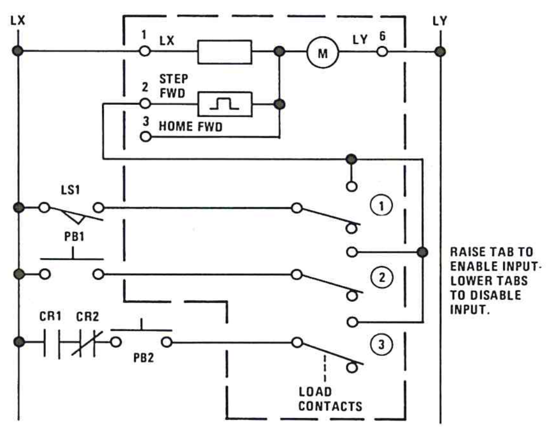
MTA QUIZ ANSWERS

1. *Answer:* When required, the MT interrupter contact creates a pulse by opening at the top of the step solenoid energizing stroke. The MTA contains circuitry to automatically create the same type pulse. Also, with the availability of a separate *home* input, the interrupter contact function is not required on the MTA.

2. *Answer:* Energizing input advances the MTA Programmer.

3. *Answer:* Move the load connection to the other contact (N.O. to N.C., and vice versa). Reverse the program tab arrangement (Raise those which are down; lower those which are up).

4. *Answer:* The indicated logic, necessary to advance from step positon 3, has been added.



5. *Answer:* For application Step positions 1-20, MTA 1 is in control at application Step position 21, the input path switches to MTA #2; MTA #1 signaled to Home Fwd; MTA #2 increments to step position 1, which represents application step position 21. MTA #1 parks in its step position #22.

Application Step position 22-40 (MTA #2 step position 2-20), MTA #2 in control of output loads.

The next step input homes MTA #2, and starts the homing of MTA #1. MTA #2 parks in step position 24 while MTA #1 homes to #1 step position, ready to resume cycle.

CHAPTER SUMMARY

1. The MTA Series Sequence Programmer is a device which utilizes a controlled motor rather than a solenoid for stepping functions.

2. The MTA may be ordered to provide bidirectional operation, if desired.

3. Control circuitry *creates* the MTA step input pulse. With the separate *home* inputs, this capability eliminates the requirement for the MT type self stepping or interrupter contact.

4. Tapswitch usage and programming requirements are similar for both the MT and MTA Series Programmer.

5. Step positions may be increased by wiring two units in a *flip-flop* configuration.

5005-023

training manual

Sequence Controls

AUGUST 1978

Eagle Signal Division
GULF + WESTERN MANUFACTURING COMPANY
INDUSTRIAL CONTROLS / 736 Federal Street, Davenport, Iowa 52803

MTA PROGRAMMER QUIZ

1. Why doesn't the MTA Programmer require an interrupter contact, while the MT Stepswitch does require one?

1. ____

2. Does the MTA advance on the *energizing* or *deenergizing* of the input circuitry?

2. ____

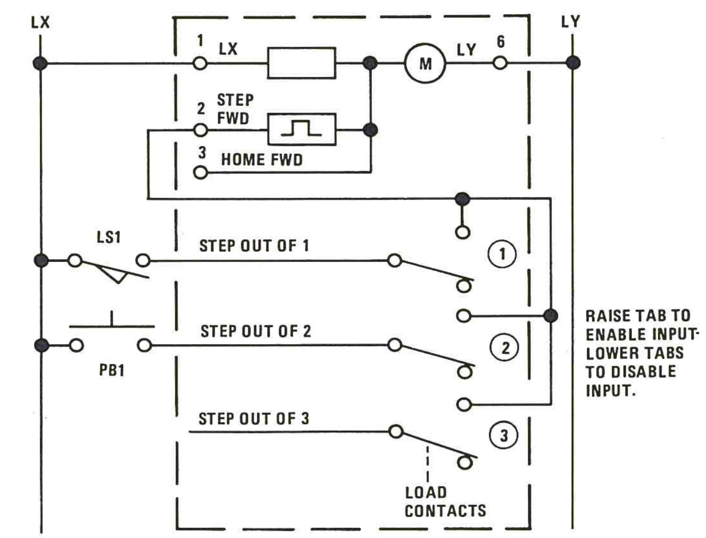
3. When designing the MTA load program, it is discovered that most of the tabs will be raised to accomplish the desired result. To save time, we wish to reverse this requirement, raising only a few tabs. How can we consider doing this?

3. ____

4. The following wiring diagram represents the circuit condition to advance the MTA Programmer. Inputs are isolated through load contacts. Draw the circuit condition to advance out of step position 3, only when the following conditions to advance are satisfied:

Relay CR1 actuated, and Relay CR2 not actuated, and switch PB2 depressed.

4. Use the following sketch for your answer.



5. Turn to Figure 3-17, *Step Position Expansion*. Write a short analysis of this circuit, defining the action of each MTA, especially in the step positions where control is being transferred from one Programmer to the other Programmer.

5. ____

UNIT #1		
SWITCH	1A	U, U + 1, U + 2, ... TO POSITION 24 MINUS 1. (24 DOWN)
	2A	U, U + 2, U + 3, ... THROUGH POSITION 24. (U + 1 DOWN)
	3A	U, U + 1
NOTE: U = FIRST UNUSED POSITION HIGHEST USABLE STEP = 20 U + 1 = RESET POSITION		

UNIT #2		
SWITCH	1B	U, U + 1, U + 2, ... TO POSITION 24 MINUS 1. (24 DOWN)
	2B	U, U + 1
	3B	POSITION 24 ONLY
NOTE: U = FIRST UNUSED POSITION POSITION 24 = RESET POSITION HIGHTEST USABLE STEP = 21		

Figure 3 - 18

Summary of control switch programming (Positions for tab *up*).

LOAD PROGRAM FOR EXPANDED STEP POSITIONS

The circuit configuration shown in Figure 3 - 17 expands step positions but not circuit capacity. Since two Programmers share step capacity, two sets of load contacts share the actuation of a particular load, transferring control when each Programmer transfers its control to the other unit.

Program the switch to be open in all positions which represent an idle condition, with control being exercised by the other Programmer. Active positions are programmed as desired.

If, by accident, tabs are programmed to close unused contact positions rather than open them, the load will be energized and the true load program represented by the other switch condition will be overridden. Obviously, this is to be avoided.

Connect each pair of load control contacts in parallel as shown in Figure 3-19. Although any combination of switch numbers are possible, they should be kept as matching pairs. (i.e., MTA #1 switch 5 should be wired with MTA #2 switch 5.)

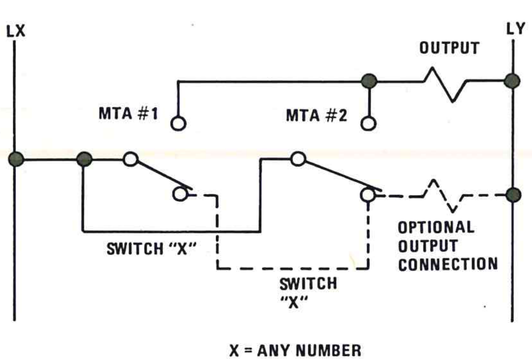


Figure 3 - 19

TABLE OF CONTENTS

The maximum usable step capacity for this configuration is 41 steps. Since 3 circuits have been taken for Programmer control, 27 circuits are available for loads with a 30 circuit unit. (Load circuits available = Total load circuits on Unit, minus 3).

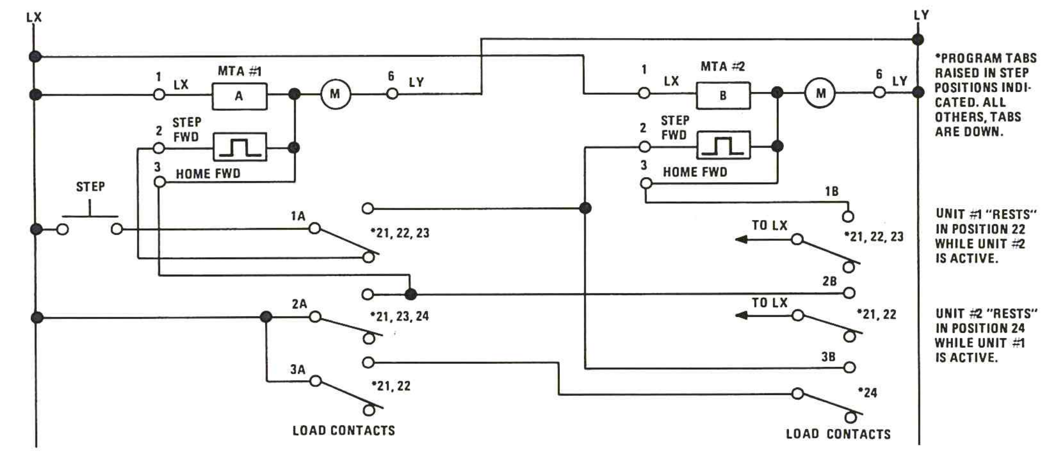


Figure 3 - 17
Step Position Expansion

The control switches serve the following purposes:

1A This contact directs the step impulse to either Unit #1 or Unit #2. Program tab is up in the *first* position to be unused. The first unused position is designated *U*. Tabs are *up* also at U+1; U+2 .... etc. Step position 24 is to be *down*.

2A This contact increments Unit #1 into a park position. It also increments Unit #1 *home* at the conclusion of Unit #2 cycle.

Program tab is *up* in Position *U*; *down* in U+1 (park position); *up* at U+2; U+3 etc. Step position 24 is also *up*.

3A This contact, in conjunction with contact 3B, is the *safety valve* circuit to insure the action described for contact 3B.

Program tab is *up* in position *U* and U+1 (park position).

1B This contact *homes* Unit #2.

Program Tab *up* at step position *U* for Unit #2; up at U+1; U+2 ..... etc. Program tab *down* at position 24.

2B This contact *homes* Unit #1 when Unit #2 cycle is completed.

Program *Tab* up at position *U* and U+1. After that, Unit #1 takes over its own home function through switch 2A.

3B This contact, by being closed in the Unit #2 rest position, insures that a pulse input of sufficient length to advance Unit #1 at its final active step position will also advance Unit #2 out of its rest position.

If this circuit did not exist, Unit #2 would have to rely on its advance signal to come from contact 1A. If the step input were not long enough (went away during the transfer of 1A), the Step Forward input of Unit #2 would not feel the signal long enough to advance. Contact 3B eliminates the need for an extra long pulse requirement during the *pass* of control from Unit #1 to Unit #2.

Program this tab to be *up* in the rest position for Unit #2. (Position 24).

In Figure 3 - 14b, notice that the home signal, obtained when the home pushbutton is depressed, will be directed to either terminals 3 or 5, dependent upon step position. Figure 3 - 14b has a potential problem, however, which must be corrected with a slight modification. The problem arises from the fact that programmer momentum could drive the mechanism past the home position. This could actuate the load switch, reversing the programmer rather than stopping it. The solution is to use an additional load switch, as shown in Figure 3 - 15, to *electrically* insure that the programmer will stop in a home position.

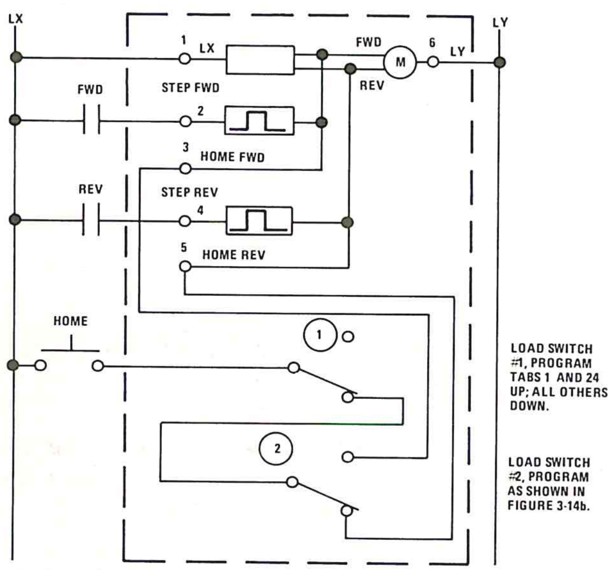


Figure 3 - 15

Also in Figure 3 - 15, if you wish only one position to represent *home*, raise only *that* program tab for load contact 1.

A circuit application may require the programmer to cycle between steps 1 and 24, then *reversing* to step position 1. This step sequence is shown as:

1,2--23, 24, 23, -- 2, 1, 2, --23, 24, 23, -- 2, 1 etc.

A single step input is shown in Figure 3-16 with the load contacts and external relay CR1 providing the logic to direct the input to either the STEP FWD or STEP REV input terminals.

In describing Figure 3-16, we will assume the starting point to be step position #1. External Relay CR1 energizes and CR1-3 closes to seal the power to CR1. CR1-1 closes the step input path to STEP FWD. In step position #2, load contact 2 opens with no circuit effect. No additional circuit efforts are noted through step position #23. In step position 24, load contact 1 opens and CR1 drops out, opening CR1-3 and CR1-1; closing CR1-2. This transfers step input path to STEP REV. After the next step input, the programmer decrements to position 23. Load contact 1 closes but the circuit is not electrically affected. The Programmer continues to decrement through position 2. In step position 1, load contact 2 closes, energizing CR1 and is sealed by CR1-3 through load contact 1. This transfers the step input path to STEP FWD and the programmer is prepared to begin its increment cycle again toward step position 24.

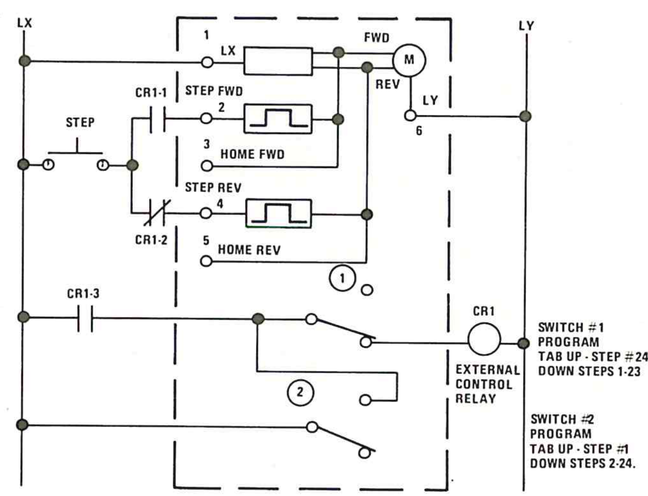


Figure 3 - 16

MTA STEP POSITION EXPANSION

Two MTA Programmers may be interconnected to provide STEP POSITION expansion. Figure 3 - 17 shows the wiring connections for this function. The Program tab configuration is shown for the specific example, but in general will be dependent upon the requirements of the application. In this example, step positions 1-20 are active for each unit. Unit #1 is to rest in position 22 while Unit #2 is active; Unit #2 is to rest in position 24 while Unit #1 is active.

CHAPTER I.

SEQUENTIAL CONTROL BASICS

INTRODUCTION

Industrial control is generally accomplished in one of two forms; Parallel Logic Control or Sequential Logic Control. Parallel Logic Control forms, such as relay logic and DC logic, use control devices that may be actuated at random. It then becomes the task of other elements, such as the wiring of pilot devices, control logic wiring, and lockout logic, to bring the actuation of loads under desired control.

Sequential Logic Control, on the other hand, uses a control element to direct the *sequence* in which certain events will occur. Pilot devices supply input signals to this control device which, unlike most Parallel Logic Control elements, has certain inherent lockout characteristics. Computers, Programmable Controllers and Step Programmers are all forms of Sequence Control. This manual is directed to the Step Programmer form of control.

Figure 1-1 shows the basic concept of sequential control in block diagram form. The number of individual steps in the sequence have been identified, along with the input condition required to terminate a step. This input may be a timer timing out, a contact closing, a value solenoid opening, etc. While a specific step is being executed, pre-determined outputs must be energized. Those remaining must be de-energized. An output *program* of some form will be used to energize and de-energize specific outputs.

Overall, sequential control can be shown as a series of steps, each occuring in a pre-determined sequence to complete the operational cycle.

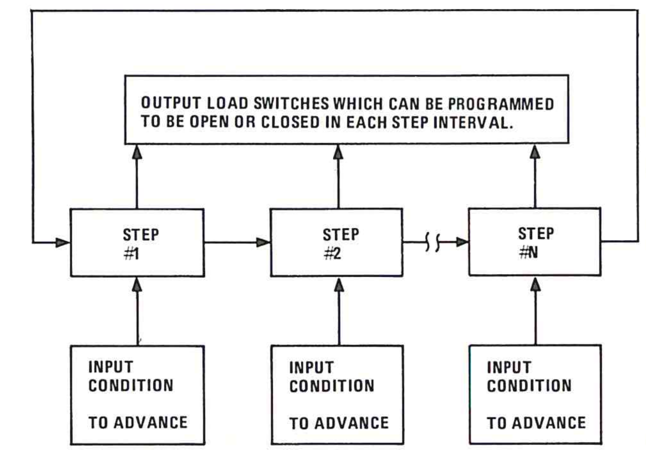


Figure 1 - 1
Sequence Program Block Diagram

Although simple in basic concept, an alternative control concept for sequential applications could result in a more complex control scheme involving interwiring of many relays and other control elements. The Stepswitch or Sequential Programmer becomes the natural control device to be considered for use in sequential applications.

The automatic commercial carwash is a very typical sequential control application. The exact steps required from the moment an auto enters until it departs are pre-determined. Alternate operations, such as hot wax, may be included, dependent upon the presence of the correct input (i.e., extra coins in the slot, etc.). The actuation of the proper combination of outputs and the inputs available to the application are all defined.

To be familiar with the application alone, however, is only part of the requirement necessary to apply sequence control devices. The remainder of the information required is a thorough knowledge of stepswitch construction and operation. The remainder of this manual will be directed to disucssions of construction, operation, programming, and wiring of both general and specific examples of sequential control devices.

When you have concluded the study of this manual, you should be able to design some basic stepswitch circuits on your own. You will then be limited only by your own interests and skills for applying stepswitch applications.

BASIC STEPSWITCH MODEL

Physically, the Stepswitch or Sequence Programmer contains certain similarities to parts of repeat cycle timers and reset counters. In fact, if it were possible to *marry* the *input* mechanism of a counter to the *output* components of a multi-circuit repeat cycle cam timer, the result would be a basic model for a rudimentary, unrefined stepswitch.

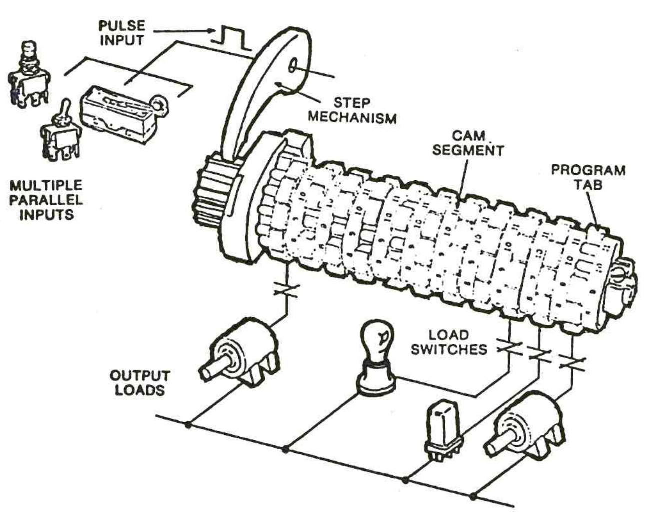


Figure 1 - 2
Basic Model - Sequence Programmer

As shown in Figure 1-2, the basic Sequence Programmer contains a set of individual load contacts or switches, each controlled by a switch actuator. For each step, these switches are forced to be opened or closed, as determined by *program* tabs. As the program drum increments, the various loads are turned on or off as required for each step in the sequence.

The input form is shown as an electrical *pulse* to the step mechanism. For each pulse, the program drum advances one step. Thus the total collection of outputs are controlled individually for each step in the overall cycle; and the cycle advances, as determined by input status.

CONTROL PROGRAM

The main control element of the Stepswitch or Sequence Programmer* is its *program*, contained on a *camshaft* consisting of individual cam *segments*. In some applications, one or more cam segments will be used for stepswitch control, leaving the remainder for control of external circuits.

Figure 1-3 shows a side view of one cam segment which has been removed from the camshaft. By some means, dependent upon the stepswitch design, a load switch will be controlled by an arm or lever which rides upon the cam segment.

The load switch may be single pole-single throw (SPST) or single pole-double throw (SPDT). In either case, external loads or stepswitch control circuits are controlled by the load switch.

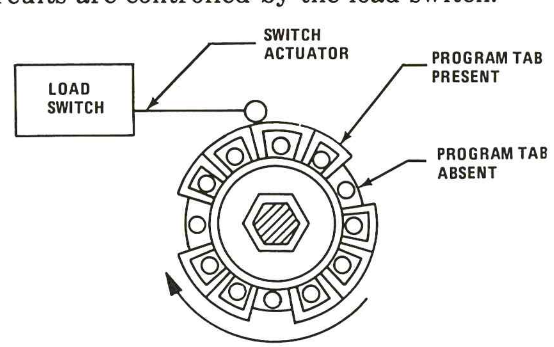


Figure 1 - 3
Program Cam Segment

* The terms Stepswitch, Sequence Programmer, and Programmer are considered to represent sequential control devices.

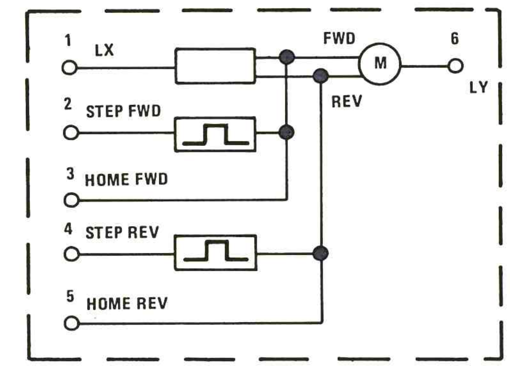


Figure 3 - 12

NOTE: Always consult the detailed circuit drawing when there is any question about the exact operational sequence. Figure 3 - 12 is only for convenience and will not resolve operational questions, or point out certain facts, such as the overriding nature of the HOME REVERSE input.

CIRCUIT EXAMPLES

Figure 3 - 13 shows a basic circuit to control both forward and reverse sequencing of the MTA Programmer.

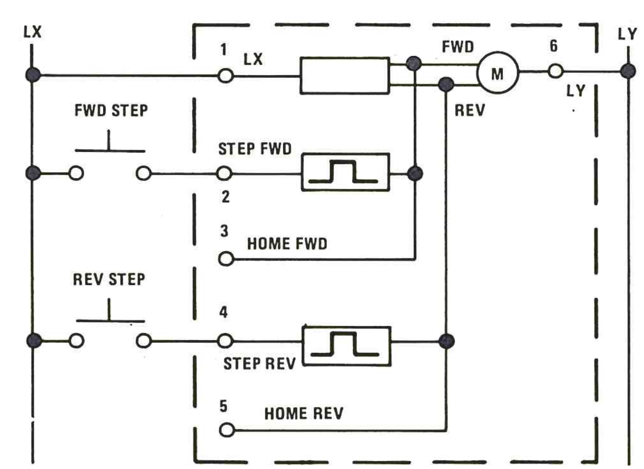


Figure 3 - 13

As expected, closing the input to terminal 2 will step the Programmer forward; closing the input to terminal 4 will step the Programmer in the reverse direction. For simultaneous application of both inputs, REV STEP may dominate FWD STEP, or the Programmer will stall.

Figure 3 - 14b shows a circuit configuration which provides step inputs similar to Figure 3 - 13, with a home signal directed to either HOME FWD or HOME REV, depending upon the step position. In this example, we will home-reverse if the Programmer is between steps 1 - 12 and home-forward if the Programmer is between steps 13 - 24. See Figure 3 - 14a.

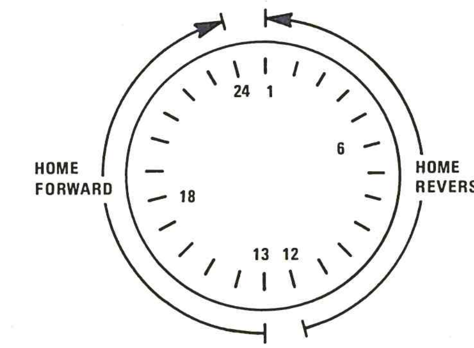


Figure 3 - 14a

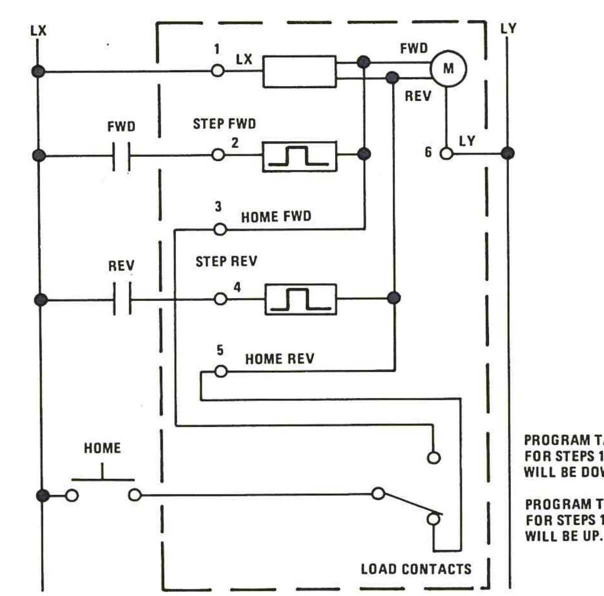


Figure 3 - 14b

FORWARD STEP - MAINTAINED CLOSURE

As before, the motor is started by the input closure. S1 closes to seal power to the motor, S2 closes to energize CR1. If the input switch is not released, CR1 remains energized through CR1 - 1. The motor stops anyway because S1 opens and CR1 - 2 is open.

FORWARD HOME

The Forward Home input is directed to the motor through the normally closed contacts or CR2 - 1. As long as CR2 is not energized, the motor will run forward, as long as the input is present. If a Home Reverse input is applied simultaneously to a Home Forward input, Home Reverse wins. This is explained in the Home Reverse section.

REVERSE STEP - MOMENTARY CLOSURE

This input appears at terminal 4, and energizes CR2 through CR1 - 4. CR2 - 1 opens; CR2 - 2 closes. Power to start the motor in reverse appears at the REV motor winding through CR2 - 3 which has closed. S1 closes, sealing power through CR2 - 2 to the REV Winding. CR2 - 1 is open and no power appears at the FWD Winding. S2 closes and power is applied at CR1. If the input has been released, energizing CR1 has no impact. The motor continues to turn until S1 opens which also releases CR2.

REVERSE STEP - SUSTAINED CLOSURE

The input energizes CR2 through CR1 - 4. CR2 - 3 closes applying power to the REV motor Winding. CR2 - 2 closes and CR2 - 1 opens. When S1 closes, power is held at the REV Winding through CR2 - 2. S2 closes, energizing CR1. If the input has been maintained, CR1 - 3 is closed, holding CR1 energized, even when S2 opens.

When S1 opens, CR2 de-energizes, opening CR2 - 2. The closed input switch cannot prevent this because CR1 - 4 is open. With CR2 - 2 and CR2 - 3 open, the REV motor Winding is open from all directions. Since S1 has also opened, power is not present at the FWD motor Winding. The motor is *stopped.*

As a circuit curiosity, notice that the HOME FWD input *would* move the motor forward after the motor is stopped, since CR2 - 1 has released, overriding the reverse mode. It is never desirable however, to present simultaneous conflicting inputs to the programmer unless the situation and final results are predictable, reliable, and desired. Therefore, while it may be interesting to *see what happens,* under these conditions in the laboratory or class room, it is highly undesirable to duplicate conflicting signal conditions in the real application.

REVERSE HOME

As input at terminal 5 energizes CR2. The CR2 - 3 contacts close, providing power to the REV motor Winding. CR2 - 2 also closes, providing power as long as S1 is closed, however power from the input itself will keep the motor running, regardless of S1 status. With CR2 - 1 open, the FWD motor Winding is cutoff. Notice that a conflicting input at HOME FWD does nothing but provide another path for power at the REV motor Winding, since CR2 - 2 is closed.

Although it is not recommended, *simultaneous* application of inputs to two different input terminals will cause the Programmer to react. The following chart gives the expected response from two simultaneous inputs.

SIMULTANEOUS POWER APPLIED TO:	TERMINALS	RESULT
Step Fwd and Home Fwd	2 and 3	Home Fwd
Step Fwd and Step Rev	2 and 4	Stall (May Rev 1 Step)
Step Fwd and Home Rev	2 and 5	Home Rev
Home Fwd and Step Rev	3 and 4	Home Rev
Home Fwd and Home Rev	3 and 5	Home Rev
Step Rev and Home Rev	4 and 5	Home Rev

As before, with the detailed operation explanation concluded, a simplified representation for the input circuitry will be utilized in wiring examples. Thus, the circuit detail of Figure 3 - 11a will be shown as Figure 3 - 12.

The presence or absence of a program tab in any particular step position will determine the resultant switch state. If the tab is present, the circuit switch is actuated, i.e. normally closed contacts are open; normally open contacts are closed. If the tab is absent, the switch is in its normal or unactuated position, i.e. normally closed contacts are closed; normally open contacts are open.

The user can ascertain the number of available Stepswitch *circuits* by counting the number of individual cam segments contained on the device. Also, the number of step *positions* is determined by counting the number of program tab positions *around* a cam segment.

If the Programmer contains ten cam segments similar to Figure 1-3, the device can control up to 10 external circuits. Remember that the total number of contacts on the stepswitch must be equal to or greater than the number required for external loads *and* Programmer control.

THE PROGRAMMER INPUT

As stated earlier, a stepswitch is similar to a multi-circuit repeat cycle timer except for the form of its input. While power must be *continuously* applied to the motor of a repeat cycle timer, the stepswitch input must consist of voltage *pulses.* Thus, the stepswitch advances one step for each input pulse, and remains stationary *in-between* pulses. The required input pulse to a stepping motor or stepping solenoid must take the general form shown in Figure 1 - 4.

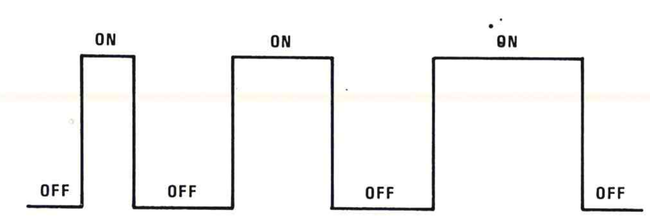


Figure 1 - 4
Input Pulse Form

To be recognized as a required Step Pulse, the stepswitch input must be present *and* absent for a minimum specified time. Depending upon stepswitch design, the stepswitch advances either on the *application* or *removal* of the input signal.

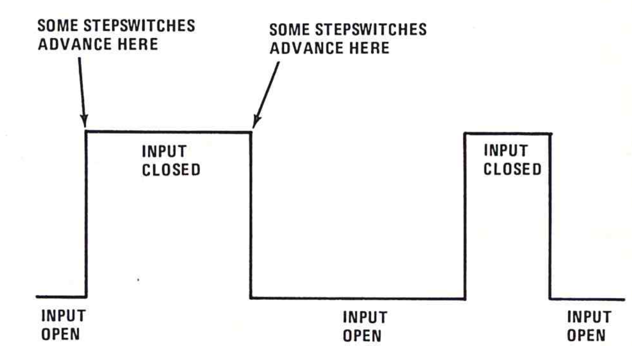


Figure 1 - 5
Stepswitch Advance Points

In either case, regardless of advance point, the stepswitch will advance two steps for the input pattern shown in Figure 1-5.

THE PROGRAMMER IN A CIRCUIT - IT'S ELECTRICAL REPRESENTATION

So far, we have discussed the load control switches as controlled by program tabs, and the step pulse input. Electrically, these elements become the major part of the Programmer wiring diagram which is shown in Figure 1 - 6. The pulse symbol at the input will remind you that the step input must result in a pulse. If you need to review electrical symbols, a summary of symbols used in this manual is presented at the end of this chapter.

The step input switch is used to apply the input voltage to the stepping mechanism. The required pulse may or may not be *automatic* as will be discussed later. Depending upon the arrangement of program tabs, load switches will be open or closed for each step interval. Although the basic circuit of Figure 1-6 will work, further examination will reveal several circuit deficiencies which probably warrant consideration.

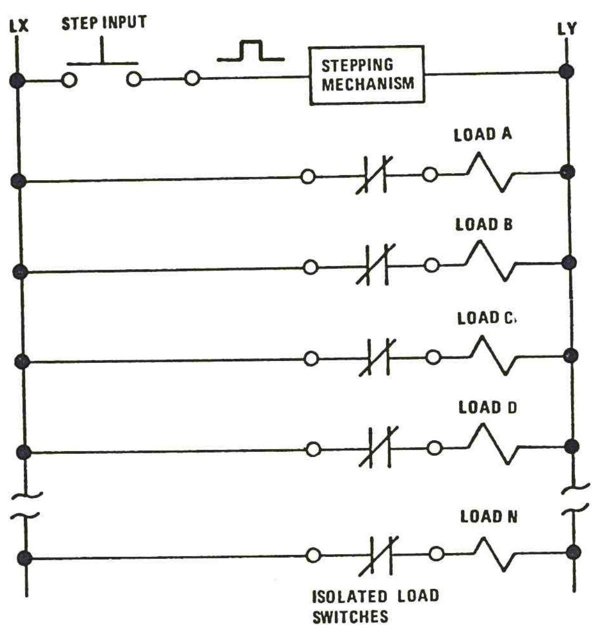


Figure 1 - 6
Basic Stepping Circuit

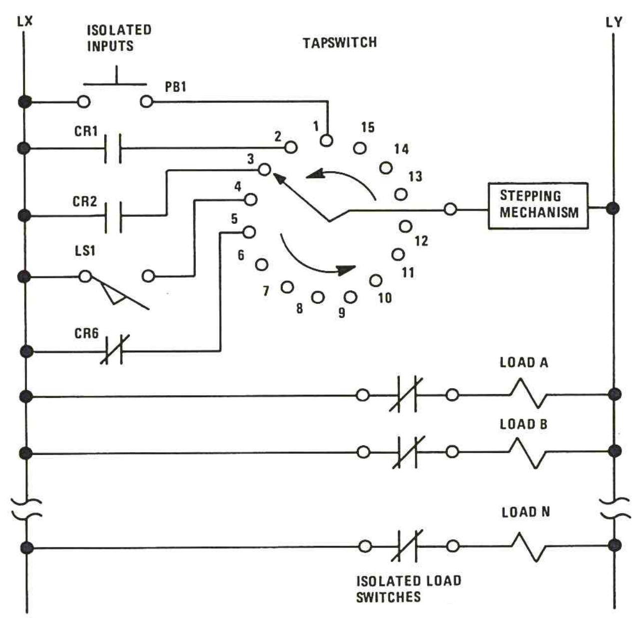


Figure 1 - 7
Tapswitch Input Isolation Circuit

INPUT ISOLATION

First, notice that only one step input switch is shown. In a real application, a separate input might be required for *each* step. Also, the designer may wish to *isolate* each input, so that other inputs are *locked out*, except the one which is to advance the stepswitch from one particular interval to the next.

Figure 1-7 shows the addition of an optional component, the *tapswitch*, which provides this necessary lockout and isolation function. The rotary *wiper* advances with the camshaft, so that the wiper is in contact with terminal 1 in step position 1; terminal 2 in step position 2; etc.

Notice the position of the tapswitch in Figure 1-7 indicates the stepswitch to be in step position 3. Only input CR2 will advance the stepswitch from this position. All other inputs are locked out, as they do not complete a circuit to the stepping mechanism.

SELF-STEPPING

While the tapswitch solves the problem of input isolation, another situation may also exist which requires consideration. Notice that the tapswitch contains fifteen terminals, indicating a fifteen interval stepswitch. But, what if our application requires only thirteen steps? What do we do with the two extra unused steps?

One solution is to provide the capability to rapidly *self-step* through the unused step positions. The self-step function may require the use of one of the available load contacts, or a special contact may be set aside on the stepswitch for this purpose. Figure 1 - 8 shows one form of special contact to implement the self-step function.

With this particular design, the *top* lever of a special switch form is forced up, opening the contacts, whenever the step solenoid is actuated.

In this configuration, to step through positions 4 - 7, load contact program tabs for these positions will be *raised*. Of course, as with the other examples, MTA Programmer action is not specified beyond the final input discussed.

BI-DIRECTIONAL OPERATION

Unlike the MT series Stepswitch, the MTA series Programmer may be ordered in a configuration which may be advanced forward or backward. Additional components and circuitry are added to provide *reverse* operation and are available when feature 02 (Bi-directional operation) is ordered.

Like the forward step operation, circuitry to provide bi-directional operation will be discussed only once. Then, a representative wiring diagram will be established, to reduce example circuit detail.

Figure 3 - 11a shows the internal wiring diagram, components, and terminal assignments for the bi-directional MTA Programmer.

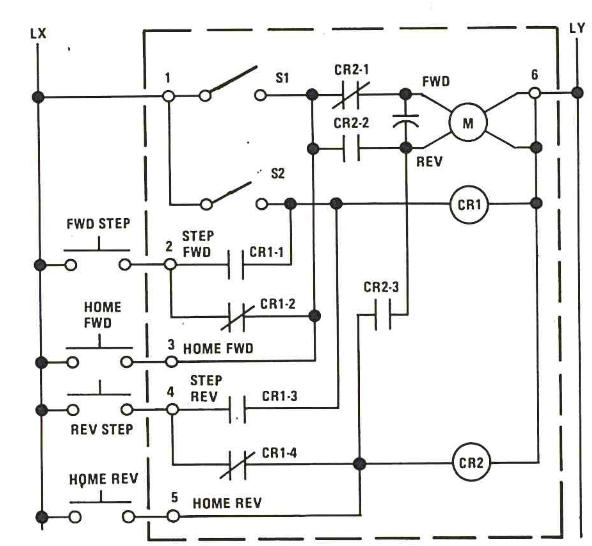


Figure 3 - 11a

Figure 3 - 11b gives the relationship between MTA switches S1 and S2. This relation is the same as discussed earlier with Figure 3 - 1b.

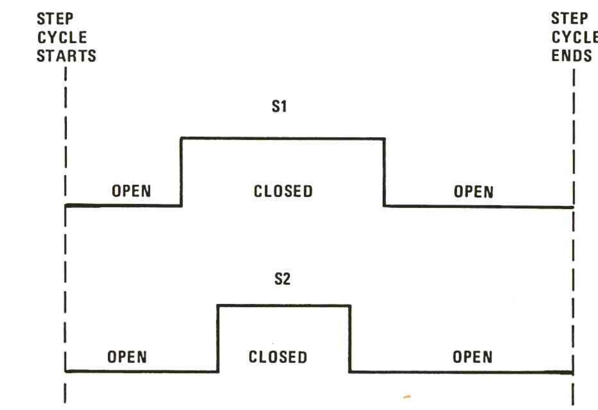


Figure 3 - 11b

Six combinations of input conditions will be considered. These are:

1. FORWARD STEP - momentary closure.
2. FORWARD STEP - closure remains after step is completed.
3. FORWARD HOME
4. REVERSE STEP - momentary closure.
5. REVERSE STEP - closure remains after step is completed.
6. REVERSE HOME

FORWARD STEP - MOMENTARY CLOSURE

From Figure 3 - 11a, it can be seen that the Forward Step input is connected to terminal 2.

When the input is closed, the motor is started forward through the N.C. contacts of CR1 - 2 and CR2 - 1. S1 closes, holding the motor energized. S2 closes a few milliseconds later, energizing CR1. Releasing the input switch has no effect since S1 and S2 are closed. S2 opens, dropping CR1. S1 opens, stopping the motor.

The circuit shown in Figure 3 - 8 will advance the Programmer as long as open inputs close in sequence. If, however, an input is already closed when the Programmer reaches that position, the Programmer will halt until the input is opened and reclosed.

For clarity, the tapswitch is being *represented*, and does not reflect the exact number of terminals which are available. The actual MTA series Programmer contains 24 step positions, and up to 30 load contacts.

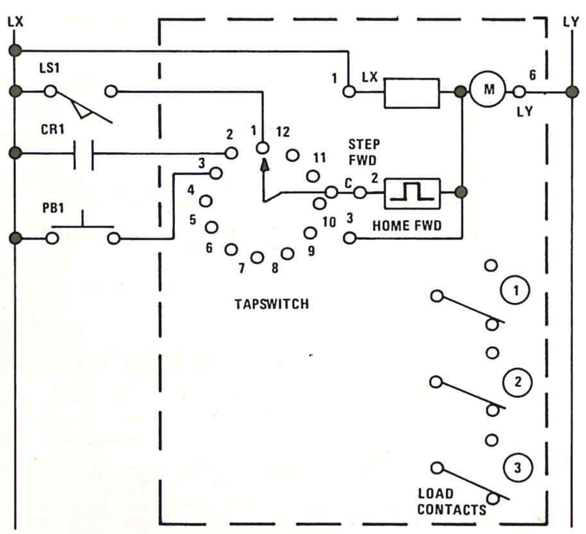


Figure 3 - 8
MTA Tapswitch Input Isolation
(Stops if closed input is encountered)

SELF STEPPING FUNCTION

With the tapswitch, the function of stepping through unused positions is similar to the equivalent MT series circuit. Unused positions are wired together on the tapswitch, with a connection made to LX. The common terminal from the tapswitch is connected to terminal 3 *(FORWARD HOME)*. Figure 3 - 9 shows this circuit.

The circuit shown in Figure 3 - 9 also has the characteristics of advancing through step positions if a closed input is encountered when the tapswitch reaches that position. Notice that in this regard, the circuit of Figure 3 - 9 differs from that in Figure 3 - 8. (The circuit of Figure 3 - 8 will *stop* at the step position rather than continuing to advance).

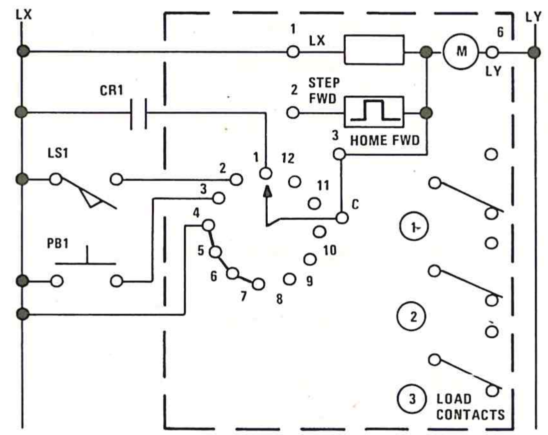


Figure 3 - 9
Tapswitch Isolation and Self Step Function
(Advances if closed input is encountered)

In Figure 3 - 9, the connection is made to *HOME FWD* rather than *STEP FWD*. If the connection were made to the *STEP FWD* terminal, a potential race condition may exist between internal control switch S2 and the tapswitch contact, which could cause the programmer to halt during the self-step mode. Halting would occur if the tapswitch contact reached its next position prior to S2 opening sufficiently to allow CR1 to drop out. In fact, this is the same circuit characteristic observed in Figure 3 - 8.

For self stepping, an alternative which utilizes the *STEP FWD* terminal, is shown in Figure 3 - 10.

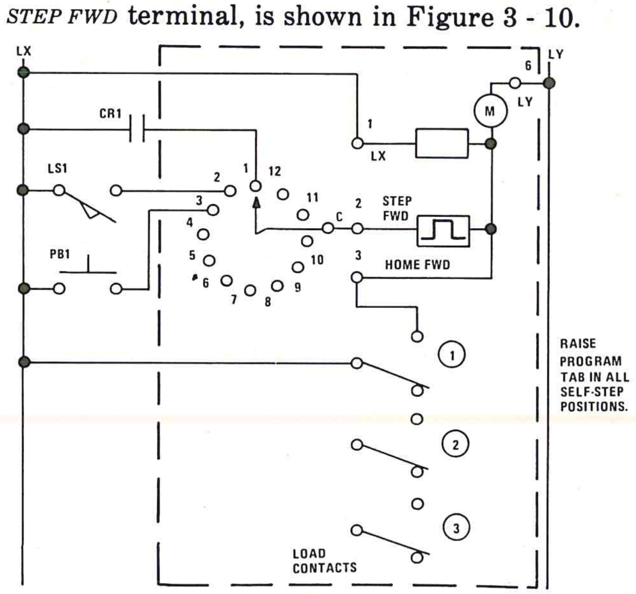


Figure 3 - 10
Input Isolation and Self Step (Using Load Contact)

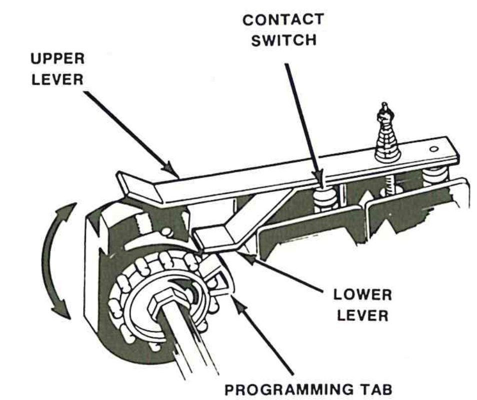


Figure 1 - 8
Self-Step Contact

The *bottom* lever follows the program tabs. If tabs are removed, the contact is allowed to reclose when the step solenoid is de-energized. If the tab is present, the switch contact is held *open.* To self-step through unused positions, tabs are removed from the positions to be *self-stepped.* A tab is present at the position which is to terminate the self-step function.

Figure 1-9 is identical to Figure 1-7, with the addition of the self-step contact. Notice that this contact is placed in series with the step input. The important action of the self-step contact is to break the circuit path leading to the stepping mechanism after the energizing stroke. This action forces de-energization, even if the input is held constant (as will be the case for inputs 6 and 7). If the stepswitch provides other components to *create* a pulse for a constant input, the self-step contact will not be necessary.

The operational sequence to advance from position 3 to position 4 is as follows:

1. **CR2 contact closes, energizing the stepping solenoid through the normally closed self-step contact.**

2. The self-step contact is forced open, completing the step, and moving the tapswitch to position 4.

3. The self-step contact recloses, as long as a program tab is not present in position 4.

4. This action repeats for steps 4 to 5 and 5 to 6, with LS1 and CR6 providing the inputs.

To step from position 6 through position 7 to position 8, the sequence is as follows:

1. The tapswitch reaches position 6. The step solenoid is activated; the step contact opens, and the tapswitch moves to position 7.

2. The same sequence occurs again, moving the tap switch to position 8.

3. The tap switch and associated inputs have not been specified beyond position 8. If the stepswitch is to step around to position 1, all remaining tapswitch terminals must be jumpered, as was done with terminals 6 and 7. The stepswitch will stop when it reaches position 1, assuming PB1 is open. (All other self-step program tabs are removed for unused step positions.)

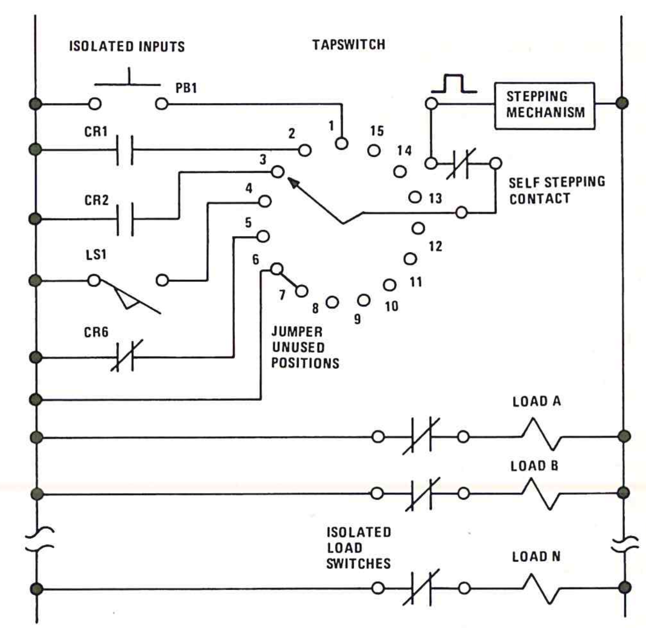


Figure 1 - 9
Self Step Circuit

The examples shown so far do not represent the entire repertoire of sequence control circuits, utilizing a tapswitch and interrupter contact. They are shown here only to familiarize you with the basic function. Additional circuit examples will be shown in later chapters when details of the Eagle Signal sequential products will be discussed.

LOAD PROGRAMMING

To determine the proper wiring scheme and output load program for a given application, consideration must be given to the circuit logic and the determination of loads which must be on and off in each step of the operation. To assist in this task, a chart is helpful to record the details of the application. This chart is shown in Figure 1 - 10. The chart provides space to briefly define each step in the operation, indicate the condition which will signal the stepswitch to advance, define the loads to be controlled, and indicate the proper pattern of load activation for the application.

SYSTEM OPERATION	CONTROL DEVICE TO ADVANCE TO NEXT STEP	STEP	LOAD SWITCHES ON STEP SWITCH CLOSED: X OPEN: O																			
			IC	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
		1																				
		2																				
		3																				
		4																				
		5																				
		6																				
		7																				
		8																				
		9																				
		10																				
		11																				
		12																				
		13																				
		14																				
		15																				
		16																				
		LOADS																				

Figure 1 - 10
Program Chart

After this chart has been filled in with the pertinent details of the operation, it serves to provide both system documentation and the information necessary to set the proper program tab sequence.

Figure 1 - 11 shows a portion of this chart properly completed. The SYSTEM OPERATION column has been used to define briefly the event or events which are to occur during a particular step in the overall cycle.

At the end of each step, something always must happen to signal that the step has been completed. In some instances, multiple conditions may be required to satisfy the logic to signal the completion of the step. This condition(s) is documented in the middle column for each step.

In Figure 3 - 6, as with any functions utilizing programmable load contacts, the correct operation depends upon the proper positioning of program tabs.

For switch (1), *self-step*, program tabs will be *raised* for unused step positions. For switch (2), *MANUAL HOME FORWARD* program tabs will be *lowered* for all step positions in which this function is to be enabled. Raise the tabs for all positions in which the Manual Home Forward function are to be disabled. Of course, if the MTA Programmer is homing toward a position which has a raised tab, the MTA Programmer will stop in that position. If all step positions are to be *home* enabled, the home position *alone* must exhibit a raised tab.

INPUT ISOLATION

Since most applications will present a multiple number of inputs to the circuit, provision should be made to isolate these various inputs. The use of load contacts for this enable/isolation function is possible, as long as the total number of inputs and outputs does not exceed the number of load switches on the MTa Programmer.

Figure 3 - 7 shows the circuit form to isolate inputs using load contacts.

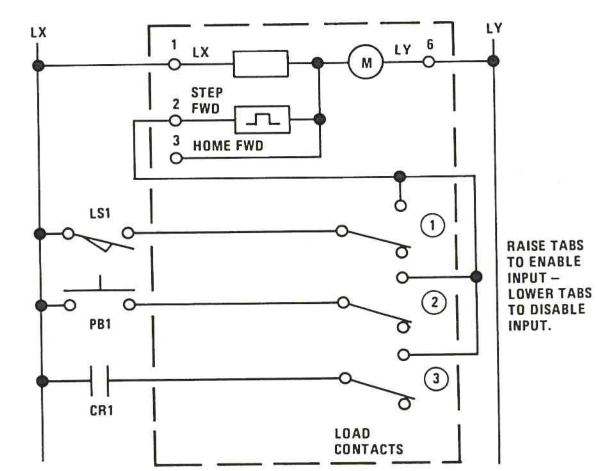
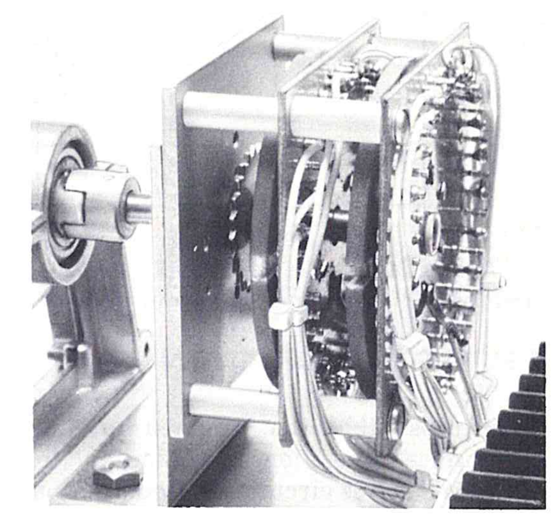


Figure 3 - 7
Input Isolation Using Load Contacts

To implement the circuit shown in Figure 3 - 7, a program tab will be raised for each step position in which a particular input is to be enabled. Tabs will be lowered in all positions for which the particular input is to be disabled.

MTA SERIES TAPSWITCH

As with the MT series Stepswitch, the MTA series Programmer offers an optional tapswitch feature, which can be used for input isolation and other functions. Input isolation, through the tapswitch, is identical in function to the MT series, previously discussed. Physically, the tapswitch is mounted at the end of the programmer and its wiper contact increments along with the camshaft rotation.



1351

MTA Series Tapswitch

Figure 3 - 8 shows three inputs, isolated through the tapswitch. Obviously, the use of a tapswitch relieves the load contacts of this function. Remember that the MTA Programmer advances (steps) on the input closure to terminal 2.

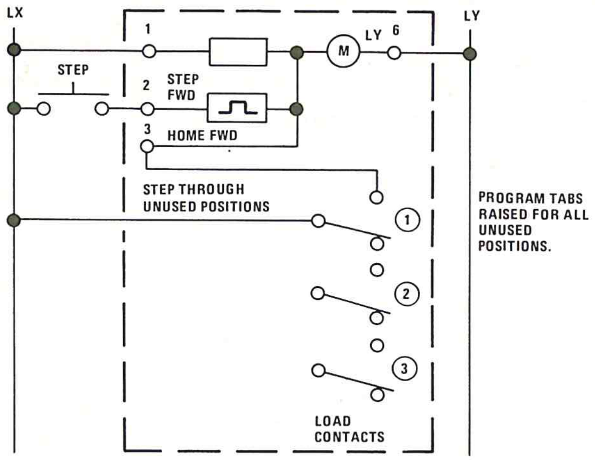


Figure 3 - 4
MTA Step Input Plus Home Forward Through Unused Step Positions

Load switch (1) is used for the self step function. LX has been connected to the common terminal, with the normally open contact connected to terminal 3 *(HOME FWD)*. Program tabs are raised for all unused steps.

Since the MTA series load switch offers both a normally open and normally closed contact, the user is offered an interesting choice. In the example shown as Figure 3 - 4, we can:

1. Use the N.O. contact and *raise* unused step position program tabs. (Active step tabs are lowered).

or

2. Use the N.C. contact and *lower* unused step position program tabs. (Active step tabs are raised).

The following figure relates load switch status to program tab status.

	PROGRAM TAB STATUS	
SWITCH TYPE	Tab Up - Switch Actuated	Tab Down - Switch Unactuated
NORMALLY OPEN CONTACT	Contact Closed	Contact Open
NORMALLY CLOSED CONTACT	Contact Open	Contact Closed

Figure 3 - 5

SELF STEP MANUAL HOME FUNCTIONS

In the next example, the unused position self step function, and a manual home forward function are shown. A second load contact is used to enable/disable the manual home pushbutton. This allows the Programmer to stop *automatically* in the home position, as long as the program tabs are positioned properly. Had we chosen to connect the manual pushbutton directly to terminal 3, the operator would have to release the pushbutton at precisely the right moment to stop at the desired step position. This type of operator requirement should be avoided, if possible.

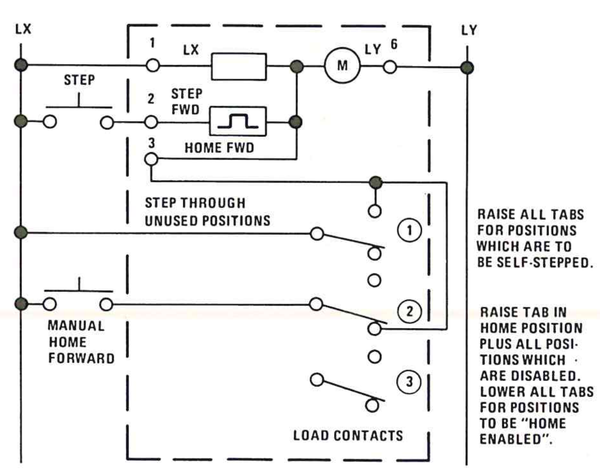


Figure 3 - 6
Manual Step, Manual Home, and Self Step

The right side of the chart is devoted to a matrix chart, with each block representing the intersection of a step position and a load circuit contact. If the contact is to be open in a step position, an *O* is placed in the appropriate block; if the contact is to be closed, an *X* is indicated.

The bottom of the matrix is used to indicate the name or number designation for each load contact.

SYSTEM OPERATION	CONTROL DEVICE TO ADVANCE TO NEXT STEP	STEP	LOAD SWITCHES ON STEP SWITCH CLOSED: X OPEN: O																		
			IC	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Home Position	AUTO DET.	1	O	O	O	O	O	O	O	O	O	O	X									
Door Up	LS 1	2	O	X	O	O	O	O	O	O	O	O	O									
Door Down	LS 2	3	O	O	X	O	O	O	O	O	O	O	O									
Water On	TIMER 1	4	O	O	O	X	O	O	O	O	O	O	O									
Brushes On	COUNTER 1	5	O	O	O	X	X	O	O	O	O	O	O									
Wax Cycle	TIMER 2	6	O	O	O	O	O	X	O	O	O	O	O									
Dry Cycle	TIMER 3	7	O	O	O	O	O	O	X	X	O	O	O									
Door Up	LS 3	8	O	O	O	O	O	O	O	O	X	O	O									
Door Down	LS 4	9	O	O	O	O	O	O	O	O	O	X	O									
		10	O	O	O	O	O	O	O	O	O	O	X									
Step positions 10-		11	O	O	O	O	O	O	O	O	O	O	X									
16 will be self-		12	O	O	O	O	O	O	O	O	O	O	X									
stepped through		13	O	O	O	O	O	O	O	O	O	O	X									
tapswitch to position		14	O	O	O	O	O	O	O	O	O	O	X									
1.		15	O	O	O	O	O	O	O	O	O	O	X									
		16	O	O	O	O	O	O	O	O	O	O	X									
		LOADS		FRONT DOOR UP MOTOR	FRONT DOOR DOWN MOTOR	WATER SOL	BRUSH MOTOR	WAX SPRAY	DRYER HEAT	DRYER MOTOR	REAR DOOR UP	REAR DOOR DOWN	READY LIGHT									

NOTES:

1. IC COLUMN IS FOR INTERRUPTER SELF-STEP CONTACT.
2. THIS CHART IS TO SHOW FORM ONLY. THE PROGRAM INDICATED IS NOT TO BE USED AS A COMPLETE RECOMMENDED SOLUTION FOR CARWASH APPLICATIONS, PER SE.

Figure 1 - 11
Completed Program Chart

CIRCUIT SYMBOLS

The following chart, Figure 1 - 12, is a summary of the electrical symbols used in this manual. Inputs are shown in their normal, unactuated position. If actuated, normally open switches are closed; normally closed contacts are open.

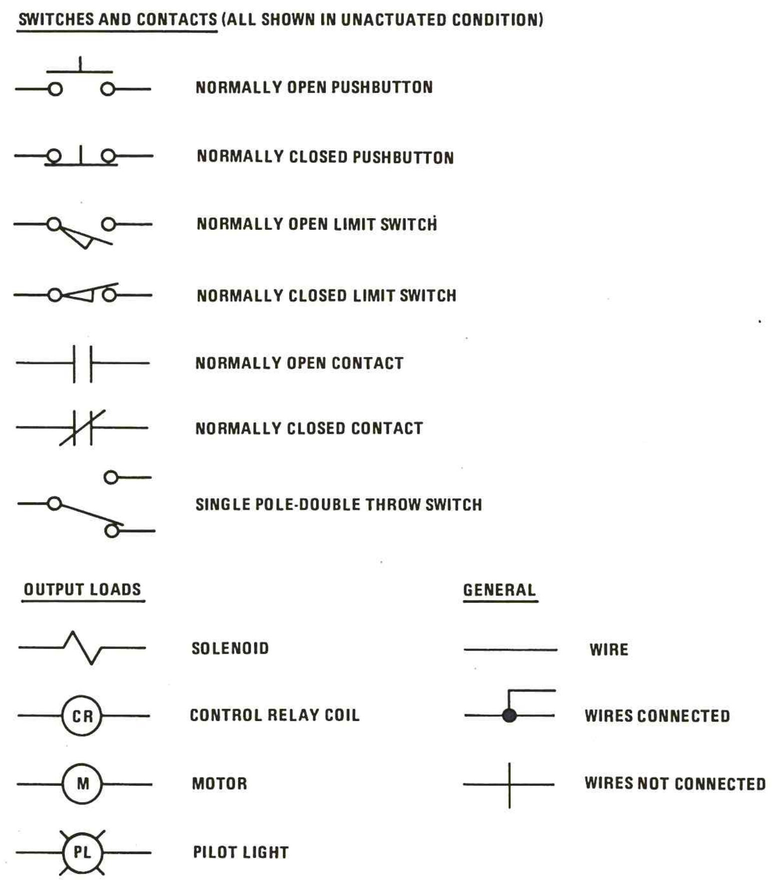


Figure 1 - 12

CHAPTER SUMMARY

1. Although basically simple in design, the electromechanical step-switch offers a potential reduction of complex relay circuitry to provide sequential control action.

2. Various control capabilities are possible, through the use of the tapswitch, self-stepping contacts, and other stepswitch components.

3. The basic parts of a stepswitch or sequential Programmer are:

   A. Cam segments and Program tabs.
   B. Camshaft and load switches.
   C. Step mechanism (motor or solenoid).
   D. Tapswitch (optional).
   E. Homing capability (self step).

will be represented by,

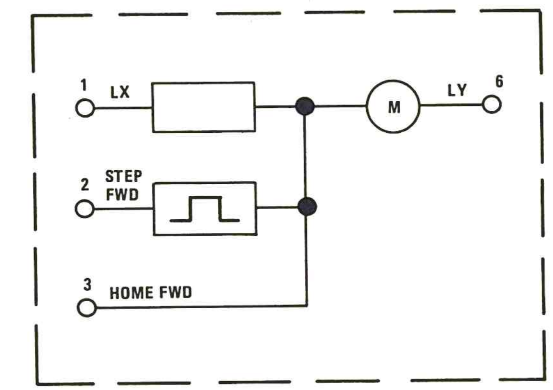


Figure 3 -1d

The open box in the LX line shows that circuit elements are present, however, the user need only connect LX to terminal 1 and LY to terminal 6. The symbol in the *STEP FWD* line indicates the *automatic* pulse nature of the step function. The straight connection from terminal 3 (*HOME FWD*) to the motor reflects the exact nature of this terminal function.

With the addition of SPDT load contacts, initial control circuits will be shown on the diagram represented in Figure 3 - 2. The dashed lines enclose MTA elements. All other circuit external to the MTA Programmer are shown outside the *box*.

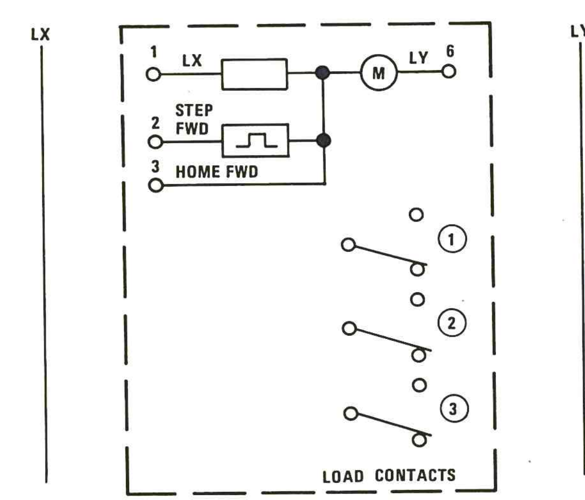


Figure 3 - 2
Basic MTA Wiring Format

MTA SERIES WIRING DIAGRAMS

The simplest example of an MTA Programmer wiring diagram consists of a single step pushbutton. This basic diagram is shown in Figure 3 - 3.

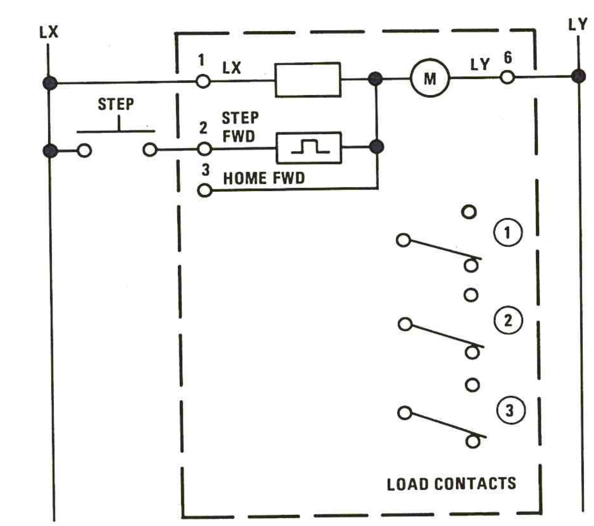


Figure 3 - 3
MTA Step Input

With the wiring example shown in Figure 3 - 3, remember that the MTA step function occurs on the switch *closure*. Also, since the internal wiring circuit, utilizing S1, S2 and CR1, creates the pulse necessary for the step, the MT Series interrupter contact function is not required.

SELF STEP CIRCUIT

Since the MTA Programmer contains a separate input to *HOME FWD*, the circuit shown in Figure 3 - 4 provides the capability to step through unused positions.

Figure 3 - 1a shows the basic MTA step function control wiring. Figure 3 - 1b shows the relative opening and closing of S1 and S2 for one step cycle. During one step cycle, cams controlling the actuation of switches S1 and S2 make one cycle as shown in 3 - 1b. Notice that S2 closes slightly later than S1 and opens slightly earlier.

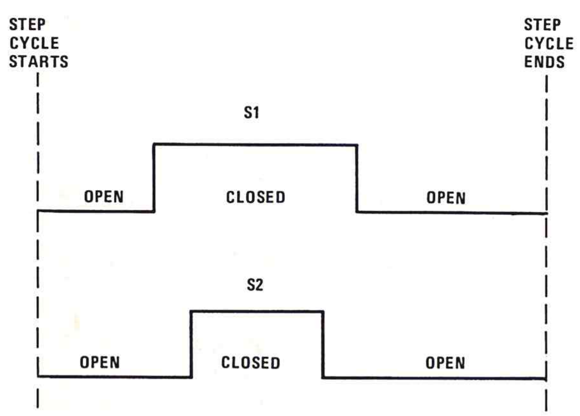


Figure 3 - 1b

MOMENTARY PUSHBUTTON ACTION

When the external remote step button is depressed, to initiate a step advance, the motor is energized through the normally closed contacts of CR1 - 2. A few milliseconds later, the S1 cam closes switch S1, holding the motor energized. At this point the step pushbutton may be released.

A few milliseconds later, the S2 cam actuates switch S2. This energizes relay CR1. CR1 stays energized and the motor keeps running because both S1 and S2 are closed.

A few milliseconds later, S2 opens, de-energizing CR1. The advance motor continues to run because S1 is still closed.

A few milliseconds later, S1 opens and the motor stops. The Programmer has completed one step advance.

MAINTAINED PUSHBUTTON ACTION

The MTA control circuit must also insure that the motor will stop after one step advance, *even* if the step pushbutton is held depressed. Lets re-analyze the action, assuming this to be the case.

The step button is depressed. S1 closes, holding the motor energized. S2 closes, energizing CR1. S2 opens, but CR1 is now held energized through the closed step pushbutton and contacts CR1-1, which are still closed because CR1 has not been allowed to drop out. However, CR1-2 contacts are open and the step pushbutton path to the motor is open. When S1 opens at the end of the cycle, the motor is stopped anyway, since *both* S1 and S2 are *open.*

The motor cannot be restarted until the step pushbutton is released, causing CR1 to release, closing contact CR1 - 2 between the step pushbutton and the motor.

In Figure 3 - 1a, notice that an input to terminal 3 *(HOME FWD)* becomes a straight connection to the motor. As long as this input is present, the motor will run.

With this explanation behind us, we are going to create a new representative drawing which will be used for all MTA wiring diagrams in this manual. This drawing eliminates the detail of the step circuitry, which would only become confusing and repetitious in our application circuits.

This MTA interval control detail,

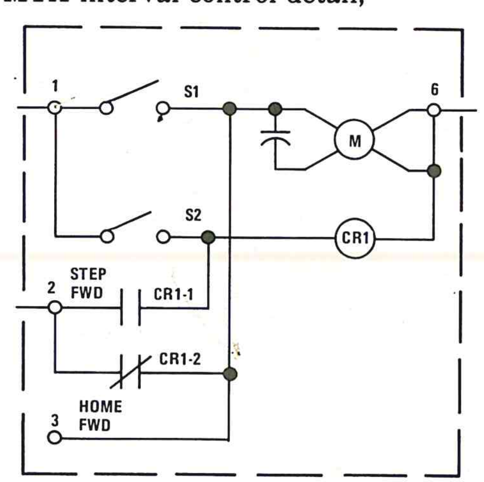
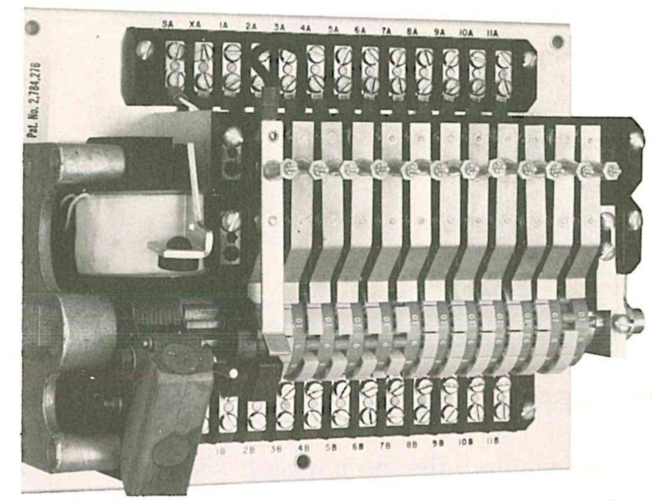


Figure 3 - 1c

CHAPTER II.

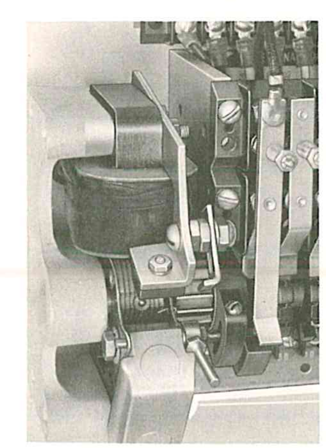
MT SERIES STEPSWITCH



575

MT Series Stepswitch

The Eagle Signal MT series Stepswitch utilizes a solenoid operated rotor and weight assembly for the step function. When the step input voltage is applied to the step solenoid, the weight is lifted, indexing a pawl into the next rachet tooth. When the step solenoid is de-energized the weight drops, rotating the camshaft ahead one step. Since the stepping action relies on gravity to drop the weight, the MT series is position sensitive and must be mounted on a vertical or near vertical plane. (See Figure 2 - 1)



1341-A

Figure 2 - 1
MT Series Stepping Mechanism

This method of camshaft advancement, used with the MT series, results in the stepswitch advancing upon the *de-energization* of the stepping solenoid. See Figure 2 - 2.

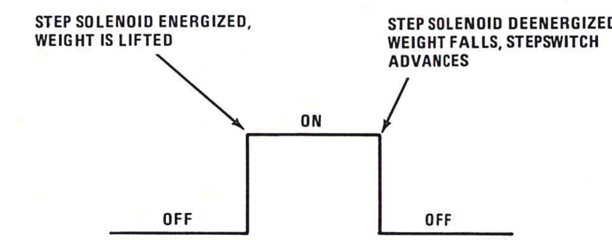
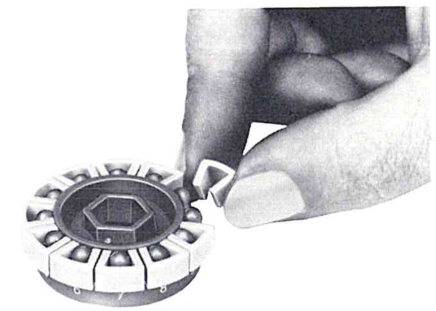


Figure 2 - 2
Mechanical Relationship to Electrical Step Input

The MT series is available in standard 12 or 16 step models. A 12 step cam is directly usable in multiples of this number by repeating the program for shorter programs. (2, 3, 4, 6, and 12 step programs). A 16 step cam can be used for 2, 4, 8 and 16 step programs. Unused program steps must be *self-stepped.*

The basic MT program camshaft is shown in Figure 2 - 3. Two to sixteen circuits are available as standard units. Figure 2 - 3 shows a camshaft to control 11 circuits plus 1 self-step position (left most tab).

574



576

Figure 2 - 3
MT Series Program Cams

The program is established for a particular load circuit by properly placing program tabs around the cam. If the tab is present, the normally closed load contacts will be open; if removed, the normally closed load contacts will be closed. The MT series load switch is of an open type construction, utilizing contact *finger* actuators. The contact form is SPST. (Single pole, single throw)

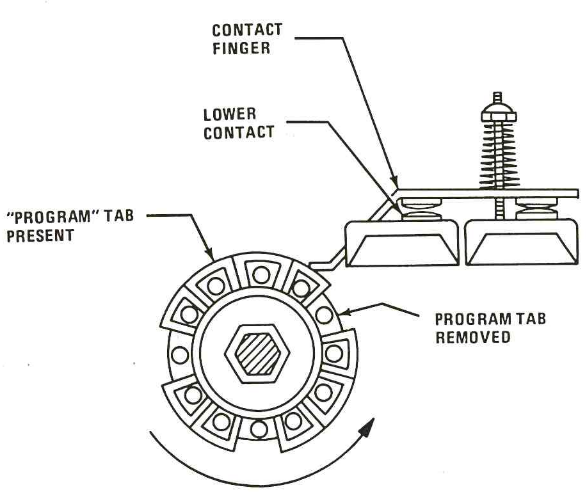


Figure 2 - 4
MT Series Program Cam - Switch Relation

MT SERIES WIRING DIAGRAMS

In this manual, the basic wiring diagrams for the MT series stepswitch will be as shown in Figure 2-5a. If the program example utilizes the optional tapswitch, the drawings will resemble Figure 2-5b. Dashed lines enclose MT Stepswitch components. All other circuit components are external to the dashed lines.

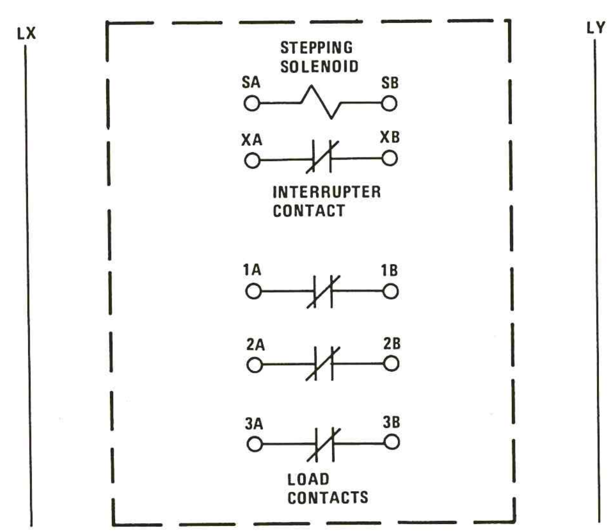
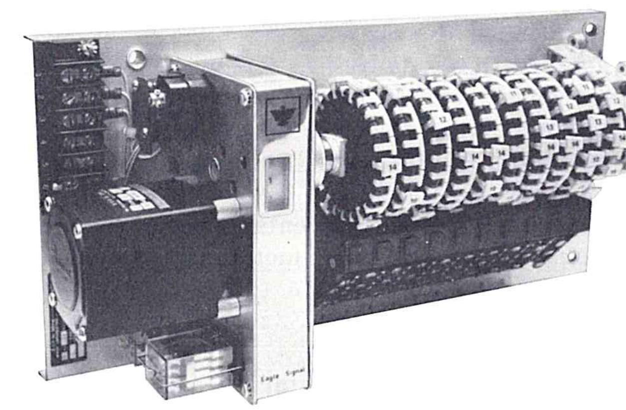


Figure 2 - 5a
Basic MT Series Wiring Diagram
(Without Tapswitch)

CHAPTER III.

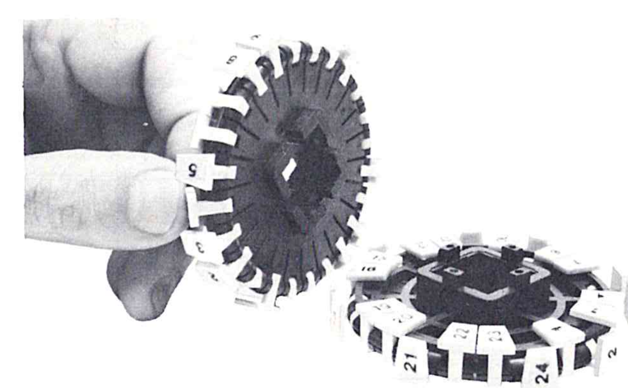
MTA SERIES STEPPING PROGRAMMER



MTA Series Programmer 1279

The MTA Series Stepping Programmer is a multicircuit sequence control device. Although similar in function to the MT series Stepswitch, some operational characteristics are implemented in a different manner.

Probably the most striking visual difference between the MT and MTA series Programmers is the presence of a heavy duty *motor* rather than a stepping solenoid. The large program tabs and tapswitch configuration, although different in appearance, have similar functions for the MTA Programmer to those elements found on the MT series. The program tabs are of a *flip-up flip-down* design.



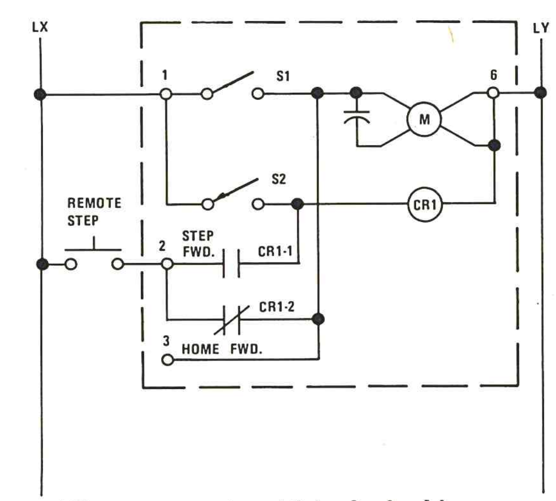
1347

This design means that the elements to program the device are always with the unit, and are never removed from the programmer.

Enclosed load control switches for the MTA series are SPDT, (single pole, double throw) with optional annunciator lights. This, of course, means that both normally closed as well as normally open contacts are available to control external loads and provide Programmer control.

STEP OPERATION

Since the MTA step operation requires the rotation of a motor, circuitry is provided to insure that the motor advances in increments, representing each step, rather than being allowed to run continuously. These internal MTA circuit elements, consisting of two cam actuated switches and a small control relay, will be shown and described here, but will not be repeated each time a Programmer circuit is discussed.



All components within dashed box are MTA elements. All items outside this box are external devices.

Figure 3 - 1a

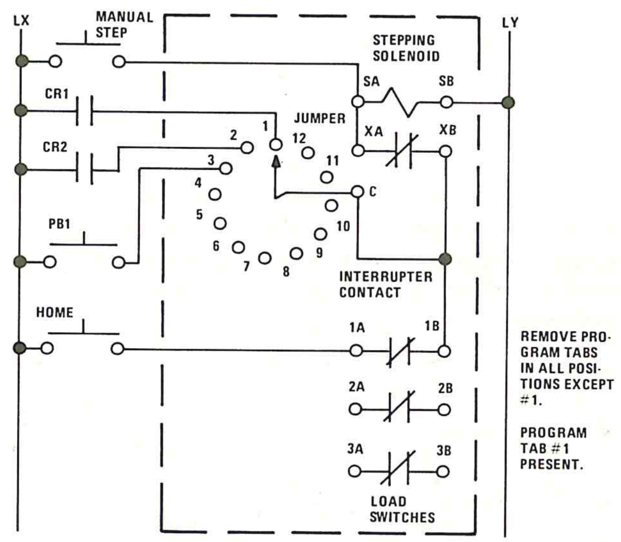
ANSWERS TO MT SERIES QUIZ

1. *Answer:* Self-stepping Interrupter contact has not been used. Step positions 1, 2 and 3 will act normally, although the Stepswitch will advance on the input opening rather than its closing. When the Stepswitch reaches position 4, the stepping solenoid will energize and *stay* energized in position 4.

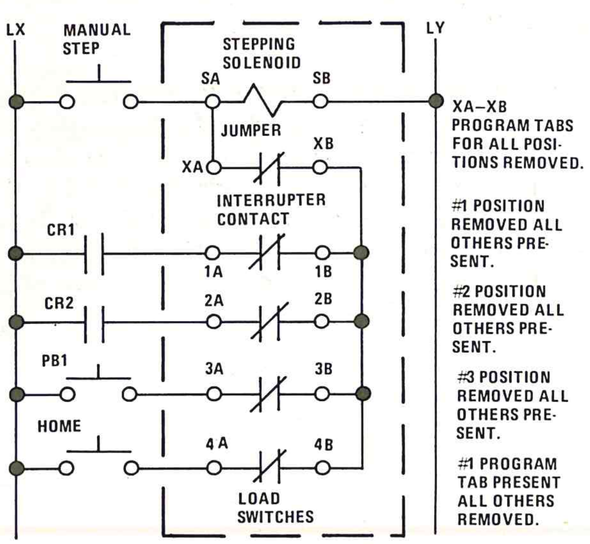
2. *Answer:* Put an XA - XB program tab in at position 1.

3. *Answer:* The XA - XB program tab in position 1 is holding the interrupter contact *open* in this position.

4. *Answer:* Bring the *home* function through a load contact (4A - 4B). Remove the XA - XB program tabs in all positions. Remove all program tabs for 4A-4B except in position 1.



5. *Answer:* Although the Stepswitch advances on the switch closure, it continues to advance as long as the input is held closed. Except for the self step or homing function, this response is not desired.



Another alternative is to use a tapswitch for input isolation, using a load contact for the overriding *home* function.

CHAPTER SUMMARY

Various wiring configurations are possible, using the stepswitch elements to create the desired circuit response.

MT series Stepswitch elements include the stepping solenoid, tapswitch, interrupter contacts, program tabs and load contacts.

Program tabs are present to cause the load contacts to be open; absent tabs cause the contacts to be closed.

The MT series interrupter contact always opens during each energization of the stepping solenoid. Program tabs are used to keep the interrupter contact open (tab present) or allow it to reclose (tab absent).

The tapswitch is like a rotary selector switch, which is mechanically linked to the camshaft. The tapswitch is normally used for input isolation, but can also be used for position annunciators.

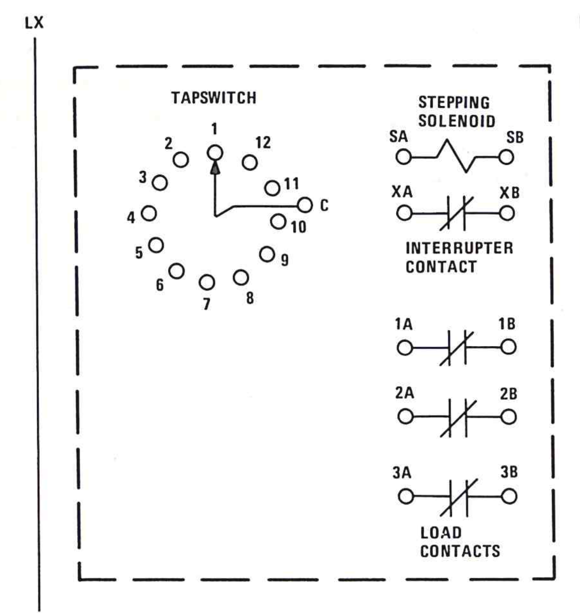


Figure 2 - 5b
Basic MT Series Wiring Diagram
(With Tapswitch)

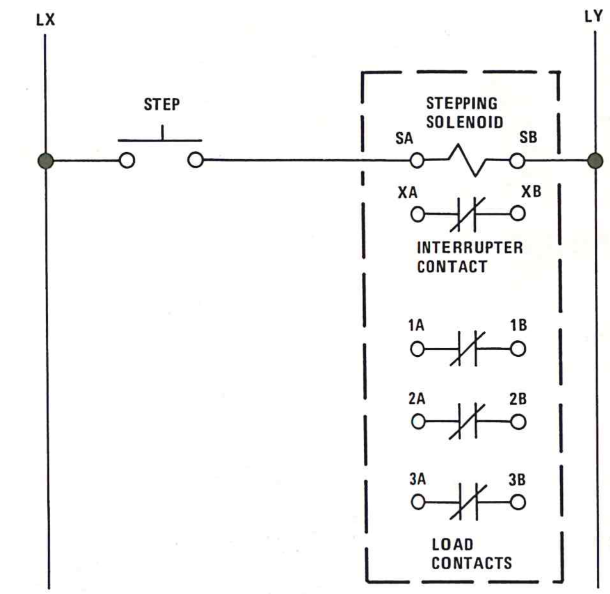


Figure 2 - 6
Basic Stepswitch Control

For the basic circuit examples utilizing the form shown in Figure 2-5a or 2-5b, user loads, connected to the load contacts, will not be shown. Load contacts will be used in circuit examples only if they are being used for *control purposes.* In actual applications, load contacts will be ON or OFF during any particular step as determined by the *load* program. In certain instances, however, one or two load contacts must be used for stepswitch control purposes and will not be available for load control.

BASIC STEP CIRCUIT

The simplest example of a stepswitch control circuit utilizes one switch to directly cause the device to step through the program. This basic circuit is shown in Figure 2 - 6.

In Figure 2 - 6, remember that the actual step will occur upon the *opening* of the input switch.

MT SELF-STEP FUNCTION

The next example is a modification of the basic diagram to provide an automatic self-step through any unused step positions.

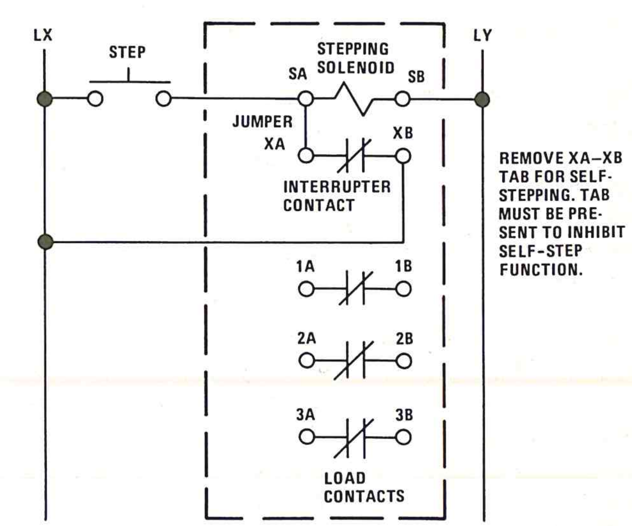


Figure 2 - 7
Stepswitch Control With Self-Stepping Through Unused Positions

The controlling factor for the self-step function will be the removal of program tabs for the interrupter contact XA - XB. For rapid stepping through all unused step intervals, remove the appropriate tabs. For normal step operation, using the pushbutton, XA - XB program tabs must be present.

MANUAL SELF STEP

Some applications require a manual input to self-step the device to the home position (which is normally designated Step No. 1). Figure 2-8 shows the addition of this *HOME* switch.

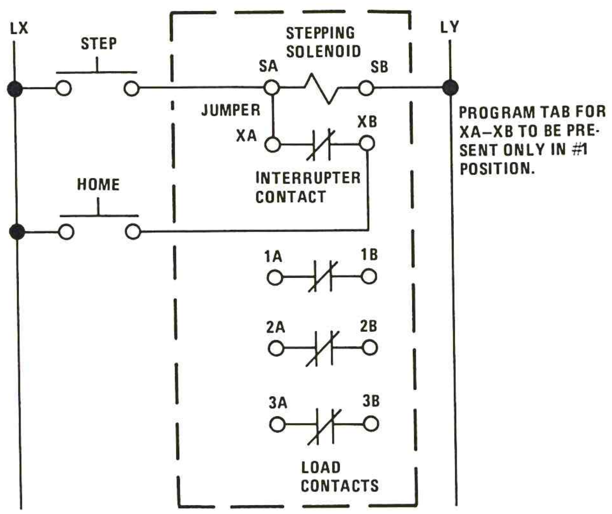


Figure 2 -8
Stepswitch Control With Manual Homing

The step pushbutton operates in the usual manner, controlling the stepswitch throughout the cycle. For the self-step or home function, the *HOME* button is depressed. The stepswitch will proceed to the #1 position and stop. Hold the *HOME* button down until the stepswitch stops in the home position.

You may wonder how the pulse is achieved to cause automatic homing. Remember that the action of the interrupter contact causes the contact to open momentarily as the weight is lifted during the energizing stroke. This automatic contact-opening breaks the circuit, causing the weight to fall, incrementing the stepswitch to the next position. If the interrupter contact recloses (no program tab) the action occurs again. Stepping will occur until a program tab is encountered, holding the XA - XB contact open.

COMBINED CIRCUITS

To combine the self stepping and manual homing functions, shown in Figures 2 - 7 and 2 - 8, a load contact is programmed and used for one of the control functions.

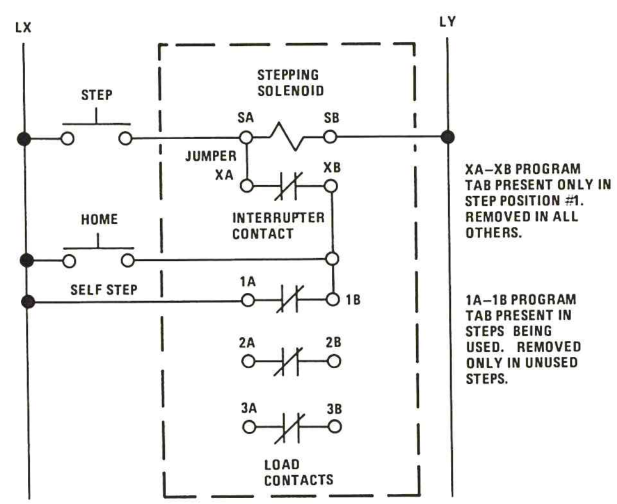
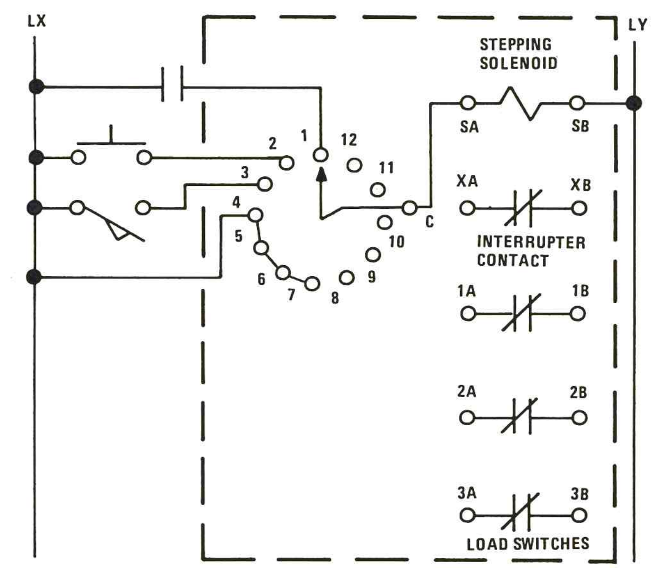


Figure 2 - 9
Stepswitch Control With Self Stepping and Manual Homing

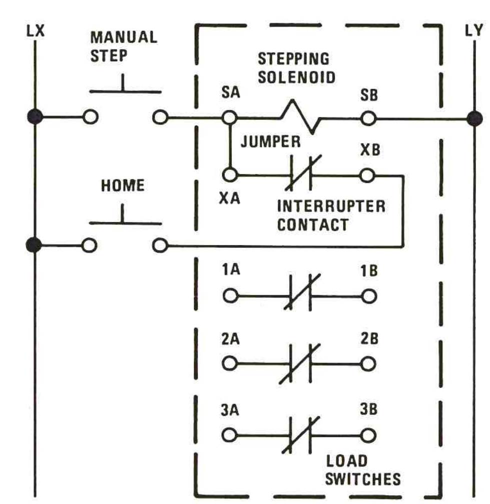
The action of a load contact is somewhat different from that of an interrupter contact, in that it does not automatically open during each energizing stroke. The pulse created by the action of the interrupter contact is required for self-stepping. Do not, however, use a normal load contact *in place of* the interrupter contact, expecting the pulse to occur and the circuit to work. It won't.

MT SERIES QUIZ

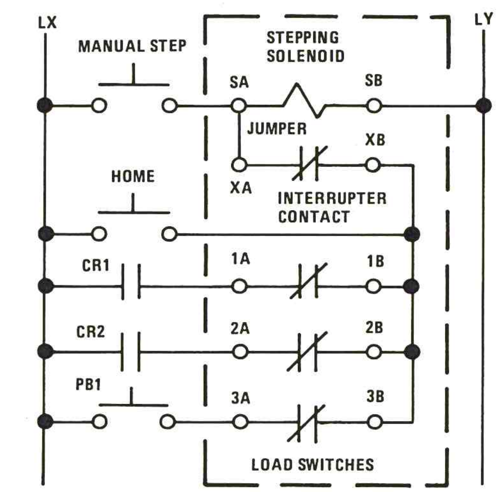
1. The following circuit has been wired using an MT Series Stepswitch. Inputs have been wired to positions 1, 2 and 3. Steps 4 - 7 are unused and will be self stepped. What is wrong with the circuit wiring? What will happen in each of the step positions?



2. *All* interrupter contact program tabs have been removed from the following circuit. The *home* function seems to overshoot the home position, which is position 1. What should be done to force the stepswitch to stop at position 1?

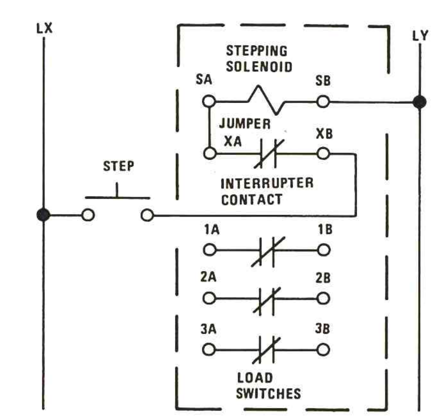


3. The following circuit is wired. XA - XB program tab present in position 1 only. 1A - 1B program tab removed only from position 1. 2A - 2B program tab removed only from position 2. 3A - 3B program tab removed only from position 3. Everything works fine except CR1 cannot move the stepswitch out of step position number 1. *CR2, PB1, HOME,* and *MANUAL STEP* work as expected. What is wrong?



4. How can the above circuit fault (#3) be corrected?

5. You have asked your technician to wire a single step input switch such that the MT Stepswitch will step on each switch closure. The technician does not have a control relay, but reasons that the XA - XB contacts can produce the necessary pulse to step on the switch closure, and the following circuit is wired. The circuit does not work properly. What will be the fault noted?



EVEN-MULTIPLE PROGRAMS

In an earlier section, it was stated that a 12 position stepswitch can be used for a 4 step program. In this instance, the Programmer would repeat the load program 3 times, so that:

Positions 1, 5 and 9 have the same program

Positions 2, 6 and 10 have the same program

Positions 3, 7 and 11 have the same program

Positions 4, 8 and 12 have the same program

Assuming that the same four inputs as shown in Figure 2 - 15 are to be used, the tapswitch must be wired to cause the same inputs to be used three times in the stepswitch cycle. Thus, tapswitch terminals 1, 5 and 9 will be connected, as will terminals 2, 6 and 10; 3, 7 and 11; and 4, 8 and 12. With this configuration, jumpered tapswitch terminals to cause rapid advance through unused positions is no longer required.

POSITION INDICATOR

Another use for the tapswitch is a position indicator. Pilot lights, driven from the tapswitch contacts, will indicate the present dwell position for the stepswitch. See Figure 2 - 16.

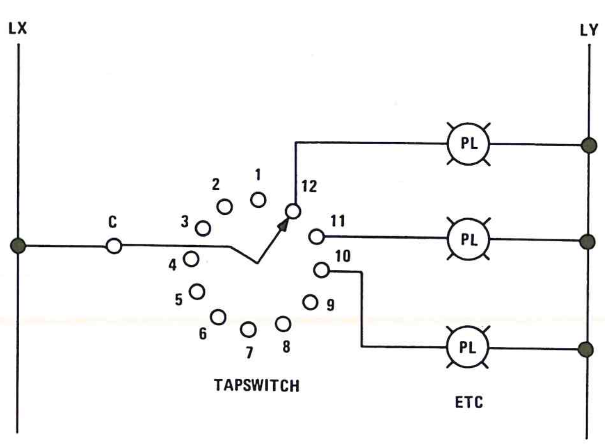


Figure 2 - 16
Tapswitch as Output Indicator Circuit

STEP ON INPUT CLOSURE

Remember that the MT series Stepswitch advances upon the *opening* of the input to the stepping solenoid. A circuit configuration, using the tapswitch, *creates* the effect of stepping upon the external input closure, however, the solenoid is still de-energized to cause the stepping action to occur.

Figure 2 - 17 shows a circuit which may be used to step once for each input closure, *without* using the tapswitch.

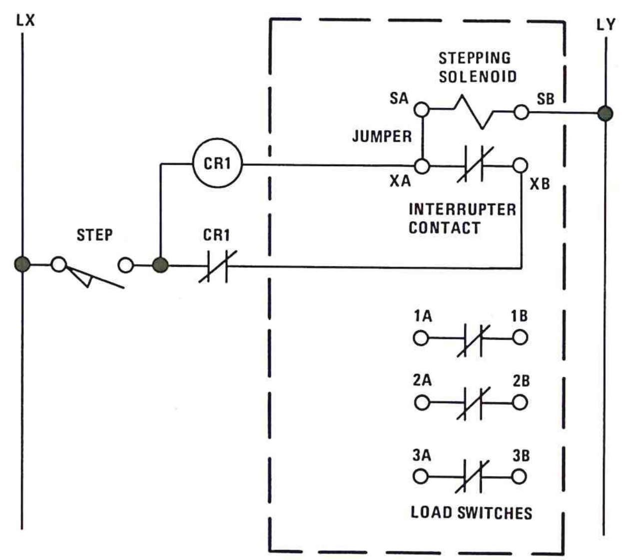


Figure 2 - 17
Step On Input Closure

When the step input switch closes, the CR1 coil is shorted by the contacts of CR1 and XA - XB and thus does not energize. At the top of the energizing stroke, XA - XB opens and current flows through the CR1 coil, energizing CR1. The normally closed contact of CR1 opens and XA - XB closes, but CR1 stays energized. The relatively much higher impedence of CR1, compared to the stepping solenoid, causes the solenoid to be de-energized from voltage and current starvation, while the CR1 coil remains energized. The step input must open before the stepswitch will again advance because only the current path through the CR1 contacts and XA - XB contacts will provide enough current to energize the step solenoid.

INPUT ISOLATION

Since load contacts may be programmed to be closed in any step position, they can be used for *input-enable* functions. Also, since the tapswitch is basically an input-enable element, programmed load contacts can be used in place of the tapswitch for input enabling. Of course, the total number of load contacts for input enabling and load control cannot exceed the total number of load contacts available.

Figure 2 - 10 shows the wiring necessary to achieve multiple input isolation/enabling, through the use of load contacts. Since the input path passes through the interrupter contact, the stepswitch advances on the *closure* of the input. Manual step is accomplished on the switch *opening*, as this input is connected directly to the stepping solenoid. Interrupter contact program tabs must be removed for all step positions being isolated in this manner. If any are accidentally left in place, the input for that step position will not advance the Programmer.

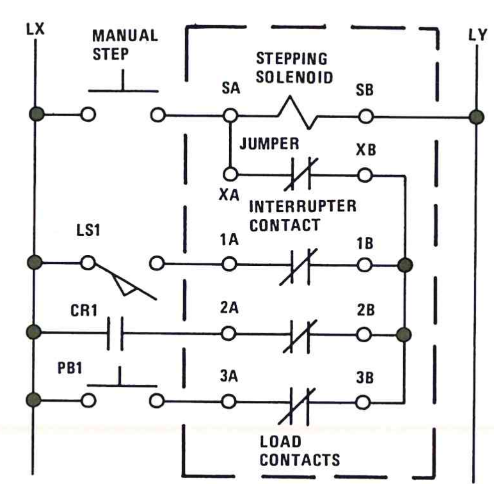


Figure 2 - 10
Load Contacts For Input Isolation

Since each load switch must be closed *only* in the enabled step position, the program tab will be removed in that position only, i.e. 1A - 1B at position 1, 2A - 2B at position 2, etc. In all other step positions, the individual program tab will be *present* to hold the switch contact open, disabling the input.

TAPSWITCH ISOLATION

If input isolation is desired, and load contacts are not available for this purpose, the tapswitch function may be used. Remember that the tapswitch is basically a rotary selector switch, with the common wiper connected directly, or through the interrupter contact, to the stepping solenoid.

Figure 2 - 11 shows the basic connection for the tapswitch, when used for input isolation.

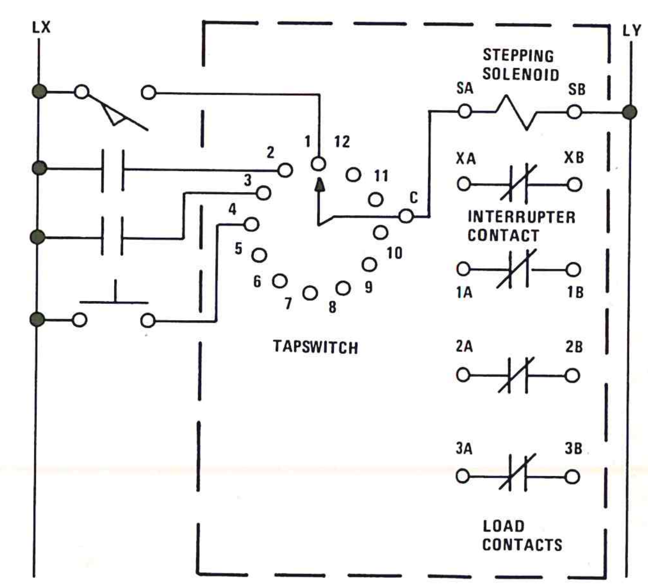


Figure 2 - 11
Tapswitch Input Isolation

Notice in Figure 2 - 11 that the inputs have been brought directly to the stepping solenoid. With this configuration, the input must close to *cock* the Stepswitch *and then open* to advance it to the next position. In other words, the input itself must *create* the required stepping pulse.

An alternate wiring arrangement will allow the interrupter contact to *create* the stepping pulse. This arrangement, shown as Figure 2 - 12, means that the step will occur when the input *closes*, since the interrupter contact will complete the pulse by opening on the *up* stroke.

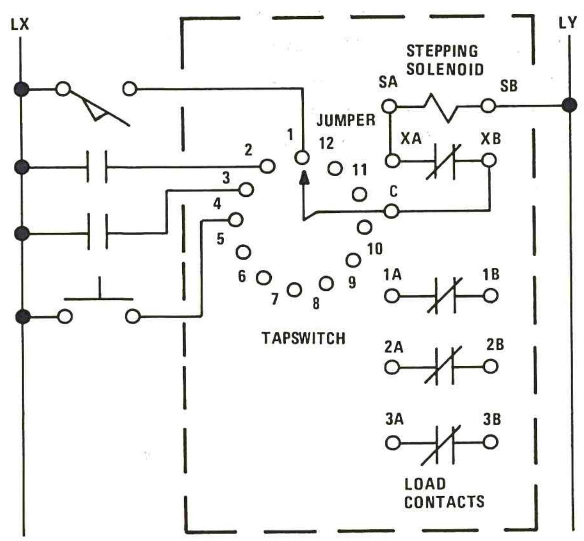


Figure 2 - 12
Tapswitch Isolation Through Interrupter Contact

Remember that in either case, the pulse requirements of the stepping solenoid have not changed and the stepswitch advances when the input voltage to the solenoid is *removed*. With the circuit of Figure 2-11, the input itself creates the pulse by closing and then opening, with the advance occuring an *opening*. With the circuit of Figure 2-12, the interrupter contact opens at the top of the energizing stroke, advancing the tapswitch and the stepswitch. Thus, the input switch may remain closed but the stepswitch advances anyway because power is interrupted to the step solenoid by the interrupter contact.

The stepping action between the circuits shown as Figures 2 - 11 and 2 - 12 is reviewed in Figure 2 - 13.

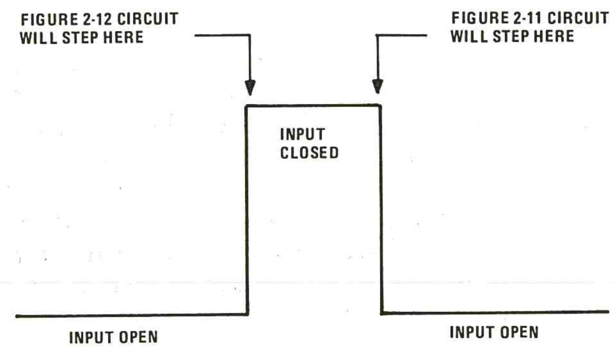


Figure 2 - 13
Stepswitch Advance Review

A manual step pushbutton may be added to the circuit shown in Figure 2 - 12 to allow for manual jogging. This circuit is shown in Figure 2 - 14.

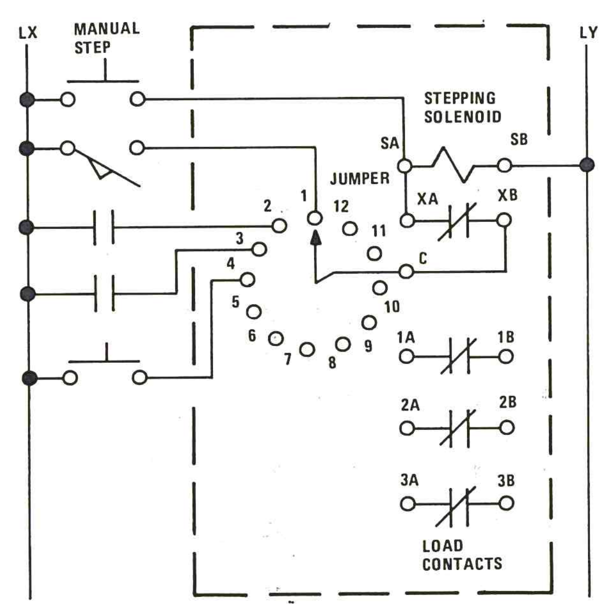


Figure 2 - 14
Input Isolation With Manual Step

For the circuit shown in Figure 2 - 14, remember that the manual step function will occur on the *opening* of the pushbutton since the input is connected directly to the stepping solenoid.

CIRCUIT SUMMARY

Figure 2 - 15 shows a recapilutation of many of the circuit functions, discussed so far. The functions include:

1. Manual step function
2. Tapswitch-isolated inputs
3. Rapid advance through unused steps
4. Rapid home to start position

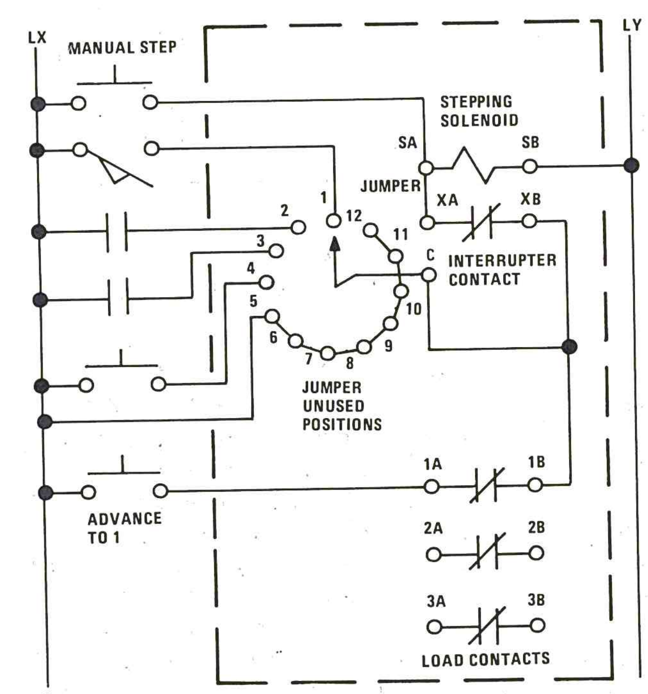


Figure 2 - 15
Circuit Recapitulation

Hopefully, by now, you can examine Figure 2 - 15 and determine the operational characteristics of the stepswitch, which are:

1. The manual step pushbutton will operate in step positions 1 through 4 to *single* step the stepswitch. The step occurs when the pushbutton opens.

2. The inputs connected to tapswitch terminals 1, 2, 3 and 4 are isolated and operate through the interrupter contact. This means that the step will occur when the input closes, as long as the tapswitch wiper is in contact with the input being actuated.

3. Unused steps are accommodated by connecting tapswitch terminals 5 - 12 to LX. This will rapidly advance the stepswitch through steps 5 - 12, utilizing the action of the interrupter contact.

4. The *ADVANCE* pushbutton will advance the stepswitch *from* and *through* any step positions which have a 1A - 1B contact program tab *removed*. The advance will stop when a program tab is encountered, forcing the 1A - 1B contact open, disabling the *advance* input. Notice that through the use of selected program tabs, the *advance* function can be active or disabled through any combination of steps. For example, the application may require rapid advance from the pushbutton *only* if the stepswitch is beyond step 4. To do this, 1A - 1B program tabs will be *present* only for steps 1, 2, 3 and 4. This forces the 1A - 1B contacts open in these step positions, disabling the advance circuit.