Bailey
control systems

Recipe Handling for Distributed Batch Control

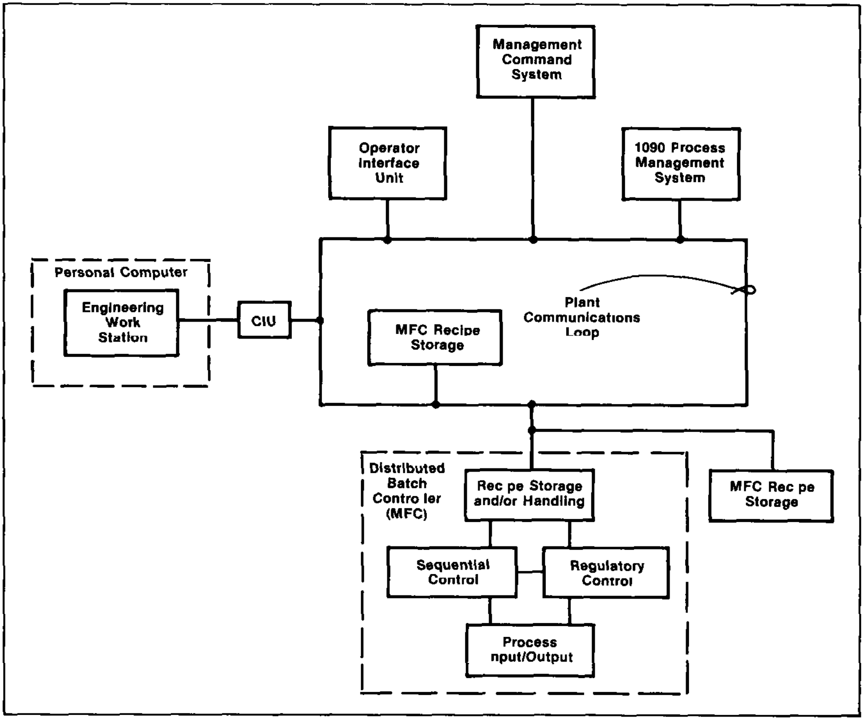

FIGURE 1 Bailey Distributed Batch Control Architecture

Introduction

The Bailey Controls NETWORK 90 microprocessor based distributed control system provides a hierarchical control system for batch processes (**FIGURE 1**). NETWORK 90 allows the actual sequential and regulatory control of the batch to be accomplished solely by a microprocessor based controller located at the lowest possible level in the control system hierarchy. Operator interface, mass storage for long term historical data, and user friendly configuration features are provided by devices at a higher level in the hierarchy (i.e. CRT based Operator Interface Unit, Management Command System, Engineering Work Station, the mainframe based 1090 Process Management System, and/or a supervisory computer with

Application Guide

AG-0000-300-03

Bailey XRS 90 interface software). However, even with these higher level functions, the distributed nature of the batch control system can be maintained, the NETWORK 90 Multi-Function Controller (MFC) provides regulatory and sequential control for the batch process, stores/executes batch recipes, and collects/stores/transmits batch historical data to the higher level devices in the system.

Effective distributed recipe handling (i.e. storage, editing, selection) is critical to effective distributed batch control. Although the recipes may be configured through the Bailey Engineering Work Station or the various NETWORK 90 CRT-based consoles, the recipes can be stored and implemented within controllers distributed throughout the NETWORK 90 control system. The purpose of this application guide is to discuss techniques for configuring these recipes for maximum operational flexibility and reliability.

Distributed Recipe Handling

For many applications, the unique design of NETWORK 90 allows recipe handling entirely within the distributed control system. Depending on the memory requirements for the actual batch control configuration, the recipes can be stored in the same Multi-Function Controller as the control configuration, or in another MFC located either on the same module bus or in a different PCU (Process Control Unit) on the plant communications loop.

This distributed recipe handling capability generally eliminates the requirement for a centralized computer to store batch recipes, and the associated costs, project complications and operational unreliabilities. Further, recipe storage can be expanded incrementally as the need arises by simply adding another MFC to the control system.

Recipe Handling in a Multi-Function Controller

Generally, recipe data includes parameters which define the quantity of each ingredient to be used in preparing the batch, the conditions under which the parameters are added (temperatures, pressures, flowrates, etc.) and the heating/cooling/mixing times and/or rates. In some processes (such as paint blending, lube oil blending and edible oil refining) only these parameters vary from recipe to recipe.

However, in processes where the same equipment is used to make several significantly different products, the steps to be used in preparing each batch, as well as the order in which the steps are executed, may vary from recipe to recipe. As a batch process controller, the MFC can easily accommodate variations in both the batch sequence and the parameter values.

In NETWORK 90, batch recipes can be configured into the MFC through the Engineering Work Station (EWS) or any of the CRT based operations consoles (OIU, MCS, or 1090 System). For the purpose of this discussion, EWS module configurations and OIU graphic displays will be used. Detailed descriptions and discussions of NETWORK 90 function blocks referenced in this application guide can be found in Bailey Product Instruction E93-900-2, FUNCTION CODE REFERENCE MANUAL.

Storing Recipe Data

A specialized function block has been provided in the MFC to simplify recipe handling. The Real Recipe Table function block (RECIPR block, Function Code 118) stores up to ten (10) values of one recipe parameter. The output of the RECIPR block corresponds to the parameter value for the recipe number input to the RECIPR block. The RECIPR block can be used to change the value of any variable parameter from recipe to recipe. Further, the RECIPR block can be used to change which steps are used for preparation of a batch, and the order in which the steps are executed.

RECIPR blocks can be linked together in parallel (each RECIPR block receives the recipe number as an input) with each block storing up to 10 values for each of the recipe parameters. If more than 10 values are required for each parameter, the RECIPR blocks can be linked together in series to provide the necessary number of values for each parameter (the second RECIPR block is slaved to the first, the third is slaved to the second, etc.). Consequently, a batch process requiring 8 recipes, each with 5 parameters, would require 5 RECIPR blocks to store the recipe data and a batch process requiring 28 recipes each with 5 parameters would require 15 RECIPR blocks (two slave RECIPR blocks connected to each of 5 master RECIPR blocks).

The specifications for the RECIPR block are shown in **FIGURE 2** as they would appear on an OIU configuration display. Specifications S1 through S10 contain the actual value of the recipe parameter for each of ten recipes. Specification S11 contains the address of the recipe selected. Specification S12 contains the block address of the next RECIPR block in a series of linked RECIPR blocks. When several RECIPR blocks are linked together in this fashion, it is necessary only to manipulate S11 for the first RECIPR block in the series (the remaining RECIPR blocks automatically receive the correct commands).

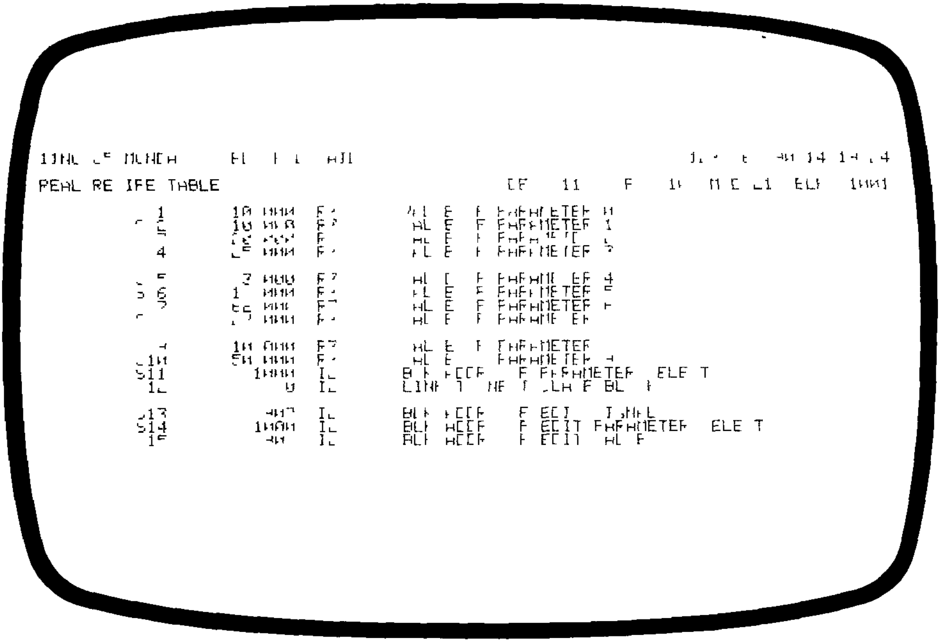

FIGURE 2   RECIPR Block Configuration Display (OIU)

Changing Step Sequences

A specialized function block has been provided in the MFC to facilitate changing the sequence of steps from recipe to recipe. The Multiple Sequence Monitor function block (MULTI MON Function Code 134) provides up to eight (8) phases which can be executed during a batch sequence. A phase is defined as a step which can vary from recipe to recipe. In this context, a step is defined as a fixed action with an associated parameter value (e.g. heating the contents of a batch reactor to 300°F) which is assigned a step number. The same step can be executed several times during a sequence of phases (each time with a different parameter value) or not at all.

MULTIMON blocks function very much like the Sequence Monitor Block (SEQMON Function Code 124) but with variable step sequences and parameter values. RECIPR blocks are used to change the step number, step type and parameter value from recipe to recipe. The various specifications for the MULTIMON block are shown and discussed in detail in **TABLE 2.**

Editing Recipe Data

To facilitate online changes in recipes, RECIPR blocks can be edited from any Bailey CRT based operations console. Referring to **FIGURE 2**, specification S14 is used to identify which recipe is being selected for editing (0-9 in the first RECIPR block of a series, 10-19 in the second RECIPR block, 20-29 in the third RECIPR block, etc.). Specification S15 is the new value of the recipe parameter for the edited recipe. Normally, Individual Remote Manual Set Constant blocks (REMSET Function Code 68) are used to feed specification S14 and S15 of the RECIPR block (only the first RECIPR block in a series). These REMSET blocks are then accessed by a single CRT display to allow convenient online editing.

Once all of the new values for the edited recipe have been entered through the CRT display and verified, they can be written into the RECIPR block by changing the boolean value of the block address identified in specification S14 from 0 to 1. This is normally accomplished using a Remote Control Memory block (RCM Function Code 62) accessed through the recipe editing display. When the RCM is changed to 1, the edited parameter value put into S15 of the RECIPR block is written into the parameter value location identified by the REMSET addressed in S14.

Reciple Handling for a Fixed Batch Sequence

In a fixed batch sequence, the order in which the steps are accomplished remains fixed from recipe to recipe, but the amounts of ingredients and the conditions under which they are added may vary. The following tube oil blending example illustrates how fixed sequence recipes can be edited. In this case, the user has a series of recipes that contain the following information:

Parameter	Description
1	Percent Component A in blend
2	Percent Component B in blend
3	Percent Component C in blend
4	Percent Component D in blend
5	Percent Component E in blend
6	Percent Component F in blend
7	Maximum flowrate of blend
8	Parameter Number of Component to be used for viscosity trim
9	Viscosity setpoint (SUS) for blend
10	Total amount of blend product on (barrels)

Each one of these parameters is stored in a RECIPR block. The recipe is stored in 10 RECIPR blocks tied together in parallel. The MFC configuration logic required to manipulate and store this recipe is shown in **FIGURE 3**. Several things should be noted about this configuration drawing:

1. Each RECIPR block contains one of the recipe parameters.

2. REMSET blocks are used to load the new values into the RECIPR function blocks. These blocks show the current value of the recipe parameter before any changes are entered.

3. The recipe being selected for use is from the REMSET at block address 1000.

4. The RCM at block address 907 is used to write the new values into the RECIPR blocks through REMSETs at block addresses 908 through 917.

5. In this configuration, both the REMSET that selects the recipe parameter for editing and the RCM that writes the new parameter value to the RECIPR block are interlocked to allow editing of the recipe only during Step 1 of the batch sequence. Without this interlock, the recipe can be edited at any time during the execution of the batch.

6. After the user has completed editing a recipe, he can "download" the edited recipe to the RECIPR blocks by triggering the RCM at block address 907 from the CRT console.

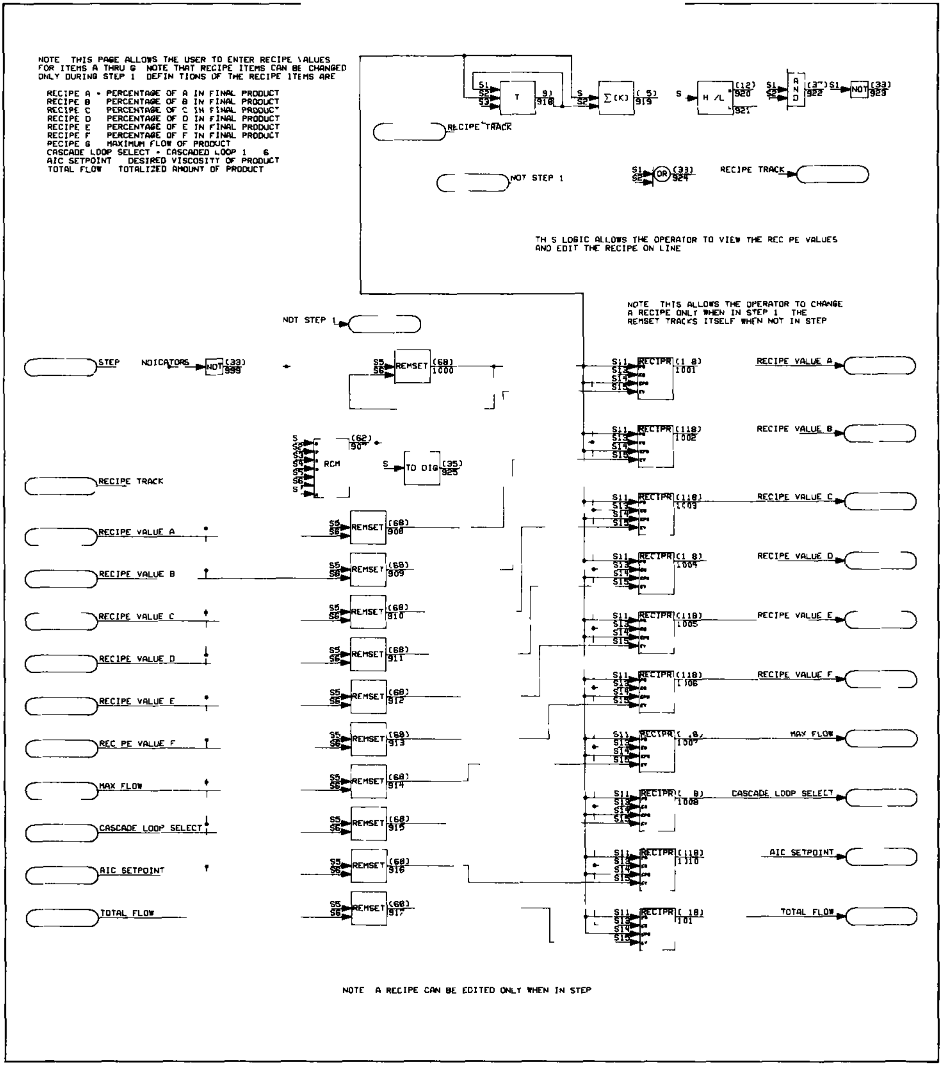

FIGURE 3 MFC Configuration Logic for Fixed Sequence Batch Control Example with On Line Editing

Recipe Handling for Variable Sequence Batches

A variable sequence batch is one in which the *order of the steps to be accomplished changes* from recipe to recipe as well as the amounts of ingredients and the conditions under which they are added. The following example illustrates how variable sequences can be stored and implemented using an MFC.

In this example, the batch reactor has the ability to execute the following steps:

Step	Description
1	Clean Reactor with Water
2	Clean Reactor with Acetone
3	Add component A
4	Add component B
5	Ramp temperature up
6	Ramp temperature down
7	Turn on agitator
8	Turn agitator off
9	Pump product out of reactor
10	End

Each recipe can use any of the above steps *once, multiple times or not at all. Two recipes* that use these steps are shown in **TABLE 1.**

The MFC configuration logic required to store and manipulate variable sequence recipes is shown in **FIGURE 4.** The logic required to change the step to be executed is shown for one phase only. This logic is repeated for all twelve phases of the example.

PHASE	DESCRIPTION	RECIPE 1		RECIPE 2	
		STEP	VALUE	STEP	VALUE
1	1st step to execute	1		2	
2	2nd step to execute	2		3	400
3	3rd step to execute	3	250	7	
4	4th step to execute	7		5	25
5	5th step to execute	4	350	4	300
6	6th step to execute	6	30	6	25
7	7th step to execute	3	250	4	200
8	8th step to execute	9		3	200
9	9th step to execute	8		9	
10	10th step to execute	10		8	
11	11th step to execute			10	
12	12th step to execute				

TABLE 1
Batch Recipes Using Different Sequences of Predefined Steps

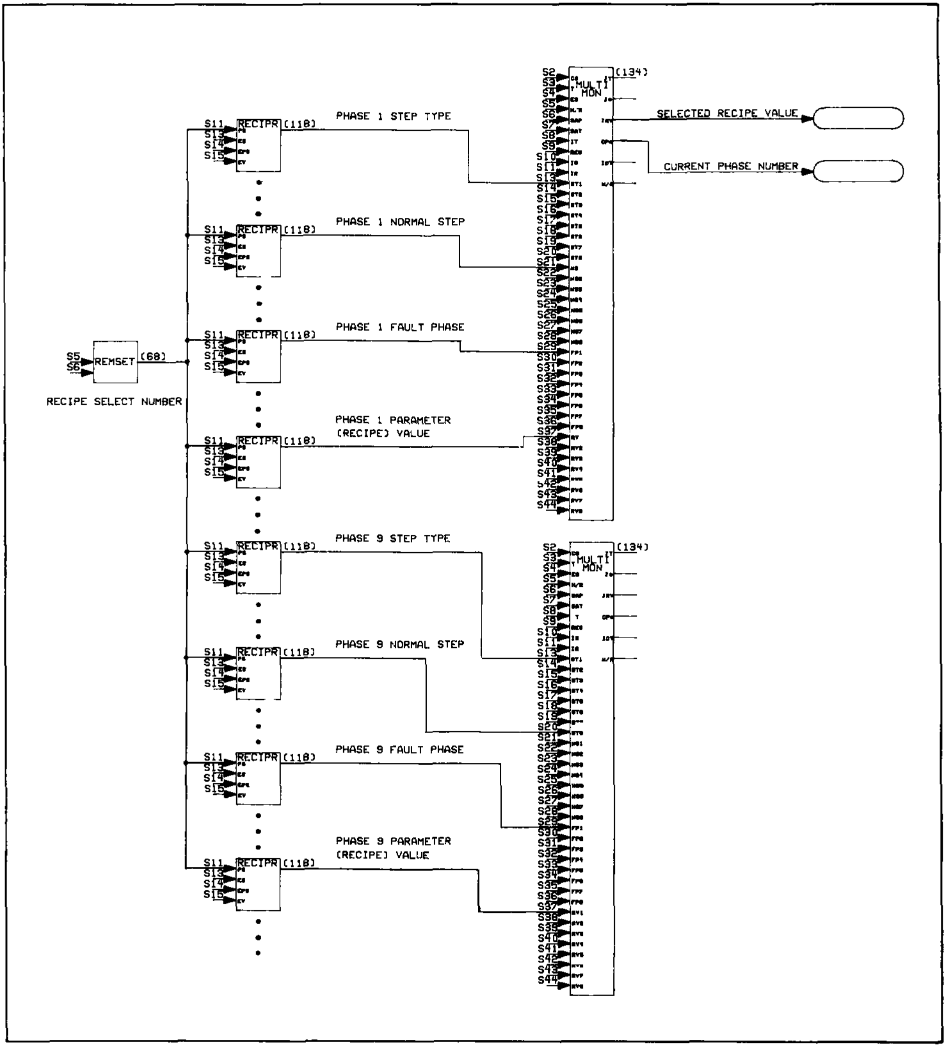

FIGURE 4 MFC Configuration Logic for Variable Sequence Batch Control Example

On-Line Recipe Editing

Although the recipes are stored and executed in the MFC the actual on line editing for the recipes is accomplished through one of the Bailey CRT based consoles A typical editing display is shown in **FIGURE 5.**

This display presents the editing data in tabular form for clarity The user can call up an RCM or REMSET mimic (lower right hand corner) for each of the editing functions through Control Select Numbers (1 12)

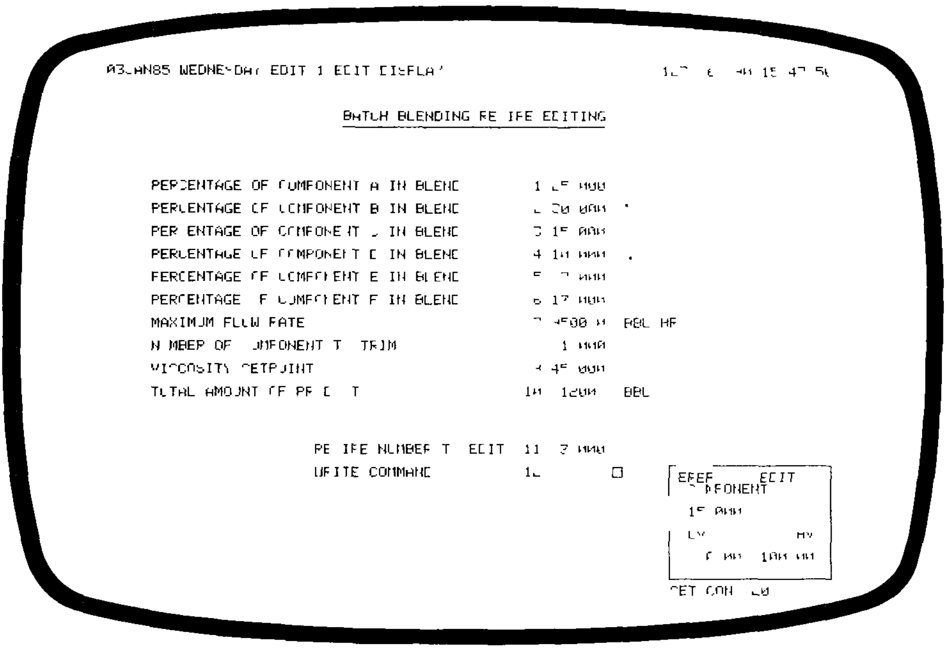

FIGURE 5 CRT Based Recipe Editing Graphic Display

Multi Function Controller Recipe Storage Capacity

To determine whether a batch recipe is suitable for storage in an MFC the following information is required

1 The number of recipes
2 The number of parameters or phases in each recipe

00 28 36 04 11 07

3 The memory requ rement for each REC PR b ock s 47 bytes of EEROM (e ectr ca y eraseab e read on y memory)

A f xed sequence batch prccess w th 200 d f ferent rec pes nc ud ng 20 parameters each wou d requ re 20 cha ns of REC PR b ocks (one cha n for each rec pe parameter) Each cha n wou d con ta n 20 REC PR b ocks n ser es Consequent y a tota of 400 RECIPR b ocks and 18 000 bytes of EEROM wou d be requ red f the MFC to be used has 20 000 bytes of ava ab e EEROM, then storage of th s rec pe data wou d ut ze approx mately 95% of the MFC s ava lab e memory The add t ona conf gurat on requ red for on ne ed t ng of rec pes wou d requ re another 570 bytes ncreas ng the MFC ut zat on to 97%

A var ab e sequence batch process w th 60 d f ferent recipes inc uding 16 phases would require 64 cha ns of REC PR b ocks (one cha n for each MULT MON spec f cat on S13 through S44 two MULT MON b ocks for 16 phases) Each cha n wou d conta n 6 REC PR b ocks to store the step number and parameter va ues for the 60 rec pes Consequent y, a tota of 384 REC PR b ocks and 18 048 bytes of EEROM wou d be requ red f the MFC to be used has 20 000 bytes of ava able EEROM then storage of th s rec pe data wou d ut ze approx mate y 90% of the MFC's ava ab e memory

FUNCTION CODE 134

MULTIPLE SEQUENCE MON TOR

Description

The Mu t p e Sequence Mon tor funct on b ock contro s the execut on of an assoc ated Sequence Generator funct on b ock (Funct on Code 161) Th s b ock contro s the execut on of e ght (8) phases n th s context a phase s def ned as a step wh ch may vary from rec pe to rec pe The purpose of the Mu t p e Sequence Mon tor s to prov de a means to change the order n wh ch predef ned steps are executed n d fferent rec pes

The phases are processed n numer ca order from one to e ght Each phase has four parameters wh ch are def ned at addresses con f gured nto the Mu t p e Sequence Mon tor spec f cat ons These parameters are (where N s from 1 to 8)

1 The b ock address of Phase N step type Step type s a two d g t rea number that def nes the cond t ons necessary for contro to advance to the next step and whether the operator s perm t ted to ho d or use a sem automat c tr gger at each step The step types a owed are as fo ows (note that th s s a two d g t spec f cat on, X or Y means that the other d g t s used to def ne other features than the one be ng current y exp a ned)

XO Advance to the next phase f a dev ces are good and the phase tr gger s on

X1 Advance to the next phase f a dev ces are good (gnore the phase tr gger)

X2 Advance to the next phase f the phase tr gger s on gnore dev ce status)

0Y Perm t the operator to HOLD the sequence and to use the sem automatic phase tr gger

1Y Perm t the operator to HOLD the sequence on y (no sem automat c phase tr gger)

2Y Do not perm t the operator to HOLD the sequence or to use the sem automat c phase tr gger

TABLE 2 Multiple Sequence Monitor

2. The block address of Phase N normal step. This parameter identifies the step number which the phase is intended to execute. For example, if the Phase 5 normal step has a value of 7, then Step 7 in the sequence generator block would be executed when Phase 5 is triggered.

3. The block address of Phase N fault phase. If the step type to be executed during Phase N requires a device status check before advancing to the next phase, when a device fails, control will go to the identified fault phase.

4. The block address of Phase N recipe value. Each step used by a phase can have an associated analog parameter value. For example, in charging a reactor, the analog value associated with that operation is the amount of ingredients to be added.

By configuring these specifications, the Multiple Sequence Monitor will automatically output (as Output N+2) the recipe value of the active phase.

In operation, the Multi Sequence Monitor is very similar to the Sequence Monitor block (Function Code 124). The operator can control the operation of the block through S5. If the block address at S5 has a value of 0, then the Multi Sequence Monitor is in the RUN mode. If the value is 1, then the sequence is in HOLD. Upon transition of S5 from 1 to 0, the first phase to be executed is the value found at S9.

Normally, the step type is X0, which means the control will go from one phase to the next when the phase trigger is on and the status of all devices is good. Note that, as in all applications using triggers, the function block representing the trigger must change from a 0 state to a 1 state.

The semi automatic mode is a special mode of operation in which an external pushbutton (the semi auto phase trigger) is one of the necessary semi auto conditions to advance to the next step. The SEMI AUTO mode is activated when the block address at S6 has a boolean value of zero and the step type permits a semi auto phase trigger. The semi auto phase trigger is normally a Remote Control Memory block (RCM, Function Code 62) at the block address indicated by S7.

The emergency stop function is normally provided by an RCM at the block address indicated by S4. When the RCM has a boolean output of one, then control will go to Step 0. Emergency stop overrides all other triggers.

When switching the MFC from CONFIGURE to EXECUTE mode, all Multiple Sequence Monitor blocks will go to Step 0. To activate the sequence, the Multiple Sequence Monitor block must be switched to HOLD and then back to RUN. The purpose of this procedure is to prevent the user from inadvertently starting a sequence when setting the MFC to RUN mode.

The Multiple Sequence Monitor also allows a step to be manually inserted into the programmed sequence. With the sequence in "hold", when the insert trigger (normally provided by an RCM block at the address identified in S8) is changed from 0 to 1, the step number specified (normally provided by a REMSET block at the address identified in S10) will be executed with the parameter value (normally provided by a REMSET block at the address identified in S11, when applicable). If the sequence is returned to "run" while the insert step is executing, the sequence will automatically resume once the inserted step generates a step trigger.

In sequences with more than eight phases, several Multiple Sequence Monitors can be linked together. Specification S1 is used to point to the next Multiple Sequence Monitor block in a series. In this case, specifications S2 through S11 must be entered for only the first Multiple Sequence Monitor in the series; the slaved blocks automatically read these specifications from the first block in the series.

OUTPUTS (Function Code 134)

Block Number	Data Type	Description
N	Real	Jump Step Number
N+1	Boolean	Jump Step Trigger
N+2	Real	Jump Step Recipe Value
N+3	Real	Current Phase Number
N+4	Boolean	Insert Step Done
N+5	Boolean	Status of Sequence (1 = HOLD, 0 = RUN)

TABLE [illegible]. Multiple Sequence Monitor (continued)

SPECIFICATIONS (Function Code 134)

Spec No	Tune	Default Value	Data Type	Range Min Max	Description
S1	No	0	2	0 to 2046	Block Address of Next Multi Sequence Monitor
S2	No	5	2	0 to 2046	Block Address of Control Status Input
S3	No	0	2	0 to 2046	Block Address of Step Trigger
S4	No	0	2	0 to 2046	Block Address of Emergency Stop
S5	No	0	2	0 to 2046	Block Address of Hold/Resume
S6	No	1	2	0 to 2046	Block Address of Semi Auto Permissive
S7	No	0	2	0 to 2046	Block Address of Semi Auto Trigger
S8	No	0	2	0 to 2046	Block Address of Insert Trigger
S9	No	5	2	0 to 2046	Block Address of Resume Phase Number
S10	No	5	2	0 to 2046	Block Address of Insert Step Number
S11	No	0	2	0 to 2046	Block Address of Insert Recipe Value
S12	No	0	R3	Full	Spare
S13	No	5	2	0 to 21	Block Address of Phase #1 Step Type
S14	No	5	2	0 to 21	Block Address of Phase #2 Step Type
S15	No	5	2	0 to 21	Block Address of Phase #3 Step Type
S16	No	5	2	0 to 21	Block Address of Phase #4 Step Type
S17	No	5	2	0 to 21	Block Address of Phase #5 Step Type
S18	No	5	2	0 to 21	Block Address of Phase #6 Step Type
S19	No	5	2	0 to 21	Block Address of Phase #7 Step Type
S20	No	5	2	0 to 21	Block Address of Phase #8 Step Type
S21	No	5	2	0 to 2046	Block Address of Phase #1 Normal Step
S22	No	5	2	0 to 2046	Block Address of Phase #2 Normal Step
S23	No	5	2	0 to 2036	Block Address of Phase #3 Normal Step
S24	No	5	2	0 to 2046	Block Address of Phase #4 Normal Step
S25	No	5	2	0 to 2046	Block Address of Phase #5 Normal Step
S26	No	5	2	0 to 2046	Block Address of Phase #6 Normal Step
S27	No	5	2	0 to 2046	Block Address of Phase #7 Normal Step
S28	No	5	2	0 to 2046	Block Address of Phase #8 Normal Step
S29	No	5	2	0 to 2046	Block Address of Phase #1 Fault Phase
S30	No	5	2	0 to 2046	Block Address of Phase #2 Fault Phase
S31	No	5	2	0 to 2046	Block Address of Phase #3 Fault Phase
S32	No	5	2	0 to 2046	Block Address of Phase #4 Fault Phase
S33	No	5	2	0 to 2046	Block Address of Phase #5 Fault Phase
S34	No	5	2	0 to 2046	Block Address of Phase #6 Fault Phase
S35	No	5	2	0 to 2046	Block Address of Phase #7 Fault Phase
S36	No	5	2	0 to 2046	Block Address of Phase #8 Fault Phase
S37	No	5	2	0 to 2046	Block Address of Phase #1 Parameter Value
S38	No	5	2	0 to 2046	Block Address of Phase #2 Parameter Value
S39	No	5	2	0 to 2046	Block Address of Phase #3 Parameter Value
S40	No	5	2	0 to 2046	Block Address of Phase #4 Parameter Value
S41	No	5	2	0 to 2046	Block Address of Phase #5 Parameter Value
S42	No	5	2	0 to 2046	Block Address of Phase #6 Parameter Value
S43	No	5	2	0 to 2046	Block Address of Phase #7 Parameter Value
S44	No	5	2	0 to 2046	Block Address of Phase #8 Parameter Value

00 28 43 04 11 07

Ba ley Controls Wickliffe Ohio 44092 a division of THE BABCOCK & WILCOX COMPANY

Bailey Controls Australia Pty Ltd Regents Park N S W Austral a
Bailey do Brasil Sao Paulo Brazil
Ba ley Controls D v of B&W Industr es td Burlington Ontar o Canada
Bailey Japan Company Ltd Sh zuoka Ken Japan
Representatives in Other Principal C t es

Form AG 0000 300 03 L tho n U S A 186