

Sequencing program

Sometimes, a customer wants the machine operator to be able to create simple motion sequences without using RMCTools.

One method is to allow the operator to make a simple sequence on the HMI, and then send the sequence to the Variable Table. To do this, a simple protocol needs to be defined, where a certain number means a certain action, such as a move or delay. Each action may need several parameters.

For example, maybe a machine operator needs to make simple sequences of moves, delays, and sinusoidal motion.

For this example, the following numbers could mean the following actions:

1 = Move to specified position at specified speed (two parameters – position and speed)

2 = Delay for specified time (one parameter – time)

3 = Sine wave for a certain number of cycles (one parameter – number of cycles)

For example, to do the following sequence:

Move to 10
Delay 1 second
Move to 5

The operator would write the following numbers to the variable table:

Variable Values	Meaning
1	1 = move 10 = position to move to 5 = speed
10	
5	
2	2 = delay 1 = time to delay
1	
5	1 = move 5 = position to move to 3 = speed
3	

Do this:

Make a user program that can interpret and run any sequence of the actions listed above (move, delay, or sine wave).

This involves the following steps:

1. Make an array in the variable table, where the operator will write the values to.
2. Create a variable called Index. Make it a DINT data type.
3. Make a user program. The first step must set the Index variable to zero.
4. Add another step. Label this step "CheckAction". This step must contain a Conditional Link Type. The link type must use the Index variable to look at the current array item. The value of the array item specifies the action (move, delay, or sine wave).
5. Based on the value, the Conditional Jump will make the user program jump to the step that does the desired action. The step that does the desired action can send the command. The command parameters must come from the array. The Index tells you which value you are at in the array, and by adding numbers to the Index value, you can the right parameters from the array.
For example, `MyArray[Index+1]` will get the number that follows the current action value in the array.
6. After the action completes, the user program must jump back to the CheckAction step. However, before the user program jumps back to the CheckAction step, an Index variable must be incremented the correct amount so the CheckAction step knows which array item to look at for the next action.
For example, for the Delay action, the Index must be incremented by 2. For the Move action, the Index must be incremented by 3.
7. If the action value is zero, that means the end of the sequence has been reached.