder logic. If you plan to use this function, you must enable the force option during configuration. (See Page 20.)

NOTE: The Force Function will only work for those pushbuttons that you have configured as "maintained" (alternate action). It will not work for momentary pushbuttons.



Automation

Mapping Assignments



Mapped Memory Location	Function
m (such as V40600:V2000)	Pushbuttons 1-16 ON/OFF
m+1 (such as V40601:V2001)	Pushbuttons 17-24 ON/OFF
m+2 (such as V40602:V2002)	LEDs 1-16 flash
m+3 (such as V40603:V2003)	LEDs 17-24 flash
m+4 (such as V40604:V2004)	LEDs 1-16 ON/OFF
m+5 (such as V40605:V2005)	LEDs 17-24 ON/OFF
m+6 (such as V40606:V2006)	Force Function Data (1-16)
m+7 (such as V40607:V2007)	Force Function Mode/Data (17-24)

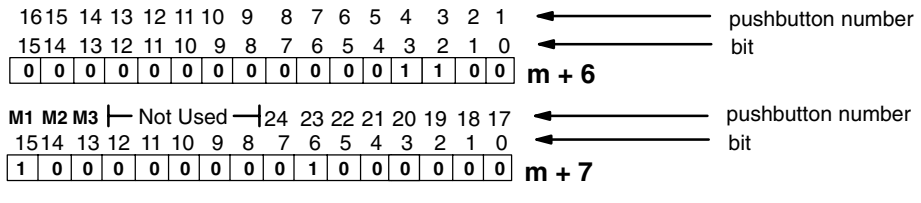
How the Memory is Used–Looking at the above memory map**, m+6** stores the forcing data for Pushbuttons 1-16 and **m+7** stores forcing data for Pushbuttons 17-24. There are three modes of the force function. These modes are controlled by the most significant bits of m+7.

Mode 1 (M1)– This forces all of the Pushbuttons to reflect the status stored in m+6 and m+7. For example, the data shown below would force Pushbuttons 3, 4 and 23 to ON and all the others would be forced OFF. Notice that bit 15 of m +7 is set to 1 for this mode. M2 and M3 are set to 0's.

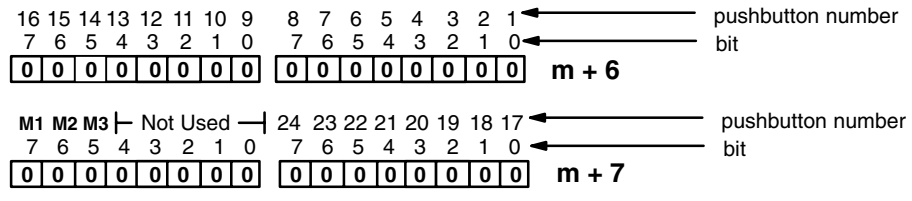
Mode 2 (M2)– This forces ON only those Pushbuttons matching the bits set in registers m+6 and m+7. The bits not set do not affect the status of the Pushbuttons. You would set M2to 1 while M1 and M3 are set to 0.

Mode 3 (M3)– This forces OFF only those Pushbuttons matching the bits set in registers m+6 and m+7. The bits not set do not affect the status of the Pushbuttons. You would set M3 to 1 while M1 amd M2 are set to 0.

Force Function Registers for DL205/DL350/ DL405



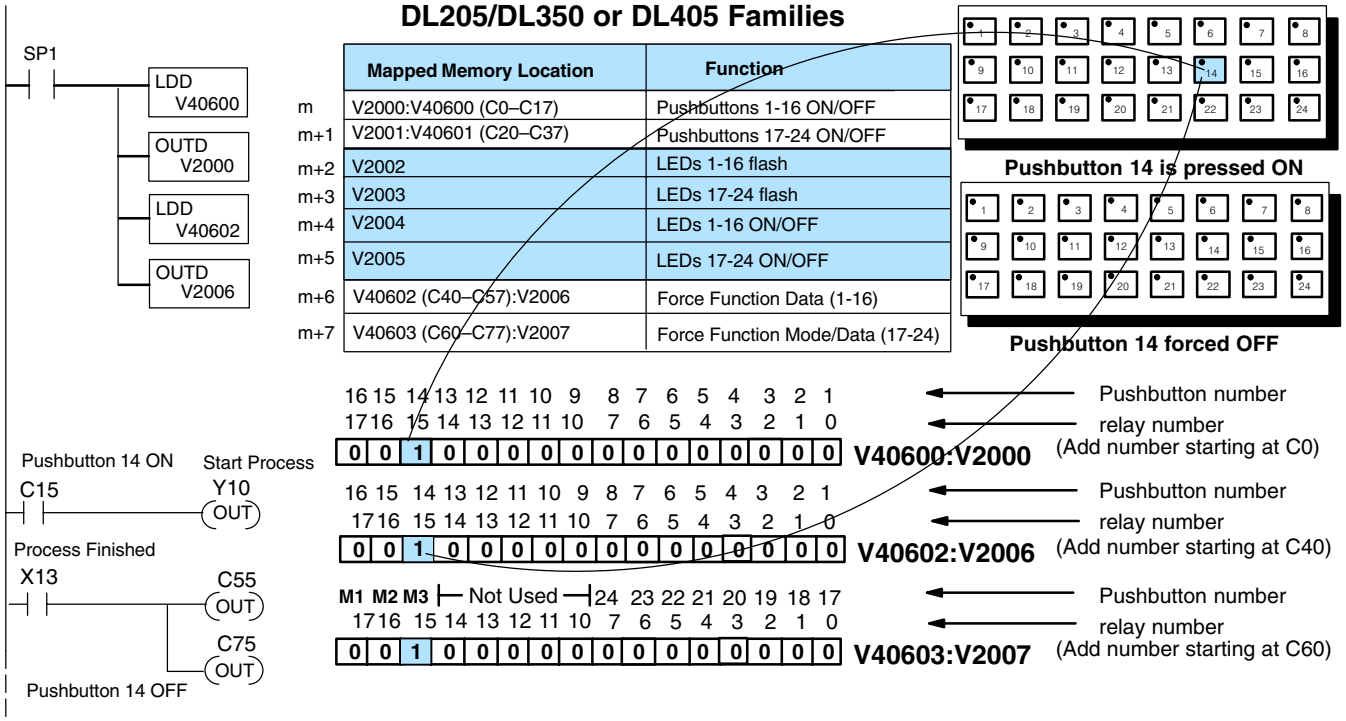
Force Function Registers for DL330/340



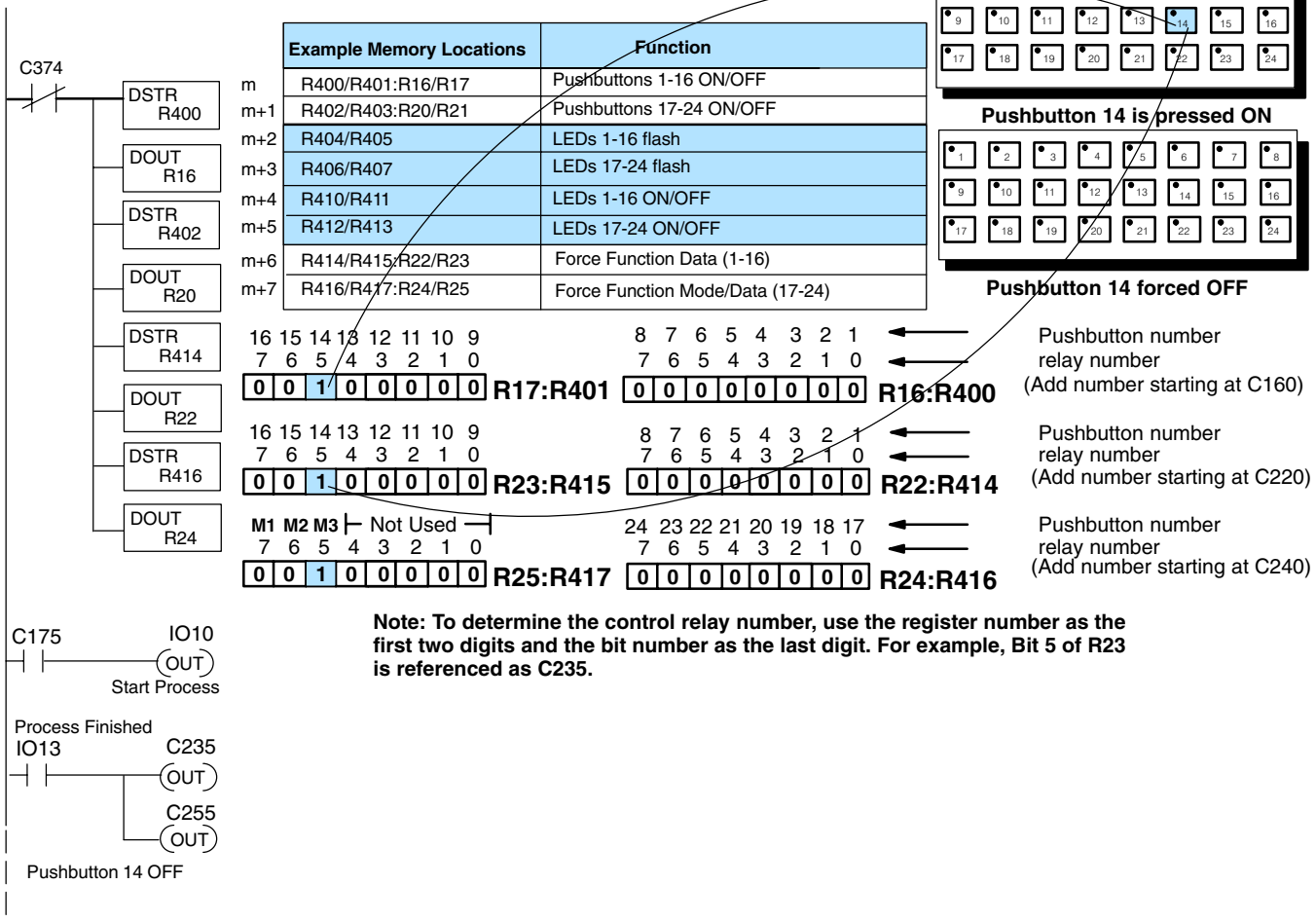
Think of forcing as a one-shot process. That is, once you have set the mode in m+7, the bit patterns in m and m+1 are changed (according to the mode selected), and then, all of the bits in m+6 and m+7 are set to zero. What this means is that all pushbuttons return to normal manual operation after the forcing is completed.

Forcing Pushbuttons ON or OFF

In this example, we have used Mode 3 of the Force Function to force Pushbutton 14 OFF when a process has been completed. Be sure and read Page 37 (if you haven't already done so) to learn the function of all three modes. For the DL205/DL405 example, we have used a base address of V40600. And for the DL305, we have used R16.



DL330/340 Family

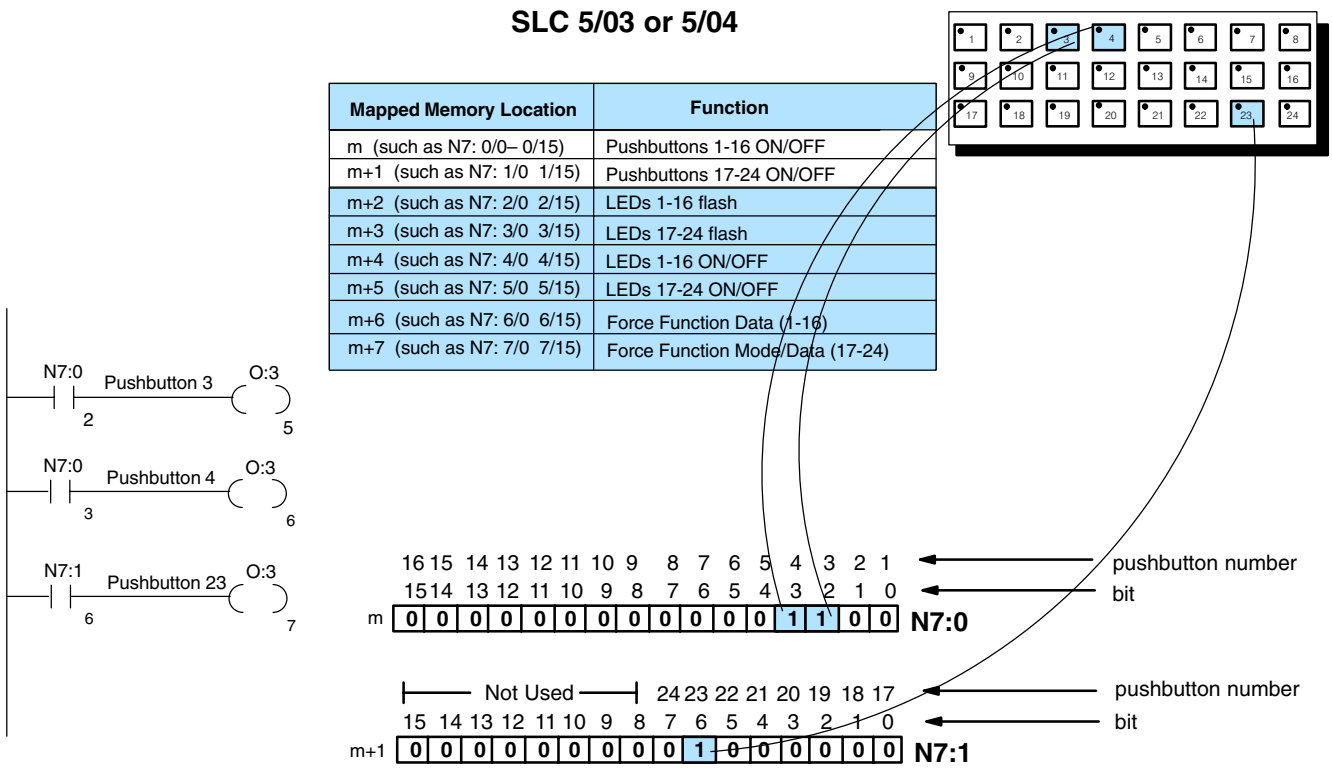


Note: To determine the control relay number, use the register number as the first two digits and the bit number as the last digit. For example, Bit 5 of R23 is referenced as C235.

Using the OP-1224 with an Allen-Bradley PLC

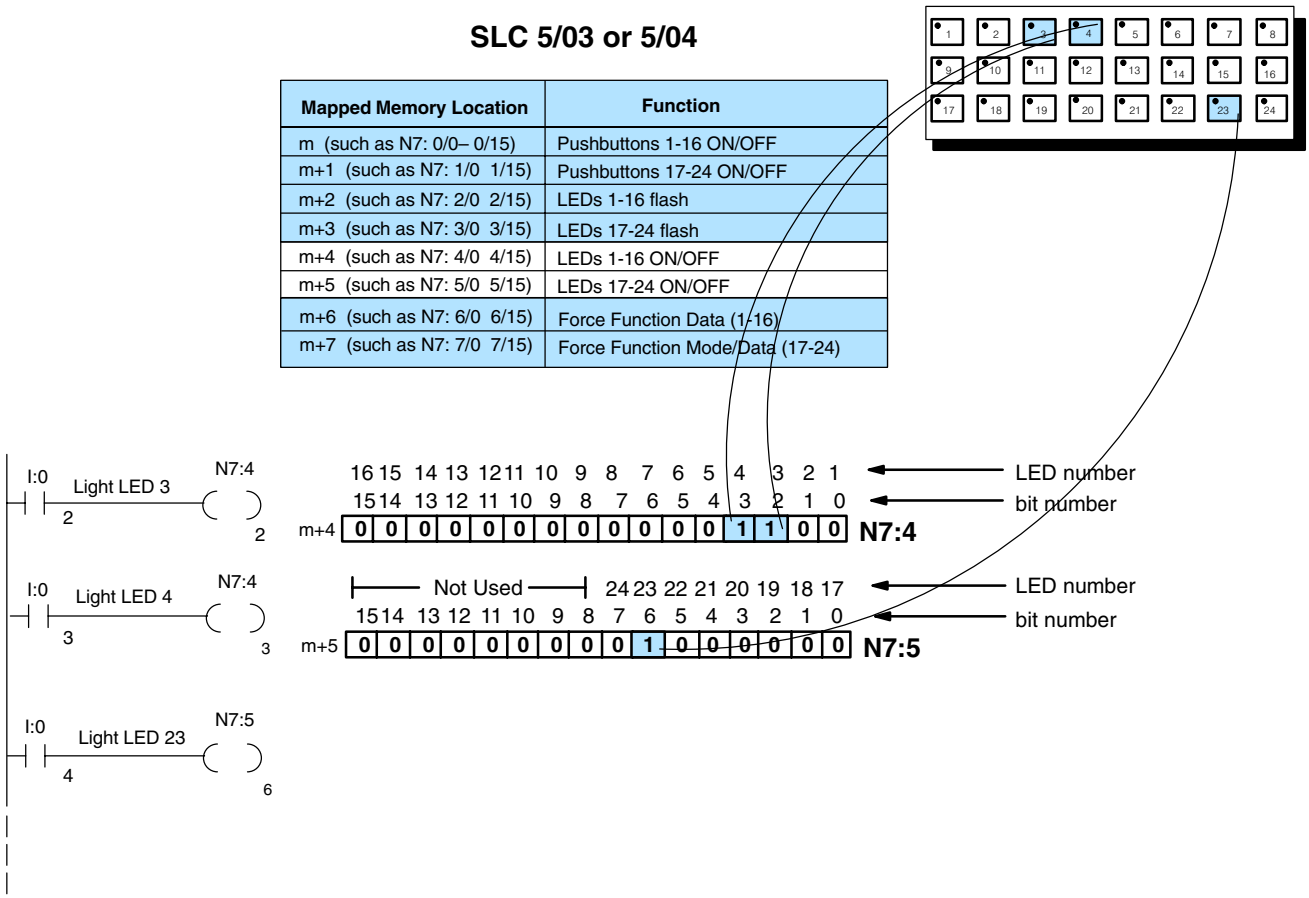
Using the Pushbutton Status

As mentioned earlier, you can map integer type files for the Allen-Bradley PLC when using it with the OP1224. In the example below, we have mapped integer file registers starting at base address N7:0. We are using Pushbutton 3 to control Output 5 (O:3/5). We are using Pushbutton 4 to control Output 6 (O:3/6). And we are using Pushbutton 23 to control Output 7 (O:3/7).



Controlling LEDs Separately

You can control the LEDs separately from the status of the pushbuttons. In the example below, we are using Allen-Bradley input type files (I:0/2, I:0/3 and I:0/4) to trigger the ON/OFF of LED3, LED4 and LED5. **Remember: Any pushbutton configured for maintained (alternate action) will ignore the bits in these two words. Independent control of the LEDs can only be accomplished if you have Enabled LED Separation during your initial configuration. (See Page 19, Step 9.)**

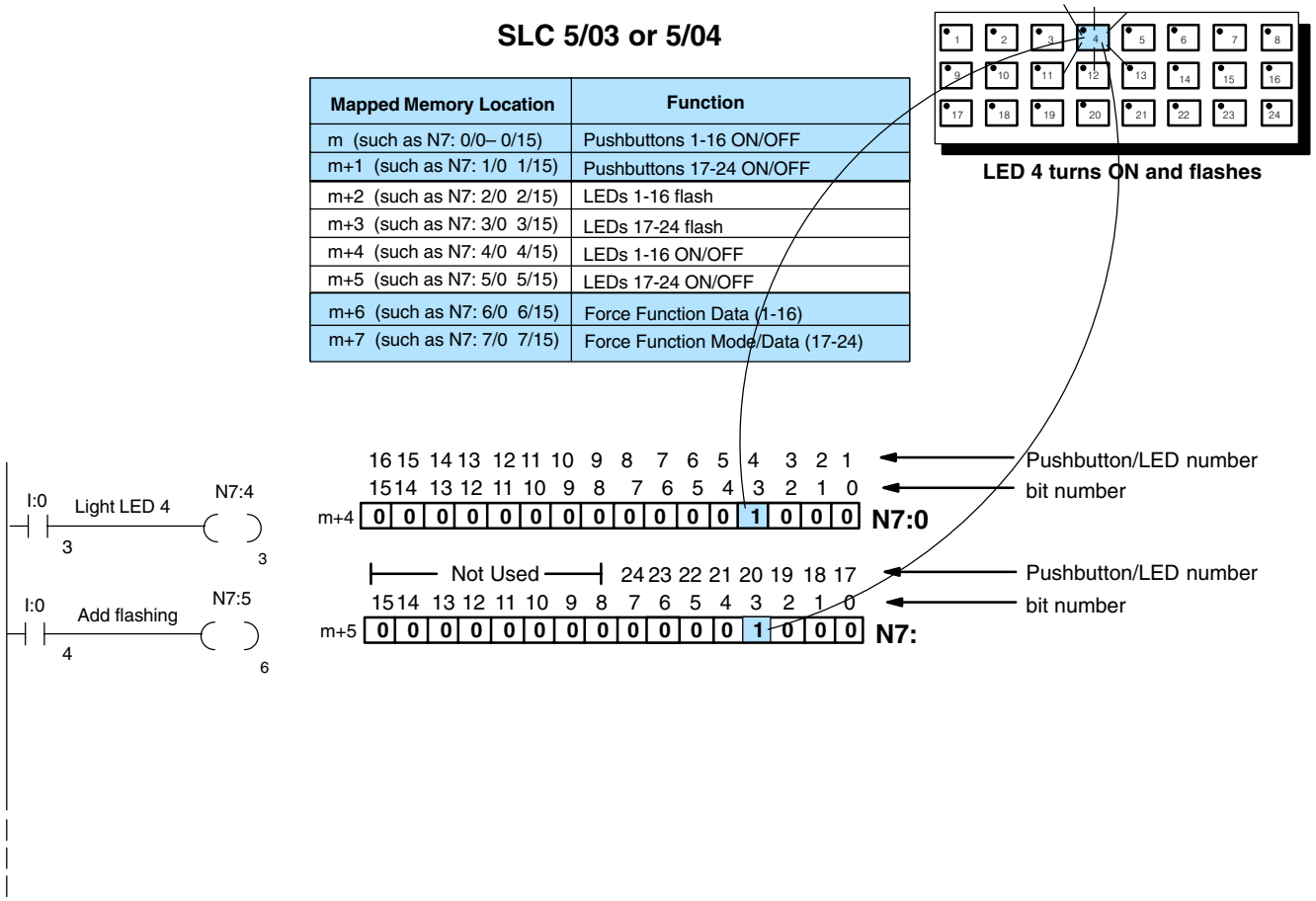


Adding Flashing

To draw extra attention to an LED that is lit, you can add flashing. If you plan to use this feature with one or more pushbuttons, there are three things you must always remember during configuration:

1. Flashing is only available for those buttons that have been configured as **Momentary**.
2. **LED Separation** must be Enabled.
3. The **Flash Option** must be Enabled.

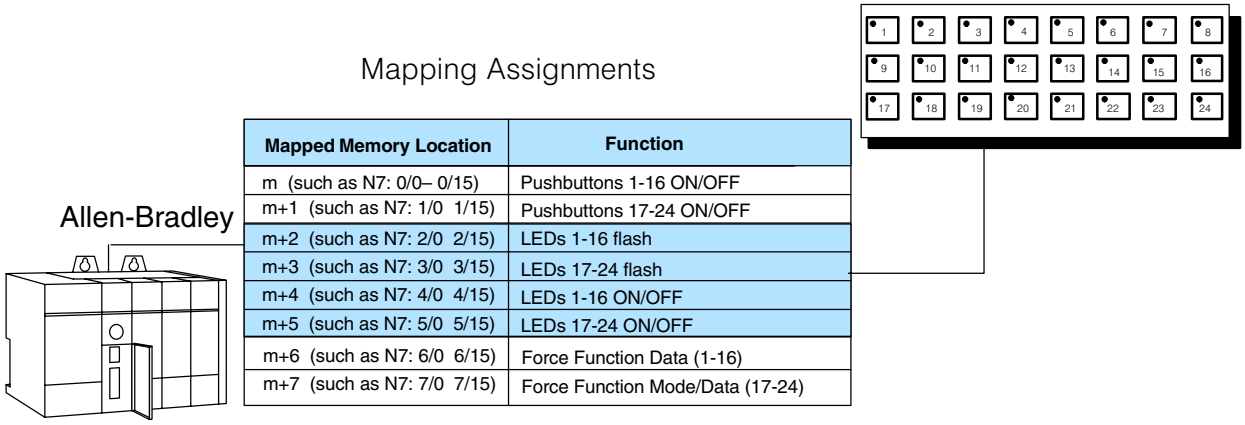
The Flashing Option is triggered through your ladder logic. On the previous page, we showed you how to turn ON an LED, this example shows you how to add flashing to an LED that has been turned ON. The flashing feature is controlled by the status of the bits in the **m+2** and **m+3** memory areas. We have mapped the user memory in these locations to internal relay memory. In the example below, we have begun our mapping at **N7:0** during the initial configuration. We are turning ON LED4 and then making it flash. Bit 3 of **m+4** turns the LED ON, and bit 3 of **m+2** makes it flash. In the example below, we are using input type files (I:0/3 and I:0/4). I:0/3 turns ON LED 4 and I:0/4 turns ON the flashing feature for that particular LED.



Force Function Registers

The OP-1224 has the capability to "force" a pushbutton ON or OFF through your ladder logic. If you plan to use this function, you must enable the force option during configuration. (See Page 24.)

NOTE: The Force Function will only work for those pushbuttons that you have configured as "maintained" (alternate action). It will not work for momentary pushbuttons.



Mapped Memory Location	Function
m (such as N7: 0/0– 0/15)	Pushbuttons 1-16 ON/OFF
m+1 (such as N7: 1/0 1/15)	Pushbuttons 17-24 ON/OFF
m+2 (such as N7: 2/0 2/15)	LEDs 1-16 flash
m+3 (such as N7: 3/0 3/15)	LEDs 17-24 flash
m+4 (such as N7: 4/0 4/15)	LEDs 1-16 ON/OFF
m+5 (such as N7: 5/0 5/15)	LEDs 17-24 ON/OFF
m+6 (such as N7: 6/0 6/15)	Force Function Data (1-16)
m+7 (such as N7: 7/0 7/15)	Force Function Mode/Data (17-24)

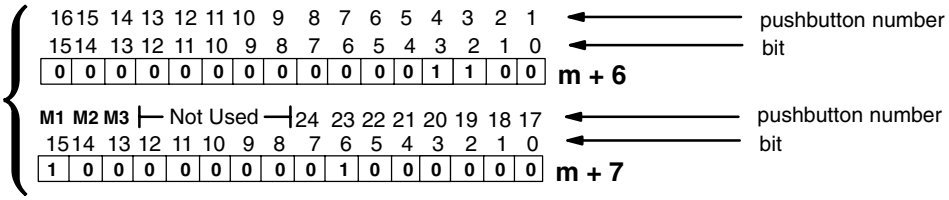
How the Memory is Used–Looking at the above memory map**, m+6** stores the forcing data for Pushbuttons 1-16 and **m+7** stores forcing data for Pushbuttons 17-24. There are three modes of the force function. These modes are controlled by the most significant bits of m+7.

Mode 1 (M1)– This forces all of the Pushbuttons to reflect the status stored in m+6 and m+7. For example, the data shown below would force Pushbuttons 3, 4 and 23 to ON and all the others would be forced OFF. Notice that bit 15 of m +7 is set to 1 for this mode. M2 and M3 are set to 0's.

Mode 2 (M2)– This forces ON only those Pushbuttons matching the bits set in registers m+6 and m+7. The bits not set do not affect the status of the Pushbuttons. You would set M2 to 1 while M1 and M3 are set to 0.

Mode 3 (M3)– This forces OFF only those Pushbuttons matching the bits set in registers m+6 and m+7. The bits not set do not affect the status of the Pushbuttons. You would set M3 to 1 while M1 amd M2 are set to 0.

Force Function Registers



Think of forcing as a one-shot process. That is, once you have set the mode in m+7, the bit patterns in m and m+1 are changed (according to the mode selected), and then, all of the bits in m+6 and m+7 are set to zero. What this means is that all pushbuttons return to normal manual operation after the forcing is completed.

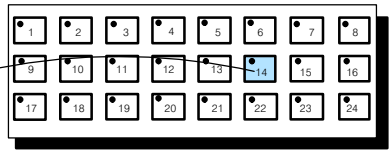
Forcing Pushbuttons ON or OFF

You can also use Allen-Bradley integer file types to force pushbuttons ON or OFF. Here we have chosen N7:0 as our base address for the mapping in the PLC. In this example, we are using Pushbutton 14 to start a process, and then forcing the pushbutton OFF when the process is completed. N7:0 holds the bit that reflects the status of Pushbutton 14. N7:6 and part of N7:7 hold the data that the force feature uses when executing one of the three selectable modes (M1, M2 or M3). These modes are selectable in the upper three bits of the mapped memory area m+7. In the example, below the mode is embedded in N7:7.

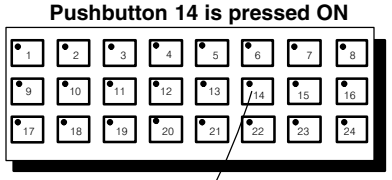
Here we have chosen to use Mode 3. Mode 3 looks at N7:6, and whichever bits are set to 1 , the corresponding pushbuttons are forced OFF. Since we set the 13th bit of N7:7 (corresponding to LED14), the OP-1224 will force LED14 OFF.

SLC 5/03 or 5/04

Mapped Memory Location	Function
m (such as N7: 0/0– 0/15)	Pushbuttons 1-16 ON/OFF
m+1 (such as N7: 1/0 1/15)	Pushbuttons 17-24 ON/OFF
m+2 (such as N7: 2/0 2/15)	LEDs 1-16 flash
m+3 (such as N7: 3/0 3/15)	LEDs 17-24 flash
m+4 (such as N7: 4/0 4/15)	LEDs 1-16 ON/OFF
m+5 (such as N7: 5/0 5/15)	LEDs 17-24 ON/OFF
m+6 (such as N7: 6/0 6/15)	Force Function Data (1-16)
m+7 (such as N7: 7/0 7/15)	Force Function Mode/Data (17-24)



Pushbutton 14 is pressed ON



Pushbutton 14 forced OFF